



---

## **Universidad de Valladolid**

**ESCUELA TÉCNICA SUPERIOR DE INGENIEROS DE TELECOMUNICACIÓN**  
**DEPARTAMENTO TEORÍA DE LA SEÑAL Y COMUNICACIONES E INGENIERÍA**  
**TELEMÁTICA**

**TESIS DOCTORAL:**

**APLICACIÓN DE TÉCNICAS NO LINEALES Y OTROS**  
**PARADIGMAS EN *SMART GRID/MICROGRID/VIRTUAL***  
***POWER PLANT***

Presentada por D. Luis Hernández Callejo para optar al grado de  
doctor por la Universidad de Valladolid

Dirigida por:  
Dra. D<sup>a</sup>. Belén Carro Martínez  
Dr. D. Antonio Javier Sánchez Esguevillas  
Dr. D. Javier Manuel Aguiar Pérez



## DEDICATORIA

Dedicado a todos y cada uno de los miembros de mi familia, por su total apoyo y comprensión durante esta maravillosa, pero a veces difícil, tarea que es la vida. En especial, a aquellos que ya no están y que ayudaron a conseguir lo que ahora soy como persona, y a los recién llegados a mi vida que me sorprenden al descubrir las cosas de importancia que transmiten con su especial luz.

Si algún estudiante leyera mi *Tesis Doctoral*, supongo que tendrá dudas respecto a si continúa con su proceso de formación, ya que es una labor larga y dificultosa. Permítame darle un consejo, la perseverancia y el sacrificio suelen tener una recompensa al final, y en caso contrario, siempre nos hará mejores personas. Sirva de ejemplo la publicación en el *Times* en 1907, donde Sir Ernest Shackleton solicitaba hombres para su expedición, en la que tras la pérdida del barco *Endurance* (“*resistencia*”), encumbró a Sir. Shackleton como uno de los héroes de la Antártida. El esfuerzo legendario de personas como él, servirá como guía y esperanza para los momentos complicados de la vida.

*“Se buscan hombres para peligroso viaje. Salario reducido. Frío penetrante. Largos meses de completa oscuridad. Constante peligro. Dudoso regreso sano y salvo. En caso de éxito, honor y reconocimiento”*



## AGRADECIMIENTOS

Muchos han sido los que han colaborado en esta *Tesis Doctoral*, por tanto, gracias a todos ellos. Pero en particular, quiero manifestar mi gratitud a:

Dra. Dña. Belén Carro Martínez, Dr. D. Antonio Javier Sánchez Esguevillas y Dr. D. Javier Manuel Aguiar Pérez, directores de *Tesis Doctoral*, por su tiempo, dedicación y apoyo en la realización de la misma, y además, por su orientación en los momentos complicados y por transmitirme su filosofía de trabajo.

Al grupo de trabajo al que pertenecen mis tutores, que forma parte del Departamento Teoría de la Señal y Comunicaciones e Ingeniería Telemática. Muy especialmente al Dr. D. Carlos Baladrón Zorita, por su colaboración en las publicaciones en revistas del mundo científico. Igualmente al Dr. D. Jorge Jesús Gómez-Sanz por su ayuda con los prototipos de simulaciones con *Multi-Agent System* e *INGENIAS*.

A D. Ramón Gavela González, Director del Departamento de Energía y Director Adjunto del Centro de Investigaciones Energéticas, Medioambientales y Tecnológicas (*CIEMAT*), y a D. Miguel Latorre Zubiri, Director del Centro de Desarrollo de Energías Renovables (*CEDER-CIEMAT*), quienes han sabido apoyarme en mis trabajos iniciados en paralelo a mi *Tesis Doctoral*, facilitando la creación de una nueva línea de investigación en la cual participo. En la actualidad, y gracias a ellos dos, desempeño el cargo de Responsable de Sistemas y Comunicaciones, así como de Coordinador de Proyectos de *Smart Grid/Smart Metering/Microgrid* en *CEDER-CIEMAT* y Coordinador de Unidad Virtual en Energía Inteligente de *CIEMAT*.

A la Agencia Estatal de Meteorología. Ministerio de Medio Ambiente (*AEMET*) por los datos facilitados de su histórico de variables climáticas.

A la *utility* española Iberdrola, a través de D. Oscar Villanueva Jefe del Negocio de Red de Iberdrola en Burgos y Soria y a Dña. Silvia Herrero Responsable de Comunicación de Iberdrola en Castilla y León, por los datos eléctricos proporcionados.

A todos y cada uno, mi sincero agradecimiento.



## NOTA PARA LECTORES

Con la presente *Tesis Doctoral*, se pretende conseguir una reflexión acerca de la necesidad de una visión multidisciplinar en ciertos problemas que rodean determinados ámbitos de la ingeniería. Es algo muy común tratar de dar una solución desde el plano donde un ingeniero se siente más cómodo, perdiendo de esta manera la perspectiva de otras disciplinas, que pueden dar valor a la solución final.

Es el caso concreto del emergente mundo de la *Smart Grid*, *Smart City*, *Microgrid*, *Smart Building*, *Smart Home*, *Smart Metering*, *Smart Customer*, entre otras, donde las soluciones a los problemas a resolver deben de partir de vertientes tan diferentes como la ingeniería eléctrica, ingeniería informática, ingeniería de las telecomunicaciones, ingeniería de diseño, ingeniería civil e incluso la sociología para el entendimiento de ciertos hábitos de comportamiento. Además de presentar los resultados de la *Tesis Doctoral*, se pretende por tanto romper la barrera de las especializaciones, y alentar a investigadores y empresas a colaborar de manera conjunta en proyectos y propuestas futuras, donde la visión conjunta de las especializaciones individuales, cree una sinergia potenciadora del problema global, y sirva para dar una solución acertada de un problema, que como se ha dicho, requiere de perspectivas expertas en diferentes áreas.





## RESUMEN

En los últimos años, la aparición de elementos de electrónica con mayor poder de cómputo y de decisión, junto a la irrupción de fuentes de generación renovable, cada vez más cercanas a los puntos de consumo, ha propiciado la aparición de nuevo espacios que albergarán demanda y generación. Por tanto, para su correcta operación se precisarán herramientas estimadoras de la demanda eléctrica.

La presente *Tesis Doctoral* tiene objetivo doble, por un lado se diseñarán algoritmos de predicción de la demanda eléctrica en entornos desagregados (ciudad pequeña, entorno rural o *microgrids*) a partir de datos históricos, y por otro lado se planteará una arquitectura, basada en Sistemas Multi-Agente (*Multi-Agent System – MAS*), para la predicción de demanda eléctrica, orientada a Hogares Inteligentes (*Smart Home*) en la Planta de Energía Virtual (*Virtual Power Plant – VPP*).

Por tanto, el objetivo primero se concretará en los siguientes objetivos específicos para estos entornos desagregados: a) necesidad de conocer la influencia de las variables climáticas vs demanda eléctrica en entornos desagregados, y como consecuencia, proponer un término aglutinador de los entornos “*Smart*” en el mundo eléctrico llamado Mundo Inteligente (*Smart World*); b) diseño y aplicación de técnicas no lineales y algoritmos de clusterizado, para la agrupación de las curvas de carga en un polígono industrial, y su uso en la predicción de la demanda eléctrica por parte de los *agregadores*; c) los nuevos entornos desagregados, precisarán una estimación de su comportamiento energético, por tanto se plantea el diseño de un modelo para Predicción de Carga a Corto Plazo (*Short-Term Load Forecasting – STLF*) basado en Red Neuronal Artificial (*Artificial Neural Network – ANN*) para la predicción de la curva de carga del día siguiente; d) a partir de los buenos resultados obtenidos en b), se plantea un diseño donde tras un reconocimiento de patrones y su posterior clusterizado, se realizará *STLF* basado en *ANN* para cada uno de los clústers, realizando una comparativa con c); e) debido a la importancia del conocimiento de la estimación de la demanda agregada (demanda total) para las Empresas Eléctricas (*utilities*) y nuevos actores del sistema eléctrico, se plantea el diseño de un modelo de predicción de la demanda agregada del día siguiente basado en *ANN*, con un estudio previo de las variables relevantes de interés para el modelo; f) el aporte de información sobre la curva de carga a predecir se convierte en fundamental, por lo tanto, se diseña un modelo para *STLF* basado en *ANN* para la predicción de la curva de carga del día siguiente, pero con información relevante del día a predecir.

Como objetivo secundario de la *Tesis Doctoral*, se plantea el diseño conceptual de herramientas de futuro basadas en el paradigma *MAS*. Primero, se presentará una aplicación basada en *MAS* para su uso en entornos desagregados, cumpliendo los objetivos de *Smart World*, cuyo objetivo será el controlar la relación entre las variables climáticas y la demanda agregada de cierta zona; posteriormente, los resultados de las relaciones se emplearán para los modelos de predicción de la demanda eléctrica. Para finalizar, se presentará un modelo basado en *MAS* para *VPP*, la propuesta no sólo se centra en la gestión de los diferentes elementos de la *VPP*, sino que incluye un conjunto de agentes integrados con *ANN* para la previsión de la demanda de energía de los usuarios domésticos finales dentro de la *VPP*.

La demanda eléctrica es totalmente dependiente de ciertos parámetros, incorporando en la misma cierta incertidumbre, que repercutirá en el precio final de la energía para los consumidores. Para tratar de reducir el coste, los gestores de los grandes sistemas de energía, necesitan conocer con precisión los mecanismos que hacen variar ostensiblemente esta demanda eléctrica, y de esta manera lograr dos objetivos principalmente: prever y adaptarse a la demanda en los diferentes plazos; tratar de gestionar la demanda.

En estos momentos, el modelo del sistema eléctrico ha comenzado a cambiar en los últimos años, al ir introduciendo inteligencia en los diferentes niveles del mismo, comenzando por el transporte, en los últimos años en la distribución, y aunque en menor medida, recientemente en las proximidades al consumo final. De esta manera surge la Red Inteligente (*Smart Grid – SG*), término traducible por red de energía eléctrica inteligente, ya que utiliza elementos de inteligencia y comunicaciones para intentar optimizar la producción y la distribución de electricidad con el fin de equilibrar mejor la oferta y la demanda entre productores y consumidores. Además de lo anterior, y aprovechando el despliegue de sensorica e inteligencia, se tratará de optimizar los flujos de energía, mejorar la algoritmia de detección de faltas, y por tanto, mejorar en la operación de la red.

Junto a *SG* surgen una serie de escenarios: Ciudad Inteligente (*Smart City – SC*), *VPP*, microrredes (*microgrids*), Edificios Inteligentes (*Smart Buildings*), etc.; cuya característica común es que su demanda eléctrica aparece desagregada si se compara con grandes extensiones (regiones) o incluso la demanda de un país. Es por tanto, que en estos nuevos espacios con demanda y generación se precisará el despliegue de herramientas de predicción tanto de la demanda como de la generación.

## ABSTRACT

In recent years, the emergence of electronic components with higher computing power and decision, together with the emergence of renewable energy sources, closer to the point of consumption, has created a favorable atmosphere for the appearance of new spaces that will house demand and generation. Therefore, estimating tools of electricity demands will be necessary for proper operation.

The present *PhD Thesis* has double aim. On one hand, prediction algorithms of electricity demand in disaggregated environments are designed (small cities, rural environments or *microgrids*) from historical data. On the other hand, architecture for electricity demand forecasting based on *Multi-Agent System (MAS)*, directed to *Smart Home* in the *Virtual Power Plant (VPP)*, will be suggested.

Therefore, the first aim will be specified in the following specific objectives for these disaggregated environments: a) the need to know the influence of climatic variables vs. electricity demand in disaggregated environments. Consequently it is going to propose an unifying term of “*Smart*” environments in the electrical world, called *Smart World*; b) design and implementation of nonlinear techniques and clustering algorithms, for clustering load curves in an industrial estate, and their use in electricity demand forecasting by *aggregators*; c) new disaggregated environments will require an estimate of their energy performance, therefore the design of model for *Short-Term Load Forecasting (STLF)* based on *Artificial Neural Network (ANN)* is considered for the prediction for the load curve of the next day; d) from good results obtained in b), it is going to propose a design in which, after a pattern recognition and subsequent clustered, a *STLF* based on *ANN* for each of the clusters will be carried out, and it will be compared with c); e) due to the importance of knowledge of the estimation of aggregate demand (total demand) for *utilities* and new agents of the electrical system, it is proposed the design of a predictive model of aggregate demand of the next day based on *ANN* with a previous study of the relevant variables of interest for the model; f) the contribution of information on the load curve to predict becomes critical, therefore, so it is designed a model based on *ANN* for *STLF* in order to predict the load curve of the next day, but with relevant information of the day-to-predict.

As secondary aim of the *PhD Thesis*, it is proposed the conceptual design of future tools based on *MAS* paradigm. First, it is presented a *MAS*-based application for disaggregated environments, meeting the objectives of *Smart World*, whose goal will be to control the relationship between

climatic variables and aggregate demand in a certain area. Then, the relationships' results will be used for the electricity demand forecasting models. Finally, a model based on *MAS* for *VPP* will be presented. This model not only focuses on the management of the different elements of the *VPP*, but includes a set of integrated agents with *ANN* for forecasting the energy demand of the end domestic users within the *VPP*.

Electricity demand is totally dependent on certain parameters, incorporating therein some uncertainty, which will affect the final price of energy for consumers. Trying to reduce the cost, managers of large power systems, need to know the precise mechanisms that make this electricity demand vary significantly. Thus two main objectives will be achieved: anticipate and adapt to the demand in different periods and try to manage it.

At present, the power system model has changed in recent years by introducing intelligence at different levels of that power system (starting from transport, in recent years in distribution, and to a lesser extent, recently in the nearness of the final consumption). Thus arise the *Smart Grid (SG)*, also known as smart power grid because it uses intelligence and communication elements to try to optimize the production and distribution of electricity in order to balance better supply and demand between producers and consumers. Besides the above mentioned and using the sensor system and intelligence deployment, the flow of energy will be optimized, improve fault detection algorithms, and thus improve the operation of the network.

Together with *SG* arise a number of scenarios: *Smart City (SC)*, *VPP*, *microgrids*, *Smart Buildings*, etc.; whose common characteristic is that its electricity demand appears disaggregated when compared with large areas (regions) or even the demand of a country. It is, therefore, that these new spaces with demand and generation will require the deployment of predictive tools both for demand and generation.

# ÍNDICE GENERAL

<b>DEDICATORIA .....</b>	<b>I</b>
<b>AGRADECIMIENTOS.....</b>	<b>III</b>
<b>NOTA PARA LECTORES.....</b>	<b>V</b>
<b>RESUMEN .....</b>	<b>VII</b>
<b>ABSTRACT.....</b>	<b>IX</b>
<b>ÍNDICE GENERAL .....</b>	<b>XI</b>
<b>ÍNDICE TABLAS.....</b>	<b>XVII</b>
<b>ÍNDICE FIGURAS.....</b>	<b>XIX</b>
<b>CAPÍTULO I. INTRODUCCIÓN .....</b>	<b>1</b>
<b>1.1. INTRODUCCIÓN .....</b>	<b>1</b>
<b>1.2. MOTIVACIÓN .....</b>	<b>3</b>
<b>1.3. OBJETIVOS.....</b>	<b>4</b>
<b>1.4. CONTRIBUCIONES ORIGINALES.....</b>	<b>5</b>
<b>1.5. METODOLOGÍA .....</b>	<b>5</b>
<b>1.6. ESTRUCTURA DE LA TESIS.....</b>	<b>7</b>
<b>CAPÍTULO II. EL SISTEMA ELÉCTRICO .....</b>	<b>11</b>
<b>2.1. EL SISTEMA ELÉCTRICO: ORÍGENES.....</b>	<b>11</b>
<b>2.2. EVOLUCIÓN DEL SISTEMA ELÉCTRICO: SMART GRID.....</b>	<b>17</b>
<b>2.3. UN PASO MÁS PARA LAS FUENTES DE GENERACIÓN RENOVABLE: GENERACIÓN DISTRIBUIDA.....</b>	<b>20</b>
<b>2.4. LA AGREGACIÓN DE FUENTES DE GENERACIÓN COMO SOLUCIÓN DE FUTURO: VIRTUAL POWER PLANT.....</b>	<b>23</b>
<b>2.5. UN NUEVO PARADIGMA DE SOSTENIBILIDAD: MICROGRID.....</b>	<b>26</b>
<b>2.6. NUEVOS ACTORES EN EL HORIZONTE. SINERGIAS ENTRE MODELOS Y NUEVO CONCEPTO AGLUTINADOR .....</b>	<b>29</b>
<b>CAPÍTULO III. LA DEMANDA ELÉCTRICA.....</b>	<b>37</b>
<b>3.1. CARACTERÍSTICAS DE LA DEMANDA ELÉCTRICA.....</b>	<b>37</b>
<b>3.2. LA EFICIENCIA DEL SISTEMA ELÉCTRICO Y LA INCERTIDUMBRE DE LA DEMANDA ELÉCTRICA.....</b>	<b>41</b>
<b>3.3. EL PAPEL DE LOS CONSUMIDORES ELÉCTRICOS ANTE LA EFICIENCIA ENERGÉTICA.....</b>	<b>42</b>

<b>3.4. NECESIDADES DE LA PREDICCIÓN DE LA DEMANDA ELÉCTRICA: ANTECEDENTES.....</b>	<b>44</b>
3.4.1. <i>La importancia de la predicción de la demanda eléctrica .....</i>	<i>45</i>
3.4.2. <i>Enfoque histórico de la demanda.....</i>	<i>47</i>
3.4.3. <i>Predicción de la demanda en los entornos de futuro .....</i>	<i>48</i>
<b>3.5. LA DIFICULTAD DE LA PREDICCIÓN DE LA DEMANDA ELÉCTRICA EN ENTORNOS DESAGREGADOS: VISIÓN DE FUTURO.....</b>	<b>49</b>
<b>3.6. CLASIFICACIÓN DE LA PREDICCIÓN DE LA DEMANDA .....</b>	<b>52</b>
3.6.1. <i>Clasificación según el horizonte de predicción .....</i>	<i>52</i>
3.6.2. <i>Clasificación según el objetivo de la predicción.....</i>	<i>55</i>
<b>CAPÍTULO IV. TÉCNICAS DE PREDICCIÓN .....</b>	<b>59</b>
<b>4.1. INTRODUCCIÓN .....</b>	<b>59</b>
4.1.1. <i>Concepto de predicción.....</i>	<i>60</i>
4.1.2. <i>Fundamentos de la predicción .....</i>	<i>61</i>
4.1.3. <i>Técnicas de predicción.....</i>	<i>66</i>
4.1.4. <i>Nomenclatura empleada.....</i>	<i>67</i>
4.1.5. <i>Predicciones .....</i>	<i>71</i>
<b>4.2. LOS MODELOS BASADOS EN LEYES.....</b>	<b>72</b>
4.2.1. <i>El concepto de modelado .....</i>	<i>72</i>
4.2.2. <i>El concepto de simulación .....</i>	<i>74</i>
4.2.3. <i>La validación y verificación de los modelos .....</i>	<i>75</i>
4.2.4. <i>Búsqueda de parámetros .....</i>	<i>76</i>
4.2.5. <i>Sensibilidad de los modelos .....</i>	<i>79</i>
<b>4.3. LA IDENTIFICACIÓN DE SISTEMAS .....</b>	<b>79</b>
4.3.1. <i>El planteamiento del problema.....</i>	<i>80</i>
4.3.2. <i>Las etapas de la identificación .....</i>	<i>82</i>
4.3.3. <i>El filtrado de los datos .....</i>	<i>83</i>
4.3.4. <i>El proceso de selección de las variables de entrada.....</i>	<i>84</i>
4.3.5. <i>El proceso de selección de los modelos.....</i>	<i>88</i>
4.3.6. <i>Ajuste de los parámetros.....</i>	<i>89</i>
4.3.7. <i>La validación.....</i>	<i>92</i>
<b>4.4. LA PREDICCIÓN CON MODELOS LINEALES .....</b>	<b>98</b>
4.4.1. <i>Modelos lineales paramétricos .....</i>	<i>98</i>
4.4.2. <i>La técnica de los mínimos cuadrados .....</i>	<i>101</i>
<b>4.5. LA PREDICCIÓN CON MODELOS NO LINEALES.....</b>	<b>103</b>
4.5.1. <i>Los modelos con no linealidades concentradas.....</i>	<i>103</i>
4.5.2. <i>Redes Neuronales: historia .....</i>	<i>105</i>
4.5.3. <i>Redes Neuronales: fundamentos .....</i>	<i>107</i>
4.5.4. <i>Características de las ANN.....</i>	<i>120</i>
4.5.5. <i>Redes neuronales supervisadas.....</i>	<i>125</i>
4.5.6. <i>Redes auto-organizadas .....</i>	<i>142</i>

4.5.7.	<i>Otros modelos de redes neuronales.....</i>	152
4.6.	<b>ANTECEDENTES SOBRE TRABAJOS DE LA PREDICCIÓN DE LA DEMANDA ELÉCTRICA CON MODELOS LINEALES Y NO LINEALES.....</b>	160
4.6.1.	<i>Trabajos con modelos lineales: Antecedentes.....</i>	161
4.6.2.	<i>Trabajos con modelos no lineales: Antecedentes.....</i>	166
4.6.3.	<i>Trabajos de predicción de la demanda en entornos desagregados: Antecedentes.....</i>	173
4.6.4.	<i>Comparativa.....</i>	175
<b>CAPÍTULO V. SISTEMAS MULTI-AGENTES.....</b>		183
5.1.	<b>LA TECNOLOGÍA DE AGENTES: ORIGEN Y VENTAJAS.....</b>	183
5.1.1.	<i>El concepto de Agente. Inicio y evolución de la tecnología.....</i>	184
5.1.2.	<i>Y a partir del siglo XXI....¿Qué?.....</i>	191
5.2.	<b>ARQUITECTURA DE AGENTES.....</b>	192
5.2.1.	<i>Agentes como sistemas intencionales.....</i>	193
5.2.2.	<i>Arquitecturas para construir agentes.....</i>	193
5.2.3.	<i>Arquitecturas Multi-Agente.....</i>	199
5.2.4.	<i>La arquitectura FIPA.....</i>	201
5.3.	<b>COMUNICACIÓN ENTRE AGENTES.....</b>	206
5.3.1.	<i>El acto de comunicarse.....</i>	206
5.3.2.	<i>Estructura de los mensajes (estándares).....</i>	208
5.3.3.	<i>Semántica de los mensajes.....</i>	211
5.3.4.	<i>Tecnologías de interacción.....</i>	213
5.3.5.	<i>Lenguaje natural: interacción persona-Agente.....</i>	215
5.4.	<b>COORDINACIÓN EN SISTEMAS MULTI-AGENTES.....</b>	216
5.4.1.	<i>Modelos para la coordinación.....</i>	217
5.4.2.	<i>Espacio de coordinación.....</i>	219
5.4.3.	<i>Toma de decisiones para la coordinación.....</i>	222
5.4.4.	<i>Colaboración y cooperación.....</i>	224
5.5.	<b>HERRAMIENTAS Y MÉTODOS.....</b>	227
5.5.1.	<i>Necesidad de herramientas y métodos para el desarrollo con Sistemas Multi-Agentes.....</i>	228
5.5.2.	<i>Cuándo desarrollar con Agentes.....</i>	228
5.5.3.	<i>Modelado con Agentes.....</i>	229
5.5.4.	<i>Implementación.....</i>	235
5.5.5.	<i>Validación y pruebas.....</i>	241
<b>CAPÍTULO VI. APLICACIÓN DE TÉCNICAS NO LINEALES Y OTROS PARADIGMAS EN SMART GRID/MICROGRID/VPP... 243</b>		
6.1.	<b>ESTUDIO DE LA RELACIÓN ENTRE LAS VARIABLES CLIMÁTICAS Y LA DEMANDA ELÉCTRICA EN EL MARCO DE SMART GRID/SMART WORLD.....</b>	244

6.1.1.	<i>Introducción al problema</i> .....	244
6.1.2.	<i>Sinergias entre modelos: Un nuevo concepto</i> .....	247
6.1.3.	<i>Análisis de datos, procesamiento de información y metodología</i> .....	248
6.1.4.	<i>Resultados</i> .....	255
6.1.5.	<i>Aplicaciones futuras de variables climáticas</i> .....	259
6.1.6.	<i>Conclusiones y estudios futuros</i> .....	263
6.2.	<b>CLASIFICACIÓN Y CLUSTERIZADO DE PATRONES DE DEMANDA ELÉCTRICA EN POPLÍGONOS INDUSTRIALES</b> .....	264
6.2.1.	<i>Introducción sobre el estudio</i> .....	264
6.2.2.	<i>Arquitectura del sistema</i> .....	265
6.2.3.	<i>Caso de estudio</i> .....	267
6.2.4.	<i>Resultados</i> .....	269
6.2.5.	<i>Conclusiones y estudios futuros</i> .....	275
6.3.	<b>SHORT-TERM LOAD FORECASTING PARA ENTORNOS DESAGREGADOS BASADO EN ARTIFICIAL NEURAL NETWORKS</b> .....	275
6.3.1.	<i>Introducción sobre el estudio</i> .....	275
6.3.2.	<i>Un modelo arquitectural para la predicción de carga en entornos desagregados</i> .....	278
6.3.3.	<i>Resultados</i> .....	282
6.3.4.	<i>Análisis de resultados</i> .....	286
6.3.5.	<i>Conclusiones y estudios futuros</i> .....	291
6.4.	<b>SHORT-TERM LOAD FORECASTING PARA ENTORNOS DESAGREGADOS BASADO EN ARTIFICIAL NEURAL NETWORKS. LA FUERZA DE LA CLASIFICACIÓN Y CLUSTERIZADO</b> .....	292
6.4.1.	<i>Introducción sobre el estudio</i> .....	292
6.4.2.	<i>Arquitectura del modelo</i> .....	293
6.4.3.	<i>Resultados</i> .....	295
6.4.4.	<i>Análisis de resultados</i> .....	299
6.4.5.	<i>Conclusiones y estudios futuros</i> .....	303
6.5.	<b>ANÁLISIS EXPERIMENTAL DE LA RELEVANCIA DE LAS VARIABLES DE ENTRADA PARA LA PREDICCIÓN DE LA DEMANDA ELÉCTRICA AGREGADA DEL DÍA SIGUIENTE POR MEDIO DE REDES NEURONALES ARTIFICIALES</b> .....	303
6.5.1.	<i>Introducción sobre el estudio</i> .....	303
6.5.2.	<i>Datos para la experiencia y metodología</i> .....	305
6.5.3.	<i>Análisis de las variables de relevancia para la producción de la demanda eléctrica agregada</i> .....	307
6.5.4.	<i>Modelos de predicción de la demanda agregada presentados</i> .....	312
6.5.5.	<i>Resultados</i> .....	313
6.5.6.	<i>Conclusiones y estudios futuros</i> .....	320



<b>6.6. SHORT-TERM LOAD FORECASTING PARA ENTORNOS DESAGREGADOS BASADO EN REDES NEURONALES ARTIFICIALES CON INFORMACIÓN DEL DÍA A PREDECIR.....</b>	<b>321</b>
6.6.1. <i>Introducción sobre el estudio .....</i>	322
6.6.2. <i>Descripción de los datos y marco metodológico.....</i>	324
6.6.3. <i>Arquitectura del modelo y evaluación del rendimiento del modelo ..</i>	325
6.6.4. <i>Resultados .....</i>	329
6.6.5. <i>Análisis de resultados .....</i>	335
6.6.6. <i>Conclusiones .....</i>	337
<b>6.7. UNA ARQUITECTURA BASADA EN SISTEMAS MULTI-AGENTES PARA LA GESTIÓN DE UNA SMART GRID Y LA PREDICCIÓN DE LA DEMANDA ELÉCTRICA EN VIRTUAL POWER PLANTS.....</b>	<b>338</b>
6.7.1. <i>Introducción.....</i>	338
6.7.2. <i>Virtual Power Plants.....</i>	340
6.7.3. <i>Tecnologías de la Información en el escenario Smart Grid.....</i>	341
6.7.4. <i>Modelo Arquitectural para la Gestión y Predicción .....</i>	344
6.7.5. <i>Conclusión .....</i>	351
<b>CONCLUSIONES GENERALES Y LÍNEAS FUTURAS .....</b>	<b>353</b>
<i>CONCLUSIONES GENERALES.....</i>	353
<i>VALIDACIÓN DE RESULTADOS.....</i>	358
<i>Publicaciones en revistas científicas .....</i>	359
<i>Proyectos en desarrollo .....</i>	360
<i>LÍNEAS FUTURAS.....</i>	364
<b>BIBLIOGRAFÍA .....</b>	<b>369</b>
<b>ABREVIATURAS .....</b>	<b>405</b>
<b>APÉNDICE.....</b>	<b>411</b>
<i>ANEXO A. CLASIFICACIÓN DE ANN.....</i>	411
<i>ANEXO B. ANN E INTERNET.....</i>	415
<i>ANEXO C. OUTLIERS DETECTADOS DEL CONJUNTO DE DATOS. DETALLES ESTADÍSTICO .....</i>	417
<i>ANEXO D. CLASIFICACIÓN Y CLUSTERIZADO DE PATRONES DE DEMANDA ELÉCTRICA EN PARQUES INDUSTRIALES. SECCIÓN 6.2. ....</i>	419
<i>ANEXO E. CLASIFICACIÓN Y CLUSTERIZADO DE PATRONES DE DEMANDA ELÉCTRICA EN MICROGRIDS. SECCIÓN 6.4. ....</i>	427



## ÍNDICE TABLAS

<i>Tabla 2.1. Previsión de la evolución de la potencia instalada de RE. Fuente Propia con Datos de la CNE.....</i>	<i>21</i>
<i>Tabla 2.2. Previsión de la evolución de la potencia instalada de RE antes de la entrada en vigor del Real Decreto-ley 1/2012 de 27 de enero. Fuente Propia con Datos de la CNE. ....</i>	<i>21</i>
<i>Tabla 3.1. Demanda eléctrica y tasa de variación anual desde 1982 a 1997. Fuente Propia con Datos de Rodríguez et al. (1998). ....</i>	<i>38</i>
<i>Tabla 4.1. Clasificación de los Sistemas, de acuerdo con diferentes criterios (Continúa). Fuente Propia con Datos de Arahal et al. (2002). ....</i>	<i>62</i>
<i>Tabla 4.2. Familias de modelos lineales paramétricos. Fuente Propia. ....</i>	<i>99</i>
<i>Tabla 4.3. Características del cerebro frente al computador convencional. Fuente Propia con Datos de Martín del Brío and Sanz (2006). ....</i>	<i>109</i>
<i>Tabla 4.4. Principales funciones de activación. Fuente Propia con Datos de Martín del Brío and Sanz (2006). ....</i>	<i>115</i>
<i>Tabla 4.5. Comparativa de los rasgos principales de los modelos lineales vs no lineales para la predicción de la demanda eléctrica. Fuente Propia.....</i>	<i>176</i>
<i>Tabla 4.6. L: carga; L+W: carga + climáticas; L+W+O: carga + climáticas + otras; L+O: carga + otras. MG: microgrid, SB: Smart Building. Fuente Propia.....</i>	<i>179</i>
<i>Tabla 4.7. Grupos formados, indicando su modelo de pertenencia así como las referencias que lo componen. Fuente Propia. (Continúa).....</i>	<i>180</i>
<i>Tabla 5.1. Componentes del mensaje FIPA ACL. Fuente Propia. ....</i>	<i>209</i>
<i>Tabla 5.2. Componentes reservados para un mensaje KQML. Fuente Propia. ....</i>	<i>209</i>
<i>Tabla 5.3. Protocolos de comunicación estándar FIPA. Fuente Propia. ....</i>	<i>211</i>
<i>Tabla 6.1. Columnas de la matriz de entrada a MatLab. Fuente Propia.....</i>	<i>253</i>
<i>Tabla 6.2. Correlación entre variables climáticas y el consumo de energía eléctrica. Fuente Propia.....</i>	<i>255</i>
<i>Tabla 6.3. Coeficientes de correlación estacionales entre las variables climáticas y demanda eléctrica. Fuente Propia. ....</i>	<i>259</i>
<i>Tabla 6.4. Análisis de los resultados por clúster. Fuente Propia. ....</i>	<i>273</i>
<i>Tabla 6.5. Distribución de errores por día. Fuente Propia.....</i>	<i>283</i>
<i>Tabla 6.6. Distribución de errores por horas. Fuente Propia. ....</i>	<i>285</i>
<i>Tabla 6.7. Relación entre el error de la fase de operación y el número de patrones de la fase de aprendizaje. Fuente Propia.....</i>	<i>289</i>
<i>Tabla 6.8. Variación del error entre las diferentes redes con respecto al número de patrones. Fuente Propia. ....</i>	<i>291</i>
<i>Tabla 6.9. Distribución de errores por días en cada clúster. Fuente Propia. ....</i>	<i>298</i>
<i>Tabla 6.10. Dependencia del error medio de la fase de operación con el número de patrones de la fase de aprendizaje. N.N.: número de la red; Ptt.: patrones; N.: número de neuronas; Error (%) error medio de la fase de operación. Fuente Propia.....</i>	<i>301</i>
<i>Tabla 6.11. Correlación entre demanda agregada y variables climáticas. Demanda Agregada (Aggregated Load –AL), Temperatura Media (Mean Temperature – MT), Humedad Relativa Media (Mean Relative Humidity – MRH), Precipitación Agregada (Aggregated Precipitation – AP), Radiación Solar Agregada (Aggregated Solar Radiation – ASR), Dirección de Viento Media (Mean Wind Direction – MWD), Velocidad de Viento Media (Mean Wind Speed – MWS) y Presión Media (Mean Pressure – MP). Fuente Propia. ....</i>	<i>311</i>
<i>Tabla 6.12. Comparativa de los MAPE mínimos entre los distintos modelos de predicción, tras 100 iteraciones de cada uno de ellos y cada función de aprendizaje. Siendo: (1) número de neuronas óptimo para modelo y función de aprendizaje; (2) MAPE (%) del número de neuronas óptimo; y (3) desviación estándar (%) del número de neuronas óptimo. Fuente Propia. (Continúa).....</i>	<i>314</i>

<i>Tabla 6.13. MAPE de los días de la fase de operación, agrupado por meses. Se muestra cada una de las redes de la primera etapa. Fuente Propia.</i>	329
<i>Tabla 6.14. Distribución de errores por día. Fuente Propia.</i>	331
<i>Tabla 6.15. Distribución de errores por hora. Fuente Propia.</i>	333
<i>Tabla 6.16. Tiempo empleado en la fase de aprendizaje y de operación para cada red. Fuente Propia.</i>	334
<i>Tabla 6.17. MAPE (%) de la fase de operación por día de la semana. Comparativa de ANN-STLF con los resultados de la sección 6.3. Fuente Propia.</i>	335
<i>Tabla 6.18. MAPE (%) de la fase de operación por meses. Comparativa de ANN-STLF con los resultados de la sección 6.3. Fuente Propia.</i>	336
<i>Tabla 6.19. MAPE (%) de los días de la sección 6.3 con error más elevado. Comparativa con ANN-STLF; al ser todos los días de 2010, la fecha se muestra con formato MM/DD. Fuente Propia.</i>	336
<i>Tabla A.1. Clasificación de las ANN monocapa. Fuente Propia con Datos de Hilera and Martínez (1995).</i>	411
<i>Tabla A.2. Clasificación de las ANN multicapa. Fuente Propia con Datos de Hilera and Martínez (1995). (Continúa).</i>	411
<i>Tabla A.3. Clasificación de las ANN con aprendizaje no supervisado. Fuente Propia con Datos de Flórez and Fernández (2008).</i>	412
<i>Tabla A.4. Clasificación de las ANN en función de la asociación. Fuente Propia con Datos de Flórez and Fernández (2008). (Continúa).</i>	412
<i>Tabla A.5. Clasificación de las ANN con aprendizaje supervisado. Fuente Propia con Datos de Flórez and Fernández (2008).</i>	413
<i>Tabla A.6. Clasificación de las ANN en función de la información de entrada y salida. Fuente Propia con Datos de Ponce (2011).</i>	414
<i>Tabla C.1. Outliers detectados con PCA. Estas curvas de carga no se emplearán en los diferentes conjuntos de datos. Fuente Propia.</i>	417

## ÍNDICE FIGURAS

<i>Figura 1.1. Sistema Eléctrico. Fuente propia.</i>	1
<i>Figura 2.1. a) Evolución de la capacidad eólica instalada en España. b) Evolución de la capacidad fotovoltaica instalada en España. c) Potencia total instalada en España. Fuente Propia.</i>	15
<i>Figura 2.2. Ubicación de las centrales nucleares y térmicas en España. Fuente Rodríguez et al. (1998).</i>	15
<i>Figura 2.3. Planificación de la TSO española para 2016. Cortesía de REE. Fuente Rodríguez et al. (1998).</i>	16
<i>Figura 2.4. Esquema de SG con elementos generadores, almacenadores y consumidores. Fuente Green Buildings Elements (2012).</i>	19
<i>Figura 2.5. Evolución en España de las diferentes tecnologías en RE. Fuente Propia con Datos de REE.</i>	20
<i>Figura 2.6. Subsistemas de VPP: a) CVPP; b) TVPP. Fuente Propia.</i>	25
<i>Figura 2.7. Recreación del concepto de microgrid. Fuente Propia.</i>	27
<i>Figura 2.8. Concepto futuro de Smart Place. Fuente Propia.</i>	33
<i>Figura 2.9. Concepto Smart World. Fuente Propia.</i>	35
<i>Figura 3.1. Evolución de la Demanda Eléctrica en el sistema peninsular desde 1982 a 1997. Fuente Propia con Datos de Rodríguez et al. (1998).</i>	38
<i>Figura 3.2. a) Curvas de carga de días de la semana de meses de verano; b) Curvas de carga de días de la semana de meses de invierno. Fuente Rodríguez et al. (1998).</i>	40
<i>Figura 3.3. Curvas de Carga del 01/03/2010 de diferentes entornos: a) Curva de Carga Diaria de España; b) Curva de Carga Diaria de ciudad española; c) Curva de Carga Diaria de Polígono Industrial; d) Curva de Carga Diaria de un Cliente Doméstico. Fuente Propia.</i>	51
<i>Figura 3.4. Clasificación de los modelos de predicción de la demanda en cuanto a su horizonte de predicción. Identificación del alcance de sus parámetros de entrada así como de su tipología. Fuente Propia.</i>	54
<i>Figura 3.5. Clasificación de los modelos de predicción de la demanda en cuanto a su objetivo. Se detalla el entorno de aplicaciones de los diferentes modelos. PL: carga pico (peak load); VL: carga valle (valley load); HL: carga de hora (hour load); HLs: carga de horas (hour loads); NDTL: carga total del día siguiente (next day total load); LP: curva de carga (load profile). Fuente Propia.</i>	56
<i>Figura 4.1. Algunos usos de los modelos atendiendo al ámbito de aplicación. Fuente Propia con Datos de Arahal et al. (2002).</i>	65
<i>Figura 4.2. Esquema de Validación y Verificación de modelos. Fuente Propia.</i>	75
<i>Figura 4.3. Cuadro de técnicas de optimización no lineal más empleadas. Fuente Propia.</i>	77
<i>Figura 4.4. Esquema general de técnica de optimización iterativa. Fuente Propia.</i>	78
<i>Figura 4.5. Esquema de los pasos de la identificación. Fuente Propia.</i>	82
<i>Figura 4.6. Esquema general del análisis del error durante la identificación. Fuente Propia con Datos de Arahal et al. (2002).</i>	93
<i>Figura 4.7. Noción geométrica de la sobreparametrización. Fuente Propia.</i>	94
<i>Figura 4.8. Comparativa de la evolución del error del modelo para los datos de entrenamiento (train), datos de validación (validation), datos de control (test) y localización del punto óptimo. Fuente Propia.</i>	95
<i>Figura 4.9. Diagrama de bloques de un modelo lineal paramétrico. Fuente Propia con Datos de Arahal et al. (2002).</i>	99
<i>Figura 4.10. Esquema de diagrama de bloques de la identificación mediante el método de mínimos cuadrados recurrente. Fuente Propia con Datos de Arahal et al. (2002).</i>	102
<i>Figura 4.11. a) Diagrama del modelo de Hammerstein; b) Diagrama de bloques de Wiener. Fuente Propia con Datos de Arahal et al. (2002).</i>	104

<i>Figura 4.12. Esquema general de una neurona biológica. Fuente Puig (2009).</i>	110
<i>Figura 4.13. a) Esquema general de una neurona artificial (Rumelhart and McClelland, 1986); b) Esquema de interacción entre neurona presináptica y postsináptica. Fuente Propia.</i>	112
<i>Figura 4.14. Esquema de modelo de neurona estándar. Fuente Propia.</i>	116
<i>Figura 4.15. Arquitectura unidireccional de tres capas: entrada, oculta y salida. Fuente Propia.</i>	118
<i>Figura 4.16. Ejemplos de arquitecturas neuronales: a) Multicapa y unidireccional; b) Monocapa y realimentada. Fuente Propia.</i>	119
<i>Figura 4.17. Clasificación de ANN en función de su aprendizaje y arquitectura. Fuente Propia.</i>	124
<i>Figura 4.18. a) Arquitectura del Asociador Lineal; b) Función de activación identidad. Fuente Propia.</i>	126
<i>Figura 4.19. a) Esquema de Perceptron simple; b) Función de transferencia de neurona Perceptron. Fuente Propia.</i>	130
<i>Figura 4.20. a) Función lógica NAND; b) Función lógica XOR. Fuente Propia.</i>	130
<i>Figura 4.21. Regla del Perceptron, cuando salida actual y objetivo no coinciden: a) Producto escalar positivo; b) Producto escalar negativo. Fuente Propia.</i>	133
<i>Figura 4.22. Regiones de decisión. Fuente Propia.</i>	133
<i>Figura 4.23. Neurona Adaline. Fuente Propia.</i>	134
<i>Figura 4.24. Superficie de error en el espacio de pesos y descenso por gradiente hacia mínimo local. Fuente Propia.</i>	136
<i>Figura 4.25. a) MLP; b) Función de transferencia de la neurona. Fuente Propia.</i>	137
<i>Figura 4.26. Arquitectura MLP. Fuente Propia.</i>	138
<i>Figura 4.27. Arquitectura SOM. Fuente Propia.</i>	145
<i>Figura 4.28. Esquema genérico del modelo Hopfield. Fuente Propia.</i>	153
<i>Figura 4.29. Neurona del modelo Hopfield. Fuente Propia.</i>	153
<i>Figura 4.30. Arquitectura RBF. Fuente Propia.</i>	157
<i>Figura 4.31. Resumen de los modelos lineales presentados en esta sección. SD: Spectral Decomposition; PLS: Partial Least-Square; ARIMA: Auto-Regressive Integrated Moving Average; ARCH: AutoRegressive Conditional Heteroscedasticity; AR: Auto-Regressive; ARMA: Auto-Regressive and Moving Average; MAM: Moving Average Model; LR: Linear Regression; SS: state-space. Fuente Propia.</i>	166
<i>Figura 4.32. Resumen de los modelos no lineales presentados en esta sección. NARX: Nonlinear AutoRegressive with eXogenous; PNN: Probabilistic Neural Network; RBF: Radial Basis Function; MLP: Multi-Layer Perceptron; GRNN: General Regression Neural Network; SVM: Support Vector Machine; SVR: Support Vector Regression; PCA: Principal Component Analysis; SOM: Self-Organizing Map; DA: Deterministic Annealing; FASE: Forecast-Aided State Estimator; AFCM: Adaptive Fuzzy Combination Model; CAS: Chaotic Ant Swarm; CABCC: Chaotic Artificial Bee Colony; CGASA: Chaotic Genetic Algorithm Simulated Annealing; CGA: Chaotic Genetic Algorithm; GA: Genetic Algorithm; CA: Chaotic Algorithm; FOA: Fly Optimization Algorithm. Fuente Propia.</i>	172
<i>Figura 5.1. Árbol de “mundos posibles”. Fuente Propia.</i>	195
<i>Figura 5.2. Arquitectura BDI. Fuente Propia.</i>	196
<i>Figura 5.3. Arquitectura subsunción. Fuente Propia.</i>	198
<i>Figura 5.4. Arquitectura TOURINGMACHINE. Fuente Propia.</i>	199
<i>Figura 5.5. Arquitectura INTERRAP. Fuente Propia.</i>	199
<i>Figura 5.6. Modelo FIPA. Fuente Propia.</i>	202
<i>Figura 5.7. a) Modelo RPC; b) Modelo agente móvil. Fuente Propia.</i>	204
<i>Figura 5.8. Tipos de paralelismo en la línea de vida. Fuente Propia.</i>	210
<i>Figura 5.9. Representación del diagrama de protocolo query. Fuente Propia.</i>	211
<i>Figura 5.10. Java y la independencia de la plataforma. Fuente Propia.</i>	214
<i>Figura 5.11. Modelo MAS-CommonKADS. Fuente (Iglesias, 1998; Cerrada et al., 2007).</i>	231

<i>Figura 5.12. Metodología INGENIAS, puntos de vista de un MAS. Fuente Propia.</i>	232
<i>Figura 5.13. Fases de MaSE. Fuente Juan et al. (2009).</i>	233
<i>Figura 5.14. Especificación AUML: protocolo de interacción FIPA Contract Net (FIPA, SC00029H). Fuente Propia.</i>	234
<i>Figura 5.15. INGENIAS: desarrollo iterativo de un módulo. Fuente Propia.</i>	237
<i>Figura 5.16. Especificación UML de arquitectura de agente ejemplo expresado según plataforma JADE. Fuente Propia.</i>	238
<i>Figura 5.17. Relación entre Mental State y las comunicaciones del agente. Fuente Propia.</i>	238
<i>Figura 5.18. Pantalla de control principal de plataforma de ejecución JADE. Fuente Mas (2005).</i>	239
<i>Figura 5.19. Editor ABLE para implementar agentes. Fuente Mas (2005).</i>	241
<i>Figura 6.1. Curvas de carga para los miércoles no festivos. Los miércoles son representados en verde. La curva de carga del día 30/06/2010 es representada en rojo, y la predicción para ese mismo día se representa en negro. Fuente Propia.</i>	249
<i>Figura 6.2. Esquema de procesamiento de la información. Fuente Propia.</i>	250
<i>Figura 6.3. a) PCA: componentes 8 y 9; b) Azul: 665, rojo: 897 y verde: 347. El número indica el número de patrón en el conjunto de datos. Fuente Propia.</i>	251
<i>Figura 6.4. Correlación a las 12h entre precipitación y presión, con el consumo eléctrico. Fuente Propia.</i>	256
<i>Figura 6.5. Correlación a las 12h entre velocidad media de viento y dirección media de viento, con el consumo eléctrico. Fuente Propia.</i>	256
<i>Figura 6.6. Correlación a las 12h de temperatura, radiación solar global y humedad relativa, con el consumo eléctrico. Fuente Propia.</i>	257
<i>Figura 6.7. Correlación a las 0h, 5h, 12h, 16h y 20h entre temperatura y consumo eléctrico. Fuente Propia.</i>	257
<i>Figura 6.8. Correlación a las 0h, 5h, 12h, 16h y 20h entre radiación solar global y consumo eléctrico. Fuente Propia.</i>	258
<i>Figura 6.9. Correlación a las 0h, 5h, 12h, 16h y 20h entre humedad relativa y consumo eléctrico. Fuente Propia.</i>	259
<i>Figura 6.10. Diseño MAS de una arquitectura para Smart World aprovechando datos climáticos. Fuente Propia.</i>	262
<i>Figura 6.11. Instancia de esquema de agentes en la simulación. Fuente Propia.</i>	262
<i>Figura 6.12. Configuración de la simulación para el escenario de Madrid. Fuente Propia.</i>	263
<i>Figura 6.13. Arquitectura del sistema. Fuente propia.</i>	265
<i>Figura 6.14. Arquitectura SOM. Fuente Propia.</i>	266
<i>Figura 6.15. a) Curvas de carga del polígono industrial “Las Casas”; b) Curvas de carga de un entorno con mayor agregación que a) (Soria capital). Las curvas verdes representan días laborables y las curvas rojas representan días no laborables (domingos y festivos entre semana). Fuente propia.</i>	268
<i>Figura 6.16. Mapa de activación según laborabilidad: a) días festivos; b) días laborables. Fuente Propia.</i>	270
<i>Figura 6.17. Mapa de activación según meses. Fuente Propia.</i>	271
<i>Figura 6.18. Mapa de activación según días de la semana. Fuente propia.</i>	271
<i>Figura 6.19. Mapa de activación según laborabilidad y día de la semana. Fuente Propia.</i>	272
<i>Figura 6.20. Curvas de carga de los cinco clústers. Las curvas rojas son días festivos, y las curvas verdes son días laborables. Fuente Propia.</i>	272
<i>Figura 6.21. a) Carga agregada media±desviación estándar (y-eje muestra los valores de consumo de energía en W y x-eje las 24 horas del día); b) box plot (y-eje muestra los valores de consumo de energía en W a través de los cinco clústers en x-eje). Fuente Propia.</i>	273
<i>Figura 6.22. Algoritmo de decisión. Fuente propia.</i>	274
<i>Figura 6.23. Arquitectura del sistema. Fuente Propia.</i>	278
<i>Figura 6.24. Operación On-Line del predictor. Fuente Propia.</i>	279

<i>Figura 6.25. Arquitectura MLP. Se muestran las variables de entrada y salida en la figura. Fuente Propia.</i>	280
<i>Figura 6.26. Errores por día. En x-eje son los días. En y-eje son los errores dados por (6.7). Fuente Propia.</i>	283
<i>Figura 6.27. Errores de la predicción de los datos. En x-eje son los errores dados por (6.7). En y-eje son las densidades de probabilidad. Fuente Propia.</i>	283
<i>Figura 6.28. Errores por horas. En x-eje son las horas del día. En y-eje son los errores dados por (6.9). Fuente Propia.</i>	284
<i>Figura 6.29. Errores de la predicción de los dato por horas. En x-eje son los errores dados por (6.9). En y-eje son las densidades de probabilidad. Fuente Propia.</i>	284
<i>Figura 6.30. Predicciones con errores medios bajos: a) 15/02/2010; b) 18/05/2010; c) 21/12/2010. Fuente Propia.</i>	285
<i>Figura 6.31. Errores por día agrupados por día de la semana. En x-eje son los errores dados por (6.7). En y-eje son cada uno de los días predichos. Fuente Propia.</i>	287
<i>Figura 6.32. Errores por día agrupados por meses. En x-eje son los errores dados por (6.7). En y-eje son cada uno de los días predichos del correspondiente mes. Fuente Propia.</i>	288
<i>Figura 6.33. Evolución del error medio de la fase de operación con respecto al número de patrones empleados en la fase de aprendizaje. La curva ajusta está en color rojo y la curva real en color azul. Fuente Propia.</i>	290
<i>Figura 6.34. Modelo general. Fuente Propia.</i>	293
<i>Figura 6.35. Arquitectura MLP para cada uno de los clústers. Fuente Propia.</i>	294
<i>Figura 6.36. Operación On–Line del predictor. Fuente Propia.</i>	295
<i>Figura 6.37. Mapa de activación para laborabilidad. El mapa representa el significado de cada neurona: número de neurona y fuerza entre neuronas. Fuente Propia.</i>	296
<i>Figura 6.38. Curvas de carga de los tres clústers: a) clúster 1; b) clúster 2; c) clúster 3. Fuente Propia.</i>	297
<i>Figura 6.39. Errores por día en clúster 1. Fuente Propia.</i>	297
<i>Figura 6.40. Errores por hora en clúster 3. Fuente Propia.</i>	298
<i>Figura 6.41. Predicciones con error elevado: a) clúster 1: 24/12/2010; b) clúster 2: 24/06/2010; c) clúster 3: 28/06/2010. Fuente Propia.</i>	299
<i>Figura 6.42. Predicciones con error bajo: a) clúster 1: 21/12/2010; b) clúster 2: 24/07/2010; c) clúster 3: 25/08/2010. Fuente Propia.</i>	299
<i>Figura 6.43. Errores por días: clúster 1. El eje de ordenadas muestra el error de predicción (por unidad), y el eje de abscisas representa el número de días predichos para el día de la semana representado. Fuente Propia.</i>	300
<i>Figura 6.44. Errores por días: clúster 2. El eje de ordenadas muestra el error de predicción (por unidad), y el eje de abscisas representa el número de días predichos para el día de la semana representado. Fuente Propia.</i>	300
<i>Figura 6.45. Errores por días: clúster 3. El eje de ordenadas muestra el error de predicción (por unidad), y el eje de abscisas representa el número de días predichos para el día de la semana representado. Fuente Propia.</i>	300
<i>Figura 6.46. Evolución del error medio de la fase de operación en función del número de patrones de la fase de aprendizaje: a) clúster 1 con 275 patrones; b) clúster 2 con 225 patrones; c) clúster 3 con 200 patrones; d) MLP sección 6.3 con 700 patrones. Fuente Propia.</i>	302
<i>Figura 6.47. Coeficientes de auto–correlación de cada uno de los días de la semana frente al resto de patrones, y separado por día de la semana. Fuente Propia.</i>	308
<i>Figura 6.48. Coeficientes medios de auto–correlación entre valores de demanda agregada según día de la semana. Fuente Propia.</i>	309
<i>Figura 6.49. Regresión lineal de la demanda agregada frente a la temperatura media, humedad relativa y radiación solar. En el eje de las y demanda agregada en kW y en el eje de las x °C, % humedad relativa y W/m<sup>2</sup>. Se indica el valor del coeficiente de correlación para cada variable climática frente a demanda agregada. Fuente Propia.</i>	311



<i>Figura 6.50. Evolución de MAPE (%) de la fase de operación de cada uno de los modelos, para cada una de las funciones de entrenamiento y número de neuronas variables. En la gráfica se indica el valor de MAPE (%) y neuronas óptimas para cada modelo. Fuente Propia.....</i>	<i>317</i>
<i>Figura 6.51. Evolución de MAPE (%) de todos los días de la fase de operación de cada uno de los modelos, para trainbr y número de neuronas óptimo. Se indica valor MAPE (%), RMSE (adimensional) y ME (W) de la fase de operación de cada modelo. Fuente Propia.....</i>	<i>318</i>
<i>Figura 6.52. Evolución de MAPE (%), RMSE (adimensional) y ME (W) de todos los días de la fase de operación de cada uno de los modelos por meses. Fuente Propia.....</i>	<i>320</i>
<i>Figura 6.53. Diagrama de bloques de la metodología propuesta. Fuente Propia.....</i>	<i>324</i>
<i>Figura 6.54. Errores por día en la fase de operación. En x-eje se representa los días. En y-eje son los errores por (6.7). Fuente Propia. ....</i>	<i>330</i>
<i>Figura 6.55. Datos de los errores de las curvas de carga en la fase de operación. En x-eje son los errores por (6.8). En y-eje son las densidades de probabilidad. Fuente Propia. ....</i>	<i>331</i>
<i>Figura 6.56. Errores por hora en la fase de operación. En x-eje son las 24 horas. En y-eje son los errores por (6.9). Fuente Propia.....</i>	<i>332</i>
<i>Figura 6.57. Datos de los errores por hora de las curvas de carga en la fase de operación. En x-eje son los errores por (6.9). En y-eje son las densidades de probabilidad. Fuente Propia. ....</i>	<i>332</i>
<i>Figura 6.58. Predicciones con error medio bajo: a) 08/02/2010; b) 23/03/2010; c) 16/12/2010. Fuente Propia.....</i>	<i>333</i>
<i>Figura 6.59. Predicción de 02/04/2010: a) con sección 6.3; b) con ANN-STLF. Fuente Propia. ....</i>	<i>337</i>
<i>Figura 6.60. Elementos de una VPP. Fuente Propia. ....</i>	<i>341</i>
<i>Figura 6.61. Inteligencia Distribuida a través de una VPP. Fuente Propia. ....</i>	<i>342</i>
<i>Figura 6.62. Modelo MAS del sistema de gestión y predicción: a) Notación INGENIAS; b) Actores; c) Interacción entre agentes. Fuente Propia. ....</i>	<i>345</i>
<i>Figura 6.63. Ejecución de la especificación MAS con INGENIAS. Fuente Propia.....</i>	<i>346</i>
<i>Figura 6.64. a) Sistema completo; b) Representación de las tres etapas; c) Entradas y salidas de los modelos MLP. Fuente Propia. ....</i>	<i>348</i>
<i>Figura 6.65. Resultados de predicción de curva de carga: a) Clusterizado de las curvas de carga (cada color es un clústers); b) Ejemplo de predicción de curva de carga: curva objetivo en negro, predicción en rojo y otras curvas del mismo grupo/clúster en verde. Fuente Propia. ....</i>	<i>350</i>



## CAPÍTULO I. INTRODUCCIÓN

En el Capítulo I se hará una breve una introducción del sistema eléctrico, ampliándose en el Capítulo II, para poder identificar los elementos motivaciones de la *Tesis Doctoral*, posteriormente detallar los objetivos que se pretenden alcanzar, enunciar las contribuciones originales y explicar la metodología empleada.

### 1.1. INTRODUCCIÓN

En la actualidad, y desde ya hace muchísimo tiempo, la manera en que se lleva la electricidad desde los orígenes de generación hasta los lugares de su consumo ha permanecido prácticamente invariable (desde el punto de vista de modelo), a esto hay que añadirle el actual modelo centralizado y muy dependiente de ciertos parámetros que pueden producir en un momento determinado la desestabilización de todo el sistema. A modo de resumen, se puede indicar que el actual sistema eléctrico, se compone por la generación, el transporte, la distribución y el consumo, como se puede observar en la Figura 1.1, como se presenta en los últimos trabajos (Ipakchu and Albuyeh, 2009; Carillo, 2010).

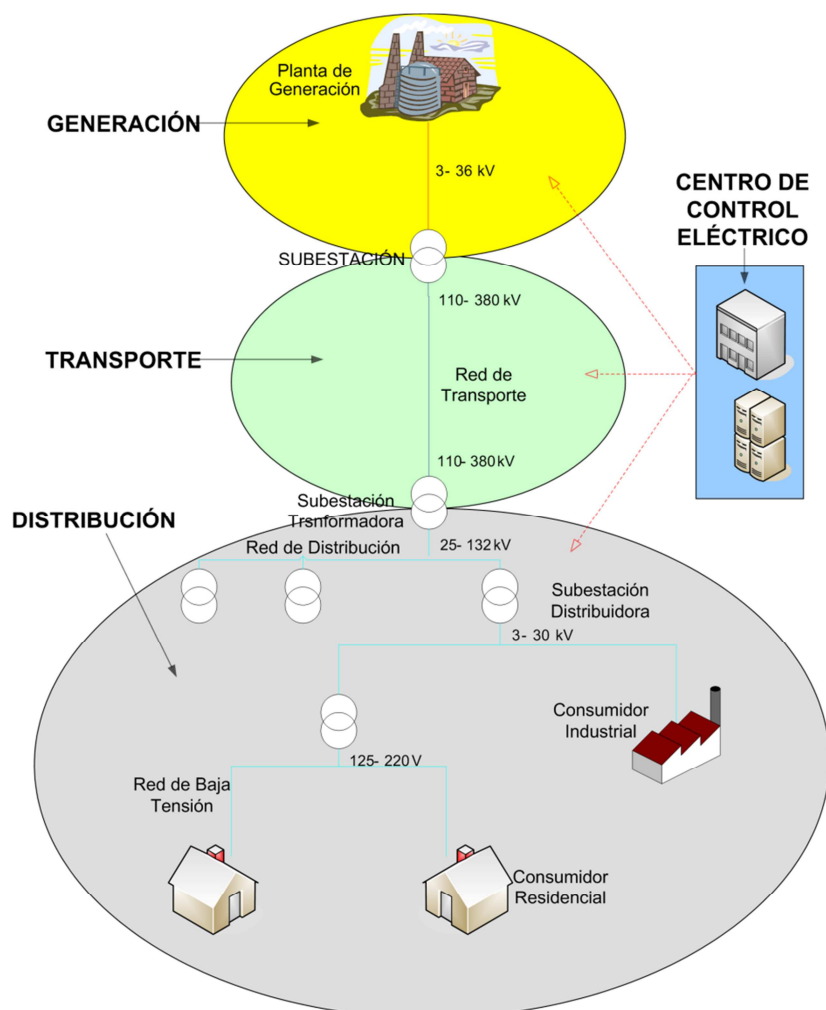


Figura 1.1. Sistema Eléctrico. Fuente propia.

En la Figura 1.1, deberíamos incluir la cada vez más creciente generación de electricidad a través de energías renovables (eólica, fotovoltaica, solar térmica, biomasa, geotérmica, etc.) principalmente en la etapa de distribución, como refleja Nestle (2009). Con este escenario planteado, se dispone de una infraestructura de red eléctrica poco flexible, antigua y que está empezando a cambiar (al menos con la intención de ello) acorde con la realidad del momento en que vivimos, como se muestra en (Al-Hader *et al.*, 2009; Kok *et al.*, 2009; Lee *et al.*, 2010; Verschueren *et al.*, 2010).

En un polo opuesto al sistema energético de electricidad comentado se encuentra el mundo de la *Tecnología de la Información y Comunicación (TIC)*, el cual está en constante evolución, muchas veces por demandas de la sociedad y otras por las propias empresas *TIC* para generar un servicio requerido o no por los ciudadanos u otros agentes del escenario económico. Lo que parece más que evidente, es que el mundo de la electricidad debería evolucionar al igual que las *TIC*, seguramente no a pasos tan agigantados debido a barreras como por ejemplo el tener que tratar con un suministro “*continuo*” frente a uno “*discreto*” que se suele emplear en las *TIC*, pero lo claro es que tanto el momento actual empresarial como la más que evidente necesidad medio ambiental, va a demandar esta actualización del mundo eléctrico, y en este punto es donde las *TIC* puede mezclarse con él para hacer actualizaciones del sistema para por ejemplo dejar de ser tan poco flexible y centralizado, como se indica en (Kim *et al.*, 2010; Laverty *et al.*, 2010; Liu, 2010). Adicionalmente, comentar que están surgiendo constantemente nuevos planteamientos de negocio y diseño, donde tienen mucho que decir la realidad de las fuentes de generación renovables y que bajo ningún concepto deben de negarse u obviar.

El mencionado despliegue de elementos *TIC*, y de determinada inteligencia en entornos eléctricos tradicionales, donde tan sólo había elementos electromecánicos, hace que aparezcan nuevos paradigmas que precisarán del desarrollo de nuevos modelos de control, planificación, monitorización y de información. De esta manera, aparecen los conceptos de *SG*, *microgrid* y *VPP*, como integradores de inteligencia para permitir la evolución del sistema de energía eléctrica.

Para finalizar esta introducción, debemos mencionar un elemento importantísimo en todo este sistema, que no es otro que los consumidores, y en concreto los hogares. Hasta la fecha los hogares de los ciudadanos no pasaban de ser un punto de consumo, sin apenas conciencia y poder de decisión, desde hace un tiempo se está pensando en dotarles de cierta información para que se involucren en todo el sistema, pensando en su propio ahorro, en formar una conciencia social medio ambiental y en la toma de decisiones que suponga incluso el desplazamiento de ciertas cargas de consumo de energía a otros horarios sin entorpecer el confort de los habitantes del hogar. Surge el concepto de *Smart Home*, introducido en (Leong *et al.*, 2009; Jahn *et al.*, 2010; Matsumoto, 2010); que en definitiva no es más que el facilitar de cierta inteligencia a los hogares de los ciudadanos, lo

cual va a permitir ciertos cambios positivos en la anteriormente comentada infraestructura del sistema de energía eléctrica, además del ya referido ahorro de energía.

## 1.2. MOTIVACIÓN

El empleo de técnicas de predicción ha experimentado en los últimos tiempos una proliferación considerable, principalmente por la mayor disponibilidad y potencia de los sistemas de cálculo automático, y en concreto por el uso de las computadoras personales. Además del cambio cuantitativo, se ha producido un cambio en la tipología de los modelos planteados y usados. En la década de 1970, el empleo de técnicas no lineales era una cuestión poco aceptada por los científicos e ingenieros, en cambio, en los últimos veinte años se han convertido en técnicas muy acertadas para la solución de problemas complejos y difícilmente abordables de otra manera. En los últimos años y a partir del concepto *SG*, han ido surgiendo nuevos entornos de futuro capaces de integrar demanda, generación y almacenamiento.

A continuación se detallarán los elementos de motivación en I+D+i que han sido seleccionados para formalizar los objetivos a conseguir en la *Tesis Doctoral*:

- Necesidad de actualización de la actual infraestructura del sistema energético de electricidad, permitiendo así una más alta penetración de fuentes de generación renovable a través de Generación Distribuida (*Distributed Generation – DG*), entre otras cuestiones. A partir de esta necesidad, surgen nuevos paradigmas como: *SG*, presentado en (Massoud and Wollenberg, 2005; Sheblé, 2008); *microgrid*, definido en (Li *et al.*, 2004; Hernández–Aramburo *et al.*, 2005); y *VPP*, concretado en (Pudjianto *et al.*, 2007; Ruiz *et al.*, 2009; Vukmirovic *et al.*, 2010); tanto la industria, las administraciones públicas, así como la cada vez mayor conciencia social y medio ambiental, demandan la aparición de un potencial mayor de fuentes de generación renovables, en cualquier punto del sistema, por lo que dicha infraestructura deberá tender a la descentralización y a un cambio de modelo, que conllevará entre otros cometidos el desplazamiento de algunas de las responsabilidades actuales más cerca de los lugares de consumo. Por ejemplo, estas nuevas *VPP* se convierten en entornos virtuales descentralizados de generación eléctrica con fuentes renovables, almacenamiento propio, Respuesta a la Demanda (*Demand Response – DR*); como indican Rahimi and Ipakchi (2010)) y consumos.
- Necesidad de mejora de algoritmos de predicción de la demanda de energía eléctrica a partir de datos históricos en entornos de *micorgrids* y *Smart Home*. Uno de las necesidades para una correcta operación en las *microgrids*, es disponer de una estimación lo más exacta posible de la demanda, ya que sin ella, no se podrá planificar sus recursos renovables y anticipar el almacenamiento necesario. Descendiendo hasta *Smart Home*, se está intentando

integrar a los consumidores domésticos en la red, de manera que dispongan de un protagonismo superior al actual, además de poder disponer de aplicaciones que ayuden a tener mayor calidad en sus vidas, como se refleja en (Gopalratnam and Cook, 2007; Rashidi and Cook, 2009; Rashidi *et al.*, 2010).

- Necesidad de comunicación bidireccional entre los consumidores finales y la red, bien sea en una *SG*, una *microgrid* o una *VPP*. Será necesaria la instalación de Medidor Inteligente (*Smart Meter – SM*) en los puntos finales de consumo para romper la forma clásica de comunicación, recogido en (Deconinck and Decroix, 2009; Godfrey and Rodine, 2010; Zhou *et al.*, 2010).
- Necesidad de propuesta de arquitectura de obtención de datos desde *Smart Home* hasta el control de *VPP*, abordado en (Choi *et al.*, 2009), así como una nueva arquitectura para la predicción de la demanda en entornos de *VPP*.

### **1.3. OBJETIVOS**

El objetivo principal de la *Tesis Doctoral* será doble, por un lado se diseñarán algoritmos de predicción de la demanda eléctrica en entornos desagregados (ciudad pequeña, entorno rural o *microgrids*) a partir de datos históricos, y por otro lado se planteará una arquitectura, basada en *MAS*, para la predicción de demanda eléctrica, orientada a *Smart Home* en las *VPP*.

Pasaremos a detallar los objetivos derivados del objetivo principal que se pretenden conseguir con esta *Tesis Doctoral*:

- Modernizar la infraestructura del actual sistema energético de electricidad, produciéndose un cambio de imagen de algunos de sus elementos, como por ejemplo subestaciones y centros de transformación, al dotarlos de elementos con inteligencia, propiciando un aumento en sus funcionalidades, así como un cambio en su imagen.
- Potenciar el empleo de las *microgrids*, como lugares para el despliegue de las fuentes de generación renovable, por medio de la *DG*, así como el almacenamiento distribuido. Además, el control y coordinación de las *microgrids*, ayudará a una mejor coordinación de las *SG*, así como un aumento de la eficiencia energética del sistema, al disponer de energía eléctrica en el lugar donde se consume, necesitando de menos energía en transporte y distribución, disminuyendo de esta manera las pérdidas asociadas a estas dos etapas.
- Con la arquitectura planteada, se pretende involucrar a *Smart Home*, teniendo en cuenta el perfil de los consumidores domésticos en la *VPP*, y participando por tanto en la toma de decisiones en la misma. Se planteará un diseño conceptual en la arquitectura óptima para el despliegue de Concentradores de Datos (*Data Concentrators – DCs*), servidores, bases de datos y elementos de comunicación.

## 1.4. CONTRIBUCIONES ORIGINALES

A continuación se van a resumir las contribuciones originales que se plantean y abordan en esta *Tesis Doctoral*, y que son los siguientes:

- Diseño y validación de diferentes modelos de predicción de la demanda eléctrica, en entornos desagregados (ciudad pequeña, entorno rural o *microgrids*), basados en ANN. Hasta la fecha, muy pocos son los trabajos en predicción de la demanda eléctrica en entornos desagregados, por lo que se ha considerado de especial interés el potenciar estos ámbitos, ya que serán una oportunidad para las fuentes de generación renovable a pequeña escala, no habiéndose producido en estos momentos su despegue.
- Realizar un cambio progresivo de la arquitectura vetusta de los elementos del actual sistema energético de electricidad, para lo cual se plantea la incorporación de dispositivos TIC e inteligencia, asociados a los sistemas de potencia ya existentes que propiciarán que estos espacios (centros de transformación y subestaciones) se conviertan en una oportunidad para albergar funcionalidades diferentes que las existentes en este momento, y adecuarlos al siglo XXI, estando en sintonía con el relativamente nuevo y emergente paradigma de SG.
- La creación de VPP supondrá la no dependencia total de generación de energía eléctrica procedente de no renovables, al tratar de igualar la demanda predicha con la generación a través de las propias fuentes de generación renovable y su almacenamiento existente de la propia VPP.
- Cada VPP representa un modelo descentralizado del sistema de control actual, tratando de predecir entre otras muchas cosas la posible demanda de energía eléctrica bajo su control. Se planteará la creación de un Control de Planta Virtual de Energía (*Virtual Power Plant Control – VPPC*), como coordinador de todas las labores existentes en la VPP, entre ellas la predicción de la demanda eléctrica de *Smart Homes*. El control de los procesos internos y externos que se originan en la VPP, se propone diseñarlos según un modelo MAS, como los ya abordados mediante simulación en SG (Karnouskos and de Holanda, 2009).

La totalidad de las contribuciones serán ampliadas y detalladas en el Capítulo VI. Estas contribuciones serán expuestas en siete secciones dentro de dicho capítulo.

## 1.5. METODOLOGÍA

A continuación se presenta la metodología llevada para la consecución de la *Tesis Doctoral*, que aunque se resuma en unas cuantas líneas, no debe de obviarse el complejo trabajo que hay detrás.

1. Contacto inicial con miembros del grupo de investigación del Departamento de Teoría de la Señal y Comunicaciones e Ingeniería Telemática de la Escuela Técnica Superior de

- Ingenieros de Telecomunicación de la Universidad de Valladolid. Se les plantea hacer una *Tesis Doctoral* en el ámbito de la Energía, pero visto desde el plano de la algoritmia y sus aplicaciones, sirviéndole a dicho grupo de investigación para ampliar sus líneas de I+D+i en este campo. El contacto se produce en septiembre de 2010.
2. Se inicia por parte del Doctorando una fase de investigación para detectar potenciales campos. Dicha fase consiste en la búsqueda de publicaciones científicas en el ámbito de la energía, detectando la novedosa línea de investigación desde hace algunos años sobre *SG*, que además, está siendo potenciado desde la *Unión Europea (UE)*. Se comienza en octubre de 2010 y finaliza en enero de 2011.
  3. Se profundiza en la temática de las *SG*, descubriendo los últimos trabajos en relación con las *microgrids*, y viendo la necesidad de trabajar en nuevos modelos de predicción de la demanda eléctrica, que en la literatura se habían empleado para el sistema eléctrico, desde la perspectiva de un país o grandes extensiones de países. Esta fase comienza en febrero de 2011 y termina a finales de marzo del mismo año.
  4. Adicionalmente, como consecuencia de las etapas 2 y 3, se descubre la existencia de *MAS*, e identificando su potencialidad en aplicaciones de *SG/Microgrid/VPP*, ya que requieren de soluciones complejas en entornos heterogéneos, siendo características deseadas para los *MAS*.
  5. A finales de marzo de 2011 se plantea al equipo de tutores el trabajar en los modelos de predicción de la demanda eléctrica en entornos desagregados (ciudad pequeña, entorno rural o *microgrids*), posibilitando de esta manera el despegue que estos lugares tendrán de manera inminente. Se detecta la figura de los *agregadores* como responsable del control y suministro eléctrico de pequeñas ciudades, pueblos, entornos industriales, campus universitarios, etc.
  6. A la par que lo planteado en 5, se propone definir una arquitectura para la gestión de la predicción de la demanda eléctrica para *Smart Homes* dentro de una *VPP*. La propuesta se plantea desde el plano teórico.
  7. Contando con la colaboración de *CIEMAT*, se solicitan datos a la *utility Iberdrola* y a la *Agencia Estatal de Meteorología (AEMET)*, para así disponer de un histórico de datos con los que poder establecer diferentes modelos de predicción. Los datos obtenidos (tres años), son de la pequeña capital de provincia de Soria, pudiéndose asemejar a un entorno acotado y desagregado (ciudad pequeña, entorno rural o *microgrids*), y por tanto sirviendo para las intenciones de la *Tesis Doctoral*. Los datos están disponibles en julio de 2011.



8. Durante el segundo semestre de 2011 se trabaja en modelo arquitectural teórico, para plantear el control de la predicción de la demanda eléctrica para *Smart Homes* en *VPP*. Esta fase finaliza en diciembre de 2011.
9. En paralelo con 8, y desde julio de 2011, se comienzan a diseñar los diferentes modelos conceptuales de predicción de la demanda eléctrica para un entorno desagregado (ciudad pequeña, entorno rural o *microgrids*). Esta etapa finaliza en diciembre de 2011.
10. Los primeros modelos de 9 son validados con datos a partir de septiembre de 2011, extendiéndose hasta septiembre de 2012, que se validan los últimos modelos.
11. Coincidiendo con los inicios de la fase 9, se realiza un estudio de la dependencia de las variables climáticas y la demanda eléctrica, considerándose de gran importancia para el devenir de los entornos desagregados (ciudad pequeña, entorno rural o *microgrids*), y en especial para el entendimiento de ciertos comportamientos de la demanda en los mismos.
12. Tras los primeros resultados satisfactorios, se comienza con la validación de los trabajos a través de Revistas (*journals*) de carácter científico, intentando que el nivel de las mismas sea elevado. Esta fase comienza en febrero de 2012 y se extiende durante todo el año 2013.
13. En octubre de 2012 se acuerda con el equipo de tutores que durante ese curso académico (2012–2013) se presentará la *Tesis Doctoral*.
14. La redacción de la memoria de la *Tesis Doctoral* comienza en diciembre de 2012.

## **1.6. ESTRUCTURA DE LA TESIS**

A continuación se detallarán los capítulos que componen esta *Tesis Doctoral*. En el Capítulo I se mostrará una introducción del problema, se indicarán los elementos motivaciones detectados por el autor, se presentarán los objetivos que se pretenden alcanzar, para posteriormente detallar las contribuciones originales y presentar la metodología empleada.

En el Capítulo II se mostrarán los detalles del sistema eléctrico clásico, mostrando las características de su diseño, pudiendo identificar potenciales limitaciones y vislumbrando posibilidades de mejora. Se continuará relatando la evolución de dicho sistema en los últimos años, promovido principalmente por el área de I+D+i de la industria privada junto a la investigación pública.

Con el Capítulo III se pretende hacer un estudio de la demanda eléctrica, contemplada desde la visión del sistema clásico, pero sin perder de vista el horizonte que se presenta, caracterizado por la masiva información en lugares donde hasta la fecha no existía. Se identificará la incertidumbre que rodea a la demanda eléctrica, así como el papel que juegan los consumidores en estos momentos. Desde la perspectiva clásica, se mostrará la necesidad de disponer de una estimación de

dicha demanda, con la dificultad que ello conlleva, presentándose nuevas oportunidades para los modelos de predicción, al aparecer nuevos entornos al evolucionar el sistema eléctrico.

Antes de mostrar las contribuciones en el Capítulo VI de la *Tesis Doctoral*, en el Capítulo IV se presentarán los principios de las predicciones, introduciendo los modelos basados en leyes, el arte de la identificación de sistemas, la predicción con modelos lineales y para terminar, los modelos no lineales, que serán la base de lo presentado en la *Tesis Doctoral*.

De la misma forma, en el Capítulo VI se presentará un nuevo modelo teórico–conceptual basado en *MAS* de la predicción de la demanda eléctrica. Para llegar a su comprensión, en el Capítulo V se presenta este paradigma de *MAS*, como alternativa a los sistemas clásicos y su posible aplicación en entornos complejos y heterogéneos, como pueda ser el mundo eléctrico. Se dará la visión de un Agente, arquitecturas de los mismos, su comunicación y coordinación, y para terminar las herramientas y metodologías empleadas.

Una vez desarrollados los capítulos que conforman la base de la *Tesis Doctoral*, en el Capítulo VI se presentarán las aportaciones de la misma, que se centran en la aplicación de técnicas no lineales y otros paradigmas en *SG/Microgrid/VPP*). Se comenzará con el estudio de las variables climáticas y su influencia en la demanda eléctrica en un entorno desagregado (ciudad pequeña, *microgrid* o similar), y de esta manera, permitir el desarrollo de nuevas aplicaciones alrededor de estos entornos u otros como la *SG/SC*, definiendo el concepto de *Smart World* como aglutinador de todos los escenarios que tienen una serie de características comunes entre sí, y que serán expuestas. A continuación, y mediante técnicas de clasificación y clusterizado, se presentará un algoritmo decisor para la predicción de la demanda eléctrica del día siguiente de un polígono industrial (*microgrid*) y poder ser usado por los agregadores, nueva figura que está comenzando a potenciar. Seguidamente, se presentará un modelo basado en técnicas no lineales para la predicción de la demanda eléctrica a corto plazo (*STLF*) de la curva de carga del día siguiente (24 valores de consumo, cada uno corresponde a cada una de las horas del día) en un entorno desagregado (ciudad pequeña, *microgrid* o similar). Para mejorar el modelo anterior, se diseñará un modelo para *STLF*, pero basado en la fuerza de la clasificación y clusterizado. Lo anterior se basó en la curva de carga del día siguiente, se mostrará un análisis detallado de las variables relacionadas con la demanda eléctrica agregada del día siguiente en un entorno desagregado (ciudad pequeña, *microgrid* o similar, y se presentará la evolución de varios modelos diseñados, y su mejora, para la consecución de tal fin. Se dará un paso más, mejorando los anteriores modelos para *STLF* en un entorno desagregado (ciudad pequeña, *microgrid* o similar), aprovechando la forma característica de la curva de carga. Para finalizar, se diseñará una arquitectura conceptual basada en *MAS*, para la gestión de una *SG*, en concreto, para la coordinación y control de la predicción de la demanda eléctrica en una *VPP* de futuro.

Para finalizar, se presentarán las conclusiones generales y las líneas futuras. Las conclusiones tratarán de identificar las principales aportaciones que se han realizado en la *Tesis Doctoral*, las cuales serán presentadas en el Capítulo VI, además, se listarán las publicaciones en revistas científicas donde se han validado los resultados de la *Tesis Doctoral*. Se presentarán posibles líneas futuras de I+D+i alrededor de los resultados alcanzados.

La lista de referencias bibliográficas se presentan en orden alfabético, durante el desarrollo del texto se hará referencia a las mismas mediante el nombre/s de/l autor/es. Se entrega un conjunto de Anexos para completar los capítulos teóricos y las aportaciones presentadas en el Capítulo VI.

Con respecto al “*estado del arte*” que compete a la temática presentada en la *Tesis Doctoral* decir, que debido a que los estudios relacionados con el planteamiento presentado son extensos y pueden enmarcarse en diferentes técnicas y/o áreas, se ha considerado el ir presentando los diferentes antecedentes relacionados con el trabajo desarrollado en la *Tesis Doctoral* de manera distribuida a lo largo de la estructura de la memoria. En concreto, el Capítulo II presenta el sistema eléctrico, donde en su parte final presenta los nuevos escenarios de futuro que seguirán precisando el disponer de modelos predictivos de la demanda y la generación eléctrica. En el Capítulo III, tras una exposición de conceptos relativos a la demanda eléctrica, se muestran los antecedentes de la necesidad de la predicción de la demanda eléctrica en la sección 3.4; posteriormente, se pone de manifiesto la dificultad de la predicción de la demanda en los entornos de futuro (*microgrids*, *Smart Buildings*, etc.); y para terminar, se presentarán las diferentes clasificaciones posibles de la predicción de la demanda eléctrica, a partir de los antecedentes encontrados en la literatura científica. El Capítulo IV presenta las principales técnicas y principios de los modelos empleados en la predicción de la demanda eléctrica, tras dicha exposición, se muestran los trabajos pasados en torno a la predicción de la demanda eléctrica, realizando igualmente una comparativa entre ellos, viendo claramente como existen pocas obras que tratan con datos de entornos desagregados (ciudades, pueblos, *microgrids*, *Smart Buildings*, etc.).

La estructura escogida para el desarrollo de la *Tesis Doctoral* permite un seguimiento claro y preciso de los objetivos planteados en la misma. En el Capítulo VI concretamente, el lector podrá seguir mediante un hilo conductor continuo, las mejoras que se van planteando en torno al problema de *STLF*. Partiendo de un modelo base, se irán planteando mejoras para dar una solución más exacta al problema, y además, se plantearán las relaciones de la demanda eléctrica con las variables climáticas, las cuales serán empleadas para abordar el problema de *STLF* y la predicción de la demanda agregada del día siguiente. Como solución de futuro a la problemática de la predicción de la demanda, se planteará un sistema de control distribuido que coopere para disponer de un control/operación acorde a las redes del futuro.



## CAPÍTULO II. EL SISTEMA ELÉCTRICO

En el Capítulo II se presentará el sistema eléctrico, realizando una visión retrospectiva en el tiempo para llegar a nuestros días, presentando los últimos avances desde la I+D+i, y que han propiciado la aparición de nuevos paradigmas alrededor del sistema eléctrico, y que permitirá el desarrollo de modelos y aplicaciones, apoyándose en diferentes ámbitos de la investigación.

### 2.1. EL SISTEMA ELÉCTRICO: ORÍGENES

Hasta finales de 1700, la electricidad era un enigma y se empleaba más como espectáculo para el divertimento que para otra cosa. Era la época de las primeras experiencias de Benjamin Franklin (1706–1790) y sus experimentos con rayos. Sin embargo, hasta que no irrumpió en el mundo científico Alessandro Volta (1745–1827), no se comenzó la evolución de la electricidad con cierto rigor matemático, y su desarrollo de la pila eléctrica.

Posteriormente, aparecen André-Marie Ampère (1775–1836) y Heinrich Lenz (1804–1865), con sus dos famosas Leyes. Estos dos científicos comienzan a investigar imanes, espiras y se empieza a desarrollar la teoría del Electro-Magnetismo. Entre 1789 y 1854 Georg Simon Ohm, físico y matemático alemán, formuló la Ley de Ohm, que se enseña hoy en día desde muy temprana edad en las escuelas de medio mundo.

Los avances del siglo XIX desembocaron en el nacimiento de las primeras centrales eléctricas a finales de ese mismo siglo. Estas centrales fueron al principio hidroeléctricas, ya que era la fuente de energía que conocían. En 1879 Edison fabricó en su laboratorio una lámpara con filamento de carbón, con una vida útil aproximada de 40 horas, era la primera bombilla incandescente. Y la primera central comercial se creó en Appleton (Estados Unidos) y que alimentaba unas 250 bombillas del tipo anterior.

En 1890 se instalaron algunas centrales en Europa, cercanas a los ríos, estando algunas en la actualidad trabajando, como la de Jaruga (Croacia). La característica de estas primeras centrales, era que su origen era en forma de corriente continua, ya que no se conocía ninguna otra forma de energía, y una segunda característica era que estas centrales abastecían zonas pequeñas y muy cercanas.

Sin embargo, llegó un momento en que se producía más electricidad de la que se consumía, e irrumpe uno de los héroes de la ciencia más habituales, y todo sea dicho, menos reconocido, Nikola Tesla (1856–1943), quien en 1888 con su magistral publicación “*A New System of Alternating Current Motors and Transformers*” (Tesla, 1888), presentaba la aplicación de la corriente alterna para el uso de motores, y el empleo de los transformadores eléctricos, sentando las bases del sistema de transporte eléctrico tal y como lo conocemos hoy. Esto permitiría distribuir la electricidad a

distancias más largas, y en definitiva poder vender energía eléctrica a más personas. En este momento, se concibió la idea de no construir las centrales en los núcleos de consumo.

La primera central eléctrica española para suministro a los consumidores se instaló en Barcelona, también en 1879. En 1881 entró en servicio la primera central madrileña, que se utilizó para el alumbrado público, entre otras aplicaciones (telegrafía, motores eléctricos, electrolisis, etc.). La aparición de la corriente alterna, y que se manejaban cantidades económicas inhabituales, propiciaron las grandes alianzas entre compañías, de hecho, la empresa Hidrocantábrico guarda el nombre de esta época. Como consecuencia del aumento de las redes (transporte principalmente), hubo que regular el mercado, tanto técnica como económicamente, originándose por tanto las primeras normas eléctricas.

La 1ª Guerra Mundial destruye la mayoría de las centrales instaladas y en operación. A pesar de la no participación de España en la guerra, tampoco se realizan grandes esfuerzos en evolucionar las instalaciones, apenas existía red de distribución en el país. A pesar de eso, se continuaba percibiendo el negocio alrededor de la electricidad, y por eso para 1936 ya había empresas privadas dedicadas a tal negocio.

Sin embargo, lo que sí que arrasó España fue la Guerra Civil, destruyendo la mayoría de las instalaciones y empobrece al país. Además, la 1ª Guerra Mundial supuso la casi destrucción de Alemania, motor industrial de Europa, por lo que EEUU se erigió como primera potencia mundial y tomó la delantera a Europa.

Volviendo a España, en 1942 el régimen pretendía tener controlado el petróleo, el transporte ferroviario y la telefonía, sin embargo, la electricidad siguió en manos privadas. Por este motivo, el Gobierno formó empresas públicas en los lugares donde el negocio privado no había llegado aún, con lo que al final, estas empresas públicas se terminaron llamando Endesa (1944), quedándose con el suministro que no estaba gestionado por las empresas privadas.

En 1946 se creó el *Sistema Integrado de Facturación de Energía Eléctrica (SIFE)*, sistema que representó la primera vez que se unificó la manera en que se vendería la energía a los clientes. ¿Qué ocurría con la competencia?, las empresas tenían su territorio, pero sin competencia entre ellas por eso, hasta un poco más de 1950, es una época de crecimiento lento y se reconstruyen algunas centrales, proyectando otras hidráulicas. Sin embargo, a mediados de los 50 hay un fuerte desarrollo económico, propiciado por el turismo, empezando a construir muchas centrales térmicas, cercanas a las minerías (Galicia, León, Asturias, etc.). Se continuó con las grandes obras hidráulicas, realizándose aprovechamientos integrales de las cuencas.

En ese mismo momento, Europa quedaba dividida en capitalista y comunista. Es en estos años 50 cuando empezaba a ser rentable la energía nuclear, creándose las primeras centrales nucleares de fisión en el mundo (URSS, EEUU, Francia, Inglaterra). En España, en los años 60–70

se crearon las centrales con aprovechamiento óptimo de la energía (Zorita, la cual cerró sus puertas en 2006 en España, y Garoña), debido a que la extracción de los carbones empezaba a ser más cara, mientras que el nivel de vida en todos los países comenzaba a aumentar. Además, existía la Guerra Fría y la amenaza de armas nucleares, lo cual creaba un gran clima de incertidumbre internacional, por lo que las naciones apostaban por la diversificación de fuentes de energía. En esta misma década, aumentó el número de abonados eléctricos, por lo que crecieron tanto la demanda como la generación, necesitando más líneas y redes más malladas, y debiéndose interconectar las compañías.

Como fruto de esto, a finales de los 70, España ya estaba dividida en zonas, donde sólo operaba una eléctrica. Además de que todo el mundo tenía derecho a la electricidad, se comenzó a hablar de calidad de suministro (suministro ininterrumpido).

Gracias a la electrónica, informática y las comunicaciones, se empezó a vislumbrar horizontes imposibles hasta el momento, y es cuando comenzó a tener más presencia el mundo de la computación. En EEUU, los más interesados en este nuevo panorama fueron los militares, los bancos y las eléctricas.

En España, con el cambio de Gobierno a partir de 1975, se desbloquearon aspectos sociales y políticos nacionales, concretamente hubo dos eventos muy importantes que afectaron al sector eléctrico. Se comenzó a animar a los inversores privados para tratar de amortiguar la dependencia económica del petróleo, por ejemplo, de esta época es la construcción de la central nuclear de Lemóniz (no es tan antigua como Garoña o Zorita), y que no llegó a ser puesta en funcionamiento. Todavía estaba cerca y presente la crisis del petróleo de 1973, provocada por la *Organización de Países Exportadores de Petróleo (OPEP)*, que se fundó pocos años antes, básicamente lo que hizo *OPEP* fue aumentar el precio del petróleo de la noche a la mañana, creando una gran crisis. Las centrales térmicas que se abastecían de petróleo se convirtieron en ruinosas y se plantearon el cerrarlas o quemar carbón, pero en 1972 se celebró la Cumbre de Estocolmo convocada por la *Organización de Naciones Unidas (ONU)*, en la que se especificaba la forma en que había que quemar carbón, disparándose el precio de quemar este segundo combustible. Por todo lo anterior, se comenzaron a proyectar las centrales térmicas de 3ª generación, y además, los recursos hidráulicos ya estaban siendo totalmente explotados.

De nuevo en España, y debido al truculento panorama político, el gobierno decidió la moratoria nuclear y paralización de casi todos los proyectos nucleares, pero además, se produce la nacionalización de la red de alta tensión (1984). Existían una serie de empresas que tenían las cuatro funciones que forman el sistema eléctrico: Generación, Transporte, Distribución y Comercialización. El Gobierno quitó a la empresa privada el Transporte, y se crea *Red Eléctrica de*

España (REE), convirtiéndose en pública. Transporte representaba y representa el manejo de alta tensión, para llevar la energía desde donde se genera hasta las ciudades.

La Crisis del Petróleo de 1973 llegó a España más o menos entre 1979 y 1984. El negocio iba tan mal que se cierran los altos hornos y las acerías (sector que tuvo mucha influencia en País Vasco y norte de España). Estos hornos se alimentaban de carbón nacional, y al desaparecer, generó crisis en esta materia prima. Además, las eléctricas compraban fuera el carbón más barato, el Gobierno no se lo puede permitir y subvencionó el carbón de las cuencas españolas, en consecuencia, las eléctricas perdieron mucho dinero principalmente por tres frentes: moratoria nuclear; nacionalización y creación de REE; y la crisis de la minería. Evidentemente, a los inversores les gusta ver su dinero de vuelta, y ese dinero sale finalmente de los clientes, se añaden unos términos especiales en las tarifas eléctricas, y esto está durando hasta nuestros días.

Las energías renovables han constituido una parte importante de la energía utilizada por los humanos desde los albores de la historia, principalmente con la solar, la eólica y la hidráulica. La navegación a vela, los molinos de viento o de agua y las disposiciones constructivas de los edificios para aprovechar la del sol, son algunos ejemplos. Con el invento de la máquina de vapor (James Watt), se van abandonando estas formas de aprovechamiento, por considerarse inestables y dependientes del recurso, y se utilizan cada vez más los motores térmicos y eléctricos, en una época con todavía escaso consumo, y por tanto no se vislumbraba un posible agotamiento de las fuentes, y por supuesto donde no se tenían en cuenta problemas ambientales derivados de su uso masivo.

Hacia la década de años 1970 las energías renovables se consideraron una alternativa a las energías tradicionales, tanto por su disponibilidad presente y futura garantizada, como por su menor impacto ambiental en el caso de las “energías limpias”, y por esta razón fueron llamadas energías alternativas. Actualmente muchas de estas energías son una realidad, no una alternativa, por lo que el nombre de alternativas ya no debe emplearse, siendo más acertado el emplear el término de energía renovable.

Según la Comisión Nacional de Energía española, la venta anual de energía del *Régimen Especial (RE)* se ha multiplicado por más de 10 en España, a la vez que sus precios se han rebajado un 11%. En España las energías renovables supusieron en el año 2005 un 5.9% del total de energía primaria, con un 1.2% de eólica, un 1.1% de hidroeléctrica, un 2.9 de biomasa y el 0.7% de otras. La energía eólica es la que más crece. En la Figura 2.1 se muestra la capacidad instalada en los últimos años de la tecnología eólica y fotovoltaica en España, así como el total de la potencia instalada agrupado por diferentes tecnologías, en España y para 2012.

En la Figura 2.2 se representan las principales centrales nucleares y térmicas instaladas en nuestro país. La principal característica de las mismas es que se ubican en zonas muy concretas de



la geografía española. Entre ellas, y dependiendo de la localización del recurso renovable, se instalan las plantas de generación renovable comentadas anteriormente.

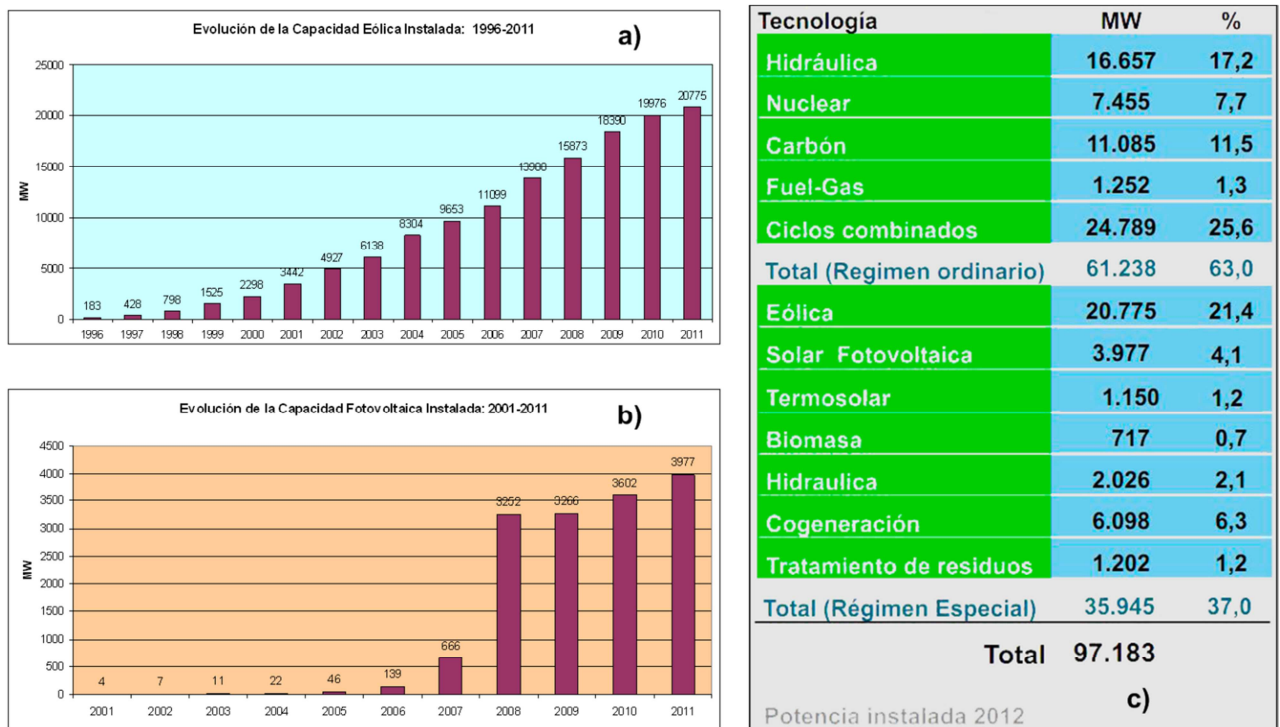


Figura 2.1. a) Evolución de la capacidad eólica instalada en España. b) Evolución de la capacidad fotovoltaica instalada en España. c) Potencia total instalada en España. Fuente Propia.

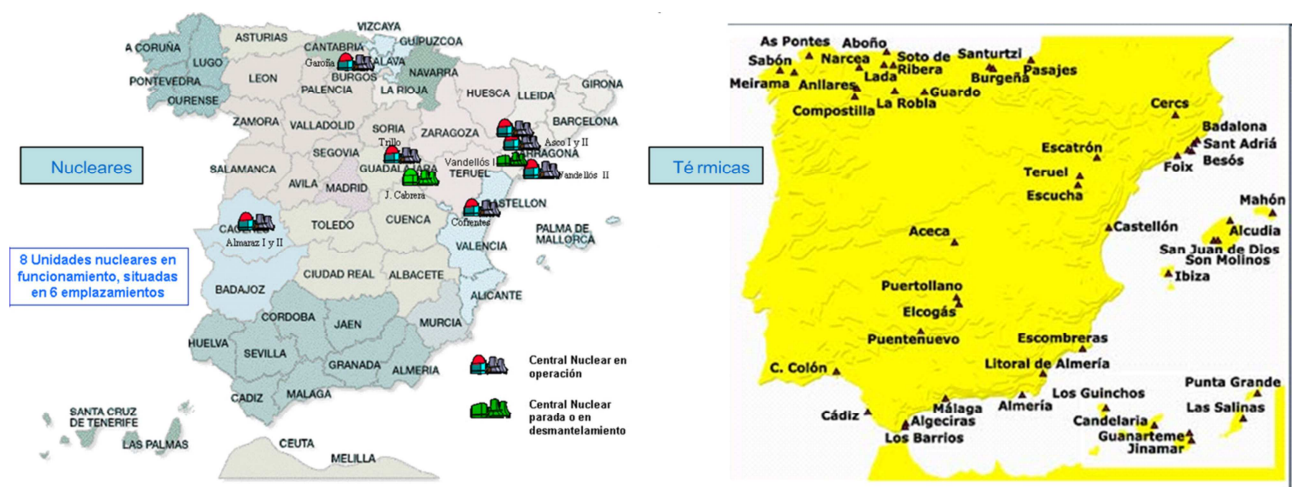


Figura 2.2. Ubicación de las centrales nucleares y térmicas en España. Fuente Rodríguez *et al.* (1998).

Para poder unir los puntos de generación de las centrales (tanto la generación clásica como las recientes de generación renovable) con los consumos, que normalmente se disponen de forma alejada de los primeros, en España REE transporta la energía eléctrica en alta tensión, gestionando para ello las infraestructuras eléctricas que componen la red de transporte, convirtiéndose en el Operador del Sistema de Transporte (*Transmission System Operator – TSO*). La red de transporte está compuesta por más de 38.000 km de líneas de alta tensión, más de 4.000 posiciones de subestaciones y más de 72.000 MVA de capacidad de transformación. Como gestor de esta red, REE

es responsable del desarrollo y ampliación de la red, de realizar su mantenimiento, de gestionar el tránsito de electricidad entre sistemas exteriores y la península y garantizar el acceso de terceros a la red de transporte en régimen de igualdad. En la Figura 2.3 se representa la red mallada de la TSO española planificada para 2016, pudiendo ver la extensa red existente actualmente y el desarrollo por hacer.



Figura 2.3. Planificación de la TSO española para 2016. Cortesía de REE. Fuente Rodríguez *et al.* (1998).

A partir de los puntos frontera de la TSO, es responsabilidad del Operador del Sistema de Distribución (*Distribution System Operator – DSO*) el tener que distribuir la energía eléctrica hasta los puntos de consumo final. En ciertas ocasiones, para grandes consumidores, se suele abastecer la energía en media tensión (fábricas en polígonos industriales, grandes centros comerciales, etc.), siendo responsabilidad de estos últimos el adquirir los elementos de protección y transformación correspondientes, que una vez instalados y validados, se cederán a la *DSO* correspondiente.

Para finalizar esta sección, resumiremos brevemente lo más destacado. En el origen del sistema eléctrico, la generación de la electricidad se realizaba, bien con corriente continua o bien con corriente alterna, en las proximidades de los consumos. Con la aparición de la corriente alterna, gracias a Tesla, y unido a la necesidad de consumo propiciado por el aumento del nivel de vida, permitió a las empresas privadas el detectar una fuente de ingresos alrededor de la electricidad, por lo que surgió la necesidad de llegar a la mayoría de los potenciales clientes. De esta manera, aparecieron las extensas redes de transporte de energía, a la vez que se creaban grandes centrales de producción eléctrica, en determinadas localizaciones. Esto permitió el poder llegar a la gran mayoría de los clientes, pero a base de grandes inversiones en infraestructura, y además, aumentando la ineficiencia del sistema, al tener asociado al transporte y la distribución pérdidas de energía. En las últimas décadas, la apuesta por las fuentes de generación renovable, supuso el

acercamiento de la generación de electricidad al lado del consumo, empleando por tanto grandes plantas de tecnología diferente (eólica, fotovoltaica, etc.), permitiendo al sistema un aporte extra de energía, al aprovechar el recurso existente, y liberando algo de pérdidas en transporte y distribución, haciendo que el sistema fuera algo más eficiente. Como se verá en las siguientes secciones, la reciente incorporación de elementos de inteligencia en las redes de transporte y distribución, junto al comentado acercamiento de la generación hacia los espacios con sólo consumo hasta la fecha, se presenta como una gran oportunidad para la I+D+i desde diferentes ámbitos.

## 2.2. EVOLUCIÓN DEL SISTEMA ELÉCTRICO: SMART GRID

A medida que las redes comenzaron a crecer por los motivos ya comentados, empezó a ser muy complejo su control, además de ser ineficientes debido a las pérdidas. Hay que añadir, que hasta ese momento, los consumidores finales, con excepción de los grandes consumidores debido a que reciben un trato diferente, no son considerados en el sistema, salvo para el pago de la factura, que además en la mayoría de los casos no es una facturación acorde con la realidad, ya que está basada en estimaciones, disponiendo de un periodo de tiempo la comercializadora para la realización de la misma.

Este modelo, ha comenzado a cambiar en los últimos años, al ir introduciendo inteligencia en los diferentes niveles del sistema eléctrico, comenzando por el transporte, en los últimos años en la distribución, y aunque en menor medida, recientemente en las proximidades al consumo final. De esta manera surgen la *SG*, término traducible por red de energía eléctrica inteligente, ya que utiliza la *TIC* para intentar optimizar la producción y la distribución de electricidad con el fin de equilibrar mejor la oferta y la demanda entre productores y consumidores. Además de lo anterior, y aprovechando el despliegue de sensórica e inteligencia, tratan de optimizar los flujos de energía, mejorar la algoritmia de detección de faltas, y por tanto, mejorar en la operación de la red, como se expone en (Fang *et al.*, 2012).

Otras definiciones válidas de *SG*, ya que no se tiene aún claro el alcance de las mismas, son:

- Según el Instituto de Investigación de Potencia Eléctrica (*Electric Power Research Institute – EPRI*): “sistema eléctrico capaz de suministrar energía a millones de clientes, dotado de una infraestructura de comunicaciones inteligente que permite establecer puntualmente el flujo de información adaptable y seguro necesario para abastecer una economía digital en constante evolución”.
- Según *General Electric (GE)*: “es Internet de la Energía que distribuye información de la red eléctrica en tiempo real para potenciar elecciones energéticas más inteligentes”.
- De nuevo *GE*: “red de transmisión y distribución avanzada que emplea información digital y tecnologías de control para mejorar la fiabilidad, la seguridad y la eficiencia”.

- Según Mantooth (2010): “*hardware y software incorporados al sistema eléctrico para conseguir una reacción más autónoma ante eventos que puedan afectar a la red y una eficiencia operativa diaria del suministro*”.

Uno de los elementos base, y que se tratará más adelante, de las *SG* es el *SM*, ya que permitirá una telediagnóstico de las curvas de carga, y posibilitará una facturación detallada de los consumos, y en un futuro, la facturación por franjas horarias, permitiendo a los consumidores no solo el elegir las mejores tarifas de entre las diferentes empresas comercializadoras, sino también discernir entre las horas de consumo, lo que permitiría un mejor uso de la red. Este sistema también permitiría tratar de igualar con más precisión el consumo con la generación, y anticipar mejor las necesidades futuras a nivel local.

La irrupción de las energías renovables en el panorama energético ha cambiado notablemente los flujos de energía en la red eléctrica, ahora los usuarios no sólo consumen sino que también producen electricidad a través de la misma red. Por tanto, el flujo de energía es ahora bidireccional, siendo otra característica fundamental de las *SG*. En la Figura 2.4 se representa un esquema de una *SG*, donde se aprecia como las fuentes de generación y almacenamiento se acercan a los consumos por medio de la red de distribución, y aparecen elementos inteligentes y novedosos en el lado del consumo, para tratar de obtener una mejor operación y conseguir eficiencia.

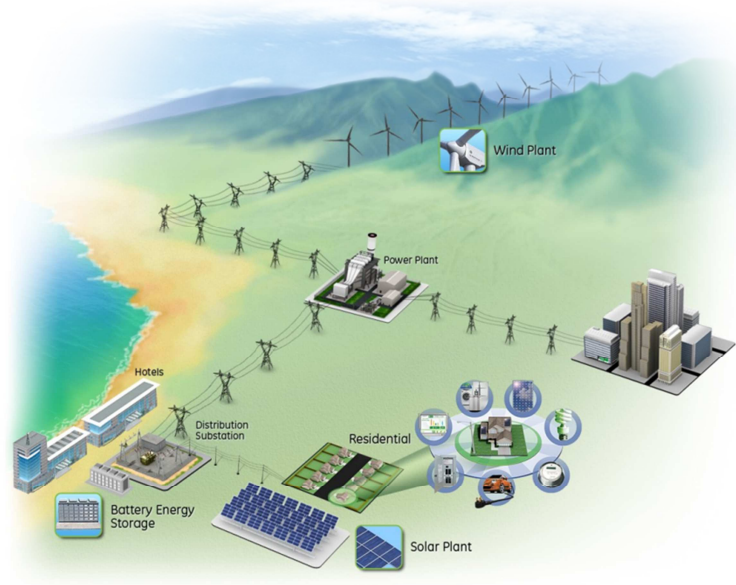
Los principales requerimientos de una red de energía eléctrica del futuro pueden resumirse, como indican Ferreira *et al.* (2010), en:

- Flexible: las redes deben satisfacer las necesidades de los clientes, mientras al mismo tiempo, responden a una medida de la incertidumbre (cambios y retos futuros).
- Accesibilidad: el acceso a la red por parte de todos los usuarios debe ser garantizado, incluyendo las fuentes de generación renovable y una alta penetración de las unidades locales de generación que están comenzando a proliferar.
- Fiabilidad: las redes deben garantizar la fiabilidad y calidad de suministro, acorde con las exigencias de una economía dependiente de la energía eléctrica.
- Consideraciones económicas: la competencia y la regularización darán lugar al desarrollo económico y la evolución de las *SG*. La innovación deberá ser la clave para una gestión de energía eficiente.

Los requerimientos antes mencionados conducen a los siguientes objetivos *SG* (Sánchez–Jiménez, 2006):

- Desarrollo de soluciones técnicas que puedan ser desplegadas rápida y con bajo coste, así como el establecimiento de estándares técnicos y protocolos que puedan asegurar la apertura de soluciones.

- Desarrollo de sistema de información, computación y telecomunicaciones, para hacer negocios y dar servicios innovadores para el sistema y sus clientes.
- Armonizar marcos regulatorios y comerciales en diferentes regiones para las soluciones adoptadas y sistemas de trabajo uniforme y conseguir economías de escala.
- Tratar de realizar un aprovechamiento de la infraestructura existente hasta el momento, consiguiendo un perfecto sincronismo entre los diseños de modelos del pasado y los nuevos.



**Figura 2.4. Esquema de SG con elementos generadores, almacenadores y consumidores. Fuente Green Buildings Elements (2012).**

Como se presenta en (Sánchez–Jiménez, 2006), las redes europeas de electricidad han proporcionado los vínculos vitales entre los productores y los consumidores con gran éxito desde hace décadas. La arquitectura fundamental de estas redes se ha desarrollado para satisfacer las necesidades de grandes plantas, principalmente las tecnologías de generación a base de carbono, ubicados de manera lejana de los centros de demanda. Los desafíos de la energía a los que actualmente se enfrenta Europa están cambiando el escenario de la generación de electricidad. El impulso de las tecnologías de generación de bajo carbono, combinado con una mayor eficiencia en el lado de la demanda, permitirá a los clientes interactuar con las redes.

En este contexto, la Plataforma Tecnológica Europea (*European Technology Platform – ETP) Smart Grids*, se creó en 2005 para formar una visión conjunta de las redes europeas de energía y tratar de conseguir los objetivos del 2020. La plataforma incluye a representantes de la industria, *TSO*, *DSO*, gestores de redes, organismos de investigación y reguladores. Tiene identificado objetivos claros y propone una ambiciosa estrategia para hacer realidad esta visión de los beneficios de Europa y de sus clientes de electricidad. Entre los objetivos marcados por Europa están:

- Construir y mantener una visión compartida para el futuro de las redes eléctricas de Europa y servir de catalizador para su desarrollo.
- Asegurarse de que la visión e implementación continúan centrándose en dar respuesta a las necesidades de los clientes, siempre según las directrices de Europa.
- Mantener una visión estratégica de alto nivel con el fin de comunicar de forma eficaz la evolución del sector, las oportunidades, así como los retos potenciales.
- Abogar por el desarrollo de una política energética sostenible para una Europa más competitiva.
- Promover la investigación de SG, su desarrollo, demostración y la ejecución de proyectos de implementación.

### 2.3. UN PASO MÁS PARA LAS FUENTES DE GENERACIÓN RENOVABLE: GENERACIÓN DISTRIBUIDA

Como se ha comentado, el aporte de las fuentes de generación renovable, ha venido impuesto principalmente por la necesidad de los países de no depender de los combustibles derivados del petróleo. Además, la conciencia medio ambiental que en los países desarrollados se tiene desde ya hace unos cuantos años, está posibilitando el despegue de diferentes tecnologías renovables.

Desde la implantación de las plantas de generación renovable, en cualquiera de sus tecnologías, su característica común ha sido similar a la de las plantas clásicas, instalaciones en zonas alejadas de los núcleos de consumo (aunque cada vez cerca de ellos), así como con un elevado valor de potencia instalada.

En España, las principales tecnologías con mayor implantación han sido la eólica, fotovoltaica e hidráulica. En la Figura 2.5 se presenta la evolución de los últimos años de la producción de las diferentes tecnologías del RE en España. En el gráfico se manifiesta el progresivo crecimiento de la eólica en este país, así como la desaceleración del crecimiento de todas ellas a partir de 2010, consecuencia de los ajustes de las primas realizadas.

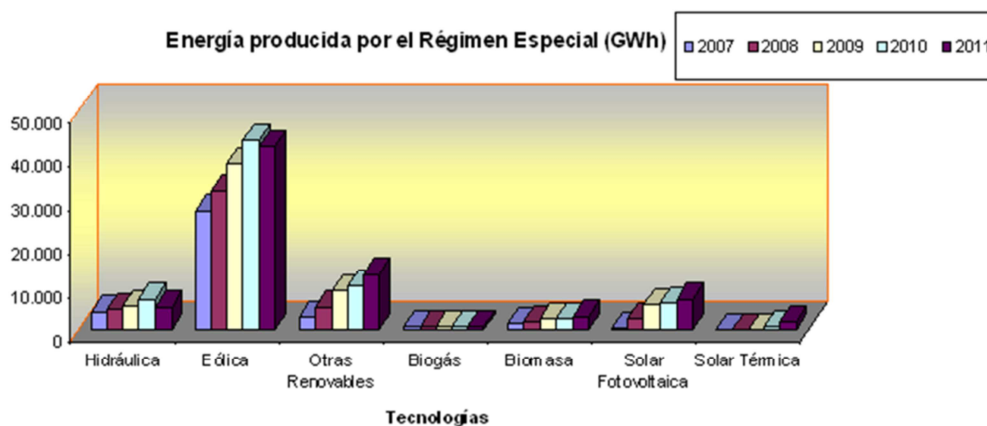


Figura 2.5. Evolución en España de las diferentes tecnologías en RE. Fuente Propia con Datos de REE.

Según la *Comisión Nacional de la Energía (CNE)*, con el Real Decreto–ley 1/2012, de 27 de enero, por el que se procede a la suspensión de los procedimientos de pre–asignación de retribución y a la supresión de los incentivos económicos para nuevas instalaciones de producción de energía eléctrica a partir de cogeneración, fuentes de energía renovables y residuos, supondrá “*una revisión a la baja de las previsiones de potencia a instalar por el régimen especial peninsular estimadas para el período 2011–2015*”. Durante 2012 solo se instalarían en España 97 MW fotovoltaicos, en cuanto a la eólica el informe dice que en 2012 se instalarían 474 MW, con respecto a la termosolar la previsión indica la instalación en 2012 de 970 MW y colapso total durante el trienio siguiente y cero megas para biomasa y cero para biogás durante los cuatro años. Esta información, y la evolución según la *CNE*, se muestran en la Tabla 2.1 para la península. En cambio, las previsiones de potencia instalada dadas por la *CNE*, con el mismo horizonte de tiempo, pero si no hubiera entrado en vigor el Real Decreto–ley, serían las mostradas en la Tabla 2.2 para la península. En ambas tablas se muestran la evolución de las diferentes tecnologías, en cuanto a potencia instalada, partiendo del año 2012 y hasta el 2015.

**Tabla 2.1. Previsión de la evolución de la potencia instalada de RE. Fuente Propia con Datos de la CNE.**

<i>Potencia Instalada MW</i>	<i>2012</i>	<i>2013</i>	<i>2014</i>	<i>2015</i>
Cogeneración	6.211	6.211	6.211	6.211
Solar Fotovoltaica	4.080	4.177	4.177	4.177
Solar Termoeléctrica	1.551	2.521	2.521	2.521
Eólica	22.470	23.944	23.944	23.944
Hidráulica	2.063	2.063	2.063	2.063
Biomasa y Biogas	775	775	775	775
Residuos	456	456	456	456
Tratamiento de Residuos	658	658	658	658
<i>Total</i>	<i>38.264</i>	<i>40.805</i>	<i>40.805</i>	<i>40.805</i>

**Tabla 2.2. Previsión de la evolución de la potencia instalada de RE antes de la entrada en vigor del Real Decreto–ley 1/2012 de 27 de enero. Fuente Propia con Datos de la CNE.**

<i>Potencia Instalada MW</i>	<i>2012</i>	<i>2013</i>	<i>2014</i>	<i>2015</i>
Cogeneración	6.285	6.432	6.582	6.731
Solar Fotovoltaica	4.172	4.560	4.948	5.333
Solar Termoeléctrica	1.501	2.281	2.521	2.771
Eólica	22.470	23.969	24.944	25.919
Hidráulica	2.082	2.119	2.158	2.198
Biomasa y Biogas	807	865	925	985
Residuos	510	515	520	525
Tratamiento de Residuos	658	658	658	658
<i>Total</i>	<i>38.485</i>	<i>41.398</i>	<i>43.256</i>	<i>45.120</i>

Del análisis anterior, se evidencia una disminución de la potencia instalada total por parte del *RE* en la península de un 0.6% en 2012, 1.5% en 2013, 6% en 2014 y 9.56% en 2015. Esta paro en la instalación del *RE*, según la *CNE* se deberá principalmente al citado Real Decreto–ley.

Con el escenario que se acaba de mostrar, parece no haber un futuro muy alentador para la energía renovable. No obstante, y tal y como se introdujo en el Capítulo I, aparece el término *DG*, dando una nueva oportunidad a las fuentes de generación renovable, y además, posicionándolas en localizaciones no contempladas hasta el momento, concretamente en las proximidades de los puntos de consumo. En muchos casos, la *DG* puede proporcionar electricidad a un menor coste, proporcionando altos niveles de fiabilidad y de seguridad, y además, con menos consecuencias ambientales que las plantas clásicas de energía (carbón, nuclear, etc.). Los sistemas con *DG* emplean numerosas plantas, de tamaño pequeño cada una, y puede proporcionar la energía en las proximidades donde se necesita, no teniendo casi dependencia de la red de distribución y la de transporte. La potencia de *DG* podrá variar en el rango de un kilovatio a los centenares de megavatio. Se puede decir por tanto que existen dos modelos de *DG*, indicadas a continuación:

- *DG* a nivel local: son las plantas de generación que conocemos en los últimos años, granjas de turbinas eólicas o módulos fotovoltaicos para la producción de electricidad. Su localización suele ser en emplazamiento donde el recurso existente permita la implantación de una potencia elevada, pero siempre en una escala inferior a las grandes plantas térmicas o nucleares.
- *DG* a nivel punto final: consiste en aplicar directamente muchas de las tecnologías renovables descritas (eólica, fotovoltaica, hidráulica, etc.) en las dependencias de los consumidores directamente.

Los defensores de los consumidores, consideran que la *DG* puede mejorar la eficiencia del suministro de energía eléctrica. Consideran que la energía que transita desde las grandes plantas de producción hasta los consumos finales está sujeta a una pérdida de entre 4% y 9%, como consecuencia de las pérdidas por efecto Joule y la obsolescencia del equipamiento e instalaciones. Al mismo tiempo, los clientes sufren de mala calidad en la energía, con las variaciones de tensión eléctrica o porque el flujo de energía se ve afectado de una variedad de factores, incluyendo operaciones de conmutación, interrupciones, transitorios, y las perturbaciones de la red por las grandes cargas. Además, los clientes ven en sus facturas reflejado el gasto de las grandes infraestructuras de transporte y distribución, por lo que el uso de equipos de generación de energía en el sitio, puede proporcionar a los consumidores de energía un nivel superior de ahorro. Además, los residentes y las empresas que generan energía a nivel local tienen la posibilidad de vender la energía excedente a la red, lo que puede generar ingresos significativos durante las horas de demanda máxima.



Las instalaciones de *DG* ofrecen ventajas potenciales para mejorar la transmisión de energía, debido a que producir energía localmente para los usuarios, ayudará a toda la red mediante la reducción de la demanda en horas punta y reduciendo la congestión de potencia en la red, una de las causas del apagón de 2003.

Los defensores del medio ambiente y los académicos, sugieren que con la *DG* se puede proporcionar beneficios adicionales a la sociedad. Las grandes plantas eléctricas centralizadas, emiten cantidades importantes de monóxido de carbono, óxidos de azufre, hidrocarburos y óxidos de nitrógeno. Además, las grandes instalaciones de energía también centran su contaminación y producen calor residual, a menudo destruyendo los hábitats acuáticos y la biodiversidad marina. Además, dado que la *DG* puede permanecer independiente de la red, posibilita proporcionar energía de emergencia para un gran número de servicios públicos, tales como hospitales, escuelas, aeropuertos, estaciones de bomberos y de policía, bases militares, prisiones, suministro de agua y plantas de tratamiento de aguas residuales y naturales de transporte de gas y sistemas de distribución y estaciones de comunicaciones. Por último, la *DG* puede ayudar a la nación a incrementar su diversidad de fuentes de energía, por ejemplo, algunas de las tecnologías de la *DG*, como las turbinas eólicas, paneles solares fotovoltaicos y turbinas hidroeléctricas, no consumen combustibles fósiles. La creciente diversidad ayuda a aislar a la economía de las variaciones de precios, interrupciones, y la escasez de combustible.

Independientemente del modelo económico, y dejando a un lado las primas del sector, el sentido común dice que si existe potencial de generación eléctrica en los emplazamientos donde se consumen de manera continuada y masiva, debería poder aprovechar ese potencial y convertirlo en energía eléctrica, para de este manera aumentar la eficiencia global del sistema, y contribuir a la mejora medio ambiental empleando tecnologías renovables. Además de lo anterior, los gobiernos deben ver la *DG* como una oportunidad de generación de empleo local, como se lleva años trabajando en este aspecto en UK, apoyado por un sistema de tarifas sostenible. La red del futuro deberá convivir con las plantas centralizadas y con la *DG*, tal y como se refleja en (Nissen, 2009), e indiscutiblemente la *DG* formará parte de los elementos disponibles entre los diferentes desarrolladores y planificadores de ciudades, teniendo siempre presente el impacto de la *DG* sobre la red actual, como se indica en (Hamedani and Arefifar, 2006), necesitando algoritmos optimizadores para el control.

#### **2.4. LA AGREGACIÓN DE FUENTES DE GENERACIÓN COMO SOLUCIÓN DE FUTURO: VIRTUAL POWER PLANT**

Dentro de la *SG*, surge un nuevo modelo de gestión de la producción de energía denominado *VPP*, como se explica en (Wille-Haussmann *et al.*, 2010), donde la planta de energía ya no es una

instalación monolítica, sino una suma de pequeños elementos cooperantes, e inteligentes. La agregación de los generadores y las cargas se unen como una unidad autónoma físico o lógico, donde todos los elementos funcionan coherentemente, como pueda ser un parque industrial, una zona residencial con módulos fotovoltaicos o los recursos distribuidos de una *utility*. Como se puede observar, el concepto de *VPP* deriva del concepto *DG*, aunque como se verá en esta sección, cubre un mayor espectro, ya que debe tener en cuenta la carga, no sólo se limita a las fuentes de generación como ocurre cuando se trata con *DG*.

Sin embargo, la gestión de estas *VPP* representa un desafío para las *TIC*. *VPP* comprende un gran número de elementos diferentes, diseñados para resolver los problemas locales, pero que necesitan interactuar juntos con el fin de comportarse como una unidad (*VPP*). Protocolos de comunicación y herramientas son necesarias para permitir que estos elementos se puedan comunicar, tomar decisiones que afectan a varias entidades, y coordinar los comportamientos para llevar a cabo tareas complejas, como las presentadas en (Sebastian and Maire, 2008). Al mismo tiempo, los dispositivos de una *VPP* son extremadamente heterogéneos: se construyen y son administrados por organizaciones muy diferentes con propósitos distintos, y seguirán normas dispares para el control y la comunicación.

Además de las plantas grandes, el paisaje de la red eléctrica del futuro, abarcará nuevos elementos más pequeños: módulos fotovoltaicos en las casas, pequeñas turbinas eólicas para áreas residenciales y pequeñas plantas de producción eléctrica de diferentes tecnologías en los parques industriales. Si bien estos elementos pueden funcionar de manera aislada, su integración facilitará su uso de manera más eficiente. Es posible imaginar la redirección de energía en una *VPP* de los módulos fotovoltaicos de las casas vacías, para cubrir los picos de demanda de viviendas con consumos en esos instantes (con la consecuente facturación), o por ejemplo, seleccionando automáticamente el mejor momento para recargar vehículos eléctricos (aprovechando los picos de viento durante la noche), o vender la energía excedente a un parque industrial cercano. Tal y como se muestra en (Bel *et al.*, 2007), la agregación de pequeños generadores y sus cargas controlables a través de *VPP*, puede dar una oportunidad a las pequeñas unidades de producción, y así alcanzar volúmenes de energía producida que les haga económicamente competitivas, y además, obtener una coordinación para ofrecer capacidad de los servicios a la red.

A continuación, se explicará con mayor detalle las partes que conforman una *VPP*, haciendo especial mención a la parte técnica y la comercial, tal y como se muestra en (Corera, 2006). Como ya se ha dejado claro, una sola unidad de generación pequeña no puede ofrecer una alta capacidad económica, fiabilidad, flexibilidad y de control en el mercado de la electricidad. Una *VPP* puede ser vistas como una planta de generación individual con su programa de generación propia y sus límites de control, así como con su costo de operación y su característica propia de la demanda.

VPP puede entonces ser utilizada para proporcionar servicios técnicos y comerciales al operador del sistema y para el mercado de la energía, como se refleja en (Pudjianto *et al.*, 2008). Por tanto, una VPP se divide en dos sub-VPP, la Planta de Energía Virtual Comercial (*Comercial Virtual Power Plant – CVPP*) y la Planta de Energía Virtual Técnica (*Technical Virtual Power Plant – TVPP*). A continuación se explicará cada uno de ellos:

- **CVPP:** una CVPP tiene un perfil agregado y una salida que representa las características de costo y de operación para la cartera de DG. El impacto de la red de distribución no se considera en el perfil CVPP. Los servicios/funciones de CVPP incluyen el comercio en el mercado mayorista de la energía, el equilibrio de las carteras de negociación y la prestación de servicios (a través de la presentación de ofertas y demandas) para el operador del sistema. El operador de CVPP puede ser cualquier *agregador* de terceros o un proveedor de energía.
- **TVPP:** la TVPP consiste en DG en una misma localización física. La TVPP incluye la influencia en tiempo real de la red local en el perfil de DG agregado, así como la representación de las características de costo y operación de la cartera. Los servicios/funciones de TVPP incluyen la gestión del sistema local para DSO, así como proporcionar el equilibrio entre TSO y los servicios auxiliares. El operador de un TVPP requiere información detallada sobre la red local (normalmente será el DSO).

En la Figura 2.6 se detallan las entradas y salidas de ambos subsistemas, así como la interacción entre ellos.

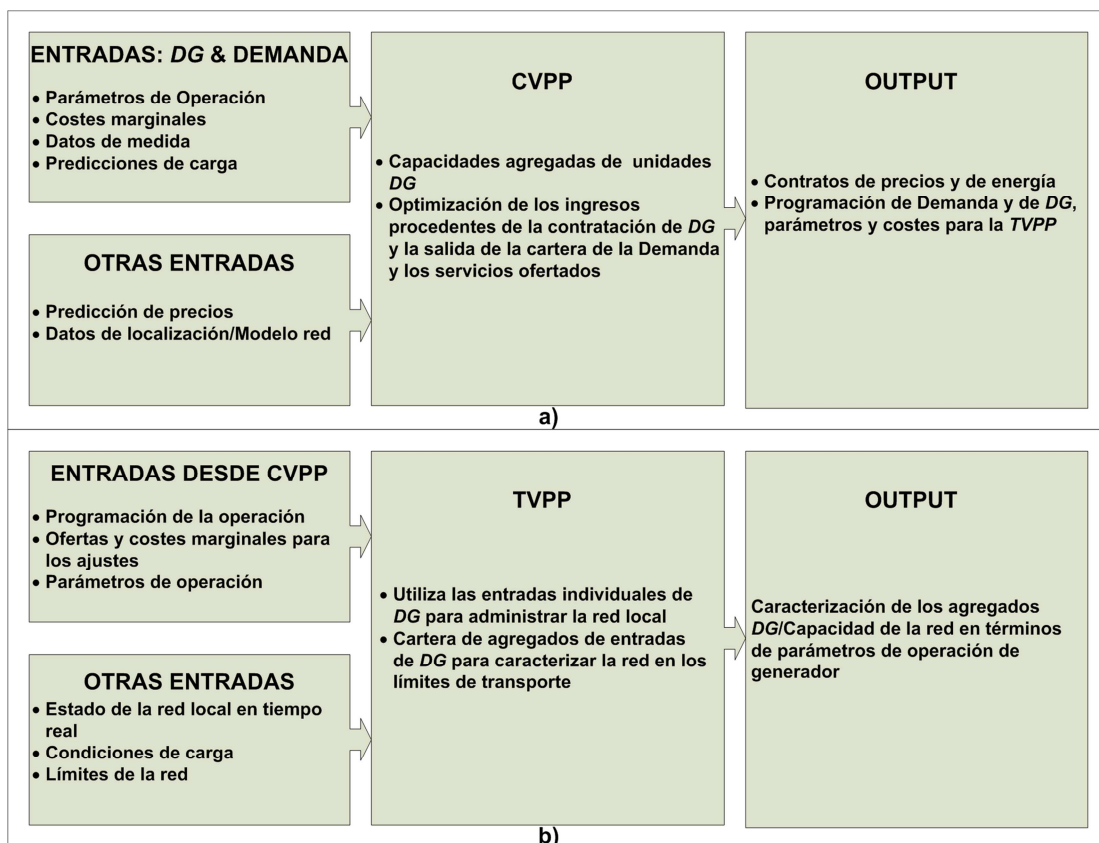


Figura 2.6. Subsistemas de VPP: a) CVPP; b) TVPP. Fuente Propia.

Para finalizar esta sección, se presenta una oportunidad para la *DG* por medio de las *VPP*, ya que en su concepción, una *VPP* agregará las pequeñas unidades de generación en su ámbito de control. Además, y sin ánimo de que caiga en el olvido, la *VPP* debe de ser conocedor de su demanda bajo su control, ya que tanto la *DG* como la demanda son parámetros fundamentales para *CVPP* y *TVPP*.

## **2.5. UN NUEVO PARADIGMA DE SOSTENIBILIDAD: MICROGRID**

Según Lassetter *et al.* (2002), se define *microgrid* como “una agregación de cargas y microgeneradores operando como un sistema único que provee tanto energía eléctrica como térmica”. Este nuevo paradigma, como sistema *agregador* de diversas tecnologías de generación y de cargas, se posiciona de manera privilegiada de cara a un aprovechamiento de las oportunidades del mercado liberalizado y, además, contribuye a tener una red eléctrica más optimizada y racional.

Con respecto a la *microgrid* eléctrica, la mayoría de los generadores de pequeña potencia conectados, lo estarán por medio de convertidores de potencia, con el fin de proporcionar flexibilidad en cuanto a su operación, y disponer de esta manera de un sistema agregado de generación. El control de los convertidores, permite a la *microgrid* cubrir las necesidades locales con fiabilidad y seguridad.

En la definición dada por *Consortium for Electric Reliability Technology Solutions (CERTS)* sobre *microgrid*, no se especifica si ésta funciona de manera aislada de la red de distribución o conectada a la misma. En (Anduaga *et al.*, 2008) se redefine *microgrid* eléctrica como “el sistema formado por fuentes de generación, equipos de almacenamiento y cargas conectadas eléctricamente, que puede funcionar tanto conectado al sistema principal como aislado del mismo en el caso de perturbaciones eléctricas, que se controla desde el operador del sistema como un sistema agregado y en el que hay que planificar y gestionar la energía generada y consumida”. Como se observa, la definición de *microgrid* no especifica si la conexión debe de ser en baja tensión o media tensión, ni fija los límites de su potencia, esto dependerá de la aplicación de la *microgrid*, del volumen de demanda que se pretenda satisfacer, así como de su ubicación física. Por tanto, una *microgrid* podría estar compuesta por los elementos existentes en un edificio, pudiendo definir el término *nanogrid*, o ser del tamaño de un pueblo o pequeña ciudad. Con independencia de lo anterior, en la Figura 2.7 se muestra una representación de una *microgrid* conceptual, donde se aprecian los elementos eléctricos y sus niveles de tensión, las diferentes tecnologías de *DG* desplegadas en la misma, almacenamiento eléctrico distribuido, elementos de comunicación y procesado, *SM*, otros elementos que la conforman (estaciones climáticas, Vehículo Eléctrico [*Electrical Vehicle – EV*], etc.) y las interconexiones (eléctricas y de comunicación) entre varias *microgrids* a través de sus controles.

Como detalle de interés, la necesidad de una planificación y una gestión de la *microgrid*, tanto de la generación como de la demanda. Además, y agregando dificultad, se debe aclarar que las potencias, tanto de generación como demanda en una *microgrid*, son muy inferiores a las que nos encontraríamos en una gran planta de producción fotovoltaica de varios *MW*, o para la demanda de un país, una extensa zona del mismo o incluso un gran núcleo urbano. Pensemos entonces que para una óptima planificación, se precisarán herramientas de predicción lo más precisas posibles, por lo que en una demanda desagregada que se presentará en una *microgrid*, no se producirán las cancelaciones y aportaciones en la curva de carga que se producen en una zona extensa o un país, presentando en éstos, un perfil mucho más suave y no tan abrupto al ir desagregando la demanda, por lo tanto, los mecanismos de predicción deben de continuar evolucionando en las *microgrids*.

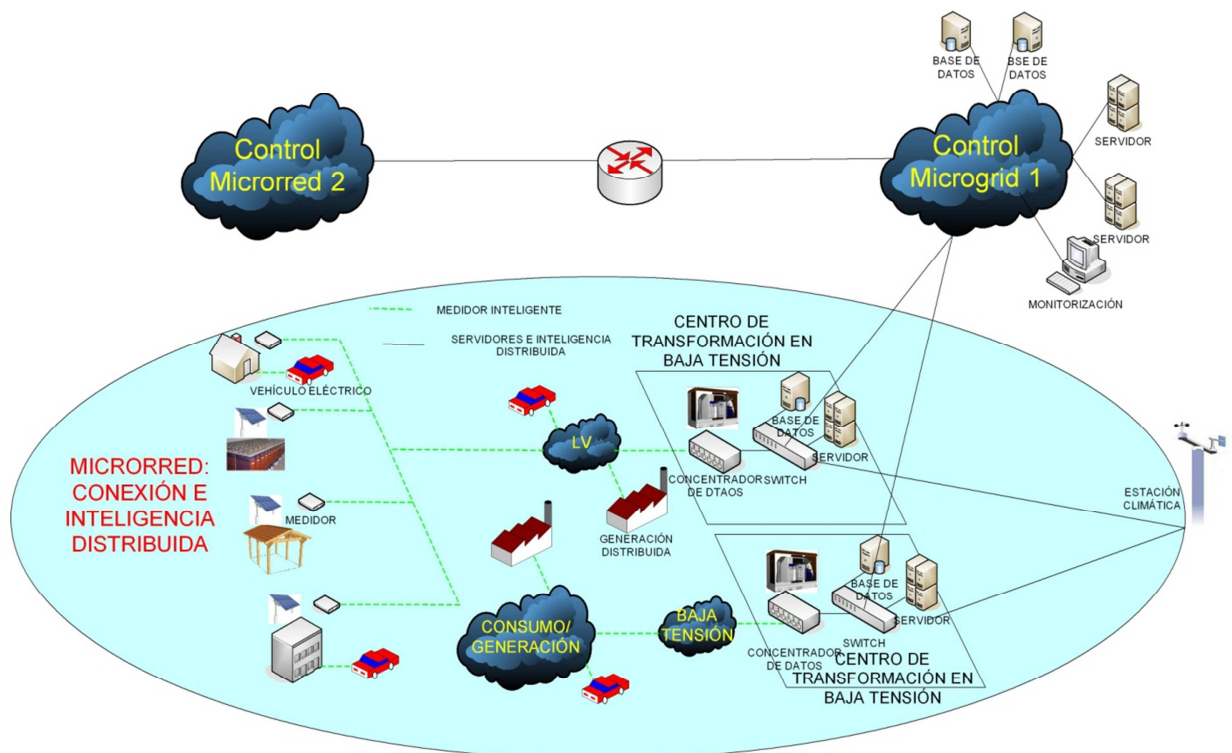


Figura 2.7. Recreación del concepto de *microgrid*. Fuente Propia.

La operación de la red eléctrica, de la *DG*, almacenamiento eléctrico y las cargas en una *microgrid*, posibilita la consecución de los siguientes objetivos:

- Con la planificación y operación de todos los elementos disponibles, tanto de la parte eléctrico como térmica, se aumentará el rendimiento energético, y por tanto se obtendrá *Eficiencia Energética*.
- Al hacer un uso más racional de los recursos existentes en una *microgrid*, se contribuye a la reducción del *Gas Efecto Invernadero (GEI)*.
- La *microgrid* facilita la penetración de las renovables, al permitir controlar las agregaciones de diferentes fuentes y sistemas de almacenamiento para tratar de allanar la irregularidad

temporal que presentas las fuentes de generación renovable. Por tanto la *microgrid*, se presenta como un entorno para la *DG*, y ésta, una oportunidad para las renovables.

- *Reducción del Coste*, ya que podrá participar en el mercado de energía, siendo una de las formas que tiene de participación por medio de la *VPP*.
- En caso de perturbaciones de la red eléctrica, la *microgrid* se podrá desconectar, por tanto, permitirán un incremento en la *Seguridad de Suministro*, y una participación en *Servicios Auxiliares* (control de frecuencia y control de tensión).
- Al disponer de *DG* y almacenamiento en las proximidades a la demanda, se minimizan las pérdidas en el transporte y distribución, por tanto, se *Minimizan las Pérdidas Eléctricas*.

Como se ha visto, y representado en la Figura 2.7, una *microgrid* se integra dentro de una red de suministro eléctrico, por lo que su sistema de control deberá coordinarse con su homólogo en la red de distribución/transporte, o incluso otra *microgrid*. Por tanto, deberá existir una jerarquía en el control y la operación de la *microgrid*, siendo la siguiente:

- Operador de la Red de Transporte, Operador de la red de Distribución y Operador de Mercado. La *microgrid*, con transporte debe coordinar los servicios complementarios, con distribución los momentos de desconexión/conexión de la misma, y con mercado la gestión de precios de la energía.
- Controlador central de la *microgrid*. Asume múltiples funciones, desde el despacho económico de la *microgrid*, hasta la coordinación de los controladores locales, entre otros.
- Controladores locales. Nivel más bajo de control, cuya función será controlar las unidades de producción, almacenamiento y cargas, siempre guiado por las indicaciones del controlador central de la *microgrid*.

Para concluir, resumiremos que el nuevo paradigma *microgrid*, se presenta como un aglutinador de *DG* en pequeñas unidades de potencia, junto a elementos almacenadores y una serie de cargas, todos ellos a planificar y gestionar en la medida de lo posible. En definitiva, se compone de unidades generadores, que si son agregadas para salir al mercado se comportarán conceptualmente como una *VPP*, y una demanda, apoyados ambos elementos por el almacenamiento. Para la total orquestación de todos los elementos de la *microgrid*, así como la interacción con el exterior (*DSO*, *TSO* y mercado), se presenta como reto fundamental el disponer de predicciones de la demanda existente y de su generación, ya que resultará imposible la toma de decisiones en torno a la energía, si no se conoce con cierta anterioridad el comportamiento de la demanda y generación bajo su control.

## 2.6. NUEVOS ACTORES EN EL HORIZONTE. SINERGIAS ENTRE MODELOS Y NUEVO CONCEPTO AGLUTINADOR

Derivados de la *SG*, y que deben de tenerse en cuenta en ellas y en el resto de escenarios anteriormente mencionados, aparecen una serie de conceptos, los cuales se detallarán a continuación:

- *DR*: por intereses de mercado o por el ajuste de la demanda a la *DG* existente, se plantea el desplazar la demanda de manera controlada, apareciendo el término *DR* y sus implicaciones. En (Saele and Grande, 2011) se presenta la experiencia de un estudio piloto centrado en *DR* diaria de hogares, empleando *SM*, control de carga remota, fijación de precios basado en precios horarios, presentado los efectos en el mercado (precios) al hacer *DR*, y marcando las bases para futuros proyectos.
- Movilidad eléctrica: la aparición del *EV* debe de tenerse en cuenta en el impacto sobre la infraestructura eléctrica, como por ejemplo en (Couillet *et al.*, 2012), donde se investiga la interacción competitiva entre los *EV*, híbridos y convencionales, en un mercado con transacciones de electricidad, y cuyo objetivo es el maximizar el beneficio de la red de distribución de electricidad existente. De la misma manera, se debe prestar atención a las nuevas posibilidades de negocios, tal y como se muestra en (Sarangi *et al.*, 2012), presentando una moneda de energía denominada *token*, que da derecho a su propietario a producir o consumir una cierta cantidad de electricidad energía, para lo cual se propone una infraestructura de *TIC* para administrar *tokens* producidos por millones de *EV* y centrales eléctricas, dentro de la infraestructura eléctrica de una *SG*.
- *SM*: medidor electrónico que registra información de consumos de una hora o periodos inferiores y comunica esta información, posibilitando de nuevo una comunicación bidireccional entre el punto de consumo y la *utility*, posibilitando el conocimiento del patrón de hábito de los consumos y la propuesta de tarifas flexibles. *SM* no sólo se plantea para la medida de energía y su posterior facturación, por ejemplo en (Dong *et al.*, 2012), se presenta como los *SM* pueden almacenar gran cantidad de información, que puede servir para poder hacer un seguimiento del consumo y comportamiento de los dispositivos existentes dentro del punto de consumo.
- *Smart Home*: las investigaciones pasadas han tratado el consumo de energía doméstico desde cuatro enfoques: *feedback* a los usuarios, emplear tecnología para detectar y controlar el uso de la energía, estrategias económicas tales como incentivos para reducir la energía y examinar factores sociales concernientes con las prácticas de energía; todas ellas se presentan como grandes retos y oportunidades para el concepto de *Smart Home*. *Smart Homes* se han convertido en un área de investigación muy popular, empleándose técnicas

computacionales avanzadas para la reducción del gasto energético, y permitiendo verla sostenibilidad desde la perspectiva del usuario. El consumo de energía es un aspecto de la vida en el hogar que no se ha tomado en cuenta, este descuido es perjudicial – entre 1973 y 2004, el consumo de energía creció a un ritmo mayor que el crecimiento de la población, como aparece en (Björkskog *et al.*, 2010). Los hogares son responsables de más del 40% del uso de energía en la mayoría de los países, como resultado, existe una necesidad urgente de desarrollar tecnologías que estudien el uso de energía en los hogares, para tratar de mejorar los comportamientos de eficiencia energética, así como desarrollar los dispositivos de eficiencia energética en los hogares.

- *Smart Customer*: está el concepto de introducir inteligencia en los hogares e incluso unirlos con *SG*, pero otro tema muy distinto es conseguir unos consumidores con conciencia de sus propias actuaciones, surgiendo el término *Smart Customer*, que trata de obtener el concepto Información y Conocimiento (*Information and Knowledge – I&K*), que consiste en crear conocimiento en los consumidores finales a través de la información, para obtener ahorro energético, mejorar en la eficiencia y crear conciencia medio ambiental, para lo cual se vuelve como indispensable, una vez más, el saber sus hábitos de consumo.
- De manera paralela a *SG*, surgen la *SC*, donde los espacios urbanos dependen de forma clara del conocimiento de la infraestructura física, y también de la posibilidad de comunicación de información y conocimiento hacia la sociedad que convive en la *SC*. Las ciudades digitalizadas y con una inteligencia orientada a la consecución de ciertos objetivos, permitirán también la creación de ciudades más sostenibles y de nuevos modelos urbanísticos, forjándose así nuevos paradigmas de ciudades con capacidad de crecimiento racional y de manera menos costosa, así como menos agresivas con el medio ambiente, al ir incorporando por ejemplo, más elementos de generación eléctrica renovable o transporte más eficiente y no contaminante. Podemos decir que las *SC* se centran en los siguientes pilares, pudiendo observar como algunos de ellos son tenidos en cuenta por *SG* también: Energía; Planificación urbana e infraestructuras *TIC*; Medio ambiente; Movilidad y transporte; Urbanismo; Personas y calidad de vida. Adicionalmente a *SC* y complementándolo, se está desarrollando en Europa en torno al concepto Inteligencia Urbana (*Smart Urban*), el proyecto “Espacios Urbanos Inteligentes” (“*Smart Urban Spaces*”) con ciudades piloto para ver la evolución de su desarrollo. Se trata esta vez de un proyecto global que plantea el interrogante del futuro de las ciudades una vez más, participando en el mismo expertos, para intentar dar como resultado final visiones, ideas y soluciones para que los estilos de vida sean sostenibles, con sociedades modernas y de futuro en los campos de la arquitectura, diseño y tecnología.



- Atomizando todavía más el problema, aparece el concepto de *Smart Building*, que además va muy asociado al término *Smart Customer*, presentando nuevos modelos de edificaciones que además de integrar fuentes de generación renovables, dispongan de inteligencia asociada para su operación y automatización en base a diferentes estrategias, que unas veces serán económicas y otras simplemente de autosuficiencia. Los *Smart Buildings* precisarán de información local de su entorno controlado, el propio edificio, y también variables y consignas exteriores, tanto de la propia red como de las condiciones climáticas existentes, para así poder realizar su funcionamiento de una manera eficiente y segura. En (Harris, 2012), se presenta varias definiciones sobre este concepto, pudiendo ser la más acertada “*Smart Building es un edificio con características cambiantes que pueden responder con un mínimo de intervención humana, para cambiar el entorno externo e interno, para el beneficio y la comodidad de los ocupantes, tomando en consideración la perspectiva económica y la reducción del uso de energía*”.
- Entorno Inteligente (*Smart Environment*) se define como aquel que es capaz de adquirir y aplicar el conocimiento sobre sus habitantes y su entorno físico para mejorar el rendimiento del mismo. *Smart Environment* percibe el estado del medio ambiente físico y los residentes por medio de sensores, razona sobre el estado utilizando diferentes técnicas de inteligencia, y actúa sobre el medio ambiente utilizando controladores. Se toman acciones para lograr un objetivo específico, como maximizar el confort de los residentes, lo que minimiza el consumo de recursos y el mantenimiento de la seguridad del edificio y los residentes. Las tecnologías *Smart Environment* se pueden utilizar en una variedad de formas para mejorarla eficiencia energética: mediante el análisis de uso de electricidad para identificar tendencias y anomalías; mediante la identificación de correlación entre el comportamiento humano y el consumo de energía, y mediante la automatización de los ambientes a ser más eficientes energéticamente. Además, el uso de inteligencia aplicada junto a sensorica se emplea en aplicaciones médicas para control de ancianos, tal y como se presenta en (Seelye *et al.*, 2012).

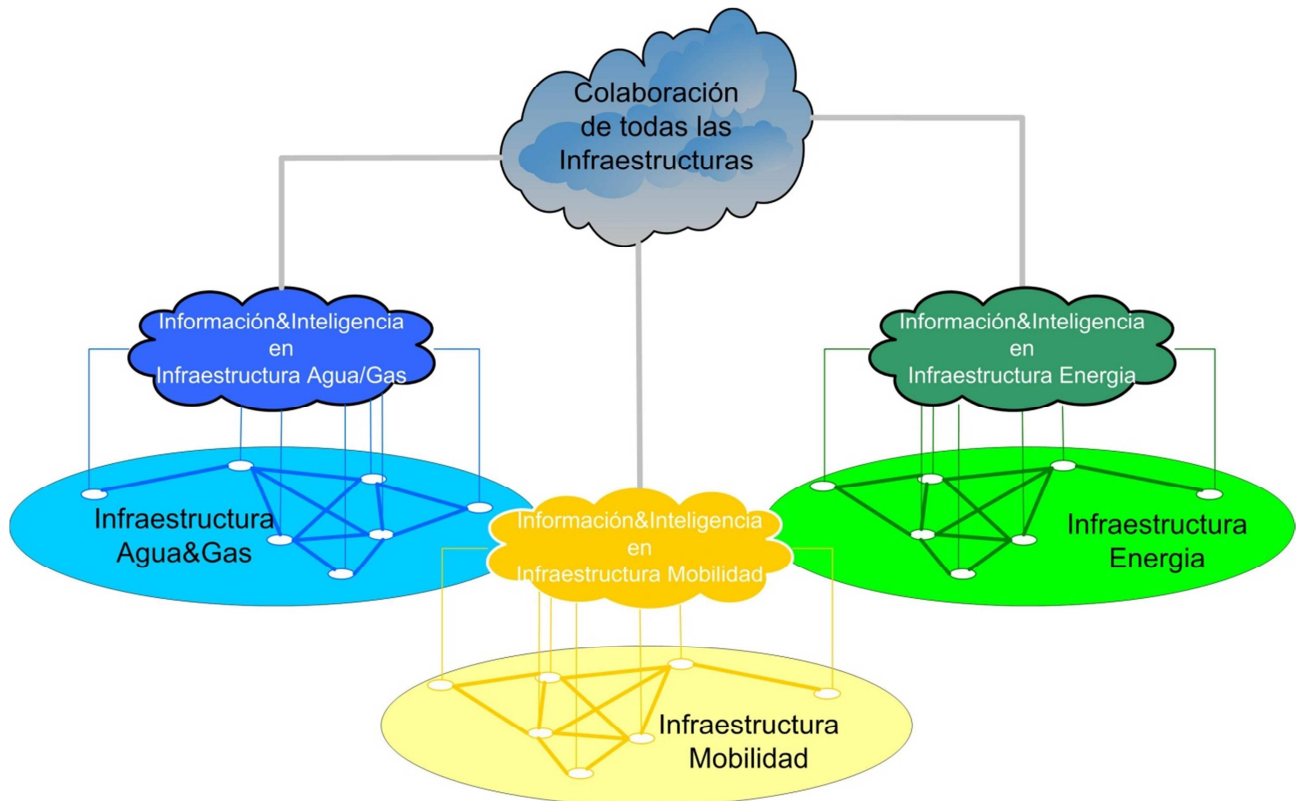
Llegado este punto, hagamos una reflexión a partir de los conceptos introducidos en esta y en anteriores secciones. *SG* es un término que, de manera resumida, introduce nuevos equipamientos en las redes eléctricas de transporte y distribución para una operación más eficiente de las mismas, adicionalmente, una *SC* es una ciudad con elementos de inteligencia, donde además del control de sus activos, se tratará de involucrar a todos los actores (sociedad, administración pública, medio ambiente, *EV*, etc.). Por tanto, se puede afirmar, que dentro del gran monstruo que suponen la *SG*, convivirán de manera coordinadas muchas *SCs*, cada una con su propia idiosincrasia, pero con la visión global de la *SG*.

Una *microgrid* no será otra cosa que un conjunto de *DG*, junto a almacenamiento, cargas, controles y la propia red, que tratan de operar de acuerdo a sus intereses, sin obviar la red de distribución. Por tanto, una *microgrid* se solapa conceptualmente con una *VPP*, independientemente de los niveles de tensión, y formará parte de la *SC* (podrá considerarse toda la *SC* como una *microgrid* o formada por muchas de éstas), y por tanto, de la *SG*. Continuando con la introspección, *Smart Home*, *Smart Building*, *Smart Environment*, etc., podrán considerarse formados por *microgrids* en mayor o menor escala, por tanto pertenecer a una *SC* y en definitiva parte intrínseca de la *SG*. A muy pequeña escala, y siempre que dispongan de elementos generadores y/o almacenadores, se les podrían considerar como mini-*VPP*.

A continuación, avanzaremos en la reflexión, para presentar un nuevo paradigma aglutinador de lo visto hasta el momento. Tanto *SG*, *SC*, *microgrid*, *Smart Building*, etc., tienen en común que son espacios con infraestructuras físicas desplegadas junto a elementos *TIC*. Una visión de futuro de estos espacios, independientemente de sus funciones y escenario físico, son lugares con sensores y *TIC* proporcionando inteligencia a las infraestructuras, y además, varias infraestructuras inteligentes colaborando mutuamente para proporcionar mejor calidad de vida de los habitantes y usuarios. Podemos ver esta situación en la Figura 2.8, donde por ejemplo la infraestructura de energía eléctrica, podría corresponderse con una *SG* o una *microgrid*, y estando a la vez cooperando con el resto de infraestructuras para una correcta convivencia dentro de una *SC*. De esta manera, se trata de unificar los objetivos individuales a nivel de infraestructura física existente en los diferentes modelos, para así obtener resultados finales con mayores potencialidades, ya que se permitirá, además de la mejor operación de las mismas, emplear la información inter-infraestructuras para un beneficio global que se sale del ámbito individual de cualquiera de las infraestructuras. Los sistemas de control e información colaborarán en conjunto para crear una infraestructura social, segura y agradable para con el medio ambiente y los ciudadanos, surgiendo de esta forma el concepto Lugar Inteligente (*Smart Place*).

Lo que se acaba de presentar es una sociedad futura que tratará de aprovechar la información procedente de infraestructuras aisladas para la consecución de unos objetivos claros que afecten a la mejora medio ambiental y sobre la propia ciudadanía. Trataremos de definir un nuevo concepto que abraque todavía más información disponible, pero antes de proseguir, hagamos una reflexión profunda sobre el concepto *Internet*, del cual la gente no tiene una idea muy clara de su significado, pudiendo ser esta duda la no existencia de una definición que resuma *Internet*, en cambio, existen varios puntos de vista para entenderlo. Por un lado, *Internet* podría ser definido en relación con sus protocolos comunes, como un conjunto de circuitos y rutinas, como un grupo de recursos compartidos o incluso como una habilidad para intercomunicarse; es decir, una red de redes. Otro enfoque quizás más adecuado, es plantear las redes como el medio a través del cual se envía y

acumula información, desde este punto de vista, *Internet* puede ser interpretado como la información y los servicios que circulan por esta red, en pocas palabras, un sistema distribuido de información, una red global de redes de ordenadores. Concluyendo, *Internet* es tanto un conjunto de comunidades como un conjunto de tecnologías, y su éxito se puede atribuir a la satisfacción de las necesidades básicas de la comunidad y a la utilización de ésta de un modo efectivo para impulsar la infraestructura, siendo a la vez una oportunidad de difusión mundial, un mecanismo de propagación de la información y un medio de colaboración e interacción entre los individuos y sus ordenadores, independientemente de su localización geográfica.



**Figura 2.8. Concepto futuro de Smart Place. Fuente Propia.**

Una vez expuestos todos los anteriores modelos, y pensando que surgirán nuevos precedidos por la palabra *Smart*, parece lógico tratar de unificar todos ellos bajo unas siglas que encuadren de manera clara su significado individual, y además, permita saber que la información disponible a partir de dicho modelo puede ser empleado para otros cometidos. Pongamos de nuevo la mirada sobre la *SG*, información proveniente de la infraestructura eléctrica cuyo principal cometido será realizar una operación más eficiente para la obtención de ahorro energético entre otros objetivos, pero esa misma realidad aparece en la *SC*, donde una de sus infraestructuras a controlar será la eléctrica. Además, una *SG* requiere realizar ajustadas predicciones de su demanda eléctrica, necesitando información de variables climáticas e incluso socioeconómicas, por lo que estamos retroalimentando a *SG* con información procedente de otros modelos. Estamos viendo por tanto, que las *Smart* se componen de una serie de comunidades interesadas en ciertos servicios y

aplicaciones para fines diversos, las cuales requieren para su funcionamiento información procedente de su propio modelo y de otras entradas al sistema que proceden de otros modelos diferentes, a través de tecnologías heterogéneas desplegadas por ciertas infraestructuras. Haciendo abstracción, las *Smart* simplemente son cajas negras que precisan información proveniente de diversos sistemas para la consecución clara de ciertos objetivos, pudiendo ser a su vez, algunos resultados de su funcionamiento, entradas para otros modelos. Pensemos de nuevo en la demanda eléctrica proveniente de una *SG*, asociémosla a la información de la generación renovable que una *SC* puede tener bajo su control, y la influencia de ambas en el comportamiento cambiante de ciertos indicadores medio ambientales que hagan evolucionar a dicha *SC* en un ranking de espacios sostenibles aún no escrito, o simplemente que sirva a *SC* para informar a sus ciudadanos de sus hábitos de generación y demanda, y la influencia en su entorno, desde una perspectiva medio ambiental, sea cada vez más saludable. El equipo investigador no está intentando presentar y definir Internet de la Energía (*Energy Internet*), introducido en (Huang *et al.*, 2011), donde presentan un escenario de un sistema de distribución eléctrica con consumidores y generaciones renovables, cuya información es transmitida de manera análoga al funcionamiento de *Internet*.

Por tanto, se define un nuevo vocablo conceptual que trate de englobar al conjunto de las *Smart*, de manera análoga al concepto de *Internet*, que como se ha visto es la interconexión de comunidades mundiales, las cuales deben de ser satisfechas a través de su interconexión, enviando y acumulando información, y ciertos servicios en un sistema distribuido. Además, será independiente de la infraestructura que se despliegue y de la tecnología empleada para transmitir la información y acciones de control, así como del lugar donde se instalen los sensores empleados. Este nuevo concepto recibirá el nombre de **Mundo Inteligente** (*Smart World*), y sus aplicaciones o servicios resultantes, independientemente del origen de su necesidad, cumplirán algunos o todos los objetivos de *Smart World*:

- Operación, mantenimiento, optimización y automatización de infraestructuras.
- Seguridad y protección.
- Movilidad y transporte.
- Urbanismo.
- Ahorro energético y eficiencia energética.
- Sostenibilidad.
- Evolución y control medio ambiental.
- Calidad de vida.

*Smart World* se representa en la Figura 2.9, donde se aprecia como la información fluye desde los sensores e inteligencia distribuida desplegada en cualquier infraestructura del planeta, bien sea en las terrestres ya desplegadas o las futuras por desplegar, en las oceánicas o en las

proporcionadas a través de los globos, aviones o satélites que monitorizan La Tierra. El concepto *Smart World* se nutre de información proveniente de: la superficie terrestre; ríos, lagos, mares y océanos; observación de La Tierra desde la atmósfera o fuera de ella. A partir de la información de *Smart World*, se plantea la creación de servicios, aplicaciones, nuevos protocolos de comunicación, tecnología y transferencia de información que puede ser explotada por los propietarios de la información o terceros, para el empleo en comunidades interesadas.

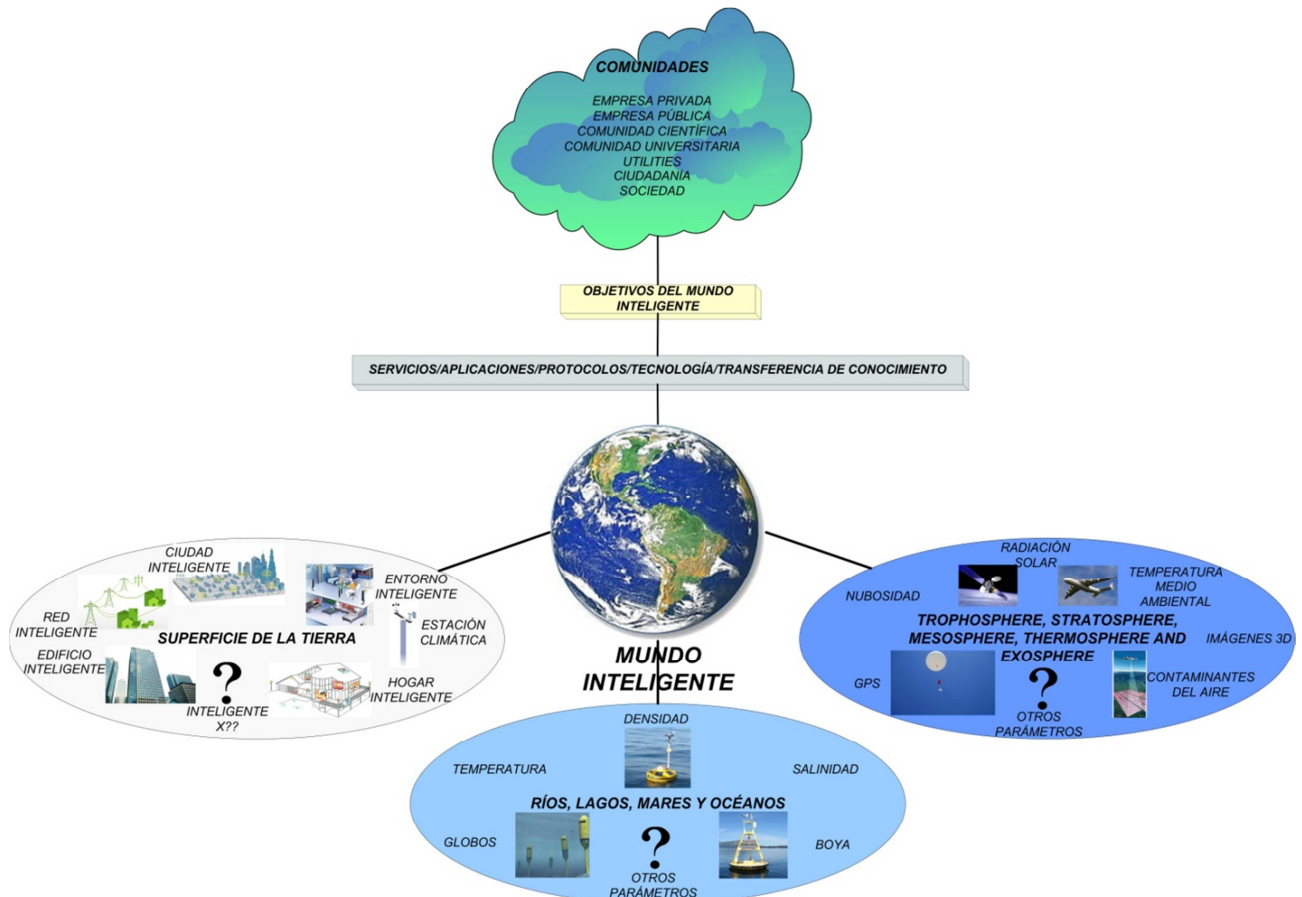


Figura 2.9. Concepto *Smart World*. Fuente Propia.

Supongamos una *SC* que desea ser conocedora de los índices medio ambientales y variables climáticas procedentes de observatorios meteorológicos, o que precisa una aplicación a medida que le permita, a partir de imágenes 3D obtenidas por *Light Detection and Ranging (LIDAR)* y cruzada con datos de radiación solar incidente procedente de satélites, disponer de un inventario de posible instalación fotovoltaica en los tejados de sus edificios y fachadas públicas de su ciudad, y que junto con la información procedente de la *SG* (red física eléctrica), poder informar a promotores instaladores de fotovoltaica de la posibilidad o no de ciertas implantaciones. En los ejemplos anteriores, se necesita información procedente de diferentes tecnologías e infraestructuras, así como de diferentes modelos *Smarts*, para cubrir algunos de los objetivos anteriormente detallados de *Smart World*, siendo los ejemplos presentados tan sólo algunos de los que a partir de ahora, tanto las *utilities*, las sociedades y las comunidades científicas van a requerir. Por lo tanto, *Smart World*

se convierte en el escenario mundial, donde convivirán diferentes tecnologías para comunicar información y actuaciones, y en el que comunidades dispares harán uso de dicha información con objetivos distintos pero a la vez perfectamente acotados dentro de los objetivos de *Smart World*, además, el progreso en I+D+i formalizará la creación de nuevos espacios *Smart*, que si se ajustan a los objetivos definidos por *Smart World*, quedarán bajo su influencia, y de esta manera se unificarán los criterios globales de la humanidad, sin por ello dejar de cubrir la necesidad específica del fin particular perseguido.

Los anteriores conceptos de *Smart Place* y *Smart World*, junto a otros temas que se presentarán en el Capítulo VI, se presentaron por el autor de esta *Tesis Doctoral* junto a sus colaboradores en (Hernández *et al.*, 2012a).

## CAPÍTULO III. LA DEMANDA ELÉCTRICA

En el presente Capítulo, se identificarán las principales características que definen la demanda eléctrica, así como la relación entre ella y la eficiencia energética del sistema eléctrico a través de la incertidumbre de la primera. Se mostrará el rol de los consumidores eléctricos con respecto a la eficiencia energética, el por qué se hace necesario el disponer de estimaciones o predicciones en cuanto a la demanda eléctrica, y para finalizar, una reflexión de la dificultad añadida de dichas predicciones en entornos desagregados. Se finalizará con una clasificación inicial de la predicción de la demanda, que posteriormente se ampliará en el Capítulo IV.

Antes de comenzar a detallar la demanda eléctrica, anticiparemos la importancia del conocimiento de la misma, tal y como muestran Rodríguez *et al.* (1998), la demanda eléctrica es totalmente dependiente de ciertos parámetros, incorporando en la misma cierta incertidumbre, que repercutirá en el precio final de la energía para los consumidores. Para tratar de reducir el coste, los gestores de los grandes sistemas de energía, necesitan conocer con precisión los mecanismos que hacen variar ostensiblemente esta demanda eléctrica, y de esta manera lograr dos objetivos principalmente: prever y adaptarse a la demanda en los diferentes plazos; tratar de gestionar la demanda.

También se expondrán las necesidades de disponer de predicción de la demanda eléctrica. Se mostrará la importancia de la misma en torno a los nuevos conceptos de futuro (*SG, DG, VPP, microgrid* y *Smart Building*). Se mostrarán los antecedentes históricos de la demanda eléctrica, así como los trabajos existentes en los citados entornos de futuro, de la misma forma que se presentará la dificultad añadida existente en estos entornos, debido principalmente a que al desagregar las curvas de carga, el perfil de la misma se hace más abrupto. Para finalizar, se clasificarán las predicciones de la demanda eléctrica, según los antecedentes vistos en la literatura.

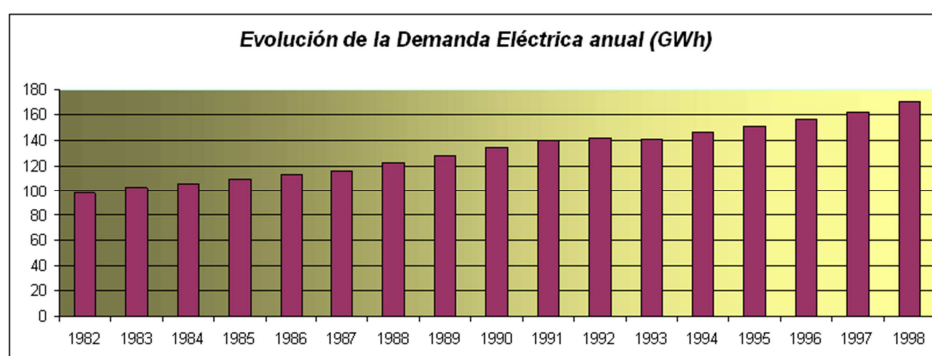
### 3.1. CARACTERÍSTICAS DE LA DEMANDA ELÉCTRICA

Existe una serie de características, que hace que el consumo eléctrico (demanda) sea muy diferente al resto de los bienes de consumo, siendo estas características las siguientes:

- La energía eléctrica no se puede almacenar, sin necesidad de dispositivos adicionales, por tanto, el consumidor eléctrico sólo demanda energía en el instante de tiempo que lo necesita, por tanto, la demanda eléctrica y su consumo coinciden en el tiempo.
- De manera teórica, el suministrador de energía eléctrica tampoco almacena la electricidad si no dispone de dispositivos para tal fin, por tanto, tanto la red como los generadores deben de estar siempre dispuestos para abastecer toda la demanda de ese instante, siendo de esta manera el sistema eléctrico dependiente de sus clientes.

- Además del consumo total de los clientes, el sistema eléctrico debe de tener información del momento en que se precisa el servicio, por tanto, la demanda eléctrica debemos decir que es dependiente del momento del día en que se solicita, además de depender de la estacionalidad.

La demanda eléctrica, además, es dependiente de factores sociales y dependientes del crecimiento del país, tal y como se muestra en (Rodríguez *et al.*, 1998), donde se muestra la evolución de España desde 1982 a 1997, representando en la Figura 3.1 la evolución en GWh de la demanda eléctrica, y en la Tabla 3.1 la demanda eléctrica de cada uno de los años y su tasa de crecimiento con respecto al año anterior.



**Figura 3.1. Evolución de la Demanda Eléctrica en el sistema peninsular desde 1982 a 1997. Fuente Propia con Datos de Rodríguez *et al.* (1998).**

**Tabla 3.1. Demanda eléctrica y tasa de variación anual desde 1982 a 1997. Fuente Propia con Datos de Rodríguez *et al.* (1998).**

<i>Año</i>	<i>Energía GWh</i>	<i>Tasa de variación anual (%)</i>
1982	98	2.40
1983	102	3.92
1984	106	3.77
1985	110	3.63
1986	113	2.65
1987	116	2.58
1988	122	4.91
1989	128	4.68
1990	134	4.47
1991	140	4.28
1992	141.50	1.06
1993	141	-0.35
1994	146	3.42
1995	151	3.31
1996	156	3.20
1997	162	3.70
<b>16 años</b>	<b>Media GWh: 126.15</b>	<b>Media %: 3.37</b>



El crecimiento no ha sido homogéneo, la demanda sufrió las situaciones económicas a las que se vio sometido el país:

- En 1986, el crecimiento se ralentiza al entrar España en la Comunidad Económica Europea, tras un periodo de recuperación tras las crisis de 1980.
- En el periodo de 1987–1991, la demanda aumenta acompañando a la bonanza de la economía.
- De 1992–1993 la economía entra en crisis, por lo que la demanda eléctrica se ve afectada, reduciendo su crecimiento.
- La economía se recupera de 1994–1997, viéndose en el aumento sostenido de la demanda eléctrica.

Además de la economía, la demanda eléctrica se ve claramente influenciada por la estacionalidad. La demanda se ve fuertemente incrementada en los meses invernales, y se suaviza en verano y entre tiempo. Hay que añadir, que la posible influencia del uso de aires acondicionados en verano, puede suponer un aumento de la demanda, principalmente el mes de julio, tratando de igualarse con los meses invernales.

Claramente la demanda eléctrica se ve influenciada por ciertas variables climáticas, en concreto, como se ha comentado anteriormente, la temperatura influye fuertemente en la variabilidad de la demanda mensual, los meses más fríos tendrán mayor demanda. Otros parámetros que afectarán a la demanda mensual serán claramente la laborabilidad, y tal y como se ha comentado, la coyuntura económica.

Por otro lado, la demanda diaria presenta grandes oscilaciones a lo largo del año, siendo las más importantes las siguientes:

- Temperatura ambiente.
- Actividad laboral.
- Evolución económica.

La evolución de la demanda diaria sigue un ciclo semanal, marcado por el nivel productivo en cada día de la semana. Los valores de demanda más elevados se presentan de martes a viernes, mientras que los fines de semana la demanda decrece, debido principalmente al ya comentado efecto de la producción industrial. El lunes, tiene una demanda inferior a los días de martes a viernes, aunque superior a los del fin de semana, posiblemente debido a la inercia del fin de semana y su demanda menor.

La demanda horaria, que producirá la curva de carga del sistema español, presenta una gráfica característica con dos puntas a lo largo del día, un profundo valle durante la noche y un segundo mínimo entre ambas puntas. Por norma general, la demanda horaria presenta un patrón estable para cada tipo de día (lunes, martes, etc.), según su laborabilidad y estacionalidad. No

obstante, a veces se separa, si se analiza por sector de la demanda (industrial, comercial, doméstico, etc.), e incluso con la luminosidad y temperatura. Las puntas históricas suelen producirse en momentos económicos alcistas y en periodos de frío.

En la Figura 3.2 se muestran las curvas de carga de los diferentes días de la semana, y separados por estacionalidad, mostrando en Figura 3.2a las curvas características de meses de verano, y en la Figura 3.2b las de invierno. En verano, el primer pico del día se produce aproximadamente a la 1:00 *post meridiem* (*p.m.* – después del mediodía), un poco más tarde para los domingos, un segundo pico hacia las 11:00 *p.m.*, y los consiguientes valles de madrugada y entre los picos anteriores. En el invierno, los días tienen curvas de carga más elevadas, a excepción del domingo que se mantiene muy similar al verano, el pico máximo se produce en el segundo pico (en verano era el primer pico el máximo), hacia las 8:00 *p.m.* (más temprano que en verano), mientras que el primer pico se produce hacia las 12:00 *ante meridiem* (*a.m.* – antes del mediodía) (anticipado igualmente), por el contrario, los valles se mantienen aproximadamente en el tiempo.

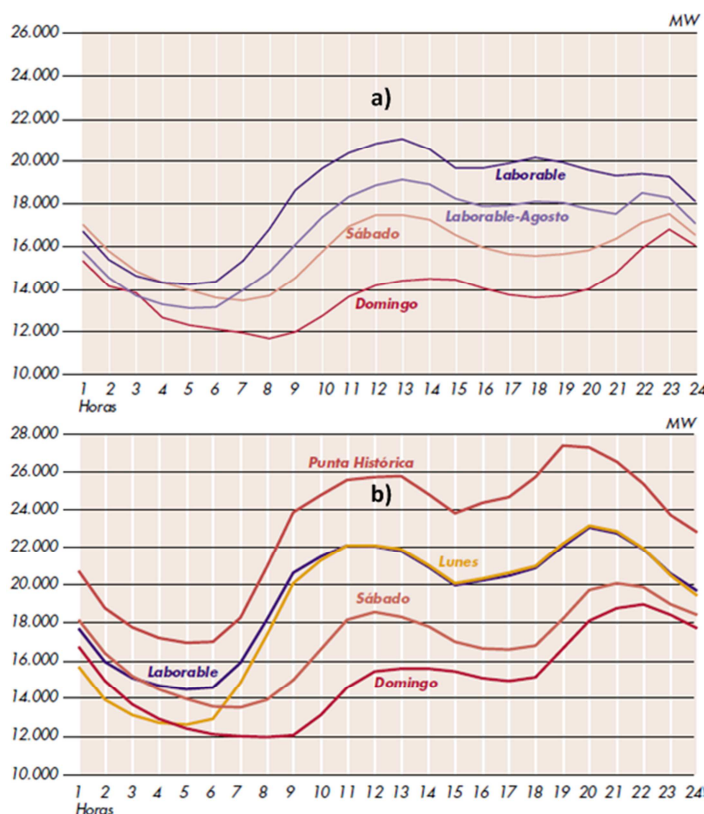


Figura 3.2. a) Curvas de carga de días de la semana de meses de verano; b) Curvas de carga de días de la semana de meses de invierno. Fuente Rodríguez *et al.* (1998).

A continuación, pasaremos a explicar algunos de los factores que afectan a la evolución de la demanda de España. De los cinco factores claves, tres de ellos corresponden con causas estacionales, y por tanto, no tienen tendencia reconocida en el tiempo. Los tres factores estacionales son:

- La temperatura ambiente que varía fuertemente en el corto plazo.

- La laborabilidad, y que además varía de un año a otro.
- La estructura estacional del sistema económico.

Despojando los tres anteriores, la demanda refleja el impacto de la actividad económica, denominando *SEDE* a la señal de la demanda eléctrica de la actividad económica, teniendo dos componentes claros:

- La evolución coyuntural, que no refleja cambios en la estructura económica, sino ajustes de la economía a corto plazo.
- Crecimiento y transformación de los sistemas productivos del país, así como de los hábitos sociales.

Para finalizar esta sección, se tratará de presentar las diferentes tipos de demanda existentes en España, presentando entre ellas claras diferencias. Los principales sectores, y que afectan a la curva de demanda de España, son:

- Sector industrial: es el sector que más consume energía eléctrica, aproximadamente entre un 60% y 70% del total. Las actividades extractivas, energéticas, metalúrgicas, químicas, manufactureras y las relativas a la construcción son las que más variabilidad aportan a la demanda.
- Sector doméstico: este sector se ve afectado claramente por el desarrollo económico del país, así como por las temperaturas invernales principalmente. La diferencia de consumo entre el invierno y el verano puede llegar a ser entre el 40% y 50%, siendo el consumo invernal claramente superior al estival.
- Sector comercial: los diferentes perfiles comerciales consumen electricidad de manera muy diferente, los establecimientos de restauración son los de mayor peso en cuanto a consumo dentro de este sector, pudiendo ser del 29% del total, les siguen los establecimientos de alimentación que pueden llegar al 20%, ya que disponen de equipos de frío y climatización.
- Sector turístico: el sector turístico tiene gran repercusión económica y en cuanto a empleo en España, es por este motivo que se suele contemplar como un sector clave cuando se analiza la demanda eléctrica. El consumo medio diario de verano puede llegar a triplicar el consumo medio diario del invierno, además, la curva de carga de verano tiene la peculiaridad que presenta tres puntas a lo largo del día, coincidiendo con las 9:00 *a.m.*, 1:00 *p.m.* y 8:00 *p.m.* del día.

### **3.2. LA EFICIENCIA DEL SISTEMA ELÉCTRICO Y LA INCERTIDUMBRE DE LA DEMANDA ELÉCTRICA**

La anterior sección pretendía presentar los rasgos más característicos de la demanda eléctrica en España, y además, presentar los principales sectores en que ésta se puede dividir. A

continuación, se tratará de acercar la demanda eléctrica, junto a la incertidumbre que asocia, a la eficiencia del sistema eléctrico.

La necesidad de nuevas instalaciones, los costes fijos y los variables del sistema eléctrico, están directamente influenciados por la curva de carga de la demanda eléctrica y las características de la misma. Por tanto, y de una manera claramente influyente, la predicción de la demanda eléctrica se presenta como fundamental, además de para los costes anteriormente mencionados, para temas de garantía de suministro y operación del sistema. El sistema eléctrico dispone de herramientas que le posibilitan reducir los anteriores costes derivados de la incertidumbre, siendo:

- Aumentar la capacidad de predecir la demanda futura. Hasta la fecha, era necesaria la colaboración de los consumidores finales de forma directa, mediante encuestas, participación en foros, etc.; en un futuro inmediato, con la instalación de medida electrónica en los consumos finales y puntos intermedios, esta información posibilitará disponer de herramientas de predicción más sofisticadas y fiables, y además, permitirá la creación de nuevas para otros modelos hasta ahora no contemplados (*microgrids, Smart Home, etc.*).
- Adaptarse a la incertidumbre de la demanda, para lo cual las compañías eléctricas deben conocer a la perfección las características de la misma, y además, deberán tratar de incorporar elementos de flexibilidad en su equipamiento y en sus procesos de explotación.
- Incentivar a los consumidores a que hagan un uso de la energía eléctrica de forma favorable a las necesidades de la explotación del sistema, principalmente por medio de la *Gestión de la Demanda Eléctrica (GDE)*. La *GDE* puede ser: directa, cuando el operador del sistema puede operar y programar el uso de la energía de los clientes; e indirecta, cuando los consumidores modifican su equipamiento y hábitos de consumo favoreciendo la eficiencia del suministro eléctrico y la de sus propios procesos, tal y como se muestra en (Ashok and Banerjee, 2001) y su cambio de hábitos en los periodos de consumo de una industria de molinos de harina.

Por tanto, el tratar de predecir la demanda eléctrica supondrá, para el sistema, una operación más eficiente del mismo. La curva de carga es la pieza fundamental de la operación, ya que permitirá la planificación de sus generaciones y, además, posibilitará tener una visión del crecimiento de las infraestructuras necesarias de una manera más realista.

### ***3.3. EL PAPEL DE LOS CONSUMIDORES ELÉCTRICOS ANTE LA EFICIENCIA ENERGÉTICA***

La totalidad de la producción de los bienes y servicios de un país dependen de manera casi exclusiva del uso de energía eléctrica. Para su producción y suministro, no sólo se incurre en costes económicos, sino también en costes medio ambientales y, además, se reducen las materias primas.

Por tanto, se hace necesario hacer un uso eficiente de la energía, es decir, reducir la cantidad de energía consumida por unidad de producto o servicio.

Para la obtención de una mayor eficiencia energética, se hace totalmente imprescindible el conocer la manera en que se irá demandando la electricidad en diferentes plazos de tiempo, permitiendo además, reducir los costes ligados a la producción. Los consumidores sólo podrán mejorar su eficiencia energética si tienen un conocimiento real de su uso, dicho de otra manera, en qué procesos o momentos del día la consumen y qué oportunidades tienen para su reducción.

Los responsables europeos y españoles de la *Administración Energética y Medio Ambiental*, están intensificando las medidas para incrementar la eficiencia energética, a través de:

- Por medio de la I+D+i.
- Por medio de la información al consumidor.
- Por medio de la incentivación económica.
- Por medio de la regulación.

La realidad es que para los consumidores españoles la eficiencia energética es un objetivo menos asumido y claro que otros de protección medio ambiental, como por ejemplo el ahorro de agua o la clasificación de los residuos. Por norma, se reconoce de manera abierta un alto nivel de despilfarro energético en los hogares o negocios de dichos consumidores, admitiendo también el no saber en qué se emplea la energía (falta de información y entendimiento de la misma), por lo que se considera difícil hacer un control de su gasto. Lo anterior es un patrón que se repite en los hogares domésticos y en las empresas de servicios, no siendo tan claro en las empresas, aunque la falta de información en estas últimas, sigue haciendo complicado el control del gasto y por tanto el hacer uso de la energía eléctrica de manera eficiente, tanto desde la perspectiva de consumidor, como desde la del sistema.

Más del 65% de los hogares españoles practican actividades activas para proteger el medio ambiente (clasificar residuos, control de agua, etc.), estas prácticas que económicamente no les son rentables y que suponen un esfuerzo para el ciudadano, no suelen ser empleadas en el ámbito de la energía eléctrica. Además, debería entenderse que la rentabilidad económica no debería ser la única motivación que moviera a los ciudadanos a realizar buenas prácticas del uso de la energía eléctrica.

Por tanto, vuelve a quedar patente que se precisa de información apropiada, además de su entendimiento, para que los consumidores finales puedan conseguir hacer *GDE*. La curva de carga, vuelve a ser primordial para conseguir lo anterior, además de ser fundamental para el funcionamiento del sistema como ya se mencionó en la sección anterior.

### **3.4. NECESIDADES DE LA PREDICCIÓN DE LA DEMANDA ELÉCTRICA: ANTECEDENTES**

La energía eléctrica es un bien indispensable y de interés estratégico en la economía de un país, por esta razón, las *utilities* deben tratar de adecuar la producción a la demanda existente con la finalidad de garantizar a sus clientes la calidad de servicio contratada a un precio competitivo. Es por tanto importante el tratar de predecir la demanda eléctrica con un error lo menor posible, sin embargo el consumo de energía eléctrica depende de muchos factores como el día de la semana, mes del año, etc., esto hace que la predicción sea un problema muy complejo que requiere de métodos que se alejan de los puramente estadísticos. Es posible considerar que una predicción con un 10% Error Absoluto Porcentual de la Media (*Mean Absolute Percent Error – MAPE*) puede ser fácil de conseguir y asumible, sin embargo el coste del error puede ser tan elevado que esfuerzos en investigar reducciones de unos pocos puntos porcentuales, estarán más que justificados.

En España, en el mercado diario de casación las compañías tienen la obligación de dar previsiones de la demanda que será empleada para casar la energía y así obtener el precio por hora. En el mercado intra-diario se negocian cantidades adicionales para intentar responder a los ajustes posteriores entre oferta y demanda, consecuencia del mercado diario. Todos los mercados funcionan de la misma manera, los agentes envían electrónicamente sus ofertas al ordenador del operador responsable, éste realiza la casación de ofertas determinando el precio y las cantidades de energía aceptadas de cada oferta, y a continuación, se comunican los resultados a todos los agentes. A partir de esto, la *utility* puede tratar de influir en el precio de sus clientes mediante diferentes incentivos, como se muestra en (Ram, 1995). En contrapartida, en (Ilic *et al.*, 2003) se presenta una simulación de un modelo híbrido para un nuevo modelo de mercado eléctrico, que emplea señales de precio de transmisión, con la finalidad de una operación óptima del sistema y del mercado. Herramientas de simulación de mercados, como la presentada en (Yu *et al.*, 2010) mediante *MAS*, permite estudiar la interacción entre los diferentes productores de un entorno, tanto entre ellos como con el mercado, comprobando además la evolución de los precios consecuencia de la entrada y salida del mercado de los diferentes productores.

A la ya comentada necesidad de ajuste entre generación y demanda, para su acertada participación en los diferentes mercados, aparecen los nuevos entornos de generación y demanda más reducidos llamados *microgrids*, lugares con consumos más desagregados que los que nos podríamos encontrar en las curvas clásicas de demanda de un país al completo, por lo que sigue siendo necesario el estudio de modelos que mejoren dichas predicciones.

Como ejemplo, véanse los polígonos industriales, que suponen una de las herramientas principales de competitividad de un país, ya que en ellos se gesta el tejido industrial que lo compone. Es por ello, que en estos espacios considerados *microgrids*, como muestran Fernández *et*

al. (2010), será necesario el control y gestión por parte de las nuevas figuras llamadas *agregadores*, que entre otras muchas cosas intentarán hacer *DR* y así obtener precios baratos de la energía eléctrica para sus consumidores. La *DR* no se concibe sin que exista un conocimiento detallado del comportamiento del polígono industrial, es por esto que se precisan nuevos mecanismos de predicción de la demanda eléctrica en estas *microgrids*.

En la mayoría de los casos, en algunas obras que servirán como antecedentes de la *Tesis Doctoral*, la previsión se realizaba por un experto humano, que trataba de analizar las curvas de carga de días y años pasados. Aunque esta solución da resultados satisfactorios, en muchos casos es aconsejable la automatización del proceso, por varios motivos:

- Realizar predicciones con intervalos de tiempo menores.
- Realizar predicciones sin depender de uno o varios individuos concretos.
- Poder realizar mediciones en tiempo real de factores meteorológicos que posibiliten ligeras modificaciones sobre los cálculos ya realizados.

### ***3.4.1. La importancia de la predicción de la demanda eléctrica***

A medida que las redes comenzaron a crecer, empezó a ser muy complejo su control, además de ser ineficientes debido a las pérdidas. Hay que añadir, que hasta ese momento, los consumidores finales, con excepción de los grandes consumidores debido a que reciben un trato diferente, no son considerados en el sistema, salvo para el pago de la factura, que además en la mayoría de los casos no es una facturación acorde con la realidad, ya que está basada en estimaciones, disponiendo de un periodo de tiempo la comercializadora para la realización de la misma.

Desde hace algunos años, han ido apareciendo nuevos actores en el panorama del sistema eléctrico que han propiciado que las predicciones de la demanda eléctrica sigan siendo de especial interés. Además de la demanda, la atención se ha puesto también en los modelos para predecir la generación eléctrica, particularizándose en las fuentes de generación renovable que a su vez dependen de la predicción del recurso energético (radiación solar, viento, etc.).

A continuación se introducirán los principales cambios y actores acaecidos en el sistema eléctrico y su implicación con la demanda eléctrica.

#### ***3.4.1.1. Smart Grid***

Los principales requerimientos de una red de energía eléctrica del futuro pueden resumirse en: flexibles; las redes deben satisfacer las necesidades de los clientes; deben responder a una medida de la incertidumbre (cambios y retos futuros); accesibilidad, incluyendo las fuentes de generación renovable y una alta penetración de las unidades locales de generación que están

comenzando a proliferar; fiabilidad, las redes deben garantizar la fiabilidad y calidad de suministro; la competencia y la regularización darán lugar al desarrollo económico.

Los requerimientos antes mencionados conducen a los siguientes objetivos *SG*: soluciones técnicas y nuevos protocolos; servicios y telecomunicaciones para los consumidores; definición de marcos regulatorios; y aprovechamiento de infraestructura pasada.

La importancia de la fiabilidad, consideración económica y, además, la importancia del cliente final, harán como fundamental el empleo de modelos de predicción para que la *utility* obtenga energía al mejor coste, y se garantice una calidad de suministro óptima. El disponer de información en tiempo real y con patrones de consumo más acorde con la realidad, posibilitará disponer de modelos predictivos más evolucionados. Además, esta constante evolución de la información, supondrá el disponer ajustes de las predicciones de manera *on-line*.

#### **3.4.1.2. Distributed Generation**

La red del futuro deberá acoger tanto las plantas clásicas de generación (pocas y de elevada potencia), como los nuevos generadores *DG*. Este nuevo escenario estará necesitado de algoritmos optimizadores para el control y la programación.

*DG* añade una incertidumbre adicional al problema del control de las redes. Por un lado se deberá estimar la predicción de la demanda bajo la operación de la red, pero teniendo en cuenta la generación bajo su control. Si las fuentes de *DG* son renovables, la aleatoriedad del recurso hará que los modelos se compliquen, viéndose afectados los resultados de los modelos predictivos de la demanda por las estimaciones obtenidas por las fuentes de generación renovable.

#### **3.4.1.3. Virtual Power Plant**

*VPP* dispondrá de una gran cantidad de generadores separados físicamente, los cuales deberán de ser monitorizados y controlados. Los diferentes elementos deberán poder comunicarse y a partir de esta comunicación poder tomar decisiones.

Por tanto, el concepto *VPP* supone la coordinación y el control de generadores de pequeña potencia y de manera distribuida. Además, y con independencia de la ofertas al mercado eléctrico por parte de *VPP*, ésta deberá de tener en cuenta la demanda bajo su supervisión. De esta manera, se vuelve a poner de manifiesto la necesidad de modelos estimadores de la demanda eléctrica y de su generación.

#### **3.4.1.4. Microgrid**

En una *microgrid* conviven elementos *DG* (renovables o no) junto a cargas y almacenamiento. Uno de los objetivos principales de estos entornos es operar los anteriores elementos de una manera óptima y respetuosa con el medio ambiente. Para poder programar la



operación de los generadores gestionables, se hace indispensable el conocimiento presente y de futuro de las cargas bajo su control. Además, las decisiones de carga o descarga del almacenamiento (si existe) estarán fundamentadas en lo anterior.

Con independencia del tamaño de la *microgrid*, será condición necesaria para su correcta operación y su interacción con la red de distribución el conocimiento de la demanda y la generación en su dominio. Por tanto, el desarrollo de aplicaciones donde los modelos predictivos supongan uno de los pilares fundamentales se presenta como prioritario.

#### **3.4.1.5. *Smart Building and Smart Environment***

La inteligencia desplegada en los edificios, junto a la integración de fuentes de generación renovable, ha supuesto la aparición de nuevos conceptos, como son el de *Smart Building* y de *Smart Environment*. El “*modus operandi*” de estos nuevos entornos supone el despliegue de sensórica e inteligencia, de manera interconectada, con la finalidad de obtener eficiencia energética (*Smart Building*) y tratar de hacer la vida de los usuarios más fácil (*Smart Environment*), entre los principales objetivos.

En los *Smart Buildings*, la integración de fuentes de generación renovables y la interacción del propio edificio con la red eléctrica, supondrá la necesidad de modelos de predicción, tanto del lado de la demanda como de la propia generación. Dichos modelos, junto a otros módulos Software, deberán ser integrados en el Sistema de Gestión de Energía (*Energy Management System – EMS*). En cuanto a los *Smart Environments*, el disponer de información del comportamiento energético (electricidad, agua, gas, etc.) de los habitantes del entorno es fundamental, por tanto, vuelve a suponer una necesidad para los modelos predictivos y su aplicación directa a las variables concretas del entorno a controlar.

#### **3.4.2. *Enfoque histórico de la demanda***

Ya desde los orígenes de la invención del sistema de transporte de energía eléctrica se intentó comprender y analizar el sistema (sencillo en aquellos momentos pero tan difícil de gestionar y controlar en nuestros días). Hamilton (1944) presentó un estudio de la adición de curvas para la determinación de la caracterización de la curva de la demanda, siendo los puntos principales la carga pico, carga valle, factor de carga o instante del pico, y presentando la predicción de la carga como un reto de futuro para la operación.

Forrest (1946) presentó la predicción como elemento fundamental para evitar roturas de instalaciones eléctricas debido a anomalías climáticas y, el conocimiento de la variación de carga con las condiciones climáticas, el tratamiento de estadísticos de datos pasados y la creación de predicciones futuras, tanto de la carga como de las variables climáticas. Se comenzaba a estudiar la

demanda como elemento de prevención de averías en las instalaciones, y sentando las bases de la creación del conocimiento alrededor de la demanda.

Al comienzo, los modelos de predicción se apoyaban en los avances en estadística, Rowson (1952) mostró una predicción de la carga en base a diferentes funciones estadísticas y, además, un gran trabajo de análisis estadístico de la influencia de la temperatura en la demanda eléctrica, para trece países y, con los últimos 50 años (1900–1950). Continuando con el aporte estadístico, Hooke and Newark (1955) presentaron un estudio de modelos que extrapolan la predicción a partir de días anteriores junto otros modelos más avanzados para la predicción de cargas futuras que una empresa tendría en el futuro, tratando de estudiar parámetros como la influencia de la población en la demanda, consumo por individuo, entre otros. Gruetter (1955) empleó tarjetas perforadas de la administración pública para la predicción de la carga, basada en unas medias estadísticas de comportamientos de consumos. La información de las tarjetas se emplearon a modo de “*hoja de cálculo*” para estimar la demanda de un área extensa.

Matthewman and Nicholson (1968) identificaron el gran esfuerzo en los últimos años alrededor del análisis de flujo de carga, programación de la generación económica y la comprobación de sistema de seguridad en sistemas de energía eléctrica, pero detectaron el poco trabajo en relación a la predicción de la demanda eléctrica. En la obra se repasaron los pocos modelos hasta la fecha, categorizados en dos niveles: métodos que utilizan la predicción climática e información meteorológica; y los que utilizan únicamente los datos de carga.

El arte de predicción de la demanda eléctrica con un error lo menor posible se convierte en labor esencial a partir de entonces. Se debe recordar que la predicción de la demanda se considera prioritaria para el buen funcionamiento del sistema eléctrico, además, la demanda eléctrica está claramente influenciada por factores económicos, temporales, climáticos y efectos aleatorios, tal y como muestran Gross and Galiana (1987).

### ***3.4.3. Predicción de la demanda en los entornos de futuro***

Como ya se ha comentado, el modelo del sistema eléctrico ha comenzado a cambiar en los últimos años, al ir introduciendo inteligencia en los diferentes niveles del sistema eléctrico, comenzando por el transporte, en los últimos años en la distribución, y aunque en menor medida, recientemente en las proximidades al consumo final. A partir de esto, se tratará de optimizar los flujos de energía, mejorar la algoritmia de detección de faltas, y por tanto, mejorar en la operación de la red, como se exponen Fang *et al.* (2012).

Como presentan Hernández *et al.* (2012a), los objetivos de los nuevos entornos pueden ser aglutinados en torno a un nuevo concepto llamado *Smart World*. Alrededor de este nuevo paradigma, se posibilitará la creación de nuevas aplicaciones, herramientas y servicios, como por

ejemplo aplicaciones que determinen las relaciones entre ciertas variables climáticas y la demanda eléctrica, con el fin de mejorar las estimaciones de la demanda.

En resumen, la aparición de nuevos entornos de futuro, según Werbos (2011), requerirán de inteligencia distribuida y, entre otras cosas, nuevos modelos y aplicaciones basados en ANN, no únicamente para el problema de la predicción de la demanda, sino para la integración de nuevos actores que aparecen en el sistema. Tal y como apuntan Yan *et al.* (2012), además de la inteligencia distribuida, tanto las comunicaciones digitales como el control, deberá realizarse de manera segura. No debemos olvidar lo presentado por Javed *et al.* (2012), que exponen la necesidad de nuevos modelos de predicción que hagan frente a las necesidades de la DR.

Esta necesidad de disponer de modelos de predicción (demanda eléctrica y generación eléctrica), aplicados a estos entornos concretos ya comentados, supondrá la posibilidad de emplear variables locales que puedan afectar directamente al comportamiento de la demanda y/o generación. Estas variables de tipo local (climáticas, sociales, económicas, hábitos, etc.) supondrán mayor complejidad al modelo global pero su objetivo será el de obtener estimaciones con mayor precisión.

Por el contrario, y volviendo a hablar de la demanda eléctrica, la desagregación que supondrá el tratar que obtener estimaciones de la demanda eléctrica en estos nuevos entornos (*microgrids*, *Smart Building*, *Smart Environment*, etc.), llevará asociada una mayor dificultad en la estimación. Por ejemplo, la curva de carga de un país presenta un perfil mucho más predecible y más suave que la que se pueda tener en un entorno desagregado (subestación eléctrica, *microgrid*, *Smart Building*, etc.), además, las curvas de carga en estos entornos desagregados presentan unos perfiles más abruptos y en ocasiones totalmente atípicos, que los encontrados a medida que se vaya agregando las curvas de carga. Por tanto, si los modelos disponen de mayor información que ayude a detectar lo anterior, se conseguirán estimaciones más precisas del entorno a analizar.

La demanda eléctrica está claramente sectorizada (industrial, comercial, doméstica, administración, etc.), por tanto, al poder desagregarla y centrarla en un entorno concreto, posibilitará el tener un entendimiento más completo de la misma por sectores. Lo anterior, posibilitará el hacer entre otras cosas DR al poder ofertar precios de energía en base a los criterios establecidos por la comercializadora. Para poder disponer de este conocimiento sectorizado, se precisará tener una idea completa de los hábitos de consumo, para posteriormente poder hacer estimaciones con respecto a ellos.

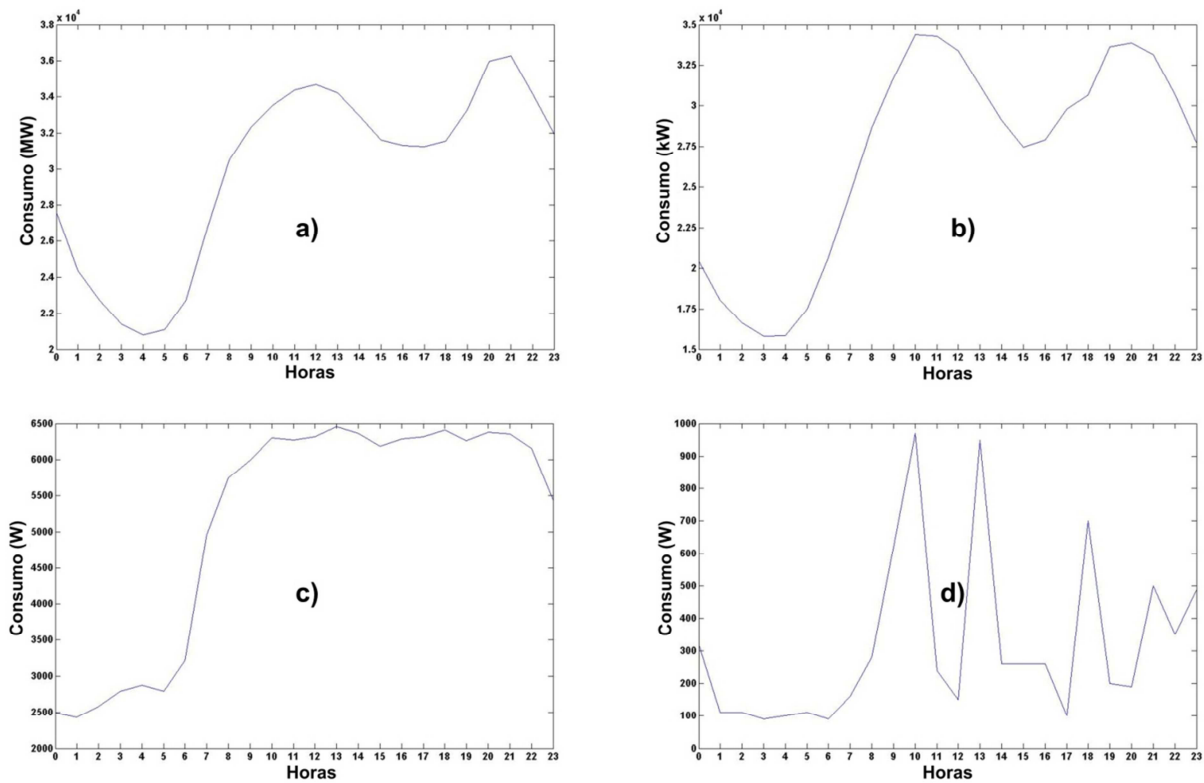
### **3.5. LA DIFICULTAD DE LA PREDICCIÓN DE LA DEMANDA ELÉCTRICA EN ENTORNOS DESAGREGADOS: VISIÓN DE FUTURO**

Una vez vista la necesidad de conocer los hábitos de la demanda, y teniendo clara la importancia de la predicción de la misma, vamos a ver como la predicción se dificulta a medida que

se pretenda estimar la demanda de espacios con menores consumos, o dicho de otra manera, cuando se comienza a desagregar la demanda. En la Figura 3.3, se representa la curva de carga diaria de la demanda correspondiente al 01/03/2010 (lunes y laborable), donde: a) muestra la de España; b) la de una capital pequeña de provincia; c) un polígono industrial; y d) un cliente doméstico con potencia contratada típica de 4.4 kW.

Con independencia de la escala en el consumo de potencia, el país entero en MW, ciudad de tamaño medio y polígono industrial en kW, y cliente doméstico en W, de las anteriores curvas de carga representadas en la Figura 3.3 se pueden obtener algunas conclusiones:

- A medida que vamos desagregando la demanda del emplazamiento controlado (país, ciudad, entorno reducido, hogar, etc.), claramente vemos como las pendientes de la curva de carga se hacen más abruptas y aparecen formas con más picos que harán que su estimación sea más complicada. Véase en la Figura 3.3, como b) comienza a ser no tan suave con respecto a a), c) además de tener una forma de curva totalmente diferente a a) y b) añade más oscilaciones con respecto a la curva suave que se presentaba en a), y en d) podemos decir que los perfiles son carentes de pendientes suaves. Esto sugiere, que con la aparición de nuevos entornos con inteligencia (*microgrids*, *Smart Buildings*, etc.), será necesario el desarrollo de nuevos modelos de predicción *Ad hoc*, tanto para la demanda eléctrica como para su potencial generación, en el caso de que exista.
- La forma de la curva de carga puede diferir entre entornos, a pesar de ser el mismo día. A pesar de las diferencias presentadas en el anterior puntos en cuanto a suavidad, a) y b) presentan una curva de carga con similitudes de forma, ambas presentan dos picos y dos valles claramente diferenciados a lo largo del día. En cambio, c) presenta una forma de curva totalmente distinta a lo esperado, tiene fácil explicación, al ser un lunes los polígonos industriales comienzan su labor productiva, por lo que después de un fin de semana donde la curva de carga comienza a decaer en cuanto a demanda a partir de la mitad del sábado, las primeras horas del lunes la demanda comienza a repuntar, manteniendo la curva casi plana a lo largo del día, con los ya consabidos altibajos (picos en la zona plana) consecuencia del arranque y parada de maquinaria particular de dicho polígono. Si nos fijamos en d), y tratamos de hacer una abstracción con respecto a la curva a), podemos ver que en esencia es una curva con ciertos rasgos similares, ya que en d) aparecen dos picos a las 9:00 y 13:00, coincidiendo con los momentos de vida en el hogar, presentando entre ellos un valle, ya que a diferencia de un país, en un hogar no hay una demanda productiva que llene ese valle durante el rango de horas de los dos picos anteriormente comentados, en cuanto a la tarde, un proceso similar al explicado se produce en este periodo.



**Figura 3.3. Curvas de Carga del 01/03/2010 de diferentes entornos: a) Curva de Carga Diaria de España; b) Curva de Carga Diaria de ciudad española; c) Curva de Carga Diaria de Polígono Industrial; d) Curva de Carga Diaria de un Cliente Doméstico. Fuente Propia.**

Las diferencias presentadas anteriormente se mantienen para cualquier día del año, e incluso si hubiéramos seleccionado otro día de la semana, se hubieran presentado evidentes diferencias, que en esencia son las arriba indicadas.

Como resumen de lo presentado y para finalizar este capítulo, podemos destacar algunas evidencias. Lo primero es que la aparición de nuevos entornos de futuro, como el caso de las *microgrids*, que además irán acompañados de inteligencia y medida desplegada, se presentan como lugares con una demanda desagregada con respecto a lo que se ha tratado de controlar hasta la fecha, que no ha sido otra cosa que países enteros o grandes extensiones, ambos casos con consumos muy superiores (del orden de mil o un millón de veces más) a los que nos encontraremos en ellos.

En segundo lugar, y como consecuencia de lo anterior, tal y como se ha presentado aquí la desagregación llevará implícito cierta complejidad añadida a la labor de las predicciones, por lo que se deberá continuar trabajando en modelos que mejoren las estimaciones, y proponiendo nuevos para estos entornos emergentes. Además, si estos entornos llevan asociados *DG*, los modelos encaminados para el control se hacen estrictamente esenciales, dentro de este control, los módulos de predicción (generación y demanda) se presentan como fundamentales.

Por último, y pensando en la figura del *agregador*, ciertos emplazamientos como puedan ser los polígonos industriales, los cuales presentan especiales características con respecto a sus curvas

de carga, requerirán herramientas que permitan hacer una planificación más exacta de sus recursos, por lo que el desarrollo de las mismas se presenta como un reto de futuro y, por tanto, de mucho interés desde la perspectiva de la investigación.

### **3.6. CLASIFICACIÓN DE LA PREDICCIÓN DE LA DEMANDA**

En esta sección se presentarán las posibles clasificaciones de los modelos de predicción de la demanda. Debido a que en el Capítulo IV se presentarán los modelos basados en técnicas lineales y no lineales respectivamente, en la presente sección se dará una clasificación desde dos perspectivas, en concreto, se clasificará en cuanto al horizonte de la predicción y el objetivo de la misma. Según los antecedentes encontrados en la literatura, esta clasificación en función del horizonte y el objetivo son los más empleados.

#### **3.6.1. Clasificación según el horizonte de predicción**

Tal y como recogían Matthewman and Nicholson (1968) en su título al referirse a la predicción de la demanda, existe un concepto claro cuando hablamos de predicción, y es *STLF*, que viene a significar obtener la predicción de la demanda eléctrica a “*corto plazo*”. Siguiendo a Hippert *et al.* (2001), la predicción de la demanda eléctrica puede clasificarse bajo criterios diferentes. Fijándonos en el intervalo a predecir, lo que normalmente se conoce como horizonte de predicción, distinguimos:

- Predicción de Carga a Muy Corto Plazo (*Very Short-Term Load Forecasting – VSTLF*): desde algunos segundos o minutos a varias horas. La aplicación de estos modelos suele ser para control de flujo.
- *STLF*: desde unas horas a semanas. La aplicación de estos modelos suele ser para ajuste entre demanda y generación, y por tanto la emisión de ofertas al mercado eléctrico.
- Predicción de Carga a Medio Plazo (*Medium-Term Load Forecasting – MTLF*) y Predicción de Carga a Largo Plazo (*Long-Term Load Forecasting – LTLF*): desde meses a años. La aplicación de estos modelos suele ser para la planificación de los activos de las *utilities*.

Los horizontes de predicción más importantes son el semanal, diario y horario. El tener una buena predicción de las siguientes 24 horas juega un papel vital para las compañías, ya que puede condicionar el funcionamiento óptimo de la planificación horaria de las unidades de generación, compra/venta en sistemas de intercambio, etc.; 24 valores de carga del día siguiente (llamado “*perfil de carga*” o “*curva de carga*” diaria). Pese a esto, Toyoda *et al.* (1970) presentan modelos de predicción para todos los horizontes de anteriormente citados.

Con independencia del modelo, que se detallará en el próximo Capítulo, la principal diferencia en estos modelos radica en el alcance de las variables a emplear. En *VSTLF* emplean

entradas recientes al modelo (minutos u horas), en *STLF* emplean entradas de días y para *MTLF* y *LTLF* valores de semanas o incluso meses. A continuación se detallarán algunos trabajos en relación a *VSTLF*, *MTLF* y *LTLF*; en esta sección no se presentan trabajos de *STLF* ya que irán apareciendo muchas obras en el resto de secciones.

Charytoniuk and Chen (2000) presentan un modelo para *VSTLF* basado en *ANN* para aplicaciones de entornos dinámicos, y por tanto, con entornos cambiantes; calculan mediante 5 redes diferentes, cinco intervalos de tiempo separados 10 minutos de tiempo, empleando además de la hora del día, tres valores de consumo de los tres intervalos inmediatamente anteriores. Guan *et al.* (2013) presentan un modelo basado en *ANN* con unos filtros previos basados en Wavelets; la Red Neuronal Wavelet (*Wavelet Neural Network – WNN*) recibe los filtrados de Wavelet, valores de calendario (hora, mes, etc.) y valores de consumo de la última hora; emplea 12 *WNN* dedicadas para predecir el valor de consumo de la hora siguiente, donde cada *WNN* predice el valor 5 minutil.

Doveh *et al.* (1999) presentan diferentes *ANN* para la obtención de *MTLF*, donde se realiza la predicción de casi un año. Las variables de entrada que se presentan al modelo son variadas: dos observaciones de la temperatura y sus correspondientes fechas de toda la serie que se introduce al modelo; cuatro variables cuyo significado es el tipo de día al que corresponde la información (día de la semana); el efecto estacional del año es introducido mediante un indicador fuzzy; factores sociales y económicos. Como dato adicional, los autores indican como problemático en los modelos *MTLF*, el no disponer de una serie continua de datos (valores de la carga), ya que el modelo precisa de una ventana de datos considerable y por tanto se precisa que dichos datos sean buenos.

Asber *et al.* (2007) demuestran la forma de emplear un marco de modelado de carga para extraer la esencia de los problemas de modelado *MTLF* y poder obtener modelos de aplicaciones; destaca la importancia de las curvas de carga de las semanas anteriores para afrontar este problema, así como la influencia climática de dichas semanas.

Kandil *et al.* (2002) presentan un sistema experto para manejar un sistema planificador de *MTLF/LTLF*; la base de conocimiento del sistema experto estará formada por hechos estáticos y dinámicos. Los hechos estáticos se refieren a información de la situación actual, interconexión con otros sistemas, etc.; empleará información de los patrones de carga para detectar ciclicidades, históricos de picos de carga anuales, histórico de consumidores de energía para detectar consumidores por sectores y los principales datos económicos, sociales y climáticos de interés. Los hechos dinámicos tienen que ver con los atributos de la carga y la energía (relaciones con las estimaciones a hacer), sistemas de pérdidas y el propio error del sistema estimador. Además de esta gran cantidad de variables de entrada al modelo, se necesitará por tanto la actuación de expertos humanos para la definición de las funciones de cambio de dominio.

Zhang and Ye (2011) emplean un modelo ANN de regresión para *LTLF*, tratando de incluir un factor económico llamado Producto Interior bruto (*Gross Domestic Product – GDP*). Como dato de interés, el trabajo necesita un histórico de 13 años para la fase de aprendizaje del modelo, para posteriormente poder predecir 5 años. Según los autores, en estos modelos para predecir *LTLF*, se precisa información complementaria a las series temporales de carga, esta información es aportada por variables económicas, en concreto, los autores la introducen por medio de *GDP*.

Daneshi *et al.* (2008) presentan dos modelos para *LTLF*. El primero emplea un método de regresión lineal donde la complicación es construir la ecuación del modelo de regresión. El segundo modelo emplea ANN donde al conjunto de datos se le aplica una función fuzzy. Con independencia del modelo, se emplean una gran cantidad de parámetros y de diferentes entornos, siendo éstos: datos históricos de las curvas y la meteorología; factores de tiempo; información de los clientes; datos económicos y demográficos y sus predicciones; precios de la energía; desarrollo regional del país; ventas nacionales de aparatos de aire acondicionado y dispositivos con clara influencia sobre la demanda eléctrica; y perturbaciones aleatorias.

En la Figura 3.4 se muestra un resumen de la clasificación en cuanto al horizonte de predicción. En base a la figura, y a lo expuesto anteriormente, el lector podrá ubicar sin problemas las obras arriba comentadas. Se muestra igualmente el alcance temporal de los parámetros de entrada, así como los tipos de parámetros empleados según el horizonte en cuestión.

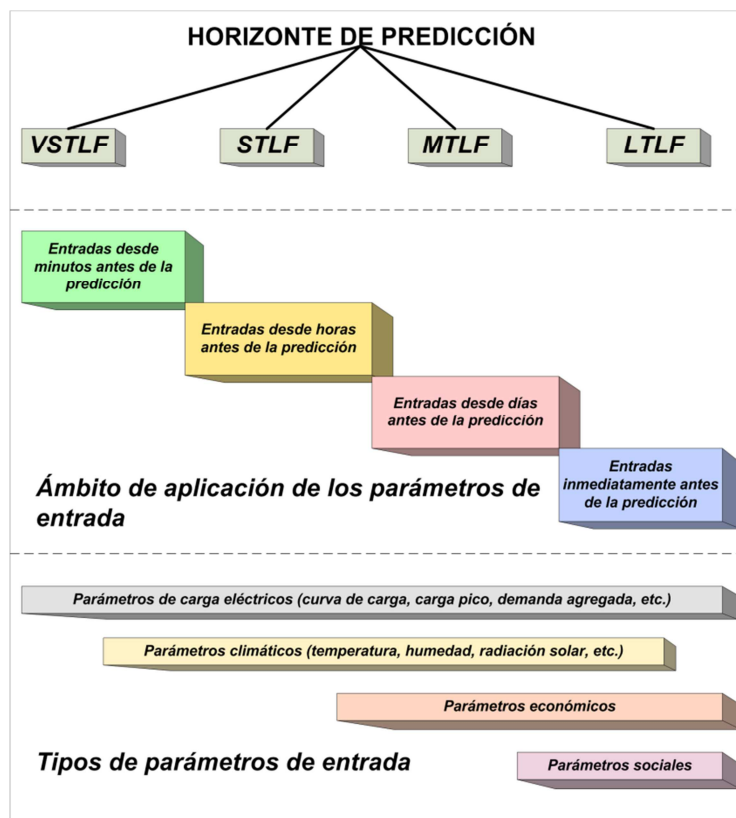


Figura 3.4. Clasificación de los modelos de predicción de la demanda en cuanto a su horizonte de predicción. Identificación del alcance de sus parámetros de entrada así como de su tipología. Fuente Propia.



### 3.6.2. Clasificación según el objetivo de la predicción

Otra clasificación de interés, y que se encuentra en la literatura encontrada hasta la fecha, es atendiendo al número de valores a predecir, destacando dos grupos principalmente. En el primer grupo se encuentran los que tienen tan sólo un valor, empleándose para predecir la carga de la próxima hora, carga pico del día siguiente, la carga total del día siguiente, etc. Con respecto al segundo grupo estarían las predicciones con más de un valor, como por ejemplo la curva de carga (*STLF* del día siguiente), varias horas siguientes, carga pico junto a otro parámetro (por ejemplo la demanda agregada) o incluso predicción de cada una de las horas del día siguiente en paralelo para la obtención del perfil de carga, que aunque podrían considerarse como del primer grupo, se muestran en el segundo debido a que la curva de carga representa la información de 24 valores concretos de un día.

Los antecedentes iniciales comienzan presentando algunos trabajos de importancia correspondientes al primer grupo, los modelos que predicen un valor. Ya desde mediados del siglo pasado se ha tenido en cuenta este tipo de retos, Gillies *et al.* (1956) presentan un proceso manual de predicción de carga pico, donde se precisa tener una predicción climatológica, convertirla a un parámetro crítico (según los autores), que es el de iluminación, y por medio de tabla iluminación vs carga pico obtener un incremento de la carga pico; y por último, aplicar dicho incremento a la tendencia de la curva para obtener la predicción. Destacar de este trabajo, la definición de un parámetro concreto de iluminación, basado en variables climáticas y su relación con la carga pico, además, todo el trabajo de su algoritmo manual se ha desarrollado en una extensa zona de Ontario.

Con respecto a obras que hacen la predicción de varios valores a la vez destacamos la de Park *et al.* (1991), donde se plantean tres pequeñas *ANNs* para predecir la carga de la próxima hora, la carga pico de la próxima hora y demanda total (agregada) del día siguiente. Ho *et al.* (1992) predicen la carga pico del día siguiente, la cual será empleada como entrada en un sistema experto de los mismos autores (Ho *et al.*, 1990) para predecir la curva de carga del día siguiente, siendo esta predicción considerada del segundo grupo.

En ocasiones, las *ANNs* con tan sólo una salida pueden emplearse de manera repetida para predecir la curva de carga como en (Drezga and Rahman, 1998; Drezga and Rahman, 1999a), o empleando un sistema en paralelo de 24 *ANNs*, tal y como se presenta en (McMenamin and Monforte, 1998).

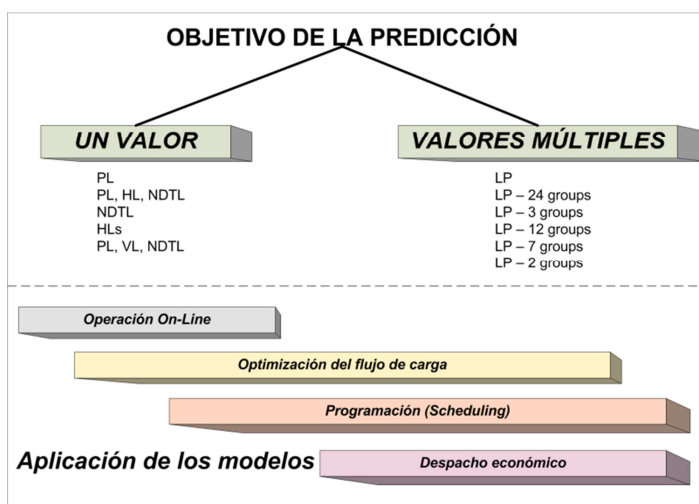
Para la obtención por ejemplo la curva de carga, se presenta el trabajo de Lee *et al.* (1992) donde se divide el día en tres periodos, empleando una *ANN* diferente para cada uno de ellos. Lu *et al.* (1993) experimentan con tres modelos de *ANN* de dos *utilities* y, se concluye, que los sistemas deben ser ajustados de manera independiente al intercambiarse, mostrando la clara dependencia de

los resultados de los modelos con los datos. Papalexopoulos *et al.* (1994) incluyen funciones de temperatura como entradas y, también, sugieren un procedimiento de mejora de festivos. Bakirtzis *et al.* (1995) presenta una mejora centrada en los días festivos con respecto a anteriores obras.

Como se ha comentado, algunas de los trabajos sugieren sistemas en el que un número de ANNs trabajan de manera conjunta para obtener la predicción. Alfuhaid *et al.* (1997) emplean una pequeña ANN que pre–procesa algunos datos y produce estimaciones de la carga pico, la carga valle y pico del día siguiente, las cuales alimentan junto a nuevos datos otra ANN para obtener la predicción de la curva de carga del día siguiente. Lamedica *et al.* (1996) proponen 12 ANNs, una para cada uno de los 12 meses del año, donde las curvas de carga fueron clasificadas a través de Mapa Auto–Organizado (*Self–Organizing Map – SOM*). En (Mohammed *et al.*, 1995) las cargas horarias fueron clasificadas de acuerdo a la temporada en siete clases, cada una de las clases fueron modeladas por una ANN independiente para formar un sistema grande. Piras *et al.* (1996) se emplea una Red Neuro–Gas (*neural–gas network*) para clasificar los datos en dos grupos (verano e invierno) los cuales se modelaron por diferentes ANNs.

En la Figura 3.5 se muestra un resumen de la clasificación en cuanto al objetivo de la predicción. Las referencias anteriormente indicadas y analizadas, las cuales marcan los antecedentes de los trabajos de predicción, pueden asociarse de manera fácil con cada uno de los grupos de la figura. De la misma forma se indica la posible aplicación de los modelos.

Se puede observar que las aplicaciones de las predicciones de un valor son para operación “en línea” (*on–line*) y optimización de flujos de carga, mientras que las de varios valores, además de las aplicaciones anteriores, su aplicación se extiende a Programación (*scheduling*) de los generadores y despacho económico.



**Figura 3.5.** Clasificación de los modelos de predicción de la demanda en cuanto a su objetivo. Se detalla el entorno de aplicaciones de los diferentes modelos. *PL*: carga pico (*peak load*); *VL*: carga valle (*valley load*); *HL*: carga de hora (*hour load*); *HLs*: carga de horas (*hour loads*); *NDTL*: carga total del día siguiente (*next day total load*); *LP*: curva de carga (*load profile*). Fuente Propia.

Como ya se ha comentado, estas clasificaciones se complementarán con lo que se presente en el Capítulo IV, donde una vez se describan los conceptos teóricos de los modelos lineales y no lineales, se detallarán las obras que servirán como antecedentes a lo planteado en esta *Tesis Doctoral*.



## CAPÍTULO IV. TÉCNICAS DE PREDICCIÓN

Como se ha indicado en el Capítulo III, tanto para las nuevas “*redes de energía inteligentes*” (SG), como para los nuevos modelos que formarán parte de las mismas (*microgrid, Smart Building, etc.*), se hace imprescindible la existencia de nuevas herramientas de predicción, tanto para la demanda desagregada, como para DG bajo su influencia. Por este motivo, se presenta este Capítulo como base previa a los aportes de la presente *Tesis Doctoral*, que se mostrarán en el Capítulo VI.

El empleo de técnicas de predicción ha experimentado en los últimos tiempos una proliferación considerable, principalmente por la mayor disponibilidad y potencia de los sistemas de cálculo automático, materializadas en las computadoras personales. Además del cambio cuantitativo, se ha producido un cambio en la tipología de los modelos planteados y usados. En la década de 1970, el empleo de técnicas no lineales era una cuestión poco aceptada por los científicos e ingenieros, en cambio, en los últimos veinte años se han convertido en técnicas muy acertadas para la solución de problemas complejos y difícilmente abordables de otra manera.

Los campos de trabajo de la ciencia donde se aplican los diferentes modelos son: física, química, ingeniería, biología, medicina, etc.; en cambio, mientras que los campos de actuación son completamente distintos, las técnicas empleadas son las mismas, ya que han tenido una evolución paralela a los citados campos, pudiéndose ver los problemas a solucionar de manera global.

Para comenzar, se presentará de forma resumida la fundamentación para el estudio de las técnicas de predicción. A continuación, se hará una descripción del modelado, que no es otra cosa que la representación matemática de un sistema. Seguidamente, se presentará el concepto de identificación, y lo que pretende resolver cuando se aplica a los sistemas dinámicos. Para finalizar, se expondrán los conceptos de los modelos lineales, ya que se emplearon en el pasado, y los conceptos en relación con las técnicas no lineales, ya que son la base de lo que se mostrará en el Capítulo VI. En base a las obras encontradas en la literatura, se procederá a realizar una comparativa entre los modelos presentados, los cuales han servido de antecedentes para la *Tesis Doctoral*.

### 4.1. INTRODUCCIÓN

En esta primera sección se presentará el concepto de predicción, vista desde diferentes perspectivas. Se continuará detallando los fundamentos de la predicción, definiendo los conceptos más significativos de la misma. Se mostrarán los principales modelos de técnicas de predicción, para que el lector se vaya haciendo una composición de las posibles soluciones de partida para los sistemas complejos. Se finalizará con la nomenclatura que se ha adoptado en este texto, y que

tratará de seguirse de aquí en adelante, y su representación en los sistemas sobre los que se harán predicciones.

Estas ideas básicas se intentarán ampliar a lo largo del Capítulo, desde varios puntos de vista, para que el lector pueda formarse una idea general alrededor de las técnicas de predicción, por lo que se considera como fundamental esta primera sección.

#### ***4.1.1. Concepto de predicción***

Según la *Real Academia Española (RAE)*, predecir es “*anunciar por revelación, ciencia o conjetura, algo que ha de suceder*”. Son muchas las situaciones donde se requiere una predicción y no sólo en la ingeniería, basta pensar en los esfuerzos dedicados por los economistas por prever los ciclos económicos o los de algunos inversores para sacar el máximo provecho en bolsa adivinando ciertas tendencias. En el ámbito de la ingeniería, la predicción se encamina al control, optimización y a la seguridad de operación, como las más importantes.

La predicción ha sido además de un fin, un medio para estudiar fenómenos como el caos. El instrumento principal para analizar los procesos que producen estas situaciones caóticas fueron las series de datos consecuencia de la observación del proceso en cuestión. La labor de enmarcar dichos datos en leyes empíricas que permitan posteriormente predicciones se remonta a muy antiguo en la historia, por ejemplo, el intento por hacer predicciones de fenómenos astronómicos. En la actualidad, el avance en tiempos de cómputo de los dispositivos de cálculo, ha permitido abordar problemas en los que la cantidad de datos y la complejidad de las relaciones, hace que la solución no sea trivial.

A comienzos del siglo XX la predicción se realizaba bien con modelos consecuencia de la aplicación de leyes y principios teóricos o a partir de observaciones, realizando ajustes globales temporales y extrapolando para el futuro. A partir de esto, técnicas autoregresivas, como las presentadas en (Yule, 1927) fueron desarrolladas en el ámbito de la predicción de manchas solares, dando pie a los modelos lineales del tipo que se presenta en (Box and Jenkins, 1970), y que ha sido ampliamente empleados desde entonces.

En la etapa final del siglo XX se ha avanzado considerablemente en el estudio de sistemas caóticos y las series temporales a que dan lugar. Las últimas técnicas, permiten considerar técnicas no lineales, separando los casos no deterministas de las dinámicas caóticas entendidas a partir de entonces como el resultado de ciertos procesos no lineales, difíciles de predecir pero con carácter determinista. Desde entonces, la aplicación de nuevas técnicas de predicción a diversos campos de la ingeniería es cada vez mayor, al tiempo que la teoría en que se apoyan sigue una evolución paralela.

### 4.1.2. Fundamentos de la predicción

Se van a mostrar los conceptos más importantes que se manejarán a la hora de hablar de técnicas de predicción. Se comenzará definiendo el concepto de sistema y sus tipos, las variables que lo conforman, los modelos y sus tipos, y para finalizar el ámbito de aplicación de los modelos.

#### 4.1.2.1. Sistemas

Según la RAE, un sistema es “conjunto de cosas que ordenadamente relacionadas entre sí contribuyen a determinado objeto”. La anterior definición coincide con la de (Aracil and Gordillo, 1997) en su esencia, definiéndolo como “objeto formado por un conjunto de cosas o partes, entre las cuales se establece alguna forma de relación que las articula en la unidad que es el sistema”.

En ingeniería se presta mucha atención a un tipo de sistema llamado dinámico, que son aquéllos en los que el tiempo juega un papel primordial, ya que las relaciones entre magnitudes se relacionan no sólo a ellas, sino también con sus derivadas. Como ejemplo, véase el Sistema Solar, formado por planetas y otros cuerpos relacionados entre sí por fuerzas, con lo que sus posiciones y velocidades varían con el tiempo, siendo estas variaciones dependientes de las leyes por las que se rigen dichas fuerzas.

Se emplea la señal variable o señal para designar a las magnitudes que pueden variar con el tiempo definiéndose, por tanto, la trayectoria como la evolución temporal de una magnitud, aunque esta magnitud no sea una posición en el espacio como da a entender el nombre. El conjunto de trayectorias seguidas por las magnitudes indica el comportamiento del sistema.

A partir de las señales, se define estado como el menor conjunto de variables cuyo conocimiento en un instante proporciona la posibilidad de determinar completamente las trayectorias futuras del sistema. El concepto estado es fundamental, ya que el comportamiento de un sistema queda descrito de manera completa por la evolución del estado. Las variables que forman el estado reciben el nombre de variables de estado y se agrupan en un vector llamado vector de estado. Normalmente, un sistema existe en un medio con el que puede tener interacciones, por lo que los cambios que se producen en el exterior afectan a su comportamiento, desde un punto de vista práctico conviene distinguir algunos tipos de relaciones:

- Internas: no se relaciona con elementos ajenos al sistema.
- Externas: los elementos del sistema interactúan con el exterior.

#### 4.1.2.2. Señales

Las variables se pueden clasificar según la relación causa–efecto que las relaciona dentro del sistema, así se habla de entradas (causa, influencia o fuerza) y salidas (efectos), dicho de otro modo, una variable es llamada de salida si su valor se considera consecuencia de los valores tomados por

una o varias variables llamadas de entrada. Las entradas son señales cuyos valores afectan al sistema, modificando su estado, tal es el caso de la intensidad de campo eléctrico sobre la intensidad de campo magnético en un medio, tensión aplicada a un circuito eléctrico y la circulación de la corriente, etc.; la trayectoria que sigue las entradas es la causante de los cambios en el sistema, cambios que se ponen de manifiesto en las trayectorias del resto de señales.

Una clasificación de las señales según lo que aparece en (Arahal *et al.*, 2006), sería:

- Externas:
  - Salidas: las variables que se desean predecir o controlar.
  - Entradas: las variables causantes de los cambios en las salidas y que pueden ser:
    - Entradas manipulables: para modificar la trayectoria de la salida del sistema.
    - Perturbaciones: escapan al control del usuario del sistema, pudiendo afectar de forma negativa a las trayectorias de la salida.
- Internas: las variables intermedias que ligan las entradas con las salidas.

Los sistemas disponen de magnitudes que casi no varían, incluso nunca, y que son conocidas como parámetros, siendo importante tener clara la diferencia entre variables y parámetros: las variables son magnitudes que cambian con el tiempo, como la corriente que circula por un circuito eléctrico; un parámetro en cambio es un valor que distingue un sistema de otro, por ejemplo, la conductividad del cobre.

#### 4.1.2.3. Tipos de Sistemas

Existen distintos criterios para clasificar los sistemas, destacando los que se muestran a continuación en la Tabla 4.1.

**Tabla 4.1. Clasificación de los Sistemas, de acuerdo con diferentes criterios. Fuente Propia con Datos de Arahal *et al.* (2002). (Continúa).**

<i>Criterio</i>	<i>Tipos</i>
Interacción con el entorno	Cerrado o autónomo: No hay relación con el entorno Abierto o no autónomo: hay relación con el entorno
Origen	Natural: no originado por el hombre Artificial: originado por el hombre
Variabilidad	Invariable con el tiempo: la relación entre las variables no cambia con el tiempo Variable con el tiempo: la relación entre variables cambia con el tiempo
Tiempo	Estático: la relación entre variables es instantánea Dinámico: la relación entre variables incluye el valor de las variables y sus derivadas
Estabilidad	Estable: el sistema se equilibra tras una perturbación Inestable: el sistema no se equilibra después de una perturbación



**Tabla 4.1. Fuente Propia con Datos de Arahall *et al.* (2002).**

<i>Criterio</i>	<i>Tipos</i>
Comportamiento	Lineal: frente a una variable, el sistema se comporta linealmente No lineal: frente a una variable, el sistema no se comporta linealmente
Respuesta al medio	Adaptable: la salida se modifica en función de lo que ocurre en el medio No adaptable: la salida no se modifica en función de lo que ocurre en el medio
Aleatoriedad	Determinista: no hay variables aleatorias Estocástico: hay variables aleatorias

#### **4.1.2.4. Modelos y Tipos de Modelos**

Según Arahall *et al.* (2006), un modelo “*es una representación simplificada de un sistema real, un proceso o una teoría con la que se pretende aumentar su comprensión, hacer predicciones y poder llegar a controlar el sistema*”. Por tanto, el propósito general de un modelo es capacitar al usuario para determinar la manera en que los cambios en aspectos del sistema modelado pueden afectar parcial o globalmente al sistema, sin necesidad de experimentar en la realidad. Los modelos se convierten de esta manera en instrumentos perfectos para hacer predicciones, y que permiten entre otras cosas:

- Hacer una predicción del sistema.
- Explicar la mayoría de los comportamientos del sistema.
- Analizar las relaciones entre las variables de entrada y las de salida.
- Analizar los cambios de estado del sistema al hacer variar algunos o todos los parámetros.
- Proponer y desarrollar algoritmos de control para obtener las salidas deseadas.

A continuación, se presenta los tipos de modelos, según Ríos and Ríos (2008) esencialmente existen tres clases de modelos:

- Modelos físicos: representación de sistemas físicos mediante variables medibles, apareciendo los siguientes modelos:
  - Modelos a escala: maquetas para estudio en un túnel de viento, maquetas de estructuras en plataformas móviles que simulan terremotos, etc.
  - Imitación: mapas de relieve.
  - Analógicos: reloj de arena para modelar el paso del tiempo.
  - Prototipos: se realizan antes de lanzar un producto al mercado, como por ejemplo un modelo de vehículo, aeronave, etc.
- Modelos mentales: procesos heurísticos que únicamente existen en la mente del ser humano.
- Modelos simbólicos: compuestos por operaciones lógicas y matemáticas para formular una solución de un problema. Dos tipos:

- Matemáticos: emplean ecuaciones matemáticas para modelar un sistema, siendo muy flexibles y adaptables, así como los más empleados en la ingeniería.
- No matemáticos: por ejemplo los que describen verbalmente un sistema, los que emplean dibujos o diagramas de flujo.

El empleo de modelos tiene como gran ventaja la evaluación de resultados de una decisión del mundo real sin tener que tomarla de manera efectiva. Otras ventajas complementarias serían:

- Posibilita la organización del conocimiento y las observaciones sobre el sistema.
- Creación de un marco para contrastar el sistema y sus alternativas.
- Presenta una perspectiva sobre detalles y aspectos a tener en cuenta, así como la facilidad para el análisis posterior.
- Con los modelos matemáticos se describen los problemas de forma más exacta que por ejemplo una descripción verbal de una persona y que deba ser entendido por terceros.
- En la mayoría de las veces, menos costoso que experimentar con el sistema real, y permite mejor y mayor interacción con el propio sistema.

Sin embargo, el desarrollo y utilización de los modelos, presentan una serie de desventajas, entre las que destacaremos:

- No hay siempre garantía plena de resultados útiles y satisfactorios del sistema modelado. Muchas veces la causa es la falta de ingenio para la consecución de los objetivos pretendidos.
- A veces se considera una visión particular del problema como la mejor representación del sistema, y puede que esto no sea totalmente cierto.

#### 4.1.2.5. *Ámbito de Aplicación de los Modelos*

Es muy importante el ámbito de aplicación del modelo de un sistema, ya que condicionará muchas de las decisiones que se deben de tomar en el diseño para plasmar el modelo, como por ejemplo: complejidad mediante sistema lineal o no lineal, número de entradas y salidas, iteraciones del proceso, precisión a la hora de reproducir el comportamiento real del sistema, etc. En la Figura 4.1 se presentan algunos usos de los modelos, tal y como se resume en (Nelles, 2001), siendo nuestro principal interés la predicción.

Un sistema o modelo se dice *monovariante*, cuando tanto la señal de entrada como la de salida son escalares. Son los sistemas en los que

$$u(t) \in \mathfrak{R} \text{ e } y(t) \in \mathfrak{R} \quad (4.1)$$

indicándose (4.1) como (1,1).

Cuando la entrada o la salida sea una variable vectorial se dice que el sistema o modelo es *multivariante* de  $n_e$  entradas y  $n_s$  salidas, o abreviadamente  $(n_e, n_s)$ , siendo

$$\mathbf{u}(t) \in \mathcal{R}^{n_e} \text{ y } \mathbf{y}(t) \in \mathcal{R}^{n_s} \quad (4.2)$$

A efectos de construir modelos, el caso *multivariante* se suele reducir a  $n_s$  casos del tipo  $(n_e, 1)$ , es decir, se obtiene un modelo por cada una de las salidas, teniendo cada modelo  $n_e$  entradas como máximo. Cuando se emplean modelos derivados de datos, el número de señales es un factor muy importante, debido a que la complejidad del modelo, el número de datos y el número de señales están fuertemente relacionados.

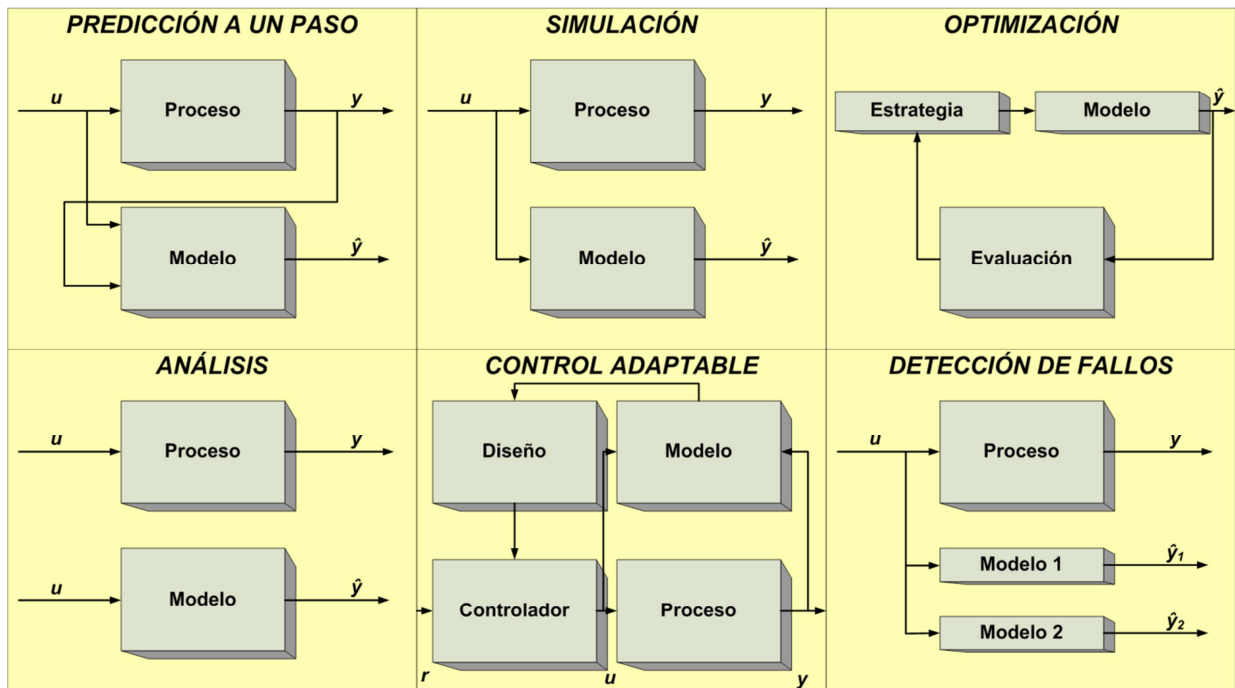


Figura 4.1. Algunos usos de los modelos atendiendo al ámbito de aplicación. Fuente Propia con Datos de Arahal et al. (2002).

Un modelo es *determinista* cuando se parte de la base de que el modelo es capaz de expresar de forma única la evolución del sistema, es decir, que conocido el modelo de un sistema y las condiciones iniciales de las que se parte, así como la evolución de las entradas, las variables del modelo siempre evolucionan de la misma forma. Por el contrario, un modelo *no determinista* o *estocástico* considera que intervienen factores aleatorios, imposibles de modelar ni predecir, pudiéndose tan sólo modelar ciertas características estadísticas de las magnitudes temporales del sistema, pero no su evolución exacta, pues varían por la influencia de eventos imprescindibles, según esto, ante unas mismas condiciones iniciales y magnitudes de entrada, las variables del modelo evolucionarán cada vez de una forma distinta, pero conservando una serie de características comunes de tipo estadístico, que son las que se pueden predecir.

Además, los sistemas se encuentran sometidos a *perturbaciones* y *ruidos*, pudiendo ser de tipo físico o incluso por las propias características del sistema. Los *ruidos* pueden afectar sólo a la lectura de variables, sólo al proceso o a ambos, por tanto, es razonable la necesidad de introducir modelos que permitan de alguna manera caracterizar el *ruido*. Los *ruidos* se consideran variables

aleatorias que vienen definidas por una serie de características, como: su forma de distribución (uniforme, *gaussiana*, etc.) y su grado de aleatoriedad, dado por el valor de parámetros que pueden inferirse por los estadísticos típicos (media, varianza, etc.).

A veces, los datos tomados de un sistema son la única fuente de información de que se dispone para realizar predicciones, resulta de gran importancia saber el origen de estos datos para depositar mayor o menor confianza en los resultados obtenidos. En ocasiones, los datos proceden de observación de fenómenos naturales, esto implica que su evolución no puede alterarse y por tanto que las características de los datos no pueden elegirse.

### **4.1.3. Técnicas de predicción**

Lo habitual, es que las técnicas de predicción empleen un modelo en forma de ecuaciones diferenciales o en diferencias. El interés está en la manera en que el modelo es obtenido, de este modo las técnicas se clasifican en función de que usen:

- Modelos derivados de leyes: consisten en ecuaciones matemáticas que ligan las derivadas del estado con las entradas y la salida. Las ecuaciones se suelen obtener por simplificaciones de la realidad. Los modelos derivados de leyes suelen tener una gran ventaja sobre los demás, y es que las leyes suelen tener dominios de validez muy amplios, por ejemplo, la ley de Newton sirve en una gran cantidad de situaciones en las que las velocidades no se aproximan a la velocidad de la luz. Esta validez de las ecuaciones, permite aplicar los modelos para la predicción, incluso cuando el escenario real se aleja mucho de las situaciones comúnmente previstas.
- Modelos de caja negra o paramétricos: son esquemas capaces de representar cualquier sistema, se trata en la mayoría de los casos de estructuras parametrizadas con la suficiente flexibilidad como para acomodar los datos observados de una gran cantidad de sistemas diferentes. Estos modelos no se obtienen, al menos no totalmente, de la aplicación de leyes, al contrario, se construyen empleando observaciones realizadas sobre el sistema para seleccionar un valor concreto de los parámetros, este valor se elige de manera que el modelo que resulte se acomode a las observaciones de que se dispone.

El proceso mediante el cual los datos registrados llevan a un modelo, recibe el nombre de identificación, que al igual que el modelado, sirve para construir modelos. Sin embargo, los modelos resultantes son relaciones matemáticas que permiten albergar los datos observados sin que la relación tenga capacidad para explicar los procesos que han generado los datos. Otra diferencia frente al modelado, es que los datos condicionan el resultado, esto es, si se dispone de un conjunto de datos diferentes de datos, se puede obtener un modelo diferente.

Los modelos de caja negra funcionan como composición de dos correspondencias, tal y como describe Sjöberg (1995), la primera maneja datos obtenidos del sistema en un vector de regresión, mientras que la segunda realiza una función sobre dicho vector de regresión para obtener un punto del espacio de salida.

En ocasiones, los datos disponibles para la identificación son valores muestreados de la evolución temporal de las entradas y salidas del sistema. Cuando se infiere un modelo a partir de tales datos se dice que se realiza un modelo de entrada-salida o abreviadamente modelo  $e-s$ . La construcción de modelos a partir de datos  $e-s$  es más rápida en muchos casos que el modelado basado en conocimiento de los procesos internos del sistema, sin embargo, la desventaja de los modelos de  $e-s$  es que no explican su funcionamiento. Otro problema relacionado con este tipo de modelos es que los datos recolectados para el desarrollo del mismo son, necesariamente, sólo una parte del espacio de  $e-s$  del problema, por ello se ha de prestar atención a conseguir datos representativos de la zona de interés.

- Sin modelo explícito: es posible calcular las predicciones directamente a partir de los datos observados, sin necesidad de construir un modelo a partir de ellos. Existen varios métodos que tienen su origen en los trabajos de Lorenz y su técnica de los análogos, como el presentado en (Lorenz, 1963), posteriormente empleados por otros investigadores de modelos no lineales interesados en el caos.

La predicción sin modelo explícito es similar a la predicción con modelos de caja negra, pero sustituyendo las dos correspondencias por algoritmos que producen predicciones directamente a partir de datos introducidos. La ventaja de este procedimiento es que se pueden añadir nuevos datos sin tener que modificar los programas de predicción.

#### ***4.1.4. Nomenclatura empleada***

A continuación, se presentará la simbología que suele emplearse más comúnmente en los textos de predicciones, aunque debe indicarse al lector que puede variar ligeramente. De manera general, y para cualquier tipo de símbolo, una letra en negrita indica la presencia de una formación vectorial, en cambio, la no presencia de negrita indicará un escalar, en esta sección no importa tanto esto, sino la nomenclatura de la simbología.

##### ***4.1.4.1. Señales***

Para comenzar, empezaremos dando algunas indicaciones con respecto a las señales. Los siguientes símbolos representan señales genéricas que pueden estar presentes en muchos tipos de sistemas:

- $y$  es la salida observada del sistema. En ocasiones, a la salida se la conoce como objetivo.

- $u$  es la señal manipulada del sistema, llamada señal de entrada y acción de control.
- $p$  es una señal de entrada que no es manipulable, es decir, una perturbación.
- $v$  ruido, señal no manipulable, no medible que afecta al sistema y que tiene un comportamiento impredecible.

#### 4.1.4.2. Ecuaciones diferenciales y en diferencias

Con respecto a las ecuaciones diferenciales y en diferencias, se debe decir que son una forma habitual de representar los modelos de sistemas dinámicos, siendo lo normal la siguiente ecuación

$$\frac{d^n y}{dt^n}(t) + a_1 \frac{d^{n-1} y}{dt^{n-1}}(t) + \dots + a_{n-1} \dot{y} + a_n y(t) = b_0 \frac{d^m u}{dt^m}(t) + \dots + b_m u(t) \quad (4.3)$$

En el caso de modelos con tiempo discreto, se emplea una ecuación en diferencias como

$$\alpha_n y(k-n) + \alpha_{n-1} y(k-n+1) + \dots + \alpha_0 y(k) = \beta_m u(k-m) + \dots + \beta_1 u(k-1) \quad (4.4)$$

Las ecuaciones en diferencias presentan algunas ventajas cuando se trabaja con sistemas digitales, en particular para la predicción es conveniente despejar la variable  $y(k)$  para obtener una expresión que permita producir valores futuros de  $y$  a partir de valores pasados de  $u$  y de  $y$ .

#### 4.1.4.3. Instantes y horizontes

Con relación a los instantes y horizontes temporales:

- $t$  instante de tiempo genérico, bien sea continuo o discreto.
- $T_m$  periodo de muestreo.
- $k$  instante de tiempo discreto, es decir, el número de periodos de muestreos transcurridos desde el instante considerado como origen. El instante de tiempo continuo equivalente es  $t = kT_m$ .
- $z^{-1}$  es el operador de retraso definido como  $z^{-1}u(k) = u(k-1)$ . De forma equivalente,  $z^{-2}$  realiza un atraso de dos periodos de muestreo, y así sucesivamente. Lo normal es emplear directamente  $u(k-1)$ .
- $h$  horizonte temporal.

#### 4.1.4.4. Predicciones

Con respecto a las predicciones tenemos:

- $\hat{y}$  es la salida predicha del sistema.
- $\hat{y}(k+1)$  es la salida predicha del sistema en el instante de tiempo discreto  $k+1$ . Normalmente se emplea este símbolo en el caso de predicción a un paso.

- $\hat{y}(k+1/k)$  es como la anterior, pero indicando que la predicción del instante  $k+1$  se realiza en el instante  $k$ . Esta notación se emplea para poder indicar predicciones en horizontes  $h$  mayores que uno, por ejemplo mediante  $\hat{y}(k+h/k)$ .
- $\hat{p}$  indica el valor predicho para una perturbación.
- $\hat{\theta}$  es un vector de parámetros a estimar en un modelo paramétrico.

#### 4.1.4.5. Datos

Con respecto a los datos tenemos:

- $x$  es una variable independiente de una función, puede indicar el vector de entrada en las redes de neuronas artificiales.
- $y$  es una variable dependiente de una función, puede indicar el vector de salida en las redes de neuronas artificiales.
- $m(k)$  es un vector regresor o vector de medidas que contiene valores conocidos en el instante  $k$ . En ocasiones se denota mediante  $x(k)$  pues coincide con el vector de entrada de una red de neuronas.
- $m$  es un vector regresor de un modelo paramétrico.
- $z = (x, y)$  es una pareja de valores de  $e-s$ , comúnmente llamado patrón de  $e-s$ .
- Con  $CE$  se denotará al conjunto de datos de entrenamiento formado por los patrones  $z_p$  con  $p = 1, \dots, N_{CE}$ . Se usa para estimar parámetros en el proceso de identificación.
- Con  $CP$  se denotará al conjunto de datos de prueba formado por los patrones  $z_p$  con  $p = 1, \dots, N_{CP}$ . Se usa para comparar distintos modelos en el proceso de identificación.
- Con  $CH$  se denotará al conjunto de datos históricos. En la mayoría de los casos  $CH = CE \cup CP$ .
- Con  $CF$  se denotará al conjunto de datos futuros. Normalmente, estos datos no están disponibles durante la construcción y prueba de modelos pues son posteriores al instante en que se construyen. Lo que se suele hacer es apartar un grupo de datos de  $CH$  para que formen parte de  $CF$ , y poder validar los modelos realizados.

#### 4.1.4.6. Medidas de error

Con respecto al error instantáneo, se puede medir de diversas formas:

- $e(k+1)$  es el error en la predicción a un paso, puede referirse a variables normalizadas o sin normalizar, en este último caso aparecerá la unidad de ingeniería en que se mide (amperios, newton, metros, etc.). La fórmula viene dada por

$$e(k+1) = y(k+1) - \hat{y}(k+1/k) \quad (4.5)$$

- $ep(k+1)$  es el error porcentual en la predicción a un paso

$$ep(k+1) = 100 \cdot \frac{y(k+1) - \hat{y}(k+1/k)}{y(k+1)} \quad (4.6)$$

Cuando operamos con un conjunto de datos, y se trata de controlar la desviación, se puede emplear alguna o varias de las siguientes medidas:

- Suma de Error Cuadrático (*Sum of Squared Error – SSE*), es la suma de errores cuadráticos y representa la discrepancia entre el objetivo y la estimación del mismo, y se calcula como

$$SSE_C = \sum_{i=1}^{N_C} e(i)^2 \quad (4.7)$$

siendo  $N_C = \dim(C)$  el número de puntos en el conjunto  $C$ . Si se quiere emplear el error porcentual, basta con emplear  $e_p$  en lugar de  $e$ .

- *MAPE*, es una medida de la precisión de un método para el ajuste de valores de series de tiempo, principalmente en estadística, especialmente para la estimación de tendencias.

$$MAPE_C = \frac{1}{N_C} \sum_{i=1}^{N_C} |e(i)| \quad (4.8)$$

siendo  $N_C = \dim(C)$  el número de puntos en el conjunto  $C$ . Si se quiere emplear el error porcentual, basta con emplear  $e_p$  en lugar de  $e$ . Es uno de los más recomendados en la industria eléctrica y, además, ha sido ampliamente empleado en la literatura, como se muestra en (Khotanzad *et al.*, 1997; Hippert, 2001).

- Error Cuadrático Medio (*Mean Squared Error – MSE*), es el error cuadrático medio y es una forma de evaluar la diferencia entre la predicción y el valor real de la cantidad que se quiere calcular. *MSE* mide el promedio del cuadrado del “error”, siendo el error el valor en la que el estimador difiere de la cantidad a ser estimada. *MAPE* es poco sensible a los atípicos (errores mayores 2 ó 3 veces la desviación estándar) tal y como se muestra en (Martín–Merino, 2005), ya que mide la disparidad entre el valor predicho y el real con el valor absoluto, sin embargo, los errores grandes y poco frecuentes son muy importantes para determinados sistemas (industria eléctrica por ejemplo), ya que inducirán a pérdidas puntuales importantes. *MSE* se calcula como

$$MSE_C = \sqrt{\frac{1}{N_C} SSE} \quad (4.9)$$

siendo  $N_C = \dim(C)$  el número de puntos en el conjunto  $C$ . Si se quiere emplear el error porcentual, basta con emplear  $e_p$  en lugar de  $e$ .



- Error Máximo (*Maximun Error – ME*), es el error máximo que pretende complementar a *MAPE* y *MSE*, y trata de evaluar la máxima desviación entre la predicción y el valor real. Ha sido empleado con anterioridad en la literatura, como se ve en (Rojas and Palomares, 2004), y vuelve a ser útil para ciertos ámbitos, como es el caso de las *utilities*, ya que un único valor elevado puede resultar nefasto para el sistema de planificación de las mismas.

$$ME_C = \max_{i=1, \dots, N_C} |e(i)| \quad (4.10)$$

siendo  $N_C = \dim(C)$  el número de puntos en el conjunto  $C$ . Si se quiere emplear el error porcentual, basta con emplear  $e_p$  en lugar de  $e$ .

#### 4.1.5. Predicciones

Sea un sistema con entrada  $u$ , sometido a perturbaciones  $p$  y salida  $y$ , la forma de obtener la predicción  $\hat{y}$  de la salida depende de la técnica empleada, que coincidirá con alguno de los tres tipos de modelos anteriormente descritos.

En los modelos de caja negra, la predicción a un paso puede obtenerse por medio de una ecuación en diferencias, de la siguiente forma

$$\hat{y}(k+1/k) = f(\mathbf{m}(k)) \quad (4.11)$$

con  $\hat{y}(k+1/k) = f(\mathbf{m}(k))$  la predicción de la salida para  $k+1$ , realizada en  $k$  y siendo  $\mathbf{m}(k)$  el vector regresor con valores conocidos de  $k$ , que pueden ser valores pasados de la propia salida y de las variables de entrada, e incluso las perturbaciones.

En la mayoría de las veces se debe producir una secuencia de predicciones, ya que la de un paso no suele ser efectiva, resultando de esta manera

$$\{\hat{y}(k+2/k), \dots, \hat{y}(k+h/k)\} \quad (4.12)$$

Para calcular  $\hat{y}(k+h/k)$  hay que conocer o poder prever los valores que tomarán las entradas no sólo en instantes pasados, sino también en  $\{k+1, \dots, k+h-1\}$ .

En el caso de la predicción sin modelo explícito,  $f$  se sustituye por un algoritmo que permita calcular  $\hat{y}(k+1/k)$ . Los modelos derivados de leyes se realizan discretizando las ecuaciones, o mediante algoritmos de integración numérica.

Con respecto a la predicción multipaso, hallar  $\hat{y}(k+2/k)$  se transforma en calcular en primer lugar  $\hat{y}(k+1/k)$  y, luego tomar esta predicción en lugar del valor correcto  $y(k+1)$  aplicando de nuevo (4.11). En esta segunda aplicación el regresor  $\mathbf{m}(k)$  contendrá valores futuros de  $u$  y  $p$  obtenidos mediante las formas ya comentadas y, en lugar de  $y(k+1)$  se empleará el valor de  $\hat{y}(k+1/k)$ . Se emplea de nuevo el mismo proceso para la obtención de  $\hat{y}(k+2/k)$ , y así

sucesivamente  $\hat{y}(k+3/k)$  y todas las demás predicciones en el horizonte fijado, llamándose este proceso predicción recurrente.

La predicción multipaso admite una solución diferente si se plantea cada predicción  $\{\hat{y}(k+2/k), \dots, \hat{y}(k+h/k)\}$  como un problema diferente a solucionar, en tal caso se trata de hallar la siguiente función

$$\hat{y}(k+j/k) = g_j(\mathbf{m}(k), \psi(k, j)) \quad (4.13)$$

y obteniendo de esta manera la predicción deseada a  $j$  pasos usando el vector regresor  $\mathbf{m}(k)$  y un vector  $\psi(k, j)$  con valores futuros (predichos) de las entradas hasta el instante de tiempo discreto  $k+j-1$ . Desde el punto de vista matemático, estas funciones  $g_j$  pueden obtenerse usando la técnica de la predicción recurrente y calculando el valor que se desea calcular, desde el punto de vista práctico, puede resultar más conveniente obtener nuevos modelos que realicen la ecuación (4.13). Se evitarán problemas con redondeos y otros factores puramente numéricos y, además, se pueden seleccionar componentes del vector  $\psi(k, j)$  que permiten obtener predicciones mejores, esto es, los predictores directos pueden dar mejores resultados ya que no acumularemos errores en el modelo al tener que emplear una predicción como dato para la siguiente.

## **4.2. LOS MODELOS BASADOS EN LEYES**

De una manera resumida, el modelado es la consecución de una realidad matemática que explique el sistema sujeto a análisis. Como norma general, se aplicarán leyes explicadas por ecuaciones diferenciales, tratando de combinarlas para simular el sistema.

### **4.2.1. El concepto de modelado**

Un modelo es por sí mismo un sistema, aunque su existencia no sea física, esto no debe dar pie a confundir las ecuaciones que describen el comportamiento de un sistema con el propio sistema. Las leyes y primeros principios que se emplean para la construcción de modelos son variados y dependen por supuesto de la aplicación concreta a la cual se va a dedicar el proyecto, en ingeniería, son comunes las aplicaciones que requieren las siguientes disciplinas: dinámica de planetas, dinámica del punto, dinámica de fluidos, termodinámica; química; electricidad; y magnetismo.

En los últimos años las técnicas de predicción se están aplicando a otros ámbitos que no son la ingeniería, como por ejemplo agricultura, mercados, marketing, etc.; como norma, suele ser imprescindible la participación de expertos en campos tales como: ecología, astronomía, meteorología, fisiología, etc.

#### 4.2.1.1. *Sistemas considerados lineales*

Estos sistemas pueden ser descritos mediante ecuaciones diferenciales lineales, cuyo tratamiento es más sencillo que el de las ecuaciones diferenciales. Por tanto, los modelos lineales consisten en ecuaciones diferenciales lineales ordinarias, formadas por operadores lineales, por lo que tienen ciertas propiedades que las hace matemáticamente más sencillas de emplear.

Lineabilidad es equivalente a cumplir el principio de superposición, que establece que la salida  $y$  de un sistema sometido a una entrada  $u = u_1 + u_2$ , es la suma de las salidas correspondientes de aplicar  $u_1$  y, posteriormente  $u_2$  por separado, con lo que tendremos la salida final  $y = y_1 + y_2$ .

Un sistema lineal se llama invariante con el tiempo, si los coeficientes de la ecuación son constantes. Existen otros sistemas en los que los parámetros varían, por ejemplo en un circuito eléctrico donde la resistencia de algún elemento es función de la temperatura del mismo y ésta de la potencia que disipa la resistencia y del calor evacuado por la misma.

#### 4.2.1.2. *Sistemas no lineales*

Son los sistemas donde no se cumple el principio de superposición, queda por tanto patente, que un sistema lineal es un caso particular dentro de una gran variedad de comportamientos, siendo la aplicación de los sistemas lineales en los entornos académicos.

Los sistemas no lineales tienen la propiedad de que la respuesta del sistema depende no sólo de la forma de la señal de entrada sino de la amplitud, no produciéndose este hecho en los sistemas lineales. Pueden modelarse por medio de ecuaciones diferenciales parciales no lineales, donde no existe solución general, complicándose mucho su estudio.

Las causas del comportamiento no lineal son varias, en muchos casos proviene de la existencia de fenómenos intrínsecamente no lineales, como el rozamiento, la histéresis, holguras entre piezas móviles, etc.; en otros sistemas, se introducen elementos no lineales a propósito como por ejemplo conmutadores para el control del nivel de un depósito o una temperatura determinada.

#### 4.2.1.3. *Linealización*

Las ecuaciones no lineales se pueden modelar para obtener otras que sí son lineales, siendo este proceso lo que se conoce como linealización, los modelos obtenidos, producen resultados aceptables en torno a ciertos puntos de operación. Como ejemplo véase una función no lineal como es el seno, si quisiéramos obtener una aproximación del mismo en un punto concreto ( $x=1$  radianes), podríamos emplear en un entorno muy reducido de dicho punto la recta  $y = 0.54x + 0.3$ , y en efecto, el teorema de Taylor permite definir el seno como una función polinomial dada por

$$\text{sen}(x) \approx \text{sen}(1) + \frac{x-1}{1!} \cos(1) + \dots \quad (4.14)$$

despreciando en (4.14) los términos de orden superior al primero, se obtiene la recta que se indicó más arriba.

La validez de la aproximación se ciñe a intervalos de operación más o menos estrechos, siendo por tanto la linealización una estrategia bastante empleada en ciertos ámbitos de la ingeniería, como en procesos de control en automática, en el estudio de la electrónica, etc.; se convierte por tanto en una situación empleada de manera cotidiana, incluso cuando se observa en la realidad que el comportamiento esperado discrepe de la linealidad.

#### **4.2.1.4. Los parámetros de los modelos**

En ciertos sistemas reales, como por ejemplo en los circuitos eléctricos y electrónicos, existen componentes donde sus propiedades varían, por ejemplo pensemos en una resistencia que tiene además del valor de resistencia, un valor de inductancia y capacitancia, o pensemos como la señal eléctrica tardará cierto tiempo en atravesarla.

Fenómenos como el anterior, suelen pasar desapercibidos y no son tenidos en cuenta, y pensando en el ejemplo anterior, como el componente resistencia no tiene parte de inductancia ni capacidad, y que además, transmite de manera instantánea los cambios de intensidad de corriente eléctrica que circula por la misma. Estos elementos ideales reciben el nombre de parámetros concentrados., que suelen ser más sencillos de desarrollar y emplear que los que emplean parámetros distribuidos, que suelen presentar un comportamiento mucho más complejo, y que aparecen para su uso cuando los concentrados no consiguen una solución satisfactoria.

#### **4.2.2. El concepto de simulación**

Simular es el arte de emplear un modelo para poder comprobar y entender la evolución de cierta variable de salida, esperando posteriormente, que en realidad se produzca una evolución similar ante idénticos estímulos. En una simulación es preciso conocer el valor de las entradas, ya que se intenta calcular el valor de las salidas que dependen de las anteriores. Para realizar la simulación se precisa la existencia de un modelo, ciertas condiciones iniciales del mismo y el valor de las entradas en cada instante de tiempo posterior al inicial. Por medio de aplicaciones informáticas u otros artificios, se calculan las trayectorias de las salidas del sistema. El modelo será útil siempre que las trayectorias resultantes sirvan para el propósito particular del usuario.

En el ámbito de la ingeniería, la simulación ayuda a entender y visualizar el comportamiento de ciertas características de los equipos diseñados antes de construirlos, e incluso comprobar resultantes de ciertas interacciones no tenidas en cuenta inicialmente. Es una manera de validad si el

modelo planteado sirve para manejar la realidad observada, y por tanto, obtener conclusiones de la misma. No todos son ventajas, existe el peligro de abusar de las simulaciones, confiando demasiado en los resultados obtenidos mediante estos métodos, y no se debe olvidar que ninguna simulación es perfecta por varios motivos:

- La propia precisión con límites de las máquinas de cómputo.
- Los modelos incluyen simplificaciones de la realidad, con lo que no cubrirán la totalidad de la misma.
- Antes entradas inconsistentes o incongruentes, algunos modelos producen un resultado, cuestión que no tiene sentido en la realidad.

En modelos matemáticos en forma de ecuaciones diferenciales, para obtener la evolución de las señales se precisa integrar las ecuaciones diferenciales del propio modelo. Por tanto, la simulación se traduce en un procedimiento donde se integran las ecuaciones diferenciales del modelo, pudiéndose llevar a cabo de dos maneras:

- Analógica: se desarrolla un circuito electrónico usando módulos analógicos que realizan ciertas funciones. El circuito electrónico obtenido es a su vez un sistema dinámico cuya evolución es análoga a la del modelo que se desea simular.
- Digital: se emplea un computador digital, con poder de procesamiento de información y capaz de generar un resultado. Es posible realizar programas que resuelvan problemas de cálculo de forma secuencial, es decir, como consecuencia de operaciones. Este tipo de algoritmos se llaman métodos numéricos.

### 4.2.3. La validación y verificación de los modelos

Con la construcción de un modelo se busca una representación válida de la realidad, por lo que se requiere una correlación alta entre lo que predice el modelo y lo que ocurre en el sistema real, de forma que los datos generados por el modelo sean iguales o muy parecidos a los posibles datos medidos en el sistema real, frente a un mismo conjunto de entradas presentadas.

En la Figura 4.2 se presentan las fases del proceso de análisis de validez de un modelo, donde destacamos lo siguiente:

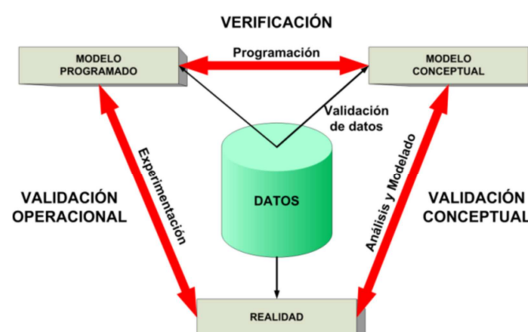


Figura 4.2. Esquema de Validación y Verificación de modelos. Fuente Propia.

- Validez operacional: medida de la validez global del modelo, verificando que los valores obtenidos por el modelo coinciden con los medidos en el sistema real.
- Validez conceptual: proporciona una medida de calidad del modelo, verificando que las hipótesis empleadas para construir el modelo conceptual a partir de la realidad son válidas.
- Verificación: valida el programa construido a partir del modelo conceptual.

A continuación se detallan varios métodos no excluyentes y complementarios para validar los modelos planteados:

- Ejercicio de vuelta a la formulación del problema, posibilitando la detección de ciertos errores.
- Comprobación de la exactitud y validez de todas las expresiones matemáticas.
- Modificar las entradas, observando la salida del modelo simulado y la realidad.
- Emplear datos históricos para reconstruir momentos pasados con el modelo y comprobar la salida real que tuvo el sistema. Se hace imprescindible no emplear datos usados para construir el modelo, tal y como se explicará más adelante.
- Recurrir a un experto humano para que corrobore que las variables del modelo presentan un comportamiento correcto.

Tal y como se ve en la Figura 4.2, para poder construir el modelo conceptual, evaluarlo y verificarlo, se precisa la existencia de datos. Por tanto, debe de existir una fase de validación de los datos, ya que en caso de no tener fiabilidad en la validez de los mismos, las pruebas para la construcción del modelo serían en vano.

#### ***4.2.4. Búsqueda de parámetros***

En ocasiones, los parámetros que aparecen en los modelos creados al aplicar leyes se pueden obtener por medición directa de la magnitud que representan, como pueda ser: una superficie, un volumen, etc.; en otras circunstancias, es posible estimar el valor de dichos parámetros por medio de medidas indirectas, a través de pruebas especiales en lugar de mediciones.

La búsqueda de parámetros es sencilla cuando éstos aparecen en las ecuaciones en forma lineal. Otras veces, las expresiones donde aparecen los parámetros no son lineales y por tanto hay que recurrir a artificios que permitan estimar valores adecuados de dichos parámetros de forma que los datos observados concuerden con los predichos por el modelo. Los valores de los parámetros de los modelos pueden obtenerse empleando datos de entrada, salida y perturbaciones, de forma que optimice el criterio de mínimos cuadrados

$$J = \left\| y_{obs} - y_{sim} \right\| \quad (4.15)$$

donde el vector  $y_{obs}$  almacena los valores medidos de la evolución del sistema e  $y_{sim}$  almacena los valores simulados por el modelo.

Algunos modelos presentan un gran número de parámetros a estimar, por lo que resulta difícil optimizar la función de costes (4.15) empleando técnicas clásicas de optimización no lineal. La Figura 4.3 muestra las técnicas de optimización no lineales clásicas, tomada de (Nelles, 2001). Puede que el algoritmo de optimización no converja por la alta dimensionalidad del problema y de las irregularidades contenidas en la función objetivo, como discontinuidades en las ecuaciones del modelo o los cambios bruscos que se producen en algunas variables. Además, el comportamiento de muchos modelos es complejo, apareciendo una interacción no lineal entre los parámetros y las distintas variables de salida, de forma que algunas de ellas son muy sensibles a unos parámetros y otras no, por lo que pequeñas variaciones en los parámetros pueden provocar grandes cambios en las salidas. De estas consideraciones, se puede concluir que el problema de minimización no tiene una solución analítica, por lo que se emplean técnicas de optimización iterativas, cuya estructura general se presenta en la Figura 4.4, tomada de (Blasco, 1999).

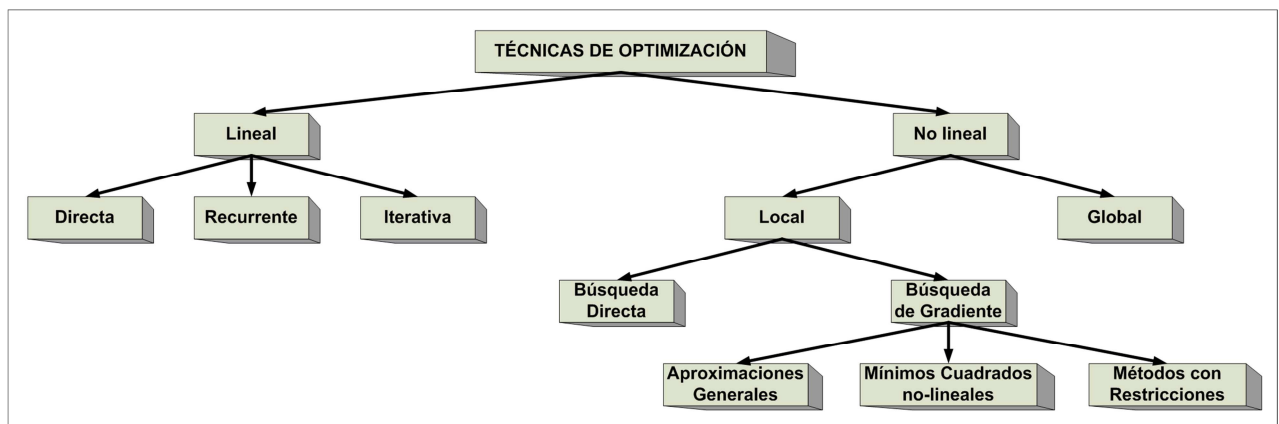


Figura 4.3. Cuadro de técnicas de optimización no lineal más empleadas. Fuente Propia.

Existen distintas técnicas de optimización de problemas multivariantes basados en métodos iterativos, el método más común es el de búsqueda secuencial directa, que consiste en un método iterativo en el que se utilizan todos los valores entre dos límites con un incremento determinado de forma que se obtenga la  $n$ -upla de parámetros que minimizan (4.15), una vez ejecutado el modelo en cuestión. Los parámetros a estimar se refinan en sucesivas ejecuciones de cada iteración del algoritmo, a la vez que disminuyen los límites del intervalo de búsqueda y el incremento de variación de los parámetros. La idea general es reducir iterativamente el tamaño de ese intervalo, partiendo de un entorno en el que se puede asegurar que se encuentre el mínimo. Esta búsqueda presenta la ventaja de que para incrementos de búsqueda no muy pequeños, el coste computacional es bajo, por lo que la obtención de límites entre los que se encuentra el mínimo se hace de forma sencilla y rápida, sin embargo para refinar el cálculo, conforme el incremento de búsqueda se hace menor, el coste computacional crece exponencialmente, resultando un tiempo de estimación

demasiado elevado, además, no son métodos muy eficientes en cuanto al número de evaluaciones a realizar de la función a minimizar. Por todo esto, se recomienda emplear otros métodos de búsqueda, se podría emplear el método del gradiente o el método de Newton, pero requieren que sea derivable la función a minimizar, y no siempre se consigue. Por estos métodos, se usan otros métodos de optimización heurísticos.

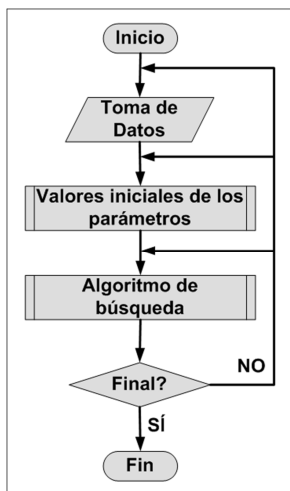


Figura 4.4. Esquema general de técnica de optimización iterativa. Fuente Propia.

Una de las opciones que se suele emplear para la obtención de una solución óptima de un problema son los algoritmos genéticos, como muestran Matous *et al.* (2000), esta técnica se ha utilizado con éxito para la obtención de los parámetros característicos de un modelo en distintas áreas de aplicación como por ejemplo en la estimación de lluvia caída presentada por Wang (1997), o en procesos de fabricación de acero presentado por Paul and Chaney (1998). El inconveniente de los algoritmos genéticos, es que no tienen la garantía de encontrar el óptimo global y, además, el coste computacional es demasiado elevado para encontrar unos márgenes de variación que aseguren el mínimo, pero una vez que se encuentran, el algoritmo de búsqueda es más eficiente que la búsqueda directa secuencial. También consiste en un método iterativo que parte de una generación aleatoria de una población inicial en unos límites dados a cada uno de los parámetros, posteriormente, se calcula la función a optimizar para cada uno de los individuos generados, se seleccionan aquellos individuos con mejores resultados, y se genera la siguiente población mediante los operadores genéticos de mutación y cruce. Estos pasos se repiten hasta que se cumpla una determinada condición de salida.

Como se ha visto, la búsqueda directa secuencial presenta el inconveniente del elevado coste computacional cuando el incremento de búsqueda de parámetros es muy pequeño, mientras que cuando es de uno o dos órdenes menor que los límites del intervalo de búsqueda, es sencillo acotar esta región, asegurando que los parámetros a encontrar se localizan en ella. Con las técnicas genéticas ocurre lo contrario, el coste computacional es demasiado elevado para encontrar unos márgenes de variación que aseguren un mínimo, pero una vez que se encuentren, el algoritmo de



búsqueda es más eficiente que la búsqueda secuencial. Por esta razón, la estimación de parámetros se puede dividir en dos etapas, combinando estas dos técnicas:

- Búsqueda directa secuencial para la definición del intervalo de variación de cada parámetro que asegure el mínimo de la función de coste.
- En el intervalo de búsqueda obtenido anteriormente, se emplea la búsqueda genética para la obtención definitiva de los parámetros óptimos.

#### **4.2.5. Sensibilidad de los modelos**

El objetivo del análisis de sensibilidad es valorar la influencia de cada parámetro del modelo con respecto a ciertos aspectos de la salida del mismo. Si se parte de un modelo de la forma  $y = f(x, u, \theta)$  donde  $y$  es la salida,  $x$  el estado,  $u$  es la entrada y  $\theta$  los parámetros, existen tres métodos para efectuar este análisis según Hamby (1994), siendo:

- Métodos analíticos: derivar la salida  $y$  con respecto al parámetro  $\theta_i$  que se desea analizar. La posibilidad de empleo o no de este método está sujeto a la continuidad o no de la función  $f(x, u, \theta)$  con respecto a  $\theta$ , o incluso la complicación de cálculo de su derivada.
- Métodos numéricos: evaluar la respuesta elegida para un determinado rango de valores del parámetro  $\theta_i$ . Como positivo del método, es que incluye la totalidad de las ecuaciones y soluciones numéricas que conforman el método y, por lo tanto, se podrá evaluar siempre sin importar la complejidad de las ecuaciones matemáticas, siendo el único escollo el tiempo que tarde en realizar la totalidad de las simulaciones.
- Método de distorsión: método para calcular perturbaciones en los valores de los parámetros, basándose en una secuencia particular de datos que se utiliza para verificar la influencia de los parámetros. La idea básica es que el modelo no se ajustará con exactitud a los datos experimentales, por tanto, el modelo deberá ser distorsionado, es decir, se deberá alterar el valor de los parámetros de una forma determinada, para que la salida del modelo coincida con las observaciones medidas.

### **4.3. LA IDENTIFICACIÓN DE SISTEMAS**

Lo que se pretende resolver con la identificación de sistemas es la problemática de la construcción de modelos matemáticos de sistemas dinámicos basados en observaciones que se encuentran en datos de la evolución del sistema real, según indica Ljung (1987). El caso más habitual es que los modelos tomen forma de ecuaciones en diferencias, aunque también pueden representarse por ecuaciones diferenciadas.

### 4.3.1. El planteamiento del problema

Sea un sistema dinámico con entrada  $u(k)$  y salida  $y(k)$ , vectores o escalares, se trata de obtener la relación existente entre las salidas y las entradas muestreadas del sistema. La relación dependerá de los valores pasados, tanto de la entrada como de la salida. Agrupando en un vector  $\mathbf{m}$  los valores pasados de entrada y salida se tiene

$$\mathbf{m}(k+1) = (y(k), \dots, y(k-L), u(k), \dots, u(k-M)) \quad (4.16)$$

en donde  $L$  y  $M$  dependen del sistema dinámico, y en caso de no conocerse de antemano, formarán parte del problema de identificación. Normalmente  $\mathbf{m}$  recibe el nombre de regresor, vector de entrada y, a veces, variable independiente.

Para el caso determinista, la relación entre la salida y el vector de entrada es

$$y(k+1) = f(\mathbf{m}(k+1)) \quad (4.17)$$

siendo  $f$  en el caso más general una función no lineal, pudiendo existir alguna información disponible de  $f$  o no.

La ecuación (4.17) presupone que el origen de los datos tiene forma concreta, esto es una hipótesis que debería ser contrastada posteriormente y por tanto no considerarse como una verdad inicialmente. La anterior ecuación puede recibir el nombre de modelo verdadero, esta denominación debe de entenderse como el mejor modelo que se obtiene bajo ciertas hipótesis de partida.

Por tanto, la identificación consistirá en hallar la relación  $f$ , o bien, una aproximación de la misma a partir de los datos recogidos del propio sistema. La totalidad de datos tomados durante la evolución del sistema se agrupan en el  $CE$ , el cual viene definido por

$$CE^N = \{(\mathbf{m}(k), y(k)) / k = 1, \dots, N\} \quad (4.18)$$

contiene pares del vector de entrada y la salida del sistema, llamándose a estas parejas como patrones, datos entrada-salida o de forma abreviada  $e-s$ . Con el anterior conjunto se puede inferir la siguiente relación

$$\hat{y}(k) = \hat{g}^N(\mathbf{m}(k)) \quad (4.19)$$

la notación anteriormente empleada, expresa el hecho de que (4.19) es una aproximación al considerado, supuesto real, sistema (4.17) realizada con los  $N$  pares del conjunto  $CE^N$ .

El siguiente paso es generar  $\hat{g}^N$  en (4.19), una forma de hacerlo consiste en buscar  $\hat{g}^N$  dentro de un conjunto de funciones parametrizadas como

$$G = \{g(\mathbf{m}(k), \theta) / \theta \in \Psi\} \quad (4.20)$$

Al no existir indicios acerca de la parametrización, se hace necesario seleccionar un conjunto de funciones lo bastante flexibles como para describir cualquier posible relación  $g$ .

En ocasiones, el conocimiento a priori acerca del sistema permite escoger la relación, las funciones suelen ser descritas como la siguiente serie

$$g(\mathbf{m}, \theta) = \sum_{j=1}^{\infty} \theta_j g_j(\mathbf{m}) \quad (4.21)$$

donde  $\theta_j$  son las componentes del vector  $\theta$ . Se hace necesario determinar ahora las funciones  $g_j(\mathbf{m})$ , las cuales deben describir a la verdadera  $f(\mathbf{m})$ , es decir, para cierto valor de parámetros  $\theta^*$  tener

$$f(\mathbf{m}) = g(\mathbf{m}, \theta^*) = \sum_{j=1}^{\infty} \theta_j^* g_j(\mathbf{m}) \quad (4.22)$$

Algo deseable es que el error que se obtiene al truncar la serie (4.22) decrezca con el número de términos sumados. Denotando el número de términos por  $n_n$  entonces la serie truncada es

$$g(\mathbf{m}, \theta, n_n) = \sum_{j=1}^{n_n} \theta_j g_j(\mathbf{m}) \quad (4.23)$$

La llamada elección de la clase del modelo consiste en definir las bases  $\{g_j\}$ , fijar el número de bases  $n_n$  constituye la elección del tamaño del modelo dentro de la clase. Una vez hecho esto, resta el ajuste del vector de parámetros  $\theta$ . Si se supone que un vector de parámetros  $\hat{\theta}$  ha sido ya seleccionado, entonces la siguiente aproximación permite predecir la salida del sistema

$$\hat{y}(k, \hat{\theta}) = \sum_{j=1}^{n_n} \hat{\theta}_j g_j(\mathbf{m}(k)) \quad (4.24)$$

Habiendo conseguido de esta manera un modelo predictivo de gran utilidad en muchas aplicaciones. Para finalizar, resta la obtención del parámetro  $\theta$ , que puede ser estimado a partir del *CE* de la manera siguiente

$$\hat{\theta}^N = \arg \min_{\theta} \sum_{k=1}^N |y(k) - \hat{y}(k, \theta)|^2 \quad (4.25)$$

llegando finalmente a la expresión del modelo estimado

$$\hat{y}(k) = \hat{y}(k, \hat{\theta}^N) = g(\mathbf{m}(k), \hat{\theta}^N, n_n) \quad (4.26)$$

En resumen, la identificación consiste en hallar una familia de modelos, de entre la cual, poder obtener una aproximación adecuada para los datos observados. En ocasiones, el conocimiento de los mecanismos que intervienen en la evolución del sistema permite orientar la búsqueda de la familia comentada.

### 4.3.2. Las etapas de la identificación

La identificación es un proceso iterativo, ya que el objetivo buscado no se alcanza hasta varias pruebas, las distintas etapas de la identificación deben de ser procesadas varias veces. Los pasos a seguir, según Ljung (1987), en la identificación son los siguientes:

- Obtención de datos.
- Filtrar datos.
- Escoger variables independientes.
- Escoger estructura del modelo.
- Estimación de parámetros del modelo.
- Validación.

Estos pasos se recorren en orden pero a veces es necesario volver atrás, como se puede apreciar en la Figura 4.5.

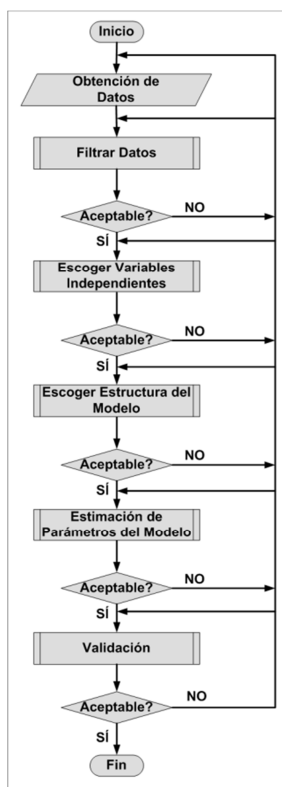


Figura 4.5. Esquema de los pasos de la identificación. Fuente Propia.

Como norma general, la identificación automática no existe salvo para situaciones particulares, se precisa por tanto la participación del humano, que resuelvan algunos dilemas, como por ejemplo la relación entre la complejidad del modelo y el ajuste del mismo. Se deberá decidir cuándo una etapa es satisfactoria y se pasará a la siguiente, e incluso la vuelta atrás y evaluación de una etapa ya pasada.

### 4.3.3. El filtrado de los datos

No es posible obtener buenos predictores a partir de datos incompletos o con un exceso de ruido. Por desgracia, no existe ningún mecanismo automático que decida esas cuestiones. Siempre que se pueda, un primer paso sería dibujar gráficamente los datos, pudiendo servir de ayuda en ciertos casos las gráficas temporales de las trayectorias de las señales. Las gráficas de algunas variables frente a otras también sirven de ayuda, detectando correlaciones entre variables.

Si se detecta alguna irregularidad, conviene estudiar el significado de ellas, antes de proseguir con el resto de etapas del proceso de identificación. Entre las irregularidades más frecuentes están:

- Falsas medidas: son fáciles de detectar ya que no siguen la distribución característica del resto. Para su detección, basta con obtener cotas de los valores alcanzables por las variables y sus derivadas. En el caso de emplear estos datos, las falsas medidas pueden causar falsas elecciones de modelos y posteriormente malas predicciones. Es importante no confundir falsas medidas con valores inusuales, como puedan ser los provenientes de arranque de equipos o procesos, días festivos, etc.; esta observación es importante si en lugar de usar cotas previamente definidas se emplean métodos como aplicaciones auto-organizadas para clasificar datos.
- Eliminación de ruido: en estadística, ruido significa la variación de la salida que es impredecible con respecto a la variación de las entradas. El ruido en los datos limita la capacidad de predicción de los modelos a que dan lugar, la mejor forma de evitar los malos efectos del ruido consiste en tener una gran cantidad de datos, de manera que las contribuciones del ruido en varias medidas similares se cancelen unas con otras. La presencia de ruido en la variable de salida provoca que el error cuadrático medio en los datos futuros, nunca pueda ser menor que la varianza del mismo, independientemente de la cantidad de datos disponibles para entrenamiento o del tipo de modelo usado.
- Normalización: consiste en escalar cada variable de forma que la serie de datos resultante tenga media nula y varianza unidad. Esta técnica tiene gran importancia a la hora de emplear algoritmos numéricos para la obtención de parámetros como es el caso de la identificación. Si se decide no emplear la normalización existen otras precauciones que deberán tomarse en cuenta: asegurarse de que los valores numéricos empleados para describir las señales están dentro del intervalo de precisión de la computadora; interesa que el resto de las variables estén expresadas en escalas parecidas, el uso de escalas muy diferentes puede provocar efectos dañinos en modelos que emplean la distancia euclídea.

- Redundancia: como se ha comentado se necesita un conjunto de datos que contenga las entradas y las salidas correctas, este conjunto llamado *CE*, es procesado por el algoritmo de entrenamiento hasta que el modelo consigue representar la relación  $e-s$  observada en los datos con error aceptable. Es conveniente tener un *CE* pequeño para minimizar la cantidad de cálculos a realizar, por otra parte, el *CE* ha de contener la suficiente información para que se pueda realizar la identificación de forma adecuada, como por ejemplo en el aprendizaje de una *ANN*. El empleo de técnicas para seleccionar ejemplos de entrenamiento tiene el beneficio adicional de mejorar la capacidad de generalización del modelo al eliminar patrones redundantes.

#### 4.3.4. El proceso de selección de las variables de entrada

Consiste en escoger un grupo de valores que permitan calcular la salida deseada del modelo, permitiendo el posterior desarrollo de modelos adecuados. Por el contrario, tomar entradas con fuerte correlación, o incluso sin interés, puede llevar a la construcción de modelos sobreparametrizados, cuya consecuencia podrá ser predicciones erróneas. Se puede dar lo contrario, dejar variables fuera que sí que tienen influencia en la salida, en tal caso, el modelo producirá predicciones pobres debido a esa falta de información vital.

La selección de variables de entrada se reduce a hallar el número de valores pasados de las variables de interés, es decir, los valores retrasados que se usarán como vector de entrada. Existen métodos basados en la inclusión paulatina de nuevas variables en un modelo con pocas entradas y, en la eliminación gradual en el modelo que incluye todas las entradas posibles, necesitando en ambos casos un alto número de ensayos de prueba y error. Esta última técnica es muy empleada, en (Czernichow and Muñoz, 1995) se emplea un método estadístico basado en las derivadas de la salida de un modelo con respecto a sus entradas, el método permite seleccionar las variables con mayor influencia en sentido estadístico sobre la salida.

Otra tendencia proviene del estudio de los sistemas caóticos, los atractores<sup>1</sup> en estos sistemas puede tener geometría fractal, por lo que pueden ser englobados en la dimensión entera inmediatamente superior a su dimensión fraccionaria. Para reconstruir la evolución de los sistemas caóticos a partir de medidas es preciso emplear un vector de entrada con dimensión adecuada, en (Kennel *et al.*, 1992) se emplea el método de los falsos vecinos para decidir la dimensión más adecuada.

A continuación detallaremos algunos de los principales métodos empleados en las aplicaciones de ingeniería, siendo: método sin modelo; método basado en datos reservados; y método basado en el gradiente.

---

<sup>1</sup> Es el conjunto al que el sistema evoluciona después de un tiempo suficientemente largo.

#### 4.3.4.1. Método sin modelo. Estimador de la varianza

Sea un conjunto de pares  $(x(k), y(k))$  para  $k = 1, \dots, N$ , se trata de derivar directamente de los datos las variables del vector  $x$  que más influencia tienen en  $y$ , para incorporarlas luego al vector de entrada  $m$ . El estimador presentado ha sido usado por Yuan and Fine (1998) para la predicción de la demanda eléctrica y está basado en un método de diferencias.

Se denota por  $x_i(k)$  a la componente  $i$ -ésima del vector  $x(k)$ , para cada variable de entrada  $x_i$  se va a obtener un índice que evalúa su utilidad  $U_i$ , y que mide los cambios en la variable de salida asociados a cambios en la entrada  $x_i$ . El índice se obtiene como sigue, las parejas  $(x_i(k), y(k))$  son reordenadas de forma que las nuevas parejas  $(x_i(r), y(r))$  verifiquen que  $x_i(r) \leq x_i(q)$  para todo  $r < q$ . El índice se calcula a partir de la secuencia ordenada de valores de salida, como sigue

$$U_i = \sum_{h=1}^N (y(h+1) - y(h))^2 \quad (4.27)$$

Este índice será alto para variables con poca relación con la salida, mientras que las que sí influyen en la salida producen valores menores.

#### 4.3.4.2. Método basado en datos reservados

Las variables de entrada se pueden seleccionar construyendo modelos con distintos vectores regresores y analizando cuál de todos los modelos es el mejor. Los métodos comentados a continuación son empleados en la etapa de validación del modelo, siendo los siguientes:

- *Validación Cruzada (VC)*: permite seleccionar modelos empleando un conjunto de datos que no han sido usados para desarrollar el modelo. Este modelo proporciona una indicación de lo buenas que serán las predicciones futuras, o dicho de otra forma, permite estimar la capacidad de generalización. El método más simple consiste en reservar parte de los datos de forma que ni la estructura ni los parámetros del modelo usen la información que contienen, posteriormente, se emplean esos datos para calcular los errores de predicción del modelo, los datos para obtener el modelo, como ya se sabe, reciben el nombre de *CE*, los datos reservados son llamados *CP*. *MAPE* (u otra medida de error) en la predicción en el *CP* es un valor que se puede asignar a cada modelo desarrollado independientemente de su estructura y vector de entradas, pudiendo comparar distintos modelos y seleccionar el mejor, y por tanto, el mejor conjunto de entrada. El método presenta algún inconveniente, en primer lugar si los datos no son abundantes el método requiere sacrificar una parte de ellos para la prueba, por otra parte, la división de los datos en *CE* y *CP* puede tener una gran influencia en los resultados.

- Validación cruzada múltiple: es una variante de la anterior, cuyo objetivo es solucionar los problemas mencionados de la VC. La validación de orden  $k$  consiste en dividir los datos disponibles en  $k$  subconjuntos y repetir el método  $k$  veces. En cada repetición se toma como  $CP$  uno de los  $k$  subconjuntos mientras que para  $CE$  se usan los restantes  $k - 1$  subconjuntos. El inconveniente de este método es que el proceso de entrenamiento es repetido  $k$  veces, por lo que la carga del cálculo aumenta proporcionalmente.
- Remuestreo: en lugar de analizar repetidamente subconjuntos de datos, se van analizando submuestras, donde cada una de ellas proviene de un muestreo aleatorio con reemplazamiento de los datos originales. Los estudios realizados mediante técnicas de remuestreo no garantizan buenos resultados en el contexto de la selección de entradas, tanto en términos de convergencia como de robustez frente a condiciones iniciales, como queda patente en (Zapranis and Refenes, 1999).

#### 4.3.4.3. Métodos basados en el gradiente

El dilema de seleccionar el número óptimo de valores pasados para usarlos en el vector de entrada se puede plantear como el proceso de seleccionar las variables que son más necesarias para explicar cambios en la señal de salida. A fin de dotar de mayor generalidad al problema, en el regresor se añaden tres tipos de variables en lugar de dos, quedando el vector de entrada de la siguiente manera

$$\mathbf{m}_{L,M,N}(k) = (y(k), \dots, y(k-L), u(k), \dots, u(k-M), p(k), \dots, p(k-N)) \quad (4.28)$$

los subíndices indican el número de valores pasados de cada una de las variables de interés y  $p$  una variable de entrada no manipulable.

Aproximando la dinámica del sistema mediante una función  $f$  donde sus argumentos son las componentes del vector, tenemos la siguiente expresión

$$\mathbf{m}_{P,Q,R} := (y(k), \dots, y(k-P), u(k), \dots, u(k-P), p(k), \dots, p(k-R)) \quad (4.29)$$

Por tanto, se deben obtener los valores de  $P, Q$  y  $R$  de modo que se obtenga una buena aproximación de la realidad, cuanto mayores sean los valores de  $P, Q$  y  $R$  mejor se podrá aproximar la salida de cualquier sistema, pero no se pueden tomar arbitrariamente grandes ya que podríamos encontrarnos con un problema sin solución.

Ya que  $L+M+N > P+Q+R$ , se puede dar el caso de que la función  $f$  resulte multivaluada, en efecto, supóngase que a partir de los datos observados de la evolución del sistema y almacenados en  $CE$  se obtengan dos valores de  $x_{L,M,N}$  tales que corresponden a un mismo valor de  $x_{P,Q,R}$ , en tal caso la función  $f$  debe de tomar dos valores distintos en un mismo punto,



añadiendo valores pasados adicionales se puede conseguir que la ambigüedad en la salida desaparezca.

El método, de manera formal será: supongamos un par de regresores tales como  $\xi^1$  y  $\xi^2$ , el valor absoluto del gradiente de la función  $f$  a lo largo de la línea que pasa por ambos podría estimarse como

$$|\nabla_{12}f| \approx \frac{|f^1 - f^2|}{\|\xi^1 - \xi^2\|} \quad (4.30)$$

mientras que la distancia  $\|\xi^1 - \xi^2\|$  sea lo bastante pequeña. Para un par  $(\xi^1, \xi^2)$  dado, el valor de  $|\nabla_{12}f|$  puede ser disminuido aumentando el regresor con algún nuevo valor pasado que explica la diferencia en los valores de salida, dicho de otra manera, añadiendo la variable que causa el cambio en la salida, aunque seguramente tan sólo de forma parcial. Por tanto, el algoritmo de puede expresar de la siguiente manera:

- Reunir un conjunto de valores de  $e-s$  de las variables de interés, obtenidas de forma que representen el comportamiento del sistema donde interese.
- Asignar valores iniciales a  $P, Q$  y  $R$ .
- Formar pares  $e-s$  de la forma  $z := (x_{P,Q,R}, y)$ , donde  $x_{P,Q,R}$  es el regresor de entrada al modelo e  $y$  la salida correcta para su entrada correspondiente. El conjunto de pares se denota por  $A_{P,Q,R}$ .
- Para los pares  $z^1, z^2$  de  $A_{P,Q,R}$  hacer: se calcula  $d_z = \|z^1 - z^2\|$ ; si  $d_z < \Delta$  estimar el gradiente según (4.29) y almacenar el par  $z^1, z^2$  y el valor del gradiente en un conjunto que contiene tríos llamado  $B_{P,Q,R}$ .
- Incrementar de manera separada, y en una unidad  $P, Q$  y  $R$ , formando los conjuntos  $A_{P+1,Q,R}, A_{P,Q+1,R}$  y  $A_{P,Q,R+1}$ .
- Repetir 3 para los conjuntos anteriores y seleccionar la combinación que produce el mayor descenso del gradiente.
- Si los resultados son aceptables se finaliza el algoritmo, en caso contrario, volver a introducir nuevas señales según lo explicado.

Si las variables del modelo se normalizan al intervalo  $[-1,1]$  y se toma  $\Delta = \sqrt{\dim(z)}\delta^2$ , el problema consiste en elegir la cantidad  $\delta$  que se puede interpretar como una fracción de la escala completa del rango de variables. La dimensión del regresor de entrada se ha incluido en la fórmula para tener en cuenta el efecto de dispersión de los datos a medida que la dimensión aumenta.

Otro aspecto es la parada del algoritmo, un posible criterio de parada puede ser cuando la dimensión del regresor de entrada  $P+Q+R$  alcanza un cierto valor prefijado. Otro criterio sería observar el descenso del gradiente al aumentar la dimensión del regresor de entrada y parar cuando dicho descenso se estabiliza o es menor que cierto valor prefijado.

Una posible estrategia sería el realizar el algoritmo varias veces obteniendo un cierto número de modelos distintos, los modelos pueden ser comprobados con el conjunto de datos y tomar el modelo que da la mejor relación entre la eficacia y el coste computacional.

#### **4.3.5. El proceso de selección de los modelos**

Los métodos constructivos permiten obtener al mismo tiempo la estructura y los parámetros del modelo. Por ejemplo, en el caso de los sistemas no lineales, existen métodos de este tipo tanto para las redes sigmoideas como de bases radiales e incluso hay métodos que permiten decidir las variables del vector de entrada. En los métodos tradicionales se separa la selección de la estructura del modelo de la selección de entrada y de la estimación de parámetros del mismo.

Para la selección de un modelo, se debe fijar la atención en el número de nodos que lo forman, si los aproximadores empleados son locales o globales y el número de capas necesarias para el funcionamiento correcto del modelo. A continuación, se darán algunos detalles a tener en cuenta para la selección del modelo, y que tiene que ver con el número de nodos, aproximadores y el número de capas.

##### **4.3.5.1. Número de nodos**

El número óptimo de nodos depende de otros factores, como el número de entradas del modelo, el número de patrones, la existencia o no de ruido en los datos, tipo de *ANN* empleada (si es el caso), número de capas del modelo, etc.; pero sin lugar a dudas, de la complejidad de las funciones que se emplean como base.

En la mayoría de los casos se recurre a un método de prueba y error para obtener el número de nodos de manera aproximada. Para complicar más las cosas, ha de tenerse en cuenta que el empleo de regularizadores<sup>2</sup> durante el entrenamiento modifica sustancialmente el número de ejemplos necesarios por parámetro. Si se emplea detección temprana del entrenamiento, es preciso añadir más nodos que los necesarios para evitar caer en mínimos locales, tal y como se presenta en (Arahal *et al.*, 2002).

---

<sup>2</sup> Términos que reducen los valores de ciertos parámetros.

#### 4.3.5.2. Aproximadores locales y globales

Sirven para seleccionar la familia de modelo de nuestro sistema. En los aproximadores locales, las funciones base tienen un valor que sólo cambia en un entorno de cierto punto  $c$  del espacio de entrada, siendo cero en el resto, como es el caso en las redes Función de Base Radial (*Radial Basis Function – RBF*). Por el contrario, los aproximadores globales emplean funciones base cuya salida es distinta de cero en todo el dominio, siendo un ejemplo las Perceptron Multi-Capa (*Multi-Layer Perceptron – MLP*). Ambas familias poseen la propiedad de ser aproximadores universales y por tanto pueden intercambiarse, sin embargo, desde el punto de vista práctico, estos aproximadores presentan diferencias, siendo las ventajas de los aproximadores globales las siguientes:

- Recurren a un menor número de funciones base que los locales.
- A medida que la dimensión del espacio de entrada aumenta, el número de funciones base necesario aumenta, pero no de manera exponencial.

En contrapartida, las ventajas de los aproximadores globales son:

- Se pueden usar con facilidad en sistemas que se adapten *on-line*.
- Los parámetros se adaptan de manera más rápida.
- El orden de presentación de los datos afecta poco a la estimación de parámetros.
- En (Gorinevsky, 1995). Se presentan las condiciones que han de tener las señales de entrada para que haya excitación persistente.

#### 4.3.5.3. Número de capas

Con una capa oculta se producen resultados más que aceptables en situaciones en las que la desviación de la linealidad presente en los datos no es severa de acuerdo con las funciones base disponibles, este tipo de redes posee la propiedad de aproximación universal, como subraya Hornik *et al.* (1989).

El empleo de más de una capa oculta proporciona mayor flexibilidad a la red, con muchos menos nodos, ocurre porque la entrada de la segunda capa consiste en señales ya procesadas por la primera y por lo tanto más complejas que las entradas, sin embargo, cuanto más capas ocultas existan más lento el entrenamiento y la aparición de mínimos locales aumenta.

#### 4.3.6. Ajuste de los parámetros

Es el problema más sencillo en la identificación, existiendo métodos numéricos para obtener aproximaciones para las soluciones de (4.25) consistentes en mínimos respecto a los parámetros. A veces, pueden ser modelos muy sencillos como el caso de los mínimos cuadrados, o más complicados, recurriendo a técnicas que requieren gran capacidad de cálculo como en los

algoritmos genéticos y evolutivos, u otros métodos incluidos en la clasificación de métodos de optimización.

A continuación se hará una clasificación de los algoritmos de ajuste más empleados, para continuar con una breve reflexión sobre la adaptación futura del modelo, y terminar con el método de adaptación empleado en el algoritmo de identificación.

#### **4.3.6.1. Clasificación**

Una clasificación típica de los algoritmos de ajuste puede ser la siguiente:

- Por lotes: se procesan bloques formados por datos históricos. Los algoritmos tienen mayor flexibilidad para manejar datos y obtener un ajuste más eficiente de los parámetros.
- En línea: los datos se emplean en el momento de su llegada, y ésta es a intervalos regulares de tiempo. Los algoritmos que trabajan en línea sacan el máximo partido de cada muestreo de señales, al mismo tiempo que estos muestreos son realizados.
- En línea con ventana: situación de compromiso entre las dos anteriores, se trabaja en línea pero se dispone de una memoria que almacena datos de una ventana temporal que va asociada al último muestreo.

Otra forma de clasificarlos se basa en el método utilizado para seleccionar los parámetros que tratan de minimizar (4.25), la mayoría de los métodos emplean una búsqueda realizando un recorrido en el espacio de los parámetros comenzando por un valor inicial y añadiendo un término corrector en función del error de la aproximación actual o el lote. Otros métodos parten de valores iniciales de los parámetros, para posteriormente combinarlos y crear nuevos individuos, sería el caso de los algoritmos genéticos y evolutivos.

En cuanto al tipo de recorrido que se realice en el espacio de parámetros los algoritmos de ajuste se pueden clasificar de la siguiente manera:

- Paso fijo: el camino hacia el mínimo es constante, debido a que el incremento que se aplica a los parámetros estimados es fijo.
- Paso adaptable: las alternativas de los parámetros estimados durante su ajuste presentan velocidades no constantes. Existen diferentes técnicas para el cambio de velocidad, como son: derivadas superiores; gradiente estimado; o algoritmo Manhattan.

#### **4.3.6.2. Frecuencia de la adaptación**

Los sistemas sufren cambios estructurales con el tiempo debido al propio paso del tiempo, cambios intrínsecos u otras causas. Sea cual sea el cambio, puede ser necesario o recomendable repetir alguno de los pasos de la identificación. Los métodos que usan modelos adaptados en línea tienen una gran ventaja sobre los que emplean datos tomados por lotes y es, que se ajustan de

manera inmediata a los cambios que puedan producirse en el sistema, pero la adaptación en línea también tiene sus desventajas:

- Alrededor de un punto de funcionamiento, las medidas proporcionan información no completa para aclarar el correcto valor de los parámetros, pudiéndose perder la exactitud de los mismos.
- El ruido de los datos puede repercutir en los valores que se estiman.
- La adaptación en línea no se realiza fácilmente para modelos del tipo *Perceptron*.

De manera alternativa a la adaptación en línea se puede recurrir a la adaptación con ventana, la idea es mantener una ventana temporal de datos que se mueve con el muestreo en curso, equivale a una pequeña memoria que permite que la adaptación de los parámetros se lleve a cabo teniendo en cuenta más datos que el error de predicción del muestreo en curso.

#### 4.3.6.3. Método de adaptación

La mayoría de las aplicaciones de técnicas no lineales de predicción emplean *ANN*, en las cuales los parámetros son estimados mediante un proceso de entrenamiento que emplea adaptación con paso fijo. Es importante tener en cuenta los problemas de estos métodos basados en el gradiente, pudiéndose resumir en:

- El algoritmo basado en gradiente emplea la derivada primera del error respecto a los parámetros, por lo que suele aparecer el problema del paso de adaptación. La dirección que indica la derivada permite avanzar en el sentido del error decreciente en el espacio de pesos, si se toman pequeños pasos en la dirección apropiada se llega a un mínimo local, pero si los pasos son muy grandes se pueden producir oscilaciones e incluso no converger al mínimo, e incluso, si los pesos son muy pequeños el entrenamiento puede llevar un tiempo infinito. Existen algoritmos que intentan resolver este problema, tratando de adaptar el tamaño del paso, como son *Quickprop* de Fahlman and Lebiere (1990) y *Rprop* de Riedmiller and Braun (1993).
- Los parámetros sometidos al ajuste tratan de evolucionar todos a un tiempo, se debe pensar que algunos parámetros no afectan a la salida, sino que generan o afectan a variables intermedias que pueden ser útiles para el objetivo global, a esta labor se le conoce como detección de características. Ya que todos los parámetros están cambiando con el tiempo, la adaptación se complica bastante, y es posible que un cambio en un parámetro ayude por un lado y perjudica indirectamente por otro, esto es conocido como objetivo móvil.

### 4.3.7. La validación

El objetivo de validar es determinar la exactitud de un modelo para un propósito determinado, realizándose habitualmente de una manera visual, comparando la salida de la realidad y la del modelo frente a determinados estímulos. Hay que tener presente, que los modelos no lineales son sensibles al tipo de entrada y que además nunca replican de manera exacta la realidad, teniendo por tanto su validez en un determinado rango. Una manera clásica de validación es por inspección visual de los errores de la predicción, pudiendo revelar por medio de una mirada si existen diferencias sistemáticas o estructurales entre modelo y sistema.

Puede resultar de interés someter al predictor a una prueba más intensa que su funcionamiento en condiciones normales, por ejemplo, si el predictor ha sido diseñado para proporcionar la predicción a un paso, puede ser interesante usarlo recurrentemente para obtener predicciones en un horizonte  $H \gg 1$  y de este modo identificar potenciales debilidades, como pudiera ser inestabilidades en las predicciones. En cualquier caso la prueba final antes de emplear el predictor, conlleva la comprobación de la capacidad de generalización del predictor diseñado.

#### 4.3.7.1. Generalización

Durante el entrenamiento la salida del estimador se amolda al  $CE$ , lo cual es interesante pero no es el objetivo buscado en la predicción, puesto lo que se desea es que la salida del modelo se asemeje a la salida observada para patrones no presentados, o dicho de otra forma, que actúe correctamente en situaciones futuras.

Nos podemos remontar al origen de la ciencia, para ver como la generalización de las estimaciones tiene ciertas implicaciones filosóficas, como las de Russel (1948), ya que no dejan de ser modelos que tratan de concluir de manera general a partir de la casuística particular.

Se han de dar tres condicionantes necesarios pero no suficientes para producirse generalización, siendo éstas:

- Los patrones  $e-s$  deben disponer de suficiente información del modelo a generar.
- La función que se intenta aproximar debe de ser continua en relación con las funciones base del estimador.
- Los ejemplos empleados en el entrenamiento deben de ser representativos de las situaciones futuras durante la fase de predicción.

La identificación de sistemas puede interpretarse como un optimizador de la complejidad del modelo, donde dicha complejidad está relacionada con el número de parámetros que contiene. La Figura 4.6 presenta un esquema básico para analizar las componentes del error de generalización en la identificación de señales,  $u$  es la entrada al modelo,  $y_u$  la salida del modelo debida a cambios en

la entrada,  $v$  se considera ruido blanco de varianza  $\sigma^2$ , que incluye la dinámica de salida del proceso no debida a la entrada  $u$ ,  $y$  es la salida observada del sistema,  $\hat{y}$  la salida estimada del modelo y  $e$  el error entre la salida del sistema y la estimada del modelo.

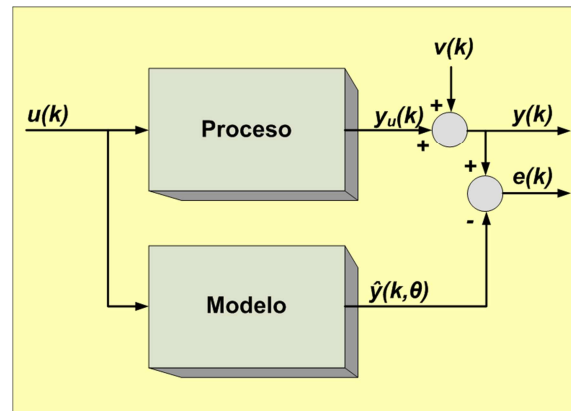


Figura 4.6. Esquema general del análisis del error durante la identificación. Fuente Propia con Datos de Arahal *et al.* (2002).

La esperanza matemática del error cuadrático, tiene dos componentes, el error de sesgo y el error de la varianza, siendo estas dos componentes las que aparecen en el segundo miembro de la siguiente igualdad

$$E\{(y_u - \hat{y})^2\} = [y_u - E\{\hat{y}\}]^2 + E\{\hat{y} - E\{\hat{y}\}\}^2 \quad (4.31)$$

La poca o mucha flexibilidad del modelo viene caracterizada por la componente del sesgo, al incluir errores sistemáticos debidos a:

- La característica particular de la entrada.
- Las particularidades de la selección de la estructura del modelo.
- El modelo de operación de la toma de datos, por ejemplo, bucle cerrado en lugar de abierto.

Por tanto, la componente de error del sesgo es independiente del ruido y las perturbaciones. Los errores de sesgo no se pueden eliminar a base de tomar más datos ( $\infty$ ) para la identificación.

La componente del error de la varianza es una medida de la desviación de los parámetros óptimos, contiene errores aleatorios debido al ruido en los datos, que no permiten al modelo reproducir exactamente la salida del sistema, estando afectado por los siguientes factores:

- Parámetros del modelo en número.
- Razón señal/ruido de la totalidad de los datos.

El error de varianza se puede minimizar con más datos, Ljung (1987) se muestra como el error de la varianza es proporcional al cociente entre el número de parámetros del modelo  $n$  y el número de datos utilizados en la identificación  $N$ , el error es  $\approx \sigma^2 n / N$ . Tras el análisis anterior, queda claro que el problema de identificación puede verse como una solución de compromiso entre errores de sesgo y de varianza para una arquitectura determinada del modelo. En general, los errores de sesgo y de varianza no son conocidos, de modo que se suelen estimar varios modelos de

diferente complejidad y se comparan los errores evaluados sobre el  $CE$ . Si se empleara el  $CE$  para la validación, no se podría detectar el error de la varianza.

#### 4.3.7.2. Sobreparametrización

Cuanto mayor es la flexibilidad de la familia de modelos empleada, mejor puede aproximarse el conjunto de datos empleados para estimar los parámetros, por ejemplo cuanto mayor es el número de nodos de una  $ANN$ . Tiene fácil explicación, a medida que se incorporan nuevas bases es posible reproducir no sólo las características del sistema real que ya subyace en los datos, sino también el ruido superpuesto en las medidas. Lo anterior es una situación peligrosa, ya que el ruido no va a repetirse en nuevas medidas de datos y, por tanto, el modelo producirá resultados incorrectos con ellos.

Veamos la interpretación geométrica de la sobreparametrización, supongamos una serie de puntos  $(x_i, y_i)$  medidos sobre una curva  $y = f(x)$  con cierto ruido en la medida como vemos en la curva azul de la Figura 4.7, donde  $f(x)$  es desconocida (suponemos las observaciones coincidentes con los picos de la curva azul). Una posible aproximación es con un polinomio de la forma  $\hat{y} = P(x)$ , los coeficientes del polinomio se pueden obtener de forma que se minimice la desviación con respecto a las medidas, en morado con un polinomio de grado tres y en rojo otra con polinomio de grado ocho. En el segundo de los casos, el error entre los datos observados y los predichos es mucho menor que con el polinomio de grado tres, sin embargo, la aproximación con este polinomio es más suave. Lo que ocurre es que el exceso de parámetros ha provocado el ajuste del modelo al ruido, conociéndose esta situación como sobreparametrización.

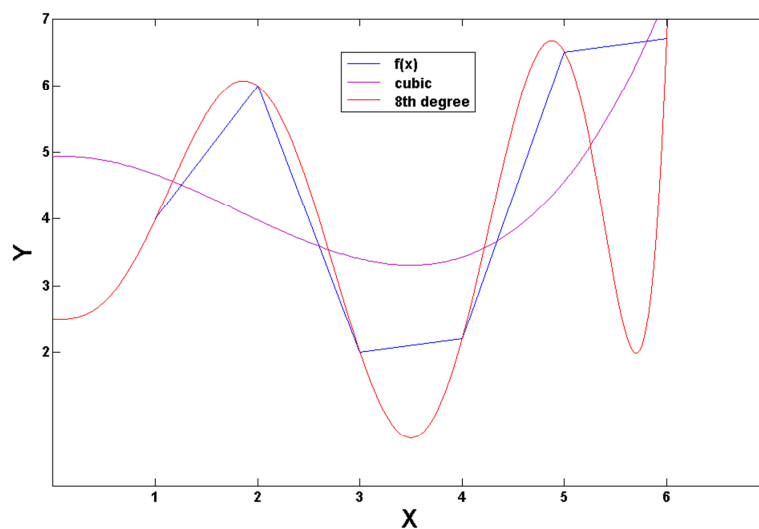
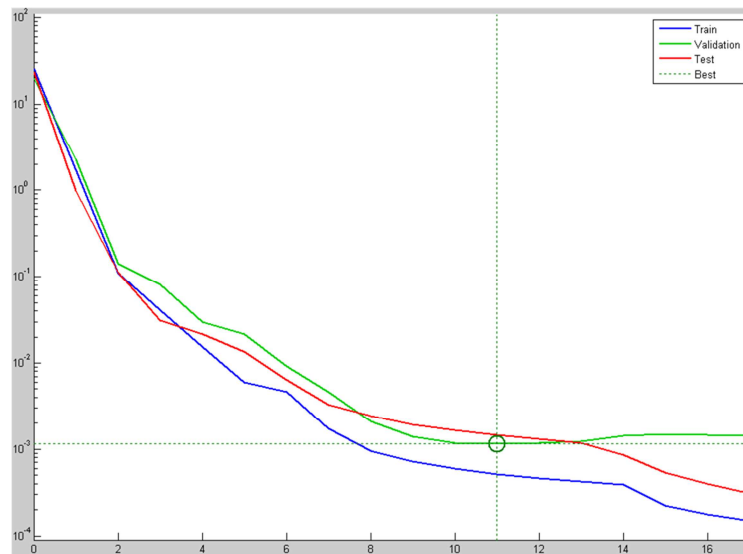


Figura 4.7. Noción geométrica de la sobreparametrización. Fuente Propia.

El Sobreentrenamiento (*overfitting*) es el efecto de entrenar un algoritmo de aprendizaje con ciertos datos para los que se conoce el resultado deseado. El algoritmo de aprendizaje debe de



alcanzar un estado en el que será capaz de predecir el resultado en otros casos a partir de lo aprendido con los datos de entrenamiento, generalizando para poder resolver situaciones distintas a las acaecidas durante el entrenamiento. Sin embargo, cuando un sistema se entrena demasiado, o se entrena con datos extraños, el algoritmo de aprendizaje puede quedar ajustado a unas características muy específicas de los datos de entrenamiento que no tienen relación causal con la función objetivo. El algoritmo ajustará con exactitud en el *CE*, mientras que comenzará a fallar para *CP*, si el error de validación se incrementa mientras que el de entrenamiento decrece puede que se esté produciendo una situación de sobreajuste. En la Figura 4.8 se puede apreciar como a partir del punto óptimo, el error de *CP* (*validation*) comienza a aumentar, mientras que el de *CE* (*train*) continúa decreciendo, momento en el que se debe de parar para garantizar la generalización.



**Figura 4.8.** Comparativa de la evolución del error del modelo para los datos de entrenamiento (*train*), datos de validación (*validation*), datos de control (*test*) y localización del punto óptimo. Fuente Propia.

El error de predicción puede estimarse por métodos de remuestreo, en particular se van a discutir la validación cruzada y el remuestreo llamado autoarranque (*Bootstrap*). Son una alternativa a los métodos de Montecarlo para realizar inferencias estadísticas, ya que no necesitan la definición a priori del mecanismo mediante el cual los datos son generados.

El método de remuestreo propuesto en (Bianchini *et al.*, 1995) consiste en aproximar la función de distribución desconocida,  $F$ , de los datos observados mediante muestreo aleatorio con reemplazamiento obteniendo,  $\hat{F}$ . El procedimiento determina un conjunto de  $k$  muestras del mismo tamaño sobre las cuales se calcula el estadístico a inferir.

Ya se ha planteado la *VC* como método para juzgar la conveniencia de un modelo u otro, la comparación de modelos permite decidir qué entradas emplear, qué número de valores pasados de cada variable, qué familia y tamaño de modelos, qué secuencia de entrenamiento emplear evitando además la sobreparametrización. Como se vio en la Figura 4.8 en la curva azul, la evolución típica de una medida del desajuste de un modelo empleando los datos de estimación, en función del

número de parámetros del modelo, se observa un descenso monótono y pronunciado en la primera parte de la curva, consecuencia de la adición de parámetros útiles para mostrar la realidad, y una parte final consistente en descensos menores ya que vienen motivados por la aproximación del ruido por parte del modelo. Fijándonos ahora en el *CP* (curva verde), se observa que el error sube al aumentar  $n$  ( $x$ -eje), esto es debido a la sobreparametrización como se habló anteriormente.

La *VC* será de gran utilidad al emplear modelos no lineales, el único inconveniente es que precisa datos no usados para la estimación. En algunas ocasiones no es posible reservar datos para el *CP*, por lo que se recurre a técnicas que intenten reproducir los resultados de la *VC* pero empleando los propios datos del *CE*. La idea básica es obtener el modelo en el que la curva del error en *CE* tiene el codo o, de manera similar, el lugar en que la curva de error en *CP* deja de descender.

La infraparametrización produce un valor mayor del sesgo de la salida, que es la diferencia entre el valor medio de un estimador y el valor correcto, mientras que la sobreparametrización reduce el sesgo y aumenta la varianza.

Para evitar el sobreentrenamiento se pueden emplear gran cantidad de datos para dicha fase, aunque no siempre es posible. Dado un *CE*, existen varias maneras de intentar evitar el sobreentrenamiento: selección de modelos; adición de ruido a los ejemplos de entrenamiento; detección temprana del entrenamiento (ciclos de entrenamiento); regularización; aprendizaje bayesiano; y uso de predictores combinados. A continuación daremos algunos detalles de los ciclos de entrenamiento y del aprendizaje bayesiano, al considerarse de gran importancia.

### Ciclos de entrenamiento

Lo normal es que se emplee un conjunto de datos históricos para el entrenamiento y otro diferente para las pruebas posteriores. La búsqueda de parámetros del modelo se realiza empleando únicamente el *CE*, pero este esquema tan sencillo se ve complicado por varios factores:

- El algoritmo de entrenamiento necesita que le sea presentado varias veces el *CE*, y así ajustar los parámetros del modelo. En cada iteración con el *CE*, el error del modelo suele disminuir, aunque existe la posibilidad de que el algoritmo quede atrapado en un mínimo local y el error aumente.
- El valor inicial que se da a los parámetros antes de comenzar el método influye, por tanto es preciso repetir el proceso para evitar este efecto.
- No es conveniente que el modelo obtenga un error ínfimo a costa de los datos, podría existir la posibilidad de que éstos estuvieran contaminados, por lo que el esfuerzo no tendría sentido.

Algunos investigadores han estudiado cuándo es interesante detener el algoritmo de entrenamiento, como Nelson and Illingworth (1991), proponiendo el método de la detección temprana en *ANN*, el cual presenta algunas ventajas. Los pasos de dicho método son:

- $CH$  se divide en  $CE$  y  $CP$ .
- Emplear una red con un número de nodos suficiente y quizá alguno más.
- Los pesos de la red inicializarlos con valores pequeños.
- Las ganancias de la adaptación que sean pequeñas.
- Calcular el error en  $CP$  mientras se hacen las iteraciones sobre  $CE$ .
- Detener el entrenamiento cuando el error en  $CP$  comience a subir, independientemente de si el de  $CE$  continúa disminuyendo.

Este método tiene las ventajas de ser fácil de aplicar, rápido y además sólo requiere el  $CP$  para ser aplicado. Como inconveniente, el error en el  $CP$  no es un buen estimador del error de generalización y además la división de los patrones en  $CE$  y  $CP$  pueden afectar a los resultados, tal y como se presenta en (Sarle, 1995).

#### Aprendizaje bayesiano

La comunidad bayesiana trata de manera diferente el aprendizaje, empleando el uso de la probabilidad para representar incertidumbres acerca de la función que genera los ejemplos. Antes de emplear los datos, el conocimiento de los parámetros ajustables del modelo se puede expresar mediante una distribución probabilística sobre dichos parámetros, como consecuencia de observar los datos, es posible ajustar las distribuciones propuestas. Los valores de los parámetros que ayudan a representar los datos observados son ahora considerados más probables que otros. La consecuencia de este aprendizaje es una distribución de los parámetros, esta distribución produce a su vez una distribución de predicciones cuando se introducen nuevos datos en el modelo.

La selección de familias modelos y la selección de estructura de los mismos, pueden solucionarse mediante la incorporación del conocimiento a priori en la distribución de parámetros. Este conocimiento a priori es empleado habitualmente de manera jerárquica empleando hiperparámetros, como se presenta en (Neal, 1992). La puesta en práctica de estos métodos es muy complicada, hallar los hiperparámetros es una tarea extremadamente más difícil que la mera estimación de parámetros. No existen métodos analíticos para seleccionar los hiperparámetros en la mayoría de los casos, en su lugar se intenta hallar hiperparámetros probables empleando métodos similares al entrenamiento con regularizadores y utilizando la información resultante para aproximar la distribución de los parámetros.

#### **4.3.7.3. Estimación del error de generalización**

El error de generalización depende de muchos factores y es complicada su estimación. Las técnicas para obtener la estimación de dicho error se pueden agrupar en tres categorías, las cuales son:

- Métodos algebraicos: los criterios para modelos lineales como Criterio de Información Akaike (*Akaike Information Criterion – AIC*), Criterio Bayesiano Schwarz (*Schwarz Bayesian Criterion – SBC*), Longitud de Descripción Mínima (*Minimum Description Length – MDL*) y Error de Predicción Final (*Final Prediction Error – FPE*), son simples fórmulas que permiten estimar el error de generalización bajo determinadas condiciones. Estos criterios pueden también emplearse en modelos no lineales en el caso de que se disponga de un *CE* suficientemente grande y, como se presenta en (Moody, 1992), ser adaptados para adecuarse a los casos no lineales, a costa de mayor carga de cómputo y algún problema teórico si se emplea para *ANN*.
- *VC*: como es sabido, el error en *CP* es una estimación del error de generalización. *CP* ha de ser representativo del tipo de situaciones en las que se usará el predictor, el problema de este método es que sacrifica datos que no son usados para obtener los parámetros del modelo.
- *VC* múltiple: alternativa a la *VC* simple.
- Remuestreo: tiene ciertas ventajas frente a las dos anteriores a costa de mayor carga de cómputo.

Como norma general, se recomienda dejar aparte un tercer grupo de datos, segundo conjunto de pruebas, que proporcione una mejor estimación del error de predicción futuro, tal y como se plantea en (Hjorth, 1994).

#### **4.4. LA PREDICCIÓN CON MODELOS LINEALES**

Durante décadas los modelos lineales han constituido la herramienta principal tanto para el control como para la identificación, debido a que existía una teoría matemática sólida donde apoyarse y, su aplicación a sistemas reales llevaba a resultados aceptables.

Mediante ecuaciones diferenciales ordinarias se puede describir cualquier comportamiento lineal, y a estos modelos, se les puede aplicar el principio de superposición mediante diferentes técnicas, como pueden ser la función de transferencia o los diagramas en frecuencia. Estas dos técnicas fallan en el momento que nos alejamos del caso lineal.

A continuación se darán los principios, y los detalles fundamentales, de los modelos lineales empleados para la identificación de sistemas dinámicos, prestando atención a los modelos lineales paramétricos y la técnica de los mínimos cuadrados.

##### **4.4.1. Modelos lineales paramétricos**

Prescindiendo de los ruidos y las perturbaciones que todo sistema real tiene (para simplificar la expresión), el modelo paramétrico para sistemas lineales de una variable vendrá dado por

$$y(k) + a_1 y(k-1) + \dots + a_n y(k-n) = b_1 u(k-1) + \dots + b_n u(k-n) \quad (4.32)$$

Seleccionando los parámetros correctos de (4.32), ésta podrá representar la totalidad de los sistemas dinámicos que cumplan con la siguiente función de transferencia

$$G(z^{-1}) = \frac{B(z^{-1})}{A(z^{-1})} = \frac{b_1 z^{-1} + \dots + b_n z^{-n}}{1 + a_1 z^{-1} + \dots + a_n z^{-n}} \quad (4.33)$$

En la Figura 4.9 se muestra la estructura básica de los modelos lineales paramétricos, donde la salida del sistema se ve afectada tanto por los valores de entrada  $u$  por medio de una función de transferencia similar a (4.33), como por una señal de ruido blanco  $v$  con estructura similar a (4.32).

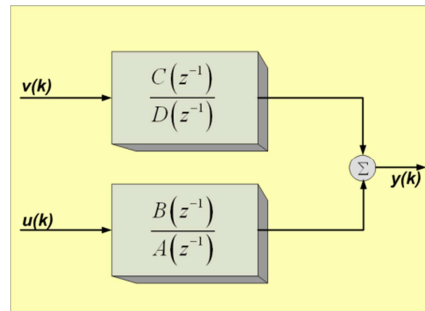


Figura 4.9. Diagrama de bloques de un modelo lineal paramétrico. Fuente Propia con Datos de Arahal *et al.* (2002).

En la Tabla 4.2 se resumen las principales familias de los modelos lineales paramétricos, indicando su acrónimo, descripción de la familia y detalle de la característica con respecto a los parámetros mostrados en la Figura 4.9.

Tabla 4.2. Familias de modelos lineales paramétricos. Fuente Propia.

<i>Acrónimo</i>	<i>Características</i>	<i>Descripción</i>
FIR	$C = D = A = 1$	<i>Respuesta de Impulso Finita (Finite Impulse Response)</i>
MA	$B = 0, D = 1$	<i>Media Móvil (Moving Average)</i>
AR	$B = 0, C = 1, D \neq 0$	<i>Auto-Regresivo (Auto-Regressive)</i>
ARMA	$B = 0, C \neq 0, D \neq 0$	<i>Auto-Regresivo y Media Móvil (Auto-Regressive and Moving Average)</i>
ARMAX	$B \neq 0, C \neq 0, D = A \neq 0$	<i>Auto-Regresivo y Media Móvil con Entradas Exógenas (Auto-Regressive and Moving Average with eXogenous inputs)</i>
ARX-LS	$B \neq 0, C = 1, D = A \neq 0$	<i>Auto-Regresivo con Entradas Exógenas – Mínimos Cuadrados (Auto-Regressive with eXogenous inputs – Least Squares)</i>
Box-Jenkins	$A, B, C, D \neq 0$	<i>Box-Jenkins</i>

Respuesta de Impulso Finita (*Finite Impulse Response – FIR*) es un tipo de modelo cuya respuesta a una señal de entrada tendrá un número finito de términos nulos, y cuya representación matemática puede ser

$$y_n = \sum_{k=0}^{N-1} b_k x(n-k) \quad (4.34)$$

Media Móvil (*Moving Average – MA*) es un modelo conocido como de la media móvil, debido a que su salida se obtiene como media ponderada de ciertos valores pasados de la señal  $v$ , siendo su expresión la siguiente

$$y(k) = c_0 v(k) + c_1 v(k-1) + \dots + c_n v(k-n) \quad (4.35)$$

En Auto-Regresivo (*Auto-Regressive – AR*) la variable de salida es función de su historia pasada, por este motivo el término autorregresivo, pudiéndose expresa de la siguiente manera

$$y(k) = a_1 y(k-1) + a_2 y(k-2) + \dots + a_n y(k-n) + v(k) \quad (4.36)$$

Auto-Regresivo y Media Móvil (*Auto-Regressive and Moving Average – ARMA*) es autorregresivo de orden  $n_a$  y de media móvil de orden  $n_e$ , por lo que también es conocido como modelo *ARMA* ( $n_a, n_e$ ). La metodología de *ARMA* fue desarrollada por Box and Jenkins (1970), por lo que a veces son llamados como modelos *Box-Jenkins*. Su expresión es

$$y(k) = a_1 y(k-1) + \dots + a_{n_a} y(k-n_a) + v(k) + c_1 v(k-1) + \dots + c_{n_e} v(k-n_e) \quad (4.37)$$

Auto-Regresivo y Media Móvil con Entradas Exógenas (*Auto-Regressive and Moving Averagewith eXogenous inputs – ARMAX*). Es el caso más general de los modelos lineales y su expresión es

$$y(k) = a_1 y(k-1) + \dots + a_n y(k-n) + b_1 u(k-1) + \dots + b_n u(k-n) + v(k) + c_1 v(k-1) + \dots + c_n v(k-n) \quad (4.38)$$

Auto-Regresivo con Entradas Exógenas – *Mínimos Cuadrados* (*Auto-Regressive with eXogenous inputs – Least Squares – ARX-LS*) es un caso particular de *ARMAX*, y asume una estructura muy simple para la fuente de perturbación, lo cual permitirá analizar las propiedades de la estimación. Su expresión es

$$y(k) = a_1 y(k-1) + \dots + a_n y(k-n) + b_1 u(k-1) + \dots + b_n u(k-n) + v(k) \quad (4.39)$$

Tanto los modelos *ARX-LS* como *ARMAX* provocan un problema de identificación paramétrica lineal. En el caso más general, como el de los modelos de *Box-Jenkins* donde  $D(z^{-1}) = A(z^{-1})$ , ya que la descripción del sistema viene dada por

$$y(k) = B(z^{-1})D(z^{-1})u(k) + A(z^{-1})C(z^{-1})v(k) \quad (4.40)$$

en los coeficientes a estimar aparecen términos no lineales, por lo que la identificación se debe realizar empleando técnicas de optimización no lineal, por ejemplo los mínimos cuadrados no lineales.

#### 4.4.2. La técnica de los mínimos cuadrados

Para presentar el método, se debe partir del modelo  $ARX-LS (n, n)$  dado por la ecuación (4.31), para a continuación, transformarla en

$$y(k) = -a_1 y(k-1) + \dots - a_n y(k-n) + b_1 u(k-1) + \dots + b_n u(k-n) \quad (4.41)$$

podemos escribirla en forma vectorial de la manera  $y(k) = \mathbf{m}(k)\boldsymbol{\theta}$ , simplemente definiendo el vector de medias de la siguiente forma

$$\mathbf{m}(k) = (-y(k-1), \dots, -y(k-n), u(k-1), \dots, u(k-n)) \quad (4.42)$$

y el vector de parámetros  $\boldsymbol{\theta} = (a_1, \dots, a_n, b_1, \dots, b_n)^T$ . Dado  $\hat{\boldsymbol{\theta}} \neq \boldsymbol{\theta}$  se podrá construir el modelo  $ARX-LS (n, n)$ , produciendo una predicción  $\hat{y} = \mathbf{m}(k)\hat{\boldsymbol{\theta}}$ , donde el error de la predicción depende únicamente de  $\hat{\boldsymbol{\theta}} \neq \boldsymbol{\theta}$  y vendrá dado por

$$e(k) = y(k) - \hat{y}(k) = y(k) - \mathbf{m}(k)\hat{\boldsymbol{\theta}} \quad (4.43)$$

Agrupando en vectores en (4.43) y desarrollándola desde  $k = n$  hasta  $k = N$  se tiene la siguiente expresión del error

$$\mathbf{E}(N) = \mathbf{Y}(N) - \mathbf{M}(N)\hat{\boldsymbol{\theta}} \quad (4.44)$$

donde  $\mathbf{E} = (e(n), \dots, e(N))^T$ ,  $\mathbf{Y} = (y(n), \dots, y(N))^T$  y  $\mathbf{M} = \begin{pmatrix} m(n) \\ \cdot \\ \cdot \\ m(N) \end{pmatrix}$ .

Por tanto, el objetivo del método será hallar el valor  $\hat{\boldsymbol{\theta}}$  que minimice el siguiente índice

$$\mathbf{J}(\hat{\boldsymbol{\theta}}) = \|\mathbf{E}(N)\|^2 = \sum_{k=n}^N e^2(k) = \sum (y(k) - \mathbf{m}(k)\hat{\boldsymbol{\theta}})^2 \quad (4.45)$$

el valor que minimiza el índice es  $\hat{\boldsymbol{\theta}}^* = (\mathbf{M}^T(N)\mathbf{M}(N))^{-1}\mathbf{M}^T(N)\mathbf{Y}(N)$ . Lo anterior indica que la matriz  $\mathbf{M}(N)$  debe de ser invertible, por lo que no puede haber combinación lineal entre sus filas.

La inversión de una matriz lleva asociado un tiempo de cálculo elevado, por lo que se dificulta su aplicación en tiempo real para la identificación en línea, sin embargo, se puede obtener la estimación que minimiza el error cuadrático ( $\hat{\boldsymbol{\theta}}^*$ ) de manera recurrente, de manera que existe un

algoritmo recurrente que lo calcula como la suma del valor anterior más un corrector que depende linealmente del error cometido en la predicción de la salida. Empleando medidas desde el instante  $n$  hasta el  $k$ , la estimación de mínimos cuadrados puede escribirse de la siguiente manera

$$\hat{\theta}(k) = \mathbf{P}(k) (\mathbf{M}^T(k-1) \mathbf{Y}(k-1) + \mathbf{m}^T(k) y(k)) \quad (4.46)$$

habiendo definido la matriz  $\mathbf{P}(k) = (\mathbf{M}^T(k) \mathbf{M}(k))^{-1} = (\sum_{i=n}^k \mathbf{m}^T(i) \mathbf{m}(i))^{-1}$ . Es sencillo

llegar a las dos igualdades siguientes

$$\mathbf{P}^{-1}(k-1) = \mathbf{P}^{-1}(k) - \mathbf{m}^T(k) \mathbf{m}(k) \quad (4.47)$$

$$\mathbf{M}^T(k-1) \mathbf{Y}(k-1) = \mathbf{P}^{-1}(k) \hat{\theta}(k-1) - \mathbf{m}^T(k) \mathbf{m}(k) \hat{\theta}(k-1) \quad (4.48)$$

y tras combinarlas se obtiene finalmente

$$\hat{\theta}(k) = \hat{\theta}(k-1) + \mathbf{K}(k) (y(k) - \mathbf{m}(k) \hat{\theta}(k-1)) \quad (4.49)$$

Con (4.49) se calcula el valor del vector de parámetros, sumando al valor anterior el error de la predicción multiplicado por la ganancia  $\mathbf{K}(k) = \mathbf{P}(k) \mathbf{m}^T(k)$ . En (4.47), si se aplica el lema de inversión de matrices se puede obtener una expresión recurrente para la matriz  $\mathbf{P}$ , pero sin necesidad de inversión, resultando las siguientes ecuaciones

$$\begin{aligned} \mathbf{K}(k) &= (\mathbf{P}(k) \mathbf{m}^T(k+1)) / (1 + \mathbf{m}(k+1) \mathbf{P}(k) \mathbf{m}^T(k+1)) \\ \hat{\theta}(k+1) &= \hat{\theta}(k) + \mathbf{K}(k) (y(k+1) - \mathbf{m}(k+1) \hat{\theta}(k)) \\ \mathbf{P}(k+1) &= (\mathbf{I} - \mathbf{K}(k) \mathbf{m}(k+1)) \mathbf{P}(k) \end{aligned} \quad (4.50)$$

Con la recurrencia evitamos la necesidad de inversiones en la matriz, por lo que podrán ser empleadas en sistemas en tiempo real para la identificación en línea, cuyo esquema de diagrama de bloques puede verse en la Figura 4.10.

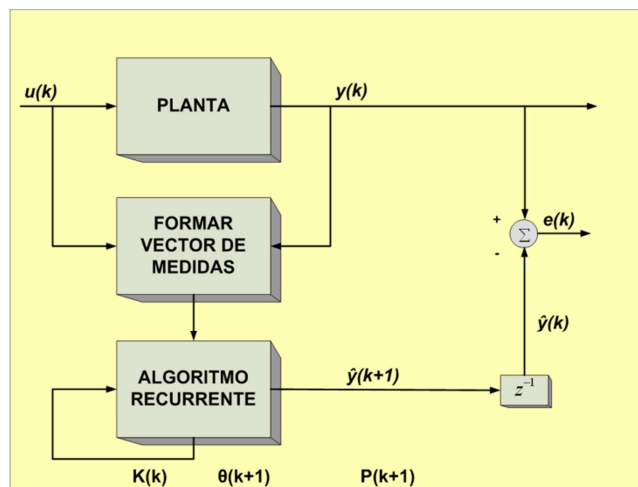


Figura 4.10. Esquema de diagrama de bloques de la identificación mediante el método de mínimos cuadrados recurrente. Fuente Propia con Datos de Arahal *et al.* (2002).



## 4.5. LA PREDICCIÓN CON MODELOS NO LINEALES

Cuando el sistema a analizar no sigue un comportamiento lineal, aplicar modelos lineales conllevará el obtener una respuesta del mismo no óptima. Los modelos no lineales no están definidos matemáticamente por ecuaciones lineales, por tanto, su resolución suele ser más complicada y tediosa, además, la selección de las mismas no suele ser una tarea trivial.

Sin embargo, los avances tecnológicos han generado una variedad de nuevos problemas y, sus necesarias aplicaciones, que son no lineales en esencia y, por tanto, requieren de una solución acorde con el problema. Sirva como ejemplo en aplicaciones de ingeniería actual, los sistemas de vuelo, robot para manipulación, sistemas de autopistas automatizadas, sistemas de inyección de combustible de alto rendimiento y predicciones de entornos multidependientes de variables (caso de la demanda eléctrica). Estos fenómenos no lineales no se pueden describir mediante modelos lineales, por lo que el empleo de modelos no lineales y, el desarrollo de conceptos y herramientas de sistemas no lineales de control, se hará totalmente necesario.

A continuación, se expondrá los modelos con no linealidades concretadas, para posteriormente, dar una explicación exhaustiva de las ANN, en las que se basarán las aportaciones que se mostrarán en el Capítulo VI. Lo interesante por tanto de trabajar con ANN, será el hacer corresponder a vectores regresores la predicción de la ANN, es decir, en lugar de obtener la predicción como combinación lineal de los componentes del vector regresor, se empleará una ANN para realizar una función no lineal de dichas componentes.

### 4.5.1. Los modelos con no linealidades concentradas

Si los modelos lineales no funcionan para el propósito perseguido, el siguiente paso en complejidad estará constituido por modelos con no linealidades concentradas, que son uniones de bloques lineales dinámicos y bloques no lineales estáticos.

Estos modelos, en el caso monovariable en tiempo continuo, se pueden representar por la siguiente expresión

$$y(t) = y_0(t) + \sum_{n=1}^{\infty} \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} k_n(t; t_1, t_2, \dots, t_n) u(t_1) \dots u(t_n) dt_1 \dots dt_n \quad (4.51)$$

donde  $k_n(t; t_1, t_2, \dots, t_n)$  son los núcleos de Volterra y son generalización de la respuesta por impulsos de los sistemas lineales. Los modelos lineales serán un caso particular del modelo de Volterra, tomando todos los núcleos nulos excepto el de primer orden.

En el caso discreto, las integrales son sustituidas por sumas y se tiene la siguiente expresión para los modelos de Volterra

$$y(t) = y_0(t) + \sum_{j=0}^{\infty} a_j u(k-j) + \dots + \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} \sum_{l=0}^{\infty} c_{ijl} u(k-i) u(k-j) u(k-l) + \dots \quad (4.52)$$

Los modelos, en la práctica, se realizan sobre un número finito de sumandos, por lo que el modelo tendrá también un orden finito, obteniéndose los modelos finitos de Volterra. Los modelos finitos de Volterra pueden aproximar cierto tipo de sistemas dinámicos llamados de memoria desvaneciente, estos sistemas son en los que el pasado lejano tiene poca influencia en el presente, siendo una particularidad que no pueden presentar múltiples estados de equilibrio, por tanto, los modelos finitos de Volterra serán adecuados para representar sistemas con estas características.

A veces, los sistemas se pueden describir de manera correcta con un modelo de dos submodelos, el primero una no linealidad estática y, el segundo, un sistema dinámico lineal. Los dos subsistemas conectados en serie forman los modelos de Hammerstein y de Wiener, que se pueden ver en la Figura 4.11.

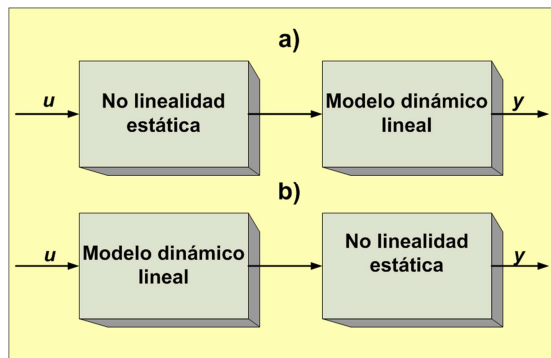


Figura 4.11. a) Diagrama del modelo de Hammerstein; b) Diagrama de bloques de Wiener. Fuente Propia con Datos de Arahal *et al.* (2002).

Para Hammerstein, la función del bloque no lineal se aproxima mediante un polinomio  $f(u(k)) = c_1 u(k) + c_2 u(k)^2 + \dots + c_n u(k)^n$ , mientras que la salida del bloque no lineal puede ser única o múltiple, en este último caso se puede emplear un polinomio  $B$  distinto para cada una de las potencias, resultando el modelo de la manera siguiente

$$A(z^{-1})y(k) = B_1(z^{-1})u(k) + B_2(z^{-1})u(k)^2 + \dots + B_n(z^{-1})u(k)^n \quad (4.53)$$

Cuando no se tiene conocimiento de la no linealidad, los modelos de Hammerstein y Wiener presentan problemas en la identificación. Es posible seleccionar diferentes funciones para el mismo, por lo que es preciso afrontar el problema de determinar la más adecuada, existiendo algunos procedimientos, como los mostrados por Westwick and Verhaegen (1996):

- Determinar en primera instancia la parte no lineal y luego la dinámica lineal. Se realizan experimentos observando la forma de la parte no lineal empleando datos de régimen permanente.

- Identificar primero la parte lineal empleando técnicas de correlación para generar la salida de dicho bloque, con posterioridad, obtener la no linealidad empleando datos de régimen permanente.
- Estimación conjunta de parámetros de la parte lineal y no lineal, según se realiza en (Wigren, 1993; Kalafatis *et al.*, 1995).

Un caso particular de lo presentado viene dado por la conexión de un sistema dinámico lineal y un bloque no lineal estático, que puede ser representado mediante trozos lineales. Esta técnica se empleó en el área de teoría de circuitos no lineales, y en la década de 1970 fueron empleados por Chua and Kang (1977), resolviendo el caso de funciones en  $R^2$ , y extendiéndose en los últimos tiempos a un caso general en  $R^n$ .

#### 4.5.2. Redes Neuronales: historia

La consecución de máquinas o dispositivos capaces de realizar tareas con cierta “*inteligencia*” ha sido una de las principales preocupaciones de la ciencia a lo largo de la historia. En un principio las investigaciones estuvieron orientadas a la obtención de autómatas, máquinas que realizan alguna labor con alguna función típica del ser humano, pero no era más que el resultado del desarrollo técnico de la habilidad mecánica de los constructores de tales dispositivos. Con la evolución de los tiempos, la habilidad mecánica ha pasado a convertirse en disponibilidad microinformática (lenguajes de programación, sistemas expertos, etc.), de tal manera que existen diferentes maneras de conseguir procesos similares a los inteligentes, pudiéndolos encuadrar dentro de la denominada Inteligencia Artificial (*Artificial Intelligence – AI*).

A pesar de disponer de potentes máquinas e cálculo y lenguajes de programación para el desarrollo de dispositivos inteligentes, la limitación principal reside en que dichos dispositivos se implementan sobre ordenadores con filosofía de funcionamiento de von Neumann, apoyándose en el tratamiento secuencial de la información. A pesar de la enorme potencia de cálculo de estos ordenadores, no dejará de ser una máquina mecánica–electrónica capaz de realizar tareas (cálculo, ordenación, búsqueda, etc.) de manera asombrosamente rápida, pero no siendo capaz de obtener resultados aceptables, para tareas sencillas, como por ejemplo las que hace un ser humano de corta edad al hablar, reconocer formas, etc.

Con los aportes de Platón (427–347 a.C.) y Aristóteles (384–322 a.C.), se comenzaron a dar las primeras ideas teóricas sobre el cerebro y pensamiento humano. Muy posteriormente, Descartes (1596–1650) y los filósofos empiristas, ampliaron aquéllas ideas sobre el proceso mental.

Hablando de *máquinas cibernéticas*, Heron de Alejandría en el siglo I d.C, inventó una máquina de vapor, conocida como *eolípila*, y un ingenio hidráulico conocido como la *Fuente de*

*Herón*, siendo ambas ejemplos de autómatas. De manera más reciente, se han desarrollado modelos de animales para demostrar el comportamiento necesidad–adaptación ante diferentes condiciones de vida, véase como ejemplo las versiones del ratón en el laberinto de Nemes (1969).

En 1936, Allan Turing fue el primero en tratar el cerebro humano como posible aplicación al mundo de la computación, además de su famosa *Máquina de Turing*, también trató el problema de la *AI*. No fue hasta 1943 con Warren McCulloch (neurofisiólogo) y Walter Pitts (matemático), cuando aparece la teoría acerca de la forma de trabajar las neuronas y su posible utilidad en entornos computacionales, mostrado en (McCulloch and Pitts, 1943). Donald Hebb, escribió un importante libro en 1949 sobre teoría de redes neuronales, siguiendo y apoyando lo comenzado por McCulloch y Pitts, en su obra (Hebb, 1949).

El aporte fundamental, debido a su aplicación desde entonces hasta nuestros días, fue en 1958, cuando Frank Rosenblatt desarrolló el modelo de red neuronal *Perceptron* en (Rosenblatt, 1958), aunque presentaba la limitación de la función *OR–exclusiva*, y que en general, no era capaz de clasificar clases no separables linealmente. Dos años más tarde, en 1960, desde Stanford, Bernard Widrow y Marcial Hoff, desarrollaron el modelo Elementos Lineales Adaptativos (*ADaptative LINear Elements – ADALINE*), que fue la primera red neuronal aplicada a problemas reales, en concreto, para eliminar ecos en líneas telefónicas, mostrado en (Widrow and Hoff, 1960).

Sin lugar a dudas, uno de los mayores investigadores de las redes neuronales desde 1960 hasta nuestros días es Stephen Grossberg. Además de escribir multitud de libros, en 1967 realizó una red llamada *Avalancha*, formada por elementos discretos con actividad variable en el tiempo, para resolver actividades como aprendizaje de movimiento de robots o reconocimiento del habla, y presentada en (Grossberg, 1982).

Desde 1969, hasta precisamente 1982, surgieron numerosas críticas que frenaron el crecimiento de las redes neuronales. De la mano de Marvin Minsky y Seymour Papert, del *Massachusetts Institute Tchnology (MIT)*, publicaron el libro *Perceptrons* (Minsky and Papert, 1969), donde consideraban que el desarrollo de un *Perceptron* multinivel tenía considerables limitaciones, se debe recordar que el *Perceptron* original tenía sólo una capa.

A pesar de la anterior campaña de desprestigio, determinados investigadores continuaron trabajando con redes neuronales, tal es el caso de James Anderson, quien presentó un modelo basado en el principio de que las conexiones entre neuronas se refuerzan cada vez que se ven activadas, llamó al modelo Cerebro–Estado–en–una–Caja (*Brain–State–in–a–Box*), descrito en (Anderson *et al.*, 1977). En ese mismo año, pero en la Universidad de Helsinki, Teuvu Kohonen presentó la evolución del modelo de Anderson, obteniendo un modelo de memoria asociativa, expuesto en (Kohonen, 1977), y siendo uno de sus primeros trabajos del gran número de ellos que

ha presentado hasta nuestros días. Tres años después y en Japón, Kunihiko Fukushima presentó un modelo de red para el reconocimiento visual de patrones, en su obra (Fukushima, 1980).

En 1982 volvió a resurgir el interés por las redes neuronales, por diferentes motivos, uno de ellos es el trabajo presentado por Hopfield (1982), donde describe claramente y de manera matemática, una red que precisamente lleva su nombre, además, mostró como dichas redes pueden trabajar. Coincidiendo con ese año, Fujitsu comenzó con el desarrollo para aplicaciones robóticas de sus *computadores pensantes*. Además, en 1985 comenzó la reunión anual de Redes Neuronales para Computación (*Neural Networks for Computing*) y, en 1987, se formó la Sociedad Internacional de Redes Neuronales (*International Neural Network Society*), iniciativa de Grossberg, Kohonen y Amari. Es a partir de ese instante, cuando el interés por esta área de la investigación se ha incrementado notablemente, avalado por el interés del mundo empresarial en el uso de esta tecnología en la solución a problemas variados. Nacieron otras iniciativas de cooperación, como en 1989 y, en Washington, la Conferencia Internacional Conjunta sobre Redes Neuronales (*International Joint Conference on Neural Networks*), en 1991 su alternativa europea, la Conferencia Internacional sobre Redes Neuronales Artificiales (*International Conference on Artificial Neural Networks*), o la reunión anual celebrada en Denver (US) desde 1987 Sistemas de Procesamiento de Información Neuronal (*Neural Information Processing Systems*).

En definitiva, y para finalizar, hoy en día existen dos corrientes importantes dentro de la “*inteligencia*” que son complementarias, más que alternativas excluyentes. Por una parte está la *AI convencional*, que se centra en algoritmos que emplean información simbólica, sobre ordenadores von Neumann, que operan sobre la lógica digital; y por otro lado, se suelen agrupar las ANN, fuzzy logic y otras técnicas, que se incluyen en la denominada *Computational Intelligence* (CI – inteligencia computacional) presentada en (Marks II, 1993), o *Soft Computing* en (Zadeh, 1994; Jang *et al.*, 1997), que tratan en cierta medida de imitar las construcciones y comportamientos de la naturaleza.

### **4.5.3. Redes Neuronales: fundamentos**

Antes de pasar a detallar los fundamentos de las ANN, es conveniente tratar de definir las mismas. A continuación, algunos ejemplos:

- “Una nueva forma de computación inspirada en modelos biológicos” de Hilera and Martínez (1995).
- “Un modelo matemático compuesto por un gran número de elementos procesales organizados en niveles” de Hilera and Martínez (1995).

- “(...) un sistema de computación hecho por un gran número de elementos simples, elementos de proceso muy interconectados, los cuales procesan información por medio de su estado dinámico como respuesta a entradas externas” de Hecht–Nielsen (1988).
- “Redes neuronales artificiales son redes interconectadas masivamente en paralelo de elementos simples (usualmente adaptativos) y con organización jerárquica, las cuales intentan interactuar con los objetos del mundo real del mismo modo que lo hace el sistema nervioso biológico” de Kohonen (1988a).

Debido a una gran cantidad de lazos de realimentación junto a no linealidades de elementos de proceso y cambios adaptativos de los parámetros, es posible la compleja operación de las ANN. De esta forma, se pueden definir incluso fenómenos dinámicos extremadamente complicados.

Las ANN, son capaces de aprender de la experiencia, generalizar de casos anteriores a nuevos casos, abstraer características esenciales a partir de entradas que representan información relevante, etc.; esto hace que presenten ciertas ventajas, como indican Maren *et al.* (1990), y su interés en aplicaciones aumente con el tiempo:

- *Aprendizaje adaptativo*: capacidad de aprender a realizar tareas basadas en un entrenamiento o experiencia inicial.
- *Auto-organización*: puede crear su propia organización de información que recibe mediante una etapa de aprendizaje.
- *Tolerancia a fallos*: la destrucción parcial de una red conduce a una degradación de la estructura, sin embargo, algunas capacidades de la red se pueden retener.
- *Operación en tiempo real*: los computadores neuronales permiten ser realizados en paralelo.
- *Fácil inserción dentro de la tecnología existente*: se pueden obtener chips especializados para ANN que mejoran su capacidad en ciertas tareas, posibilitando la integración modular con sistemas ya existentes.

Anticipándonos a lo que se explicará en el modelo biológico, en las redes neuronales biológicas, las células que componen el cerebro, llamadas neuronas, se encuentran interconectadas por medio de unas ramas de salida llamadas axones, produciendo un número variable de conexiones con otras neuronas llamadas sinapsis. Tras esta breve descripción, y para entender el potencial real de la computación neuronal, será preciso hacer ciertas aclaraciones, como las propuestas por Hilera and Martínez (1995):

- “Los sistemas neurológicos biológicos no aplican principios de circuitos lógicos o digitales”. Un sistema computacional digital deberá ser síncrono o asíncrono. Si fuera asíncrono, los impulsos neuronales deberían ser variables para mantener los valores binarios indefinidos, lo cual no es el caso. Si fuera síncrono, sería preciso un reloj global para sincronizar pulsos, no siendo el caso tampoco. Las neuronas no pueden ser circuitos de

umbral lógico, ya que hay miles de entradas en las neuronas y el umbral es variable con el tiempo. Por esto y algunos motivos más, el cerebro debe ser un computador analógico.

- “Ni las neuronas ni la sinapsis son elementos de memoria biestables”. Los hechos fisiológicos posicionan a las neuronas como integradores analógicos, y la eficiencia de la sinapsis cambia gradualmente.
- “En la computación neuronal no hay instrucciones de máquina ni códigos de control”. Por lo ya comentado, el formato de los códigos no podrá ser mantenido durante períodos largos de tiempo, mucho menos durante el período de crecimiento.
- “Los circuitos del cerebro no implementan computación recursiva, y por tanto no son algorítmicos”. Debido a cierta inestabilidad de los circuitos neuronales, las funciones recursivas no podrán definirse.
- “Incluso en el nivel más alto, la naturaleza del proceso de información es diferente en el cerebro y en los computadores digitales”. Tratar de emular una con otra requerirá acceder a los estados internos de dos sistemas de computación. Esta equivalencia no es posible entre cerebro y sistemas de programación, los computadores artificiales no pueden adquirir ni interpretar las experiencias humanas en las cuales se basa el asentamiento de los valores.

Por tanto, y para finalizar, el cerebro debemos entenderlo como un complejo sistema de procesamiento, no lineal, masivamente paralelo, pero en cambio, maravillosamente eficiente desde la perspectiva energética. En la Tabla 4.3 se presentan las principales características del cerebro frente a los computadores, según Martín del Brío and Sanz (2006).

**Tabla 4.3. Características del cerebro frente al computador convencional. Fuente Propia con Datos de Martín del Brío and Sanz (2006).**

	<i>Cerebro</i>	<i>Computador</i>
<i>Estilo del procesamiento</i>	<i>Paralelo</i>	<i>Secuencial</i>
<i>Número de procesadores</i>	$10^{11}$ a $10^{14}$	<i>Pocos</i>
<i>Velocidad de proceso</i>	$10^{-2}$ segundos	$10^{-9}$ segundos
<i>Número de conexiones</i>	<i>10.000 por procesador</i>	<i>Pocas</i>
<i>Almacenamiento del conocimiento</i>	<i>Distribuido</i>	<i>Direcciones fijas</i>
<i>Tolerancia a fallos</i>	<i>Grande</i>	<i>Escasa</i>
<i>Tipo de control</i>	<i>Auto-Organizado</i>	<i>Centralizado</i>

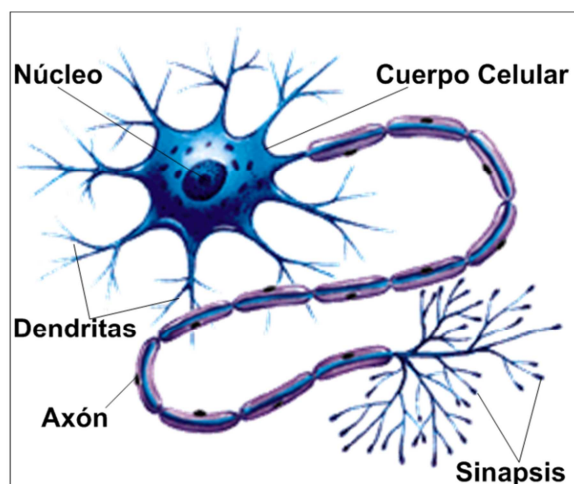
#### 4.5.3.1. Modelo biológico

Como ya se ha comentado, la teoría de las ANN está basada en la estructura y funcionamiento de los sistemas nerviosos de los seres vivos, siendo la neurona elemento fundamental.

Gracias al científico aragonés Santiago Ramón y Cajal, y su premio Nobel en 1906, se descubrieron los mecanismos que gobiernan la morfología y los procesos conectivos de las células nerviosas, comprobando como el tejido cerebral estaba compuesto por células individuales, pero a la vez interconectadas. En su obra “*Textura*” (Ramón y Cajal, 1899–1894) realiza un magistral estudio y descripción del sistema nervioso de los vertebrados, obra que después de un siglo continúa publicándose en numerosos idiomas. Posteriormente, y gracias al descubrimiento de la microscopía y su evolución, así como a la introducción de otras importantes técnicas, se ha profundizado mucho más en el estudio neuronal de Kandel *et al.* (1999).

El sistema nervioso puede contener alrededor de cien mil millones de neuronas, que vistas en el microscopio pueden presentarse de diferentes tamaños, longitudes y formas, características que determinarán la función y utilidad de cada una de ellas. Desde la perspectiva funcional, las neuronas constituyen procesadores de información sencillos. El córtex cerebral se organiza en forma de capas horizontales, coexistiendo una organización vertical en forma de columnas de neuronas. Existen grupos de neuronas (millones) pertenecientes a una región concreta del cerebro, que constituyen unidades funcionales especializadas en determinadas tareas (visión, audición, etc.) y, donde todos estos subsistemas forman el encéfalo. Se ha demostrado que el procesamiento en el sistema nervioso supone la actuación de muchos de estos subsistemas, intercambiando información constantemente.

En la Figura 4.12 se muestra una representación de una neurona biológica, donde sus elementos principales son el cuerpo celular, del que sale una rama principal llamada axón, y varias ramas más cortas llamadas dendritas cuya principal función es recibir los impulsos de otras neuronas y enviarlas hasta el soma de la neurona. Las neuronas se comunican con sus semejantes, las dendritas y el cuerpo celular reciben señales de entrada, el cuerpo celular las procesa y emite señales de salida. Estas señales son transportadas por el axón hasta los terminales axónicos y, a su vez, distribuir información hasta un conjunto nuevo de neuronas.



**Figura 4.12.** Esquema general de una neurona biológica. Fuente Puig (2009).



Las señales referidas anteriormente son de naturaleza eléctrica y química, la señal generada por la neurona y transportada por el axón es un impulso eléctrico, mientras que la que se transmite entre los terminales axónicos y las dendritas de otras neuronas son químicos, concretándose por medio de moléculas transmisoras, los neurotransmisores, a través de unos contactos especiales llamados sinapsis (Stevens, 1979).

La generación de señales eléctricas está ligada a la propia composición de la membrana celular, el proceso de generación de la señal se puede simplificar de la siguiente manera, en el interior de la neurona existe un líquido de composición diferente al líquido que la rodea, entre otras cosas, se produce una diferencia de concentraciones de iones sodio y potasio, el medio exterior es unas diez veces más rico en sodio que el interno, mientras que el interno es superior en la misma cantidad pero en potasio, esto hace que se produzca a ambos lados de la membrana una diferencia de potencial de setenta milivoltios, con la parte negativa en el interior de la membrana. Este potencial se conoce como potencial de reposo de la neurona.

El potencial de reposo baja ligeramente al acumularse las señales llegadas de otras neuronas a través de las dendritas, de esta manera, se modifica la permeabilidad de la membrana, y llegado a un punto crítico se produzca una entrada masiva de iones sodio invirtiendo la polaridad de la membrana. Esta inversión de voltaje, cierra el paso de iones sodio y abre el de iones potasio hasta restablecer el equilibrio, esta inversión de voltaje se conoce como potencial de acción, que se propaga a través del axón, provocando la emisión de los neurotransmisores.

La sinapsis puede ser de dos tipos:

- Excitadora: los neurotransmisores provocan disminuciones de potencial en la membrana, permitiendo la transmisión de impulsos a mayor velocidad.
- Inhibidoras: los neurotransmisores tienden a estabilizar el potencial en la membrana, dificultando la emisión de impulsos.

En cada instante, cada neurona recibe los estímulos de las sinapsis excitadoras e inhibidoras, estando algunas de ellas activadas y otras no. La decisión de si se emitirá o no un tren de impulsos, y su velocidad, vendrá determinada por la suma de todos los aportes sinápticos de la neurona.

Para finalizar, trataremos de establecer una similitud entre la actividad sináptica y las ANN. Las señales llegadas a la sinapsis serán las entradas a la neurona, siendo atenuadas por un parámetro llamado *peso*, asociado con la sinapsis correspondiente. Estas entradas pueden excitar a la neurona (peso positivo) o inhibirla (peso negativo), y será el efecto de todas las entradas las que, si sobrepasan el umbral de la neurona, las que harán que la neurona se active (dará salida).

Las sinapsis son susceptibles a muchos elementos (oxígeno, fatiga, etc.), e incluso a otros agentes (ejemplo de los anestésicos), e incluso puede ser modulada a escala temporal mucho más amplia que la del disparo de una neurona (minuto, hora o día). Esta habilidad será representada por

un mecanismo de *aprendizaje*, indicado en (Shepherd, 1997; Arbib, 1998), tal y como ya había postulado Hebb (1949), y posteriormente demostrado de manera experimental en (Nieto, 1989; Kandel and Hawkins, 1992). A lo largo de la vida del ser humano, el cerebro se modela, por lo que existen cualidades del mismo que no son innatas, el modelado se produce por el establecimiento de nuevas conexiones, ruptura de otras, modelado de las intensidades sinápticas o por muerte cerebral. Estas acciones, en especial la variación de la intensidad sináptica, serán las que empleen las ANN para su aprendizaje.

#### 4.5.3.2. Elementos de una neurona artificial

En esta sección y la siguiente, se presentarán los elementos y la estructura de la neurona, componente principal de una ANN. Lo propuesto, está basado en la estructura clásica de McCulloch–Pitts, mostrado en su obra (McCulloch and Pitts, 1943). A pesar de que algunos sistemas neuronales biológicos son lineales, como ocurre en la retina del cangrejo y presentado en (Brodie *et al.*, 1978), por lo general, la respuesta de las neuronas biológicas es de tipo no lineal, característica que es emulada en ANN, ya desde la neurona de McCulloch–Pitts.

A continuación describiremos la estructura genérica de una neurona artificial, según lo establecido por el grupo Procesamiento Distribuido Paralelo (*Parallel Distributed Processing – PDP*), presentado entre otros por McClelland y Rumelhart en las obras (McClelland and Rumelhart, 1986; Rumelhart and McClelland, 1986).

Comenzaremos definiendo *neurona* o *procesador elemental*, como un dispositivo simple de cálculo, que a partir de un vector de entrada procedente del exterior o de otras neuronas, da como resultado una salida o respuesta. El esquema general de una neurona artificial se muestra en la Figura 4.13a.

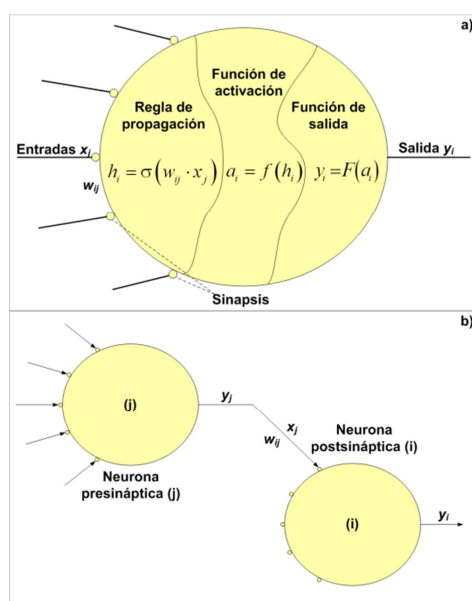


Figura 4.13. a) Esquema general de una neurona artificial (Rumelhart and McClelland, 1986); b) Esquema de interacción entre neurona presináptica y postsináptica. Fuente Propia.

Los elementos que forman la neurona  $i$  de la Figura 4.13a son los siguientes:

- *Entradas* a la neurona,  $x_j(t)$ . Como se aprecia, es dependiente del tiempo, y en un instante dado  $t+1$ , el vector regresor de entrada vendrá dado por  $\mathbf{m}(t+1) = (x_1(t+1), \dots, x_n(t+1))$ , entendiendo que la neurona tiene  $n$  entradas a través de  $n$  diferentes sinapsis.
- *Pesos sinápticos*  $w_{ij}$  de la neurona  $i$ . Representa la fuerza de la interacción entre la neurona presináptica  $j$  y la postsináptica  $i$ , tal y como se aprecia en la Figura 4.13b, donde los círculos representan los espacios sinápticos donde la neurona presináptica deja información a la postsináptica.
- El valor del potencial postsináptico  $h_i(t)$  de la neurona  $i$  en función de los pesos con que se ve afectada y sus entradas, es lo que se conoce como *regla de propagación*, y viene dado por  $\sigma(w_{ij}, x_j(t))$ .
- El estado de activación actual  $a_i(t)$  de la neurona  $i$ , función de su estado anterior  $a_i(t-1)$  y de su potencial postsináptico actual, se conoce como *función de activación* o *función de transferencia*, y se expresa por  $f_i(a_i(t-1), h_i(t))$ .
- La salida actual  $y_i(t)$  de la neurona  $i$  en función de su estado de activación  $a_i(t)$ , la proporciona la *función de salida*, y se expresa por  $F_i(a_i(t))$ .

Dicho de otro modo, la operación de la neurona  $i$  puede expresarse con la siguiente ecuación

$$y_i(t) = F_i\left(f_i\left[a_i(t-1), \sigma_i(w_{ij}, x_j(t))\right]\right) \quad (4.54)$$

### Entradas y salidas

Dependiendo del modelo de ANN y su aplicación, las variables pueden ser binarias (digitales) o continuas (analógicas), así por ejemplo, para tareas de clasificación las salidas podrían ser digitales  $\{0, +1\}$ , pero para un ajuste funcional de una aplicación multivariable continua, se emplearían salidas continuas en un cierto intervalo.

Las neuronas pueden tomar diferentes nombres, dependiendo el tipo de salida que tienen, tal y como recogen Müller and Reinhardt (1990). Si la salida toma valores 0 ó 1, se llaman *neuronas de tipo McCulloch–Pitts*. Si las salidas pueden ser -1 ó +1 se llaman *neuronas tipo Ising*. Si pueden tomar diversos valores discretos en su salida (-2, -1, 0, +1, +2) serán *neuronas tipo Potts*.

### Regla de propagación

La función más habitual para la regla de propagación es la de tipo lineal, y se basa en la suma ponderada de las entradas y pesos sinápticos de la neurona

$$h_i(t) \sum_j w_{ij} x_j \quad (4.55)$$

de manera formal puede expresarse como el producto escalar de los vectores de entrada y pesos, tal y como se muestra a continuación

$$h_i(t) \sum_j w_{ij} x_j = \mathbf{w}_i^T \cdot \mathbf{x} \quad (4.56)$$

Si el peso sináptico de una entrada es positivo tenderá a excitar a la neurona postsináptica, mientras que si el peso es negativo la inhibirá, por esto, hemos hablado anteriormente de sinapsis excitadora e inhibidora.

Otra regla diferente, de tipo no lineal es la siguiente

$$h_i(t) = \sum w_{i_1 j_1 \dots j_q} x_{j_1} x_{j_2} \dots x_{j_q} \quad (4.57)$$

donde se aprecia una interacción de tipo multiplicativo entre todas las entradas de la neuronas junto a sus pesos. Una neurona con esta regla se conoce como *neurona de orden superior* o *neurona sigma-pi*, expuesta en (Rumelhart and McClelland, 1986), *i* implica mayor complejidad tanto para el estudio de la ANN como para su posible implementación *Hardware*.

Una regla especial, normalmente empleada en modelos *RBF*, *SOM* o Cuantificación Vectorial de Aprendizaje (*Learning Vector Quantization – LVQ*) es la *distancia euclídea*

$$h_i^2(t) = \sum_j (x_j - w_{ij})^2 \quad (4.58)$$

cuando la distancia del vector de entradas y de pesos es pequeña, (4.58) aumenta, y al contrario cuando la distancia es grande. Por tanto, esta regla opera de manera diferente a las vistas anteriormente. Se pueden emplear otras distancias, como la de *Manhattan* o la de *Mahalanobis*.

#### Función de activación o transferencia

Las neuronas biológicas, presentan funciones de activación  $f(\cdot)$  deterministas, monótonas crecientes y continuas, por este motivo se emplearán de este tipo en las ANN. En la Tabla 4.4 se muestran las principales funciones de activación. En la tabla, y para abreviar, además de prescindir de algunas constantes, se designa por  $x$  al potencial postsináptico, y con  $y$  al estado de activación. La función más simple es la identidad y se suele emplear en la red *Adaline*, la función escalón se emplea en la red *Perceptron* simple o *Hopfield* discreta. La lineal a tramos es de gran sencillez computacional, y resulta recomendable desde el punto de vista biológico, ya que las neuronas se activan a mayor excitación, y es lo que dicha función pretende imitar. Muchos de los algoritmos de aprendizaje exigen que las funciones de activación sean derivables, por este motivo se emplean las del tipo sigmoideo, principalmente con el algoritmo *backpropagation*. Otra función es la *gaussiana*, que es posible emplearla también en *backpropagation*. En ciertas ocasiones, se emplean las funciones sinusoidales, cuando se requiere expresar una periodicidad temporal.

Tabla 4.4. Principales funciones de activación. Fuente Propia con Datos de Martín del Brío and Sanz (2006).

	<i>Función</i>	<i>Rango</i>
<i>Escalón</i>	$y = \text{signo}(x)$	$\{-1, +1\}$
<i>Identidad</i>	$y = x$	$[-\infty, +\infty]$
<i>Lineal a tramos</i>	$-1, \text{ si } x < -l$ $x, \text{ si } -l \leq x \leq +l$ $+1, \text{ si } x > +l$	$[-1, +1]$
<i>Sinusoidal</i>	$y = A \cdot \text{seno}(\omega x + \phi)$	$[-1, +1]$
<i>Sigmoidea</i>	$y = \frac{1}{1 + e^{-x}}$ $y = \text{tgh}(x)$	$[0, +1]$ $[-1, +1]$
<i>Gaussiana</i>	$A \cdot e^{-Bx^2}$	$[0, +1]$

Función de salida

Con esta función se obtiene la salida global de la neurona, una vez se tienen en cuenta todas las entradas, pesos y funciones anteriores. Lo normal es emplear como función de salida la identidad, de modo que la salida se considera similar al estado de activación

$$y_i(t) = F_i(a_i(t)) = a_i(t) \tag{4.59}$$

En otros modelos, como en la máquina de Boltzmann de Hinton and Sejnowski (1986), se puede emplear una función estocástica, con lo que la neurona podrá tener un comportamiento probabilístico.

**4.5.3.3. Modelo estándar de una neurona artificial**

El modelo *PDP* es muy general, en la práctica se emplea uno más sencillo, el cual llamaremos neurona estándar, con regla de propagación la suma ponderada y la identidad como función de salida de la neurona. Por tanto, la neurona estándar de la Figura 4.14 consiste en:

- Conjunto de entradas  $x_j(t)$  y sus correspondientes pesos sinápticos  $w_{ij}$ .
- Regla de propagación  $h_i(t) = \sum w_{ij}x_j$ .
- La función de activación  $y_i(t) = f_i(h_i(t))$ , que representa simultáneamente la salida de las neuronas y su estado de activación.

A los pesos de la neurona se añade un parámetro adicional,  $\theta_i$ , llamado *umbral (threshold)*, que se resta del potencial postsináptico, por lo que el argumento de la función de activación resulta

$$\sum_j w_{ij} - \theta_i \tag{4.60}$$

lo que hacemos es añadir un grado de libertad a la neurona.

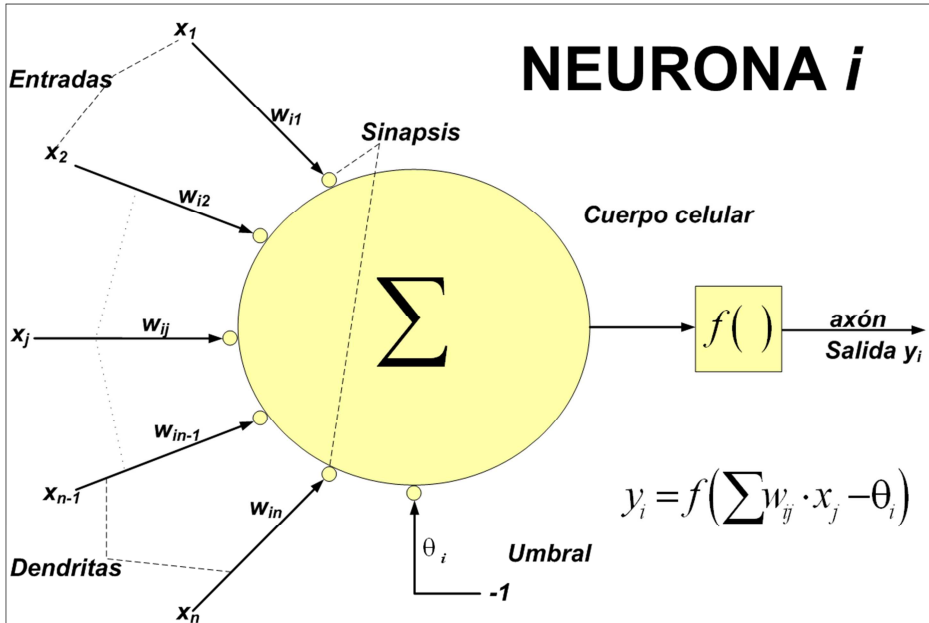


Figura 4.14. Esquema de modelo de neurona estándar. Fuente Propia.

En conclusión, el modelo de neurona estándar resulta

$$y_i(t) = f_i \left( \sum_j w_{ij} x_j - \theta_i \right) \tag{4.61}$$

En (4.61) si hacemos que los dos índices comiencen en 0, podemos definir  $w_{i0} \equiv \theta_i$  y  $x_0 \equiv -1$ , por lo que el potencial postsináptico se obtiene realizando la suma desde  $j=0$  y, finalmente, nos quedará definida la neurona estándar, simplemente con establecer la función de activación de entre las dadas en la Tabla 4.4, resultando la salida de la neurona

$$y_i(t) = f_i \left( \sum_{j=0}^n w_{ij} x_j \right) \tag{4.62}$$

A partir del modelo de la neurona estándar y, dependiendo del tipo de entradas y salidas, obtendremos dos modelos, derivados del estándar, llamados: *neurona todo–nada* y *neurona continua sigmoidea*.

Neurona todo–nada

Partiendo de la neurona estándar, sean las entradas de tipo digital (por ejemplo  $\{0,1\}$ ), función de activación la escalón *Heaviside* definida entre 0 y 1 ( $H(x)=1$  si  $x \geq 0$ ;  $H(x)=0$  si  $x < 0$ ), se tiene

$$y_i(t) = H \left( \sum_j w_{ij} x_j - \theta_i \right) \tag{4.63}$$

y al tener en cuenta los valores de la función de *Heaviside*, (4.62) se convierte en

$$y_i = \begin{cases} 1, & \text{si } \sum w_{ij}x_j \geq \theta_i \\ 0, & \text{si } \sum w_{ij}x_j < \theta_i \end{cases} \quad (4.64)$$

Dicho de otra forma, si el potencial de la membrana supera el valor del umbral de disparo  $\theta_i$  la neurona se activa, en caso contrario, la neurona no se activará. Este modelo es la *neurona Perceptron* original.

Si en este modelo la acción de las entradas inhibitoras es absoluta (una sola entrada inhibitora hará que no dispare la neurona), y se introducen retardos de propagación de las señales, obtenemos el modelo original de *neurona McCulloch–Pitts* (McCulloch and Pitts, 1943), considerado el primer modelado de operación de una neurona artificial.

#### Neurona continua sigmoidea

Partiendo de la neurona estándar, con entradas tanto digitales como continuas (analógicas) y salidas exclusivamente continuas, se puede emplear como función de activación la sigmoidea, que es una función continua y diferenciable en cierto intervalo. Las funciones más habituales de este tipo son las siguientes

$$y = f(x) = \frac{1}{1 + e^{-x}}, \quad \text{con } y \in [0,1] \quad (4.65)$$

$$y = f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}, \quad \text{con } y \in [-1,1] \quad (4.66)$$

Este modelo de neurona es el más empleado en el *Perceptron* multinivel, además, el requisito de trabajar con funciones diferenciables lo marcan las propias reglas de aprendizaje, como el ya comentado *backpropagation*.

Si la *neurona sigmoidea* se interpreta desde el punto de vista probabilístico, la operación deja de ser determinista y se convierte en la *neurona estocástica*.

#### **4.5.3.4. Estructura de una ANN**

Una vez presentados en la sección anterior los elementos que conforman una *ANN*, sobre todo, en torno al nodo de la red (microestructura), en esta sección se detallará la organización de la red (macroestructura), atendiendo al:

- Número de niveles o capas.
- Forma de conexión entre neuronas y el flujo de la información.

Antes de nada, se denomina arquitectura de la *ANN*, a la topología, estructura o patrón de conexionado de la red. Como ya se ha visto, en *ANN*, los nodos se conectan por medio de la sinapsis, es precisamente la estructura de las conexiones sinápticas la que determinan el comportamiento de la propia red. Se debe tener en cuenta que las conexiones sinápticas son

direccionales, es decir, la información fluye en un único sentido, desde la neurona presináptica a la postsináptica.

Las neuronas dentro de una red se organizan formando niveles o capas, teniendo cada capa un número determinado de neuronas, e incluso, dentro de una capa pueden formarse grupos de neuronas, siendo todas las neuronas de una capa del mismo tipo. Se distinguen tres tipos de capas:

- Capa de *entrada* o *sensorial*: capa que recibe la información directamente desde el exterior. La procedencia puede ser de sensores que transmiten información del medio ambiente, de sistemas complejos o incluso de datos procedentes de una base de datos.
- Capa *oculta*: puede existir más de una e incluso cero, y son aquellas que, siendo interiores a la red, no tienen contacto con el exterior. Las neuronas de las capas ocultas estarán interconectadas de distintas formas, lo que determinará junto al número de ellas, diferentes topologías de red.
- Capa de *salida*: formada por neuronas que proporcionan la respuesta de la red al exterior, por medio de efectores<sup>3</sup>.

En la Figura 4.15 se presenta una estructura de una posible red multicapas, en la que las neuronas se encuentran conectadas con todas las neuronas de niveles diferentes, y la información fluye desde la capa de entrada hasta la de salida, y no al revés.

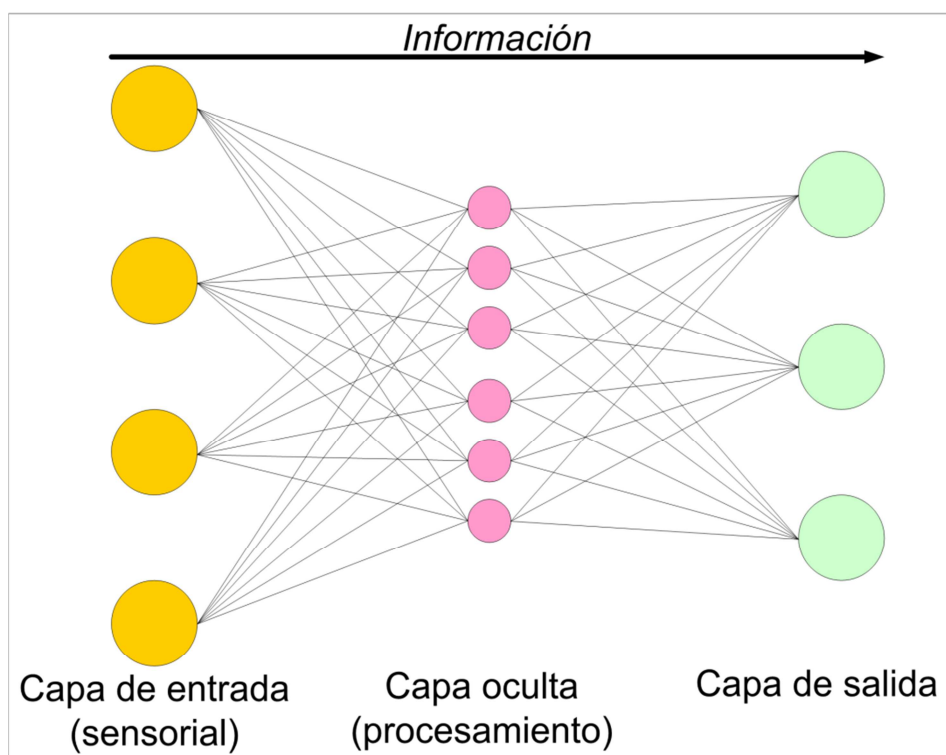


Figura 4.15. Arquitectura unidireccional de tres capas: entrada, oculta y salida. Fuente Propia.

De lo visto en la Figura 4.15, cuando las conexiones entre las neuronas se hacen entre neuronas de diferentes capas, se conoce como *conexión inter-capas*. Si la unión se hace entre

<sup>3</sup> Efectores son células para ejecutar respuestas.



neuronas de la misma capa, se conoce como *conexión intra-capa* o *laterales*. También puede producirse que la alimentación de la información vaya en sentido contrario a la dirección *entrada-salida* (dirección contraria al flujo de información de la Figura 4.15), en ese caso se conoce como *conexiones realimentadas* o *autorecurrente (feedback)*, en contraposición a las *unidireccionales (feedforward)*.

Si atendemos al número de capas, nos podemos encontrar redes con tan sólo una capa llamadas *redes monocapa*, o las formadas por más de una, las conocidas por *redes multicapas (Multi-Layer)*. En la Figura 4.16 se muestran algunos ejemplos de arquitecturas con diferentes números de capas y flujos de información, en la Figura 4.16b la única capa existente es de entrada y de salida, sin existir capa oculta.

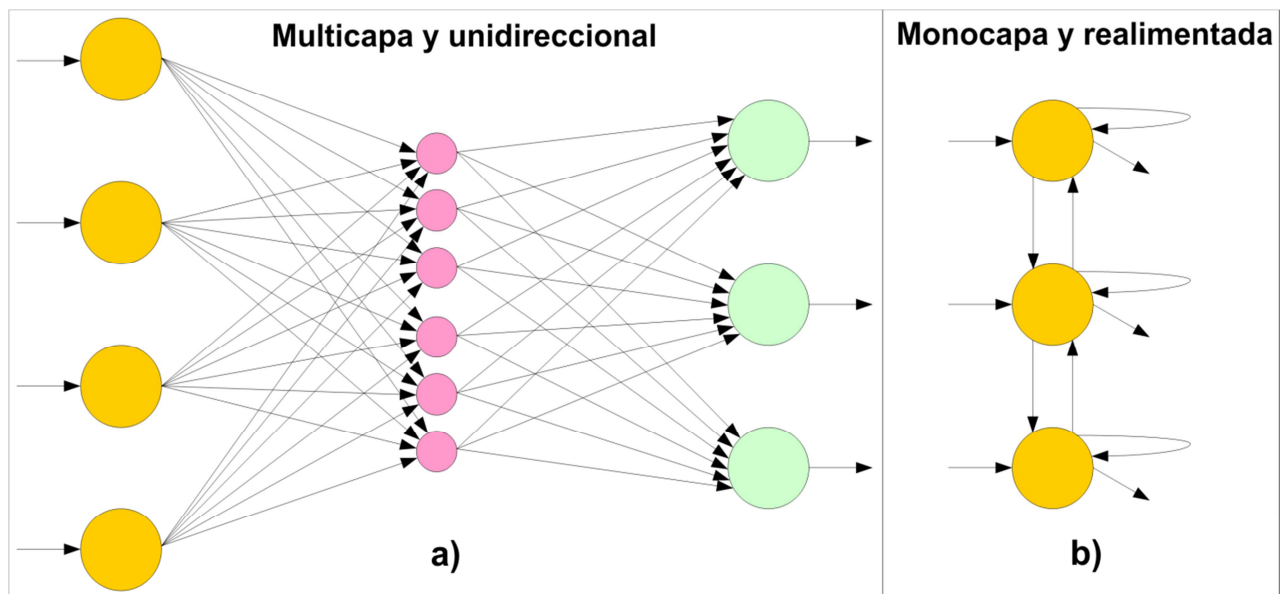


Figura 4.16. Ejemplos de arquitecturas neuronales: a) Multicapa y unidireccional; b) Monocapa y realimentada. Fuente Propia.

Por último, indicar que se habla con frecuencia de redes *autoasociativas* y *heteroasociativas*. A menudo se interpreta la operación de una ANN como la de una memoria asociativa, cuando ante un determinado patrón de entrada responde con cierto patrón de salida. Una red será heteroasociativa, si se entrena de tal manera que ante la presencia del patrón **A**, responda con otro diferente **B**. Por el contrario, si la red es entrenada para que asocie el patrón **A** consigo mismo, estaremos ante una red autoasociativa, como es el caso de la red *Hopfield* descrita en (Hopfield, 1982), que ante un patrón de la forma  $A' = A + ruido$ , la respuesta será el patrón original **A**, eliminando el ruido existente en la señal de entrada.

#### 4.5.4. Características de las ANN

##### 4.5.4.1. Definición formal de la ANN

Llegado este punto, vamos a dar una última definición de ANN, una vez se han tenido en cuenta los últimos aportes de las secciones anteriores. Desde una perspectiva matemática, según Müller and Reinhardt (1990), una ANN es:

“Un grafo dirigido con las siguientes propiedades:

- A cada nodo  $i$  se le asocia una variable de estado  $x_i$ .
- A cada conexión entre los nodos  $i$  y  $j$  se le asocia un peso  $w_{ij} \in \mathfrak{R}$ .
- A cada nodo  $i$  se le asocia un umbral  $\theta_i$ .
- Para cada nodo  $i$  se define una función  $f_i(x_j, w_{ij}, \theta_i)$ , que depende de los pesos de sus conexiones, del umbral y de los estados de los nodos  $j$  a los que está conectado el nodo  $i$ .

Por tanto, y en base a esta nueva definición, algunos de los conceptos introducidos quedarían redefinidos de la siguiente manera:

- Una *neurona de entrada* será aquella carente de sinapsis de entrada.
- Una *neurona de salida* será aquella carente de sinapsis de salida.
- Las neuronas que no son como las dos anteriores serán *neuronas ocultas*.
- Si no hay bucles cerrados de conexiones la red será *unidireccional*.
- Cuando el flujo de información puede evolucionar de atrás hacia delante, la red será *recurrente*.

##### 4.5.4.2. Actualización del estado de las neuronas

Existen dos dinámicas de actualización de los estados de las neuronas: *dinámica síncrona* y *dinámica asíncrona*. Ambas dinámicas pueden aplicarse para la misma ANN, pero para un mismo patrón pueden dar resultados diferentes. Lo anterior queda demostrado en (Bruck, 1990), quien sobre un modelo de *Hopfield* discreto aplica ambas dinámicas. Para la dinámica asíncrona, si la matriz de pesos es simétrica, la red siempre converge a un estado estable, mientras que si sobre la misma red aplicamos una dinámica síncrona, la red puede, o bien converger a un estado estable, o permanecer en un ciclo límite de longitud dos.

Como ya se habrá intuido, se puede introducir una *dinámica no determinista*, haciendo que la salida de la neurona tenga carácter probabilístico, por ejemplo, considerando neuronas con activación sigmoidea, donde la salida de la neurona  $i$  en ese caso será

$$y_i(t+1) = f(h_i(t)) = \frac{1}{1 + e^{-h_i(t)}} \quad (4.67)$$

que pertenece al rango  $[0,+1]$ , siendo la neurona determinista. Pero si consideramos neuronas de salida discreta  $\{0,+1\}$ , interpretaremos el valor obtenido por (4.67) como la probabilidad de que su salida sea  $+1$ , o de otro modo

$$p[y_i(t+1) = +1] = \frac{1}{1 + e^{-h_i(t)}} \quad (4.68)$$

con lo que se ha introducido una dinámica probabilística en la operación de las ANN.

#### 4.5.4.3. Modos de operación: recuerdo y aprendizaje

Se distinguen dos maneras de operar con las ANN, el *modo de aprendizaje* o *entrenamiento*, y el *modo de ejecución* o *recuerdo*. De especial interés será el de entrenamiento, ya que se trata de uno de los pilares de las ANN, ya que las convierte en sistemas entrenables, capaces de hacer un determinado procesamiento, tratándolo de aprender a partir de un conjunto de patrones, para posteriormente intentar reproducirlo en el modelo de ejecución con patrones desconocidos.

##### Fase de entrenamiento o aprendizaje

En el contexto de la ANN, esta fase es el proceso donde se produce el ajuste de los parámetros libres de la red, a partir de un proceso de estimulación por parte del entorno que rodea a la propia red. El entrenamiento lo determinará la manera en que son adaptados dichos parámetros. En la mayoría de los casos tan sólo consistirá en determinar los pesos sinápticos, de manera que permita a la red realizar de manera correcta el procesamiento que se pretende.

El nivel convencional del entrenamiento consistirá por tanto en modelar la sinapsis, modificando los pesos sinápticos siguiendo una regla de aprendizaje o entrenamiento, construida a partir de la optimización de una función de error, que intenta medir la eficacia de la operación de la red. Sea  $w_{ij}(t)$  el peso que une la neurona presináptica  $j$  con la postsináptica  $i$  en la iteración  $t$ , el algoritmo de entrenamiento en función de las señales en dicho instante  $t$  que llegan del entorno, proporcionará un valor  $\Delta w_{ij}(t)$ , que corresponde con la modificación a incorporar en el correspondiente peso, quedando actualizado de la siguiente manera

$$\Delta w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij}(t) \quad (4.69)$$

Como se evidencia por tanto, el modo entrenamiento es iterativo, tratando de actualizar los pesos sinápticos de manera análoga a lo indicado anteriormente, en cada iteración, hasta que la ANN consigue el rendimiento deseado.

Un segundo nivel de aprendizaje que emplean algunos modelos, consiste en la creación o destrucción de neuronas, a la vez que se modifica la propia arquitectura de la ANN. En cualquier caso, durante el entrenamiento, la información que contienen los datos de entrada queda

impregnada en la propia estructura de la ANN, la cual trata de almacenar en cierta manera la representación de la imagen de su entorno.

Los dos tipos básicos de entrenamiento son el *supervisado* y el *no supervisado*, diferenciándose en la manera en que tratan los patrones. El entrenamiento supervisado proporciona cierta información de la función  $e-s$  a estimar, en forma de distribución de las clases o las etiquetas de los patrones de salida, mientras que en el no supervisado no existe tal información. Como norma, las reglas de aprendizaje supervisado suelen ser más complejas, pero también más exactas en los resultados. Además de estas dos, se formalizan otras en (Haykin, 1999), destacando el aprendizaje *híbrido* y el *reforzado*. A continuación se detallará cada uno de los cuatro modos:

- *Aprendizaje supervisado*: llamando  $E[W]$  a la función que representa el error que se espera al operar la red y  $W$  los pesos sinápticos, en este aprendizaje se trata de estimar una función multivariable, no conocida,  $f : \mathfrak{X}^n \rightarrow \mathfrak{X}^m$  a partir de las muestras  $(\mathbf{x}, \mathbf{y})$ , con  $\mathbf{x} \in \mathfrak{X}^n$  e  $\mathbf{y} \in \mathfrak{X}^m$  tomadas de manera aleatoria, a través de la minimización iterativa de  $E[W]$ , mediante aproximación estocástica, como se ve en (White, 1989).

Dicho de otra forma, en este aprendizaje, los patrones se presentan a la red, junto a la salida deseada u objetivo (*target*), y de manera iterativa se van ajustando los pesos hasta que su salida tienda a la deseada, empleando para esto información del error que se comete en cada paso. Por tanto, la red es capaz de estimar relaciones  $e-s$  sin necesidad de detallar un punto de partida.

- *Aprendizaje no supervisado* o *auto-organizado*: es la estimación de la función densidad  $p(\mathbf{x})$ , que describe la distribución de patrones  $\mathbf{x}$  pertenecientes al espacio de entrada  $\mathfrak{X}^n$ , a partir de patrones a modo de ejemplo.

En este aprendizaje se muestra a la red únicamente los patrones, sin necesidad de presentar objetivos de los mismos, a partir de esto, la red mediante la regla de aprendizaje, estima el valor  $p(\mathbf{x})$ , comenzando a detectar regularidades en el conjunto de entradas, extraer rasgos, o incluso agrupar patrones según cierta similitud.

- *Aprendizaje híbrido*: son los aprendizajes donde coexiste el modelo supervisado y no supervisado a la vez, normalmente en distintas capas de neuronas.
- *Aprendizaje reforzado*: a medio camino entre supervisado y no supervisado, como sucede en el primer método, se emplea información sobre el error cometido, pero existiendo tan sólo una señal de error, que representa un índice global del rendimiento de la red. Como en el no supervisado, no se suministra la salida deseada. En ciertas ocasiones es conocido como aprendizaje por premio-castigo.

Por lo general, los algoritmos de aprendizaje son métodos numéricos iterativos que intentan minimizar la función de coste, pudiéndose producir problemas de convergencia del algoritmo, por tanto, no se puede abordar de manera general, sino que deben particularizarse para cada algoritmo concreto. La convergencia es una manera de comprobar si cierta arquitectura, junto a la regla de aprendizaje, puede solventar el problema, ya que el error medido en el entrenamiento describe la precisión del ajuste del mapeado (*mapping*).

Durante el entrenamiento hay que distinguir el error conseguido con el *CE*, al final de dicha fase, y el error que la red ya entrenada comete frente a patrones no pertenecientes a dicho conjunto, que mide el error de *generalización*. Resulta más interesante un error bajo de generalización que un error muy pequeño en el entrenamiento, ya que es indicativo de que la red habrá captado correctamente el *mapping* subyacente en los datos.

#### Fase de recuerdo o ejecución

Una vez finalizada la fase de entrenamiento, los pesos y la estructura de la *ANN* quedan fijados, quedando la red dispuesta para el proceso de nuevos datos, esta fase es la conocida como *modo recuerdo* o de *ejecución*.

En las redes unidireccionales, cuando un patrón es presentado, las neuronas responden proporcionando una salida, al no existir bucles realimentados, no existe el problema de su estabilidad. En cambio, en las redes con realimentación se requieren ciertas condiciones para la convergencia de la respuesta a un estado estable. En (Simpson, 1989) se indican las condiciones para la estabilidad en una amplia gama de *ANN*, en determinadas condiciones.

Para la demostración de la estabilidad, los métodos clásicos consisten en integrar el sistema de ecuaciones diferenciales que lo describen, sin embargo, en (Simpson, 1989) se presenta el *método de Lyapunov*, como alternativa. Este método establece que si un sistema dinámico (por ejemplo una *ANN*), con variables de entrada  $(x_1, \dots, x_n)$  y descrito por el sistema de ecuaciones diferenciales

$$\dot{x}_i \equiv \frac{dx_i}{dt} = F(t, x_1, \dots, x_n) \quad (4.70)$$

se cumplen las condiciones:

- Sistema en reposo solamente en el origen.
- En todo su dominio, existen las derivadas de las ecuaciones que describen el sistema.
- Las variables están acotadas

y se puede encontrar la *función de Lyapunov*  $V$ , con  $V : \mathfrak{R}^n \rightarrow \mathfrak{R}$  tal que

$$\dot{V} = \sum_{i=1}^n \frac{\partial V}{\partial x_i} \dot{x}_i \leq 0, \forall \dot{x}_i \quad (4.71)$$

entonces el sistema converge para todas las variables de entrada  $(x_1, \dots, x_n)$  y es globalmente estable.

Esta función es a veces conocida como *función energía de Lyapunov*, ya que es una generalización del concepto físico de energía. Este método, por tanto, constituye una manera asequible de tratar el estudio de la estabilidad de un sistema dinámico. Una estrategia similar fue empleada por Hopfield (1982; 1984) para demostrar la estabilidad de su modelo completamente conectado, en el caso de una matriz con pesos sinápticos simétrica y con diagonal nula.

La misma técnica fue empleada en los teoremas de Cohen, Grossberg y Kosko, en las obras (Cohen and Grossberg, 1983; Kosko, 1988; Kosko, 1992a; Kosko, 1992b), para demostrar la estabilidad de un gran número de ANN realimentadas, *autoasociativas* y *heteroasociativas*.

#### 4.5.4.4. Clasificaciones de los modelos ANN

De lo planteado hasta el momento, se concluye que existen diferentes modelos de ANN, dependiendo del modelo de neurona, la arquitectura o topología de conexionado y del algoritmo de aprendizaje.

Las características fundamentales de los modelos son su tipo de aprendizaje y la arquitectura de red, por tanto, en la Figura 4.17 se muestra una clasificación en base a estas características. La misma clasificación se adopta en (Simpson, 1989), tratando aquí de actualizarla y completarla.

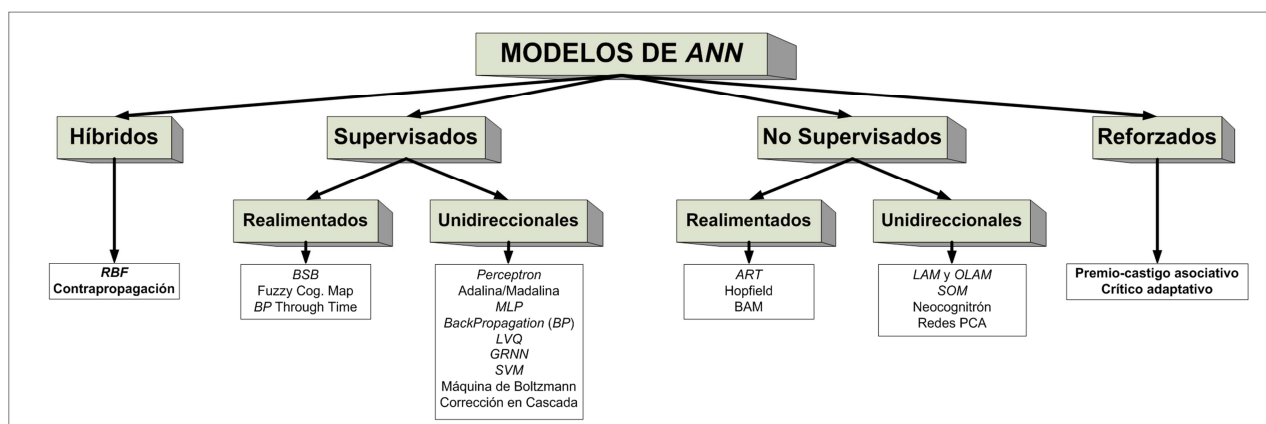


Figura 4.17. Clasificación de ANN en función de su aprendizaje y arquitectura. Fuente Propia.

Primeramente se hace una distinción en cuanto al aprendizaje, quedando la clasificación en modelos supervisados, no supervisados, híbridos y con aprendizaje reforzado. Dentro de cada grupo, se tendrá en cuenta la topología de la red, por lo que se distinguirá entre redes realimentadas y unidireccionales.

El grupo más numeroso es el de modelos de red no realimentados y con aprendizaje supervisado, siendo los más destacados el *Perceptron* simple, *Adaline* y el *MLP*. Estos modelos resultan de especial importancia por varias razones: históricas, generalidad, aglutinar gran cantidad de aspectos que se repiten frecuentemente en las ANN y, además, son muy empleados en las

aplicaciones prácticas. Dentro del grupo de los modelos no supervisados, destacan los modelos *Hopfield* y *SOM*.

En la presente *Tesis Doctoral* se detallarán las principales características de los modelos aquí empleados, para ampliar esta información e incluso encontrar detalles de los no expuestos, se recomienda la lectura de los siguientes textos (Hecht–Nielsen, 1990; Hertz, 1991; Haykyn, 1999; Principe *et al.*, 2000).

#### **4.5.5. Redes neuronales supervisadas**

En esta sección se estudiarán los modelos de *ANN* más populares, comenzando por la amplia gama de modelos unidireccionales organizados en capas (*feed-forward*), y con aprendizaje supervisado, que se suelen emplear como estimadores de función. En la literatura, estos modelos son conocidos como Mapeo de Redes Neuronales (*mapping neural networks*), como detalla Hecht–Nielsen (1990), o *ANN* para ajuste funcional.

Dentro de este gran grupo destacan el *Perceptron* simple, *Adaline* y *MLP*. En el modelo *MLP* se emplea el popular algoritmo de aprendizaje conocido como *Retropropagación* (*backpropagation*), convirtiéndose en el modelo neuronal más empleado en aplicaciones prácticas, sirva como dato el aportado por Gedeon *et al.* (1995), quienes estimaron que más del 70% de los desarrollos con *ANN* hacían uso de alguna de sus variantes.

##### **4.5.5.1. Las redes unidireccionales**

Es muy familiar encontrar el problema de estimar o aproximar funcionalmente un problema, a partir de un conjunto de entradas para proporcionar una salida deseada. Ejemplos en este sentido serán la predicción de cotizaciones en bolsa, donde mediante una *ANN* se intentará encontrar una función que relacione las variables de entrada con la cotización en bolsa de cierta empresa, o por ejemplo, un reconocedor de caracteres manuscritos, donde el objetivo será encontrar una función que trate de asociar una imagen de letra o carácter escrito con la clase a la que pertenece.

Como ya se ha comentado, dentro de este grupo de redes unidireccionales, son de especial importancia el *Perceptron* simple, *Adaline* y *MLP*. En primer lugar, se debe destacar su importancia histórica, ya que su evolución misma define la historia de las *ANN*, así, el *Perceptron* simple y *Adaline* se propusieron en los años cincuenta del pasado siglo, para sufrir un duro revés al final de los sesenta con el trabajo de Minsky y Papert, poniendo sus limitaciones de manifiesto. Tras una época oscura de las *ANN*, volvieron a resurgir en la década de los ochenta, apoyado el resurgimiento por la potencia de cálculo de los computadores y, por supuesto, los trabajos variados que en este campo se crearon en esa época. Es en este momento cuando surge *MLP*, entrenado mediante

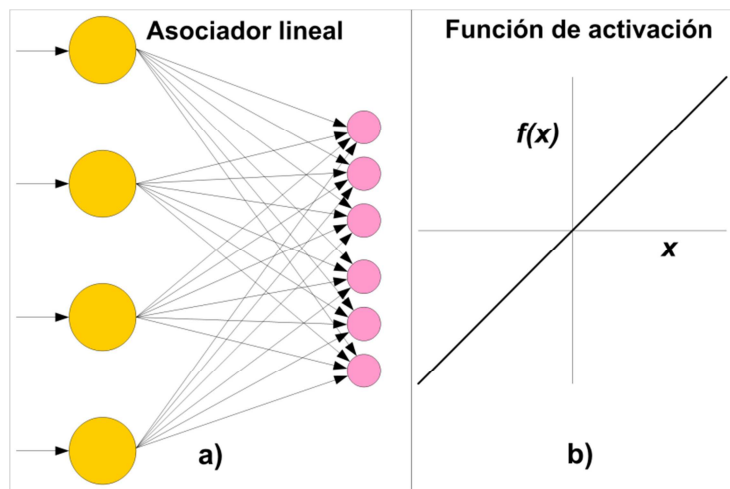
*backpropagation*, que superó las limitaciones de sus modelos predecesores, anulando muchas de las objeciones históricas de Minsky y Papert.

Pero sin duda, lo más importante de estos modelos es su generalidad y su aplicabilidad práctica y, además, en diferentes ámbitos de la ingeniería. Además, resulta interesante saber que ciertos modelos de aprendizaje empleados en ellas, se reutilizan en el entrenamiento de los sistemas borrosos.

**4.5.5.2. El aprendizaje hebbiano**

Antes de abordar los modelos ya citados, estudiaremos el *asociador lineal*, un sencillo ejemplo de red unidimensional que ayudará a centrar conceptos relacionados, principalmente, con el aprendizaje en las ANN. Este modelo, tras una transformación lineal, asocia un conjunto de patrones de entrada a otros diferentes de salida.

En la Figura 4.18 se muestra este modelo, que consta de una capa de neuronas lineales, donde las entradas las nombraremos por  $x$ , y sus salidas por  $y$ , vector que constituye además la respuesta de la ANN. La matriz de pesos sinápticos será  $W = \{w_{ij}\}$ , donde cada fila contiene los pesos de una neurona  $w_i$ .



**Figura 4.18. a) Arquitectura del Asociador Lineal; b) Función de activación identidad. Fuente Propia.**

Por tanto, la operación del asociador lineal vendrá expresada por

$$y = W x ; y_i = \sum_{j=1}^n w_{ij} x_j \tag{4.72}$$

Cada neurona del asociador lleva a cabo la suma ponderada de las entradas, junto a sus pesos sinápticos, dicho de otra forma, la neurona calcula el potencial postsináptico por medio de la suma ponderada, cantidad a la que aplica finalmente la función de activación de tipo identidad.

La asociador lineal debe tratar de aprender a asociar  $p$  pares  $e-s$ ,  $\{(x^\mu, t^\mu) / 1 \leq \mu \leq p\}$ , ajustando sus pesos sinápticos, de modo que ante cada patrón  $x^\mu$  responda con  $t^\mu$ , y que ante



entradas similares ( $\mathbf{x}^\mu + \varepsilon$ ), responda con salidas ( $\mathbf{t}^\mu + \delta$ ) también próximas. Por tanto, el problema se centrará en encontrar la matriz de pesos óptima, según lo anteriormente indicado, para ello, se hace uso de la *regla de aprendizaje*, la cual a partir de las entradas y salidas, proporciona el conjunto óptimo de los pesos.

Regla de aprendizaje de Hebb

Hebb (1949) postuló un mecanismo de aprendizaje para la neurona biológica, que básicamente consiste en que “cuando un axón presináptico causa la activación de cierta neurona postsináptica, la eficacia de la sinapsis que las relaciona se refuerza”. Posteriormente, se confirmó en parte esta teoría en la obra (Kandel and Hawkins, 1992), demostrando la existencia de este tipo de aprendizaje en la neurona biológica, pero influenciado por otros sistemas como la interacción a nivel presináptico, crecimiento y debilitación del axón, desarrollo y muerte celular, entre otros, como evidenció Alkon (1989). Este tipo de aprendizaje es simple y local, siendo pionero tanto en neurociencia como en neurocomputación, y algoritmos más complejos lo han tomado como punto de partida.

La forma de aprendizaje que involucra la modificación de pesos  $\Delta w_{ij}$ , proporcional al producto de una entrada  $i$  por una salida  $j$ , se denomina *aprendizaje hebbiano*, y dicha implicación es

$$\Delta w_{ij} = \varepsilon y_i x_j \tag{4.73}$$

$\varepsilon$  es el parámetro conocido por *ritmo* o *tasa de aprendizaje*, cantidad que varía entre 0 y 1. La elución (4.73) suele considerarse la representación matemática del *aprendizaje de Hebb*.

En el caso del ejemplo, el asociador lineal, la *regla de Hebb* se expresa por  $\Delta w_{ij}^\mu = t_i^\mu x_j^\mu$  y, por tanto,  $w_{ij}^{new} = w_{ij}^{old} + \Delta w_{ij}^\mu$ . Si los pesos de partida son nulos, el valor final de  $W$  para las  $p$  asociaciones será  $W = \mathbf{t}^1 \mathbf{x}^{1T} + \mathbf{t}^2 \mathbf{x}^{2T} + \dots + \mathbf{t}^p \mathbf{x}^{pT}$ . Empleando la *regla de Hebb*, con vectores de entrada ortonormales se cumple

$$W \mathbf{x}^\mu = (\mathbf{t}^1 \mathbf{x}^{1T} + \mathbf{t}^2 \mathbf{x}^{2T} + \dots + \mathbf{t}^p \mathbf{x}^{pT}) \cdot \mathbf{x}^\mu = \mathbf{t}^1 (\mathbf{x}^{1T} \cdot \mathbf{x}^\mu) + \dots + \mathbf{t}^p (\mathbf{x}^{pT} \cdot \mathbf{x}^\mu) = \mathbf{t}^\mu \tag{4.74}$$

y por tanto, con el *aprendizaje de Hebb* se consigue que la red haga la asociación deseada. La cuestión es que las condiciones son muy restrictivas, ya que para unas respuestas correctas las entradas deben de ser ortonormales, debido a esto, si el espacio de entradas es  $n$ , se podrán realizar  $n$  asociaciones.

Manteniendo todos los anteriores requisitos, salvo el de ortogonalidad, obtenemos la expansión señal-ruido

$$W \mathbf{x}^\mu = \mathbf{t}^1 (\mathbf{x}^{1T} \cdot \mathbf{x}^\mu) + \dots + \mathbf{t}^p (\mathbf{x}^{pT} \cdot \mathbf{x}^\mu) = \mathbf{t}^\mu + \sum_{v \neq \mu} \mathbf{t}^v (\mathbf{x}^{vT} \cdot \mathbf{x}^\mu) = \mathbf{t}^\mu + \delta \tag{4.75}$$

proporciona la salida deseada más un término adicional, que asociamos a ruido en la señal. Técnicas más sofisticadas que la de *Hebb*, como la de la *pseudoinversa* o la de *Widrow–Hoff* obtendrán una matriz de pesos que permitirá además que el ruido  $\delta$  sea pequeño comparado con la señal.

### Regla de la pseudoinversa

De manera general, los algoritmos de aprendizaje se intentarán deducir a partir de un criterio a optimizar, el aprendizaje se planteará como un proceso para alcanzar el conjunto de pesos óptimo que resuelva un problema dado. Por tanto, se hace preciso definir “*óptimo*” para cada caso, en otras palabras, hay que proponer un criterio que evalúe el rendimiento de la red para encontrar la regla de actualización de pesos que lo optimice. Una forma es emplear el error cuadrático medio de las salidas actuales de la red respecto de las deseadas, que para nuestro asociador lineal se tendrá

$$E\{w_{ij}\} = \left(\frac{1}{p}\right) \sum_{\mu=1}^p / t^{\mu} - W x^{\mu} / ^2 = \left(\frac{1}{p}\right) \sum_{\mu=1}^p \sum_{i=1}^n \left(t_i^{\mu} - W x_i^{\mu}\right)^2 \quad (4.76)$$

de esta forma, para el asociador lineal, el algoritmo de aprendizaje deberá obtener un conjunto de pesos que minimicen (4.76).

Si nombramos  $X$  a la matriz  $n \times p$ , con columnas los vectores de entrada  $x^{\mu}$ , e  $Y$  a la matriz  $m \times p$ , con columnas los vectores de salida  $y^{\mu}$ , la ecuación (4.76) se convierte en

$$E\{w_{ij}\} = \left(\frac{1}{p}\right) \|Y - WX\|^2 \quad (4.77)$$

Una regla de aprendizaje basada en la *matriz pseudoinversa* puede expresarse por

$$W = YX^+ \quad (4.78)$$

donde  $X^+$  denota la pseudoinversa de  $X$ , empleado en (Kohonen, 1989; Hecht–Nielsen, 1990). En (Ritter *et al.*, 1991) se deduce (4.78) a partir de la minimización algébrica de (4.77).

La *regla de Hebb* almacena hasta  $n$  vectores ortonormales, con la *pseudoinversa* se pueden almacenar hasta  $n$  vectores linealmente independientes. Si se pretenden almacenar más pares  $e-s$ , surgirán errores, pero el *mapping* lineal seguirá siendo óptimo en el sentido del error cuadrático medio.

Ambas reglas, *Hebb* y *pseudoinversa* están muy relacionadas, por ejemplo, si se considera un conjunto de vectores de entrada ortonormal, (4.78) se convierte en la de *Hebb*. Por otro lado, si se hace la expansión en serie de (4.78), como se realiza en (Rumelhart and McClelland, 1986), el primer término de la misma es la ecuación de la *regla de Hebb*, es decir, la *regla de Hebb* representa un caso particular de la más general *regla pseudoinversa*.

Se ha demostrado en (Hecht–Nielsen, 1990), que para el cálculo de la *pseudoinversa*, puede emplearse una aproximación del teorema de Greville, local e iterativa, y que proporciona resultados correctos

$$\mathbf{w}_i^{new} = \mathbf{w}_i^{old} + \varepsilon \cdot \left( \mathbf{t}_i^\mu - (\mathbf{w}_i^{old})^T \mathbf{x}^\mu \right) \cdot \mathbf{x}^\mu \quad (4.79)$$

$0 < \varepsilon < 1$  es la tasa de aprendizaje e indica la rapidez de actualización. En este esquema iterativo, los patrones se presentan a la red repetidamente, obteniéndose por tanto una aproximación a la matriz pseudoinversa con cálculos simples y locales.

La anterior expresión coincide con la de Widrow–Hoff de *Adaline*. Asimismo, en (Rumelhart and McClelland, 1986) se relacionan ambos algoritmos de aprendizaje, mostrándose que ambas reglas son equivalentes en aprendizaje estadístico, consistente en lugar de asignar a un patrón de entrada uno de salida, toda una clase de vectores de entrada a una clase del espacio de salida.

La *regla pseudoinversa* se ha empleado en redes mucho más complejas, como la de *Hopfield* en el trabajo (Personnaz *et al.*, 1986), obteniendo mejores resultados que con *Hebb*, inicialmente propuesta y, más conocida y estudiada.

#### 4.5.5.3. El perceptrón simple

La estructura de *Perceptron* está inspirada en las primeras etapas de los sistemas sensoriales animales, donde la información atraviesa numerosas capas, realizando procesamiento progresivo de más alto nivel. El modelo lo introdujo Rosenblatt a finales de los años cincuenta del pasado siglo, por medio de su trabajo (Rosenblatt, 1958).

Es un modelo unidireccional, formado por dos capas, una la sensorial o de entrada y, otra de salida, tal y como se representa en la Figura 4.19. El modo de operar de esta red, formada por  $n$  neuronas en la capa de entrada y  $m$  en la de salida, se puede expresar como

$$y_i(t) = f \left( \sum_{j=1}^n w_{ij} x_j - \theta_i \right), \forall i, 1 \leq i \leq m \quad (4.80)$$

Las neuronas de la capa de entrada, en principio, no realizan ningún computo, simplemente envían la información (consideremos señales discretas  $\{0, +1\}$ ) a las neuronas de salida. La función de activación de las neuronas de la capa de salida es de tipo escalón como se aprecia en la Figura 4.19b, por tanto, la operación de un *Perceptron* simple podrá escribirse de la siguiente manera

$$y_i = H \left( \sum_{j=1}^n w_{ij} x_j - \theta_i \right), \forall i, 1 \leq i \leq m \quad (4.81)$$

$H(\cdot)$  es la función de *Heaviside* o escalón. *Perceptron* puede emplearse tanto como clasificador como para la representación de funciones booleanas, pues en esencia, es una neurona de tipo McCulloch–Pitts, con salida binaria.

La importancia histórica del modelo es su carácter de dispositivo entrenable, gracias al algoritmo de aprendizaje introducido por Rosenblatt y que se verá más adelante, que permite ajustar

automáticamente los pesos sinápticos que clasifican un conjunto de patrones a partir de unos elementos etiquetados.

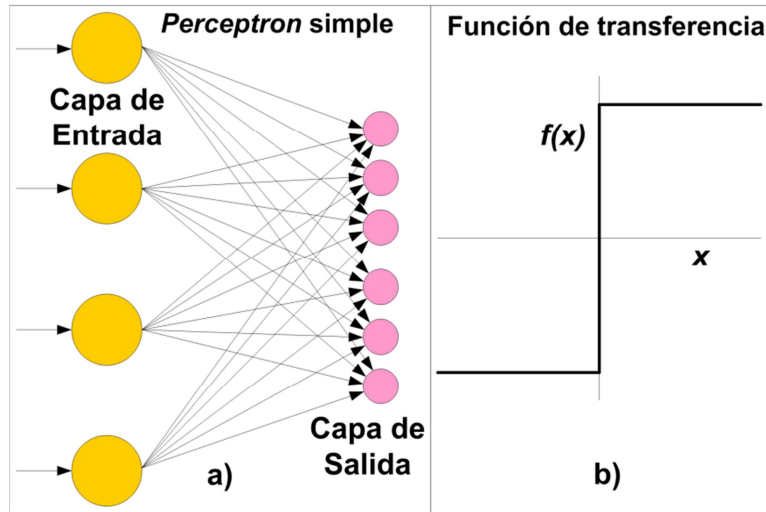


Figura 4.19. a) Esquema de *Perceptron* simple; b) Función de transferencia de neurona *Perceptron*. Fuente Propia.

Consideremos la función lógica *NAND*, *AND* negada de dos entradas, representada en la Figura 4.20a en el plano. Es posible encontrar unos parámetros  $w_1$ ,  $w_2$  y  $\theta$  que determinan una recta que separa perfectamente las regiones que corresponden con los valores lógicos 0 y 1. Por esto, la función lógica *NAND* se dice separable linealmente y, por ejemplo, un *Perceptron* con los parámetros  $w_1 = w_2 = -2$ ,  $\theta = -3$ , implementará dicha función.

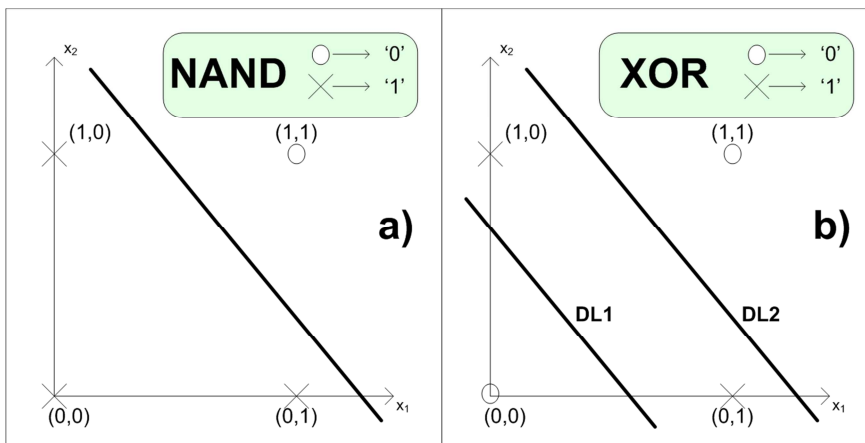


Figura 4.20. a) Función lógica *NAND*; b) Función lógica *XOR*. Fuente Propia.

Sin embargo, consideremos la función lógica *XOR*, su salida es un 0 si las variables de entrada son iguales y 1 si son diferentes, representada en la Figura 4.20b en el plano. No es posible encontrar una única condición lineal que separa las regiones de los valores 0 y 1, por lo que dicha función no es separable linealmente. Como la neurona *Perceptron* representa un discriminador lineal, esta neurona no podrá por sí sola implementar *XOR*, por lo que, las funciones no separables linealmente no pueden ser representadas por un *Perceptron* simple.

Volvamos la mirada al problema de *XOR*, se puede pensar en emplear dos neuronas *Perceptron*, una que implemente la decisión lineal *DL1* y la otra *DL2* (ver Figura 4.20b). Considérese una capa adicional, con tan sólo una neurona *Perceptron* encargada de componer las regiones en que el plano queda dividido por las dos neuronas anteriores, esta neurona se activará únicamente cuando la neurona de *DL1* lo esté y *DL2* desactivada, por lo tanto, tendremos una red de tres capas, que implementará la función *XOR*. La conclusión es, que una posible solución al *XOR*, es emplear una *MLP*, o lo que es lo mismo, una arquitectura *Perceptron* pero con más capas.

Según Lippmann (1987), los tipos de regiones de decisión por medio de *Perceptron* son: sin capa oculta la región de decisión es un hiperplano (dos regiones); con una capa oculta la región se pueden formar regiones polinomiales convexas; y con dos capas ocultas se pueden formar regiones arbitrarias.

Demostraciones más formales de los conceptos expuestos se pueden encontrar en (Müller and Reinhardt, 1990), donde por ejemplo, se demuestra que “*toda función booleanas puede ser representada por una red neuronal unidireccional con una única capa oculta*”, detalle de mucho interés, ya que en muchas ocasiones los modelos con tan sólo una capa oculta, son capaces de resolver la mayoría de los problemas complejos. Lo anterior fue demostrado por primera vez por Denker *et al.* (1987).

Al terminar la década de los setenta del siglo pasado, ya se vislumbraba la solución de las limitaciones del *Perceptron* simple, introduciendo varias capas en su arquitectura, el problema radicaba en que sí se disponía de un algoritmo de aprendizaje para el simple, pero no existía procedimiento alguno para obtener de manera automática los pesos de las neuronas de las capas ocultas. Este problema se denominó de *asignación de crédito*, ya que había que darles crédito (pesos) a las neuronas de las capas ocultas y sin conexión con el exterior. El problema lo resolvió más tarde Werbos (1974), pero debió esperarse hasta mediados de la década de los ochenta del siglo pasado, cuando el grupo *PDP* reformularan un algoritmo similar, que llamaron *backpropagation*, presentado por Rumelhart and McClelland (1986), y sirvió para que la comunidad científica conociera el gran potencial que se les presentaba para solucionar problemas prácticos de difícil solución.

#### *Perceptron y su algoritmo de aprendizaje*

El algoritmo de aprendizaje que emplea *Perceptron*, es de los llamados de *corrección de errores*. Estos algoritmos ajustan los pesos en proporción a la diferencia entre la salida actual de la red y la deseada, con el objetivo de minimizar el error actual de la red.

Sea un conjunto de  $p$  patrones,  $\mu = 1, \dots, p$ , de entradas junto a sus salidas. Ambas señales, tan sólo pueden tener valores  $\{-1, 1\}$  (o  $\{0, 1\}$  según los niveles lógicos definidos). Se dispone de un *Perceptron* simple, al que se da pesos iniciales de manera aleatoria, y se requiere que clasifique de

manera correcta todos los patrones del  $CE$ . Se actuará como sigue, ante la presentación del patrón  $\mu$ -ésimo, si la respuesta que proporciona la red es correcta, los pesos no son actualizados, en cambio, si es incorrecta, se modificarán en función de la *regla de Hebb*

$$\Delta w_{ij}^{\mu}(t) = \varepsilon \cdot (t_i^{\mu} - y_i^{\mu}) x_j^{\mu} \quad (4.82)$$

que es la *regla del Perceptron*. Es necesario llegar a un compromiso entre la tasa de aprendizaje  $\varepsilon$ , ya que un valor pequeño implicará un aprendizaje lento y, uno excesivamente grande podría conducir oscilaciones durante el entrenamiento, al variar excesivamente los pesos. Al ser las señales de entrada y salida discreta, lo serán también las actualizaciones de (4.82), que tomarán valores 0 o  $\pm 2\varepsilon$ .

La *regla del Perceptron* se puede explicar de una manera más gráfica, se la neurona  $i$  con el vector de pesos  $w_i$ , para el patrón de entrada  $x^{\mu}$  su salida objetivo es  $t_i^{\mu}$ ; la operación de la neurona será

$$y_i^{\mu}(t) = \text{signo}(\|w_i\| \cdot \|x^{\mu}\| \cos(\phi)) \quad (4.83)$$

habiendo considerado el umbral como un peso adicional de entrada -1 y, siendo el ángulo  $\phi$  el que forman los pesos de entrada y salida. Durante el aprendizaje, ante el patrón  $\mu$ -ésimo en la iteración  $t$  se pueden dar los siguientes casos:

- La salida objetivo de la neurona es +1, pero su salida actual es -1, en este caso el productor escalar  $w_i \cdot x^{\mu}$  debería ser positivo, pero es negativo, lo que indica que el ángulo entre ambos vectores es mayor de  $90^{\circ}$  ( $\phi \in [\frac{\pi}{2}, \frac{3\pi}{2}]$ ), como se ve en la Figura 4.21. En esta situación, la regla debería acercar el vector  $w_i$  a  $x^{\mu}$  para reducir el ángulo que forman, y de manera eventual, tratar de que sea menor de  $90^{\circ}$  ( $w_i \cdot x^{\mu} > 0$ ), lo cual se puede conseguir según la Figura 4.21a con  $w_i^{\mu}(t+1) = w_i^{\mu}(t) + \alpha \cdot x^{\mu}$ .
- La salida objetivo de la neurona es +1, pero su salida actual es -1. Razonando al revés que en el caso anterior, la regla debería alejar el vector  $w_i$  de  $x^{\mu}$ , lo cual se puede conseguir según la Figura 4.21b con  $w_i^{\mu}(t+1) = w_i^{\mu}(t) - \alpha \cdot x^{\mu}$ .
- La regla de aprendizaje no actuará, si la salida objetivo de la neurona coincide con su salida actual.

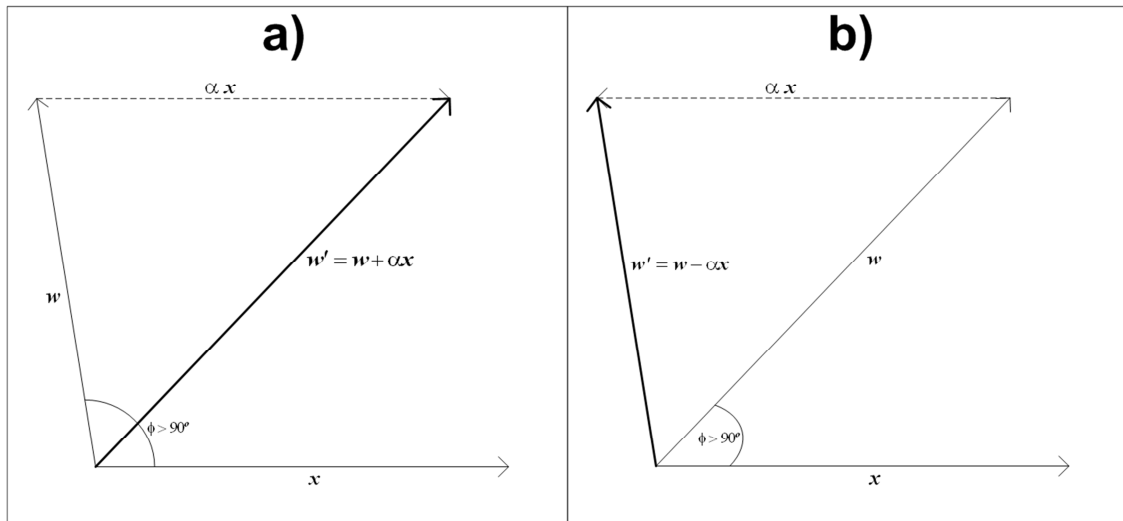
Es fácil demostrar que los tres anteriores casos se pueden agrupar en la siguiente regla

$$w_i(t+1) = w_i(t) + \left(\frac{\alpha}{2}\right) \cdot x^{\mu} (t_i^{\mu} - y_i^{\mu}) \quad (4.84)$$

y si llamamos  $\varepsilon \equiv \alpha/2$ , tenemos

$$\Delta w_i(t) = \varepsilon \cdot x^\mu (t_i^\mu - y_i^\mu) \quad (4.85)$$

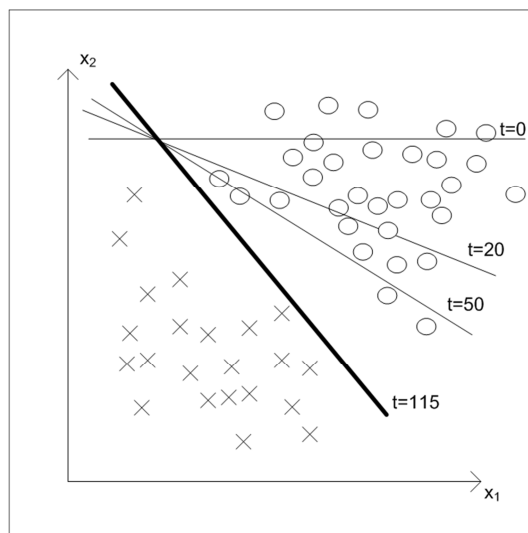
que es la regla del Perceptron (4.82) ya conocida.



**Figura 4.21. Regla del Perceptron, cuando salida actual y objetivo no coinciden: a) Producto escalar positivo; b) Producto escalar negativo. Fuente Propia.**

Debemos de recordar que el proceso es iterativo, a partir de una situación sináptica de partida y, presentando repetidamente los patrones, tratando de ajustar los pesos según (4.82), hasta conseguir que queden bien clasificados. El hiperplano límite entre dos clases, se desplaza lentamente hasta separarlas de manera correcta, como se ve en la Figura 4.22. El ajuste de los pesos en la  $t$ -ésima iteración, debido a todo el  $CE$  será

$$w_{ij}(t+1) = w_{ij}(t) + \sum_{\mu=1}^p \Delta w_{ij}(t) \quad (4.86)$$



**Figura 4.22. Regiones de decisión. Fuente Propia.**

Si la función a representar es linealmente separable, Rosenblatt demostró que el algoritmo siempre converge, y en un tiempo finito, con independencia de los pesos de partida. Por otra parte, si la función no es linealmente separable, el proceso de aprendizaje oscilará. El algoritmo se detiene

tan pronto como consigue clasificar correctamente los patrones, por lo que es posible que la línea de discriminación quede cerca de las muestras de uno de los grupos, como ha pasado en la Figura 4.22, donde ha quedado cerca de “x”. Para obtener una discriminación óptima, se han planteado algoritmos como el llamado *Adatron* (Principe *et al.*, 2000).

#### 4.5.5.4. *Adaline*

Otro modelo clásico es *Adaline*, introducido por Widrow en los trabajos (Widrow and Hoff, 1960; Widrow and Winter, 1988), emplea una neuronas similar a *Perceptron*, pero con respuesta lineal y con posibilidad de que sus entradas sean continuas, tal y como se ve en la Figura 4.23.

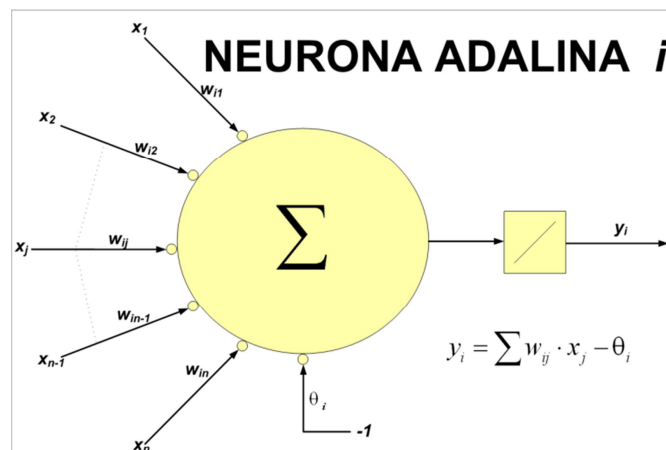


Figura 4.23. Neurona *Adaline*. Fuente Propia.

A diferencia del nodo del asociador lineal, *Adaline* incorpora un parámetro extra llamado *bias*, comúnmente conocido como umbral, pero no se trata de un umbral de disparo como en *Perceptron*, sino un parámetro que proporciona un grado de libertad adicional, pudiendo de esta manera realizar transformaciones afines (rotación y dilatación) al asociador lineal. La ecuación de *Adaline* resulta

$$y_i(t) = \sum_{j=1}^n w_{ij} x_j - \theta_i, \forall i, 1 \leq i \leq m \quad (4.87)$$

La principal diferencia con *Perceptron* y el asociador lineal, es la regla de aprendizaje que emplea, empleando la *regla de Widrow–Hoff*, o conocida como *regla de Mínimos Cuadrados Medios (LMS – Least Mean Squares)*, que conduce a actualizaciones de tipo continuo.

Este modelo se viene empleando desde la década de los setenta del siglo pasado como filtro adaptativo, como por ejemplo, y como ya se ha dicho, para cancelar ruido en la transmisión de señales, empleándose en cantidad importante de módems. Su limitación es que se trata de un sistema lineal, por lo que separará correctamente patrones linealmente independientes, fallando en ocasiones ante patrones separables, no obstante, ante patrones no separables linealmente los resultados son incluso mejores a los de *Perceptron*, quedando patente en (Widrow and Lehr, 1990; Hertz *et al.*, 1991; Gallant, 1993).



Regla LMS

Esta regla, en su caso particular es conocida como *regla delta*, constituye el algoritmo de aprendizaje asociado a *Adaline*. La *regla LMS* conduce a la asociación perfecta cuando los pares de los patrones  $e-s$  sean linealmente independientes, y cuando no lo sean, proporcionando una matriz de pesos óptima desde el punto de vista de mínimos cuadrados. Así, la *regla delta* puede considerarse una versión iterativa aproximada de la basada en la *pseudoinversa*, para el caso de vectores estocásticos. A continuación se expondrá la forma de derivar la *regla LMS* a partir de la optimización de la función coste, que es la forma habitual de obtener algoritmos de aprendizaje.

Se propone una función de error o coste, que mide el rendimiento de la red, dependiendo dicha función de los pesos sinápticos, el procedimiento de optimización tratará de dar una configuración de pesos que corresponda a un mínimo de la función propuesta. Al optimizar la función, se proporcionará una regla de actualización de pesos, que en función de los patrones de aprendizaje modifique iterativamente los pesos hasta alcanzar el punto óptimo de la red.

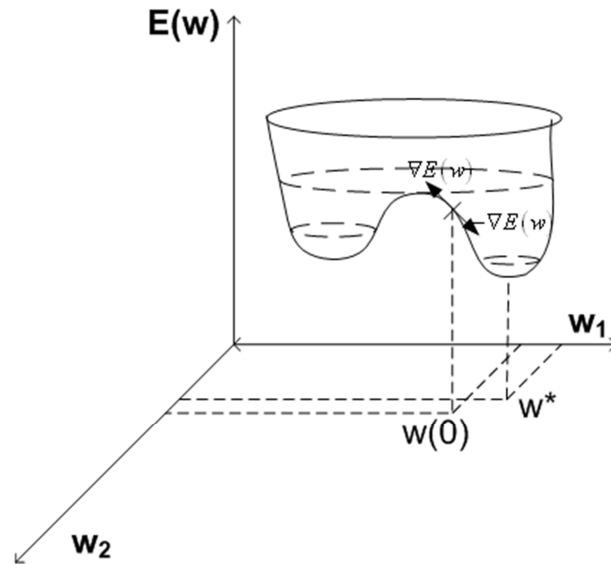
El método de optimización más empujado es el de *descenso por el gradiente*, donde se precisa definir una función coste  $E(\cdot)$ , que proporcione el error actual  $E$  que comete la red neuronal, que será función del conjunto de pesos sinápticos  $W$ ,  $E = E(W)$ ,  $E : \mathfrak{R}^n \rightarrow \mathfrak{R}$ , pudiendo representar esta función como una hipersuperficie, como vemos en la Figura 4.24, donde el valle es una configuración de pesos localmente óptima, al ser un mínimo de la función error. El aprendizaje deberá tratar de conseguir los pesos que correspondan con el mínimo global de la función de error. Para ello, se parte en el instante 0 de cierta configuración, dada en la figura por  $w(0)$ , el sentido del máximo gradiente apuntará hacia una colina de  $E(\cdot)$ , posteriormente, se modifican los parámetros  $W$  en sentido contrario a la dirección del gradiente.

El proceso anterior se itera hasta alcanzar un mínimo, que matemáticamente se expresa por  $W(t+1) = W(t) - \varepsilon \cdot \nabla E(W)$ , donde  $\varepsilon$  es el tamaño del peso en cada iteración, que debería ser infinitesimal.

A continuación se aplicará la regla al caso de *Adaline*, siendo la respuesta de la neurona de este tipo de modelo lineal y similar a (4.87), con salidas continuas, por ejemplo  $[0,+1]$ , si se define

$w_{i0} \equiv \theta_i$  y  $x_0 \equiv -1$ , reescribiendo la ecuación (4.87) por  $y_i = \sum_{j=0}^n w_{ij} x_j$ , con lo que se puede tratar el umbral como peso adicional, pero con entrada constante -1.

La *regla de Widrow-Hoff* para la aproximación estocástica, tras su desarrollo, resulta en una actualización de pesos dado por  $w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij} = w_{ij}(t) + \alpha (t_i^{\mu} - y_i^{\mu}) x_j^{\mu}$ . Es la *regla LMS*, y es la versión iterativa de la *pseudoinversa* para vectores estocásticos.



**Figura 4.24. Superficie de error en el espacio de pesos y descenso por gradiente hacia mínimo local. Fuente Propia.**

Si se emplea la función sigmoidea en lugar de la lineal, obtenemos la *regla delta*, mostrada en (Widrow and Lehr, 1990). La actualización de los pesos se aplica de la misma forma a los umbrales.

Widrow y Hoff demostraron la convergencia del algoritmo a cambio de que  $\alpha(t)$  cumpla

$$\sum_{t=1}^{\infty} \alpha(t) = \infty, \quad \sum_{t=1}^{\infty} \alpha^2(t) < \infty \tag{4.88}$$

siendo suficiente que se cumpla  $0 < \alpha < 1$ , con lo que el aprendizaje no se hará ni demasiado despacio ni demasiado deprisa. Grossberg (1982) planteó el *dilema de la plasticidad* frente a la *estabilidad*, que básicamente dice que el sistema debe aprender de manera suficientemente estable para recordar los patrones antiguos, pero suficientemente plástico como para aprender los nuevos.

Una fórmula más simple se obtiene cuando no se tiene en cuenta de manera explícita la estocasticidad del problema, para lo que se plantea la siguiente función de error

$$E[w_{ij}] = \frac{1}{2} \sum_{\mu=1}^p \sum_{i=1}^n (t_i^{\mu} - y_i^{\mu})^2 \tag{4.89}$$

El anterior error proporciona el error cuadrático medio, emplearemos de nuevo el descenso por el gradiente para la optimización, para obtener el siguiente incremento de pesos

$$\Delta w_{ij} = -\epsilon \frac{\partial E|_{w_{ij}}}{\partial w_{ij}} = \epsilon \sum_{\mu=1}^p (t_i^{\mu} - y_i^{\mu}) x_j^{\mu} \tag{4.90}$$

La expresión (4.90), vuelve a ser la *regla LMS*, donde debemos destacar que mientras en la *regla Perceptron* se actualizan discretamente los pesos, en *Adaline* se producen actualizaciones continuas. Otra diferencia, es que en *Perceptron* se converge en un número finito de iteraciones, mientras que en *Adaline* se acerca asintóticamente a la solución.

Dada la linealidad de la neurona *Adaline*, la función de error define una superficie paraboloide, que normalmente tiene un único mínimo, aunque a veces presenta más de uno, pero con la misma profundidad. En cualquiera de los casos, la función (4.90) lleva directamente al mínimo, ya que siempre descenderá por la superficie de error., como se demuestra en (Hecht–Nielsen, 1990), por ello, la *regla LMS* obtiene siempre el mínimo global, sin importar los pesos de partida, siendo una de las pocas *ANN* en las que se puede hacer dicha afirmación.

La versión multicapa de *Adaline* se conoce como *Many Adalines (Madaline)*, introducida en (Widrow and Winter, 1988; Freeman and Skapura, 1992) para solventar las carencias de *Adaline* monocapa.

#### 4.5.5.5. El perceptron multicapa

El *MLP* resulta de añadir más de una capa al *Perceptron*, donde el algoritmo empleado para su aprendizaje es el *backpropagation* o alguna de sus variantes, es por esto, que a veces a la arquitectura *MLP* se le conoce como *red backpropagation*.

La estructura y función de transferencia (sigmoidea) empleada en *MLP* se representa en la Figura 4.25, llamando a las entradas  $x_i$ , a las salidas de la capa oculta  $y_i$ , las de la capa final  $z_k$  y las salidas objetivo de la red  $t_k$ .

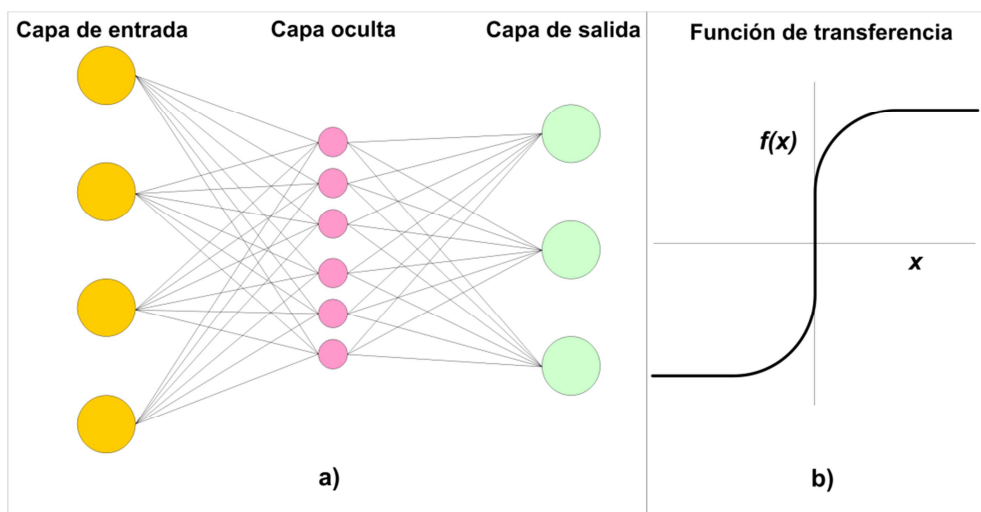


Figura 4.25. a) *MLP*; b) Función de transferencia de la neurona. Fuente Propia.

En la Figura 4.26 se muestra la arquitectura *MLP*, donde vemos las anteriores señales y, además,  $w_{ij}$  son los pesos de la capa oculta,  $\theta_j$  sus umbrales,  $w'_{ij}$  los pesos de la capa de salida y  $\theta'_j$  sus umbrales. Con una capa oculta y neuronas de salida con función lineal, *MLP* se puede expresar matemáticamente

$$z_k = \sum w'_{kj} f \left( \sum_i w_{ji} x_i - \theta_j \right) - \theta'_i \quad (4.91)$$

siendo  $f(\cdot)$  del tipo sigmoideo.

En (Hecht–Nielsen, 1987) se aplica el teorema de Kolmogorov para demostrar que una arquitectura similar en características a *MLP*, con tan sólo una capa oculta, resulta ser un aproximador universal de funciones. Posteriormente, en (Funahashi, 1989; Hornik *et al.*, 1989), se demostraron varios teoremas que proponían a un *MLP* con una capa oculta como aproximador universal de funciones.

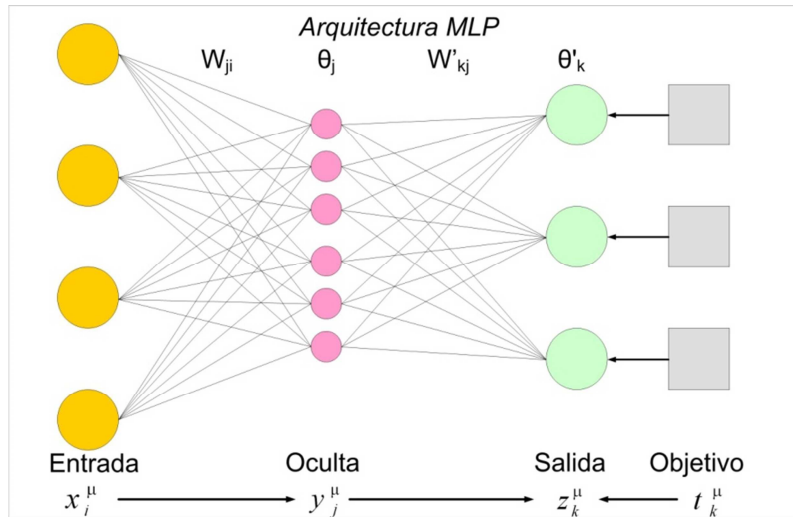


Figura 4.26. Arquitectura *MLP*. Fuente Propia.

### Aprendizaje por backpropagation

*Backpropagation* aparece como consecuencia natural de la *regla LMS*, ya que es preciso extender ésta a las redes multicapa. Se planteará por tanto una función de error similar a (4.89), pero además de derivarla para la función de pesos de salida, se hará para la de pesos de las neuronas ocultas, empleando la regla de la cadena, por lo que las funciones de transferencia de las neuronas deberán ser derivables.

Nos basaremos en la arquitectura de la Figura 4.26, donde dado un patrón de entrada  $\mathbf{x}^\mu$ , con  $\mu = 1, \dots, p$ , la operación global de la *MLP* será

$$z_k^\mu = g \left( \sum_j w'_{kj} f \left( \sum_i w_{ji} x_i^\mu - \theta_j \right) - \theta'_k \right) \quad (4.92)$$

$f(\cdot)$  es la función de activación de las neuronas de la capa oculta y  $g(\cdot)$  de las neuronas de la capa de salida. Ambas pueden ser sigmoideas, pero las neuronas de salida se las suele considerar lineal. El error cuadrático medio se empleará como función de coste

$$E[w_{ji}, \theta_j, w'_{kj}, \theta'_k] = \frac{1}{2} \sum_\mu \sum_k \left[ t_k^\mu - g \left( \sum_j w'_{kj} y_j^\mu - \theta'_k \right) \right]^2 \quad (4.93)$$

se emplea el descenso del gradiente para la minimización de (4.93), habiendo dos gradientes, uno para los pesos de la capa de salida y otro los pesos de la capa oculta

$$\delta w'_{kj} = -\varepsilon \frac{\partial E}{\partial w'_{kj}}; \quad \delta w_{ji} = -\varepsilon \frac{\partial E}{\partial w_{ji}} \quad (4.94)$$

Si se deriva (4.94) se obtiene las actualizaciones de los pesos de ambas capas y, como se ha dicho, empleando la regla de la cadena

$$\delta w'_{kj} = \varepsilon \sum_{\mu} \Delta_k^{\mu} y_j^{\mu}; \quad \text{con } \Delta_k^{\mu} = [t_k^{\mu} - g(h_k^{\mu})] \frac{\partial g(h_k^{\mu})}{\partial h_k^{\mu}} \quad \text{y } h_k^{\mu} = \sum w'_{kj} y_j^{\mu} - \theta'_k \quad (4.95)$$

$$\delta w_{ji} = \varepsilon \sum_{\mu} \Delta_j^{\mu} x_i^{\mu}; \quad \text{con } \Delta_j^{\mu} = \left[ \sum_k \Delta_k^{\mu} w'_{kj} \right] \frac{\partial f(h_j^{\mu})}{\partial h_j^{\mu}} \quad \text{y } h_j^{\mu} = \sum w_{ji} x_i^{\mu} - \theta_j \quad (4.96)$$

los potenciales postsinápticos vienen dados por  $h_k^{\mu}$  y  $h_j^{\mu}$ , las actualizaciones de los umbrales se hacen empleando las funciones (4.95) y (4.96), considerado como caso particular del peso sináptico, con entrada igual a -1.

A continuación se indican los pasos de la *regla backpropagation*:

1. Establecer pesos y umbrales aleatorios.
2. Para cada patrón  $\mu$  del *CE*: obtener la respuesta de la red de dicho patrón según (4.92); calcular las señales de error ( $\Delta_k^{\mu}$  y  $\Delta_j^{\mu}$ ) según (4.95) y (4.96); y calcular el incremento de los umbrales y pesos para dicho patrón según (4.95) y (4.96).
3. Obtener el incremento total (para todos los patrones) actual de los pesos  $\delta w'_{kj}$  y  $\delta w_{ji}$  según (4.95) y (4.96); hacer lo mismo para umbrales.
4. Actualizar pesos y umbrales.
5. Obtener el error actual (4.93),  $t := t + 1$  y volver al paso 2 si aún no es satisfactorio.

El procedimiento anterior se realiza para todos y cada uno de los patrones del *CE*, se calcula la variación en los pesos que origina cada uno de los patrones, acumulándose, y es entonces cuando se actualizan los pesos. Este aprendizaje, según la manera explicada anteriormente, se conoce como *aprendizaje por lotes (batch)*; otra opción es actualizar los pesos sinápticos tras la presentación de cada patrón, conociéndose entonces como *aprendizaje en serie (on line)*. Wilson and Martínez (2003) han demostrado que el *aprendizaje en serie* estima mejor el gradiente, permite cadencias de entrenamiento superiores y generalmente, es más rápido. En este aprendizaje, se deben presentar los patrones de manera aleatoria, para evitar que el entrenamiento se vea influenciado por el último patrón, por lo que la actualización de pesos estaría influenciada siempre por el mismo patrón, predominando su influencia sobre la del resto de patrones. Esta aleatoriedad, además, permite escapar de mínimos locales y obteniendo mínimos más profundos dentro de la función de superficie, tal y como presenta Bishop (1994).

El algoritmo *backpropagation* es un método de gran generalidad, permitiendo su aplicación a multitud de problemas sin destinar gran esfuerzo al tiempo de desarrollo. Un inconveniente es su

lentitud de convergencia, aunque debe de tenerse en cuenta que no requiere tanto coste computacional como si se obtuvieran los pesos mediante la evaluación directa de sus derivadas. Por otra parte, es necesario tener presente que *backpropagation* puede detenerse en un mínimo local, sin obtener el mínimo global pretendido.

Se han presentado variantes de *backpropagation* para entre otras cosas tratar de solucionar el problema de la convergencia lenta. Rumelhart *et al.* (1986) incluyeron en el algoritmo el *término de inercia (momentum)*, que incluye en la variación de los pesos un término adicional que es proporcional al incremento durante la iteración anterior, tratando de introducir en el método una inercia relativa

$$\delta w'_{kj}(t+1) = -\varepsilon \left. \frac{\partial E}{\partial w'_{kj}} \right|_t + \alpha \cdot \delta w'_{kj}(t-1); \quad \delta w_{ji}(t+1) = -\varepsilon \left. \frac{\partial E}{\partial w_{ji}} \right|_t + \alpha \cdot \delta w_{ji}(t-1) \quad (4.97)$$

donde  $\alpha$  es el *momentum*, que varía entre 0 y 1, de forma que si los incrementos de un peso tienen siempre el mismo signo, las actualizaciones serán mayores para cada iteración; pero si los incrementos oscilan, el incremento se reduce al cancelarse. Dicho de otra forma, se incrementa el ritmo de aprendizaje en las direcciones de interés.

La elección de los pesos iniciales también tiene importancia en el tiempo empleado en el entrenamiento, tal y como se demostró en (Thimm and Fiesler, 1995; LeCum *et al.*, 1998).

Otras técnicas para acelerar el entrenamiento se centran en el preprocesamiento de los datos de entrada. Un grupo de estos métodos se dedican a asignar tasas de aprendizaje diferentes a cada peso, tasas que además son adaptativas, la variante más conocida es *SuperSAB* que se expone en (Tollenaere, 1995). Un segundo grupo realiza el descenso por el gradiente empleando información procedente del ritmo de cambio de la pendiente, algunos ejemplos son los algoritmos de *gradientes conjugados*, *gradientes conjugados escalados*, *Levenberg–Marquardt* y *Newton*; siendo todas técnicas más robustas que *backpropagation* que pueden acelerar en varios órdenes de magnitud la convergencia del aprendizaje. Un algoritmo que se encuentra entre los dos grupos es el de *Quickprop*, que introduce información de la forma de las derivadas segundas sin derivar.

Para ampliar conocimientos en torno a los algoritmos mencionados, una buena fuente de información puede encontrarse en la obra de Beale *et al.* (2010) del entorno matemático MATLAB®, donde se destaca el algoritmo *Levenberg–Marquardt*, así como los derivados a partir de dicho algoritmo, como puede ser el algoritmo Regulación Bayesiana (*Bayesian Regularization*).

#### 4.5.5.6. Capacidad de generalización de la red

El problema de generalización se ha tratado en la sección 4.3.7.1, mientras que la VC se ha introducido en las secciones 4.3.4.2 y 4.3.7.2; a continuación se extenderán estos conceptos ya que son fundamentales para la ANN.

La *generalización* se enfrenta a la *memorización*, mientras que generalizar es la capacidad de la red para hacer frente a patrones no empleados y dar una respuesta óptima, memorizar consiste en dar esa respuesta pero tan sólo a patrones conocidos. A continuación se tratarán los aspectos que ayudarán a conseguir una mayor generalización, siendo: *VC* (*cross-validation*); número de patrones del entrenamiento; y tamaño de la arquitectura de la *ANN*.

### *Cross-validation*

Como ya se expuso, en un proceso de aprendizaje debe de tenerse en cuenta el *error de aprendizaje*, que depende del *CE*, pero además, existe el *error de generalización*, que se mide empleando patrones diferentes a los empleados en el entrenamiento. Por tanto, es posible entrenar una red con un *CE*, y comprobar su eficacia real, o error de generalización, mediante unos datos no conocidos, llamado conjunto de test (*CT*). Normalmente, el *CT* es una parte que se excluye del total del *CE*,  $CT \subset CE$ .

En la Figura 4.8 quedaba de manifiesto la importancia de la generalización, ya que se apreciaba como el modelo seguía ajustando con el *CE*, mientras que llegaba un punto, que pese a la disminución del anterior error, el modelo comenzaría a ajustar peor para patrones no conocidos. Para tratar de conseguir la generalización, *cross-validation* entrena y valida de manera simultánea, empleándose en la fase de aprendizaje de los modelos neuronales supervisados. En la práctica, es posible que se presenten varios mínimos seguidos dentro de *CT*, debiendo detener el entrenamiento en el punto más óptimo de mínimo de error de generalización, y no quedarnos en el primer mínimo que aparezca, por lo que habrá que guardar las configuraciones intermedias de los pesos, para entregar el que obtuvo el error mínimo, esto es conocido como *parada temprana* (*early stopping*) y algunas técnicas se muestran en (Prechelt, 1998).

En las herramientas lineales la complejidad del modelo viene dada por el número de parámetros libres a ajustar (coeficientes), mientras que en *ANN* la complejidad del modelo no depende tanto del número de parámetros como de su valor actual, como se muestra en (Principe *et al.*, 2000). Una *ANN* tiene tal capacidad de modelado, que puede aprender casi cualquier cosa, y si además, le unimos su faceta de poderosos estimadores no lineales, capaces de modelar situaciones extremadamente complejas, las hace indiscutiblemente útiles y aplicables.

Por tanto, y como se ha discutido, *cross-validation* se considera fundamental para el aprendizaje de *ANN*, debiendo disponer de tres conjuntos a partir de *CE*: el verdadero conjunto de patrones empleado para la fase de entrenamiento; el *CV* que se empleará para la validación cruzada; y el *CT* que determina de forma objetiva el rendimiento final de la red. En (Wilson and Martínez, 2003) se presenta la división óptima de cada conjunto anteriormente citado, junto a detalles de *cross-validation*.

### *Número de patrones de entrenamiento*

Según Haykyn (1999), la capacidad de generalización de la red depende en gran medida de la arquitectura de la red, número de patrones de entrenamiento y la complejidad del problema. Los tres están estrechamente relacionados, cuanto más complejo sea el problema, se precisará un tamaño de red mayor y, por tanto, mayor número de patrones en la fase de aprendizaje se requerirán.

Baum and Haussler (1989) demostraron que para un red  $N$  nodos de entrada, un total de pesos  $N$  y un error de generalización que cumpla  $0 < \varepsilon \leq 1/8$ , requerirá un número de patrones dado por  $m \geq O\left(\frac{W}{\varepsilon} \cdot \log\left(\frac{N}{\varepsilon}\right)\right)$ .

#### Tamaño de la arquitectura de la ANN

En (Bishop, 1994) y en (Haykin, 1999), se introdujo el concepto de *maldición de la dimensionalidad* (*curse of dimensionality*) que viene a decir que el número de patrones para hacer *mapping*, crece exponencialmente con la dimensión del espacio de entrada, por lo que redes con un espacio de entrada elevado, requerirán de gran cantidad de patrones para su entrenamiento. Una forma de proceder consistirá en limitar el número de entradas a la ANN, ya que implicará la disminución del número de pesos sinápticos necesarios en la arquitectura.

Otras técnicas de la reducción del número de parámetros de la red consiste en:

- Compartir los pesos entre neuronas, técnica llamada *compartir pesos* (*weight sharing*), explicada en (Hecht–Nielsen, 1990).
- Entrenar la red hasta determinado nivel, eliminando los pesos que no aportan a la operación, técnica conocida como *podado* (*pruning*) de la red, analizada en (Haykyn, 1999).
- Si durante el aprendizaje se deja tender a los pesos a cero, de manera lenta, para que los no actualizados se anulen, estaremos ante una técnica variante de *podado* conocida por *decaimiento de pesos* (*weight decay*), explicada en (Principe *et al.*, 2000).

#### **4.5.6. Redes auto–organizadas**

Los modelos no supervisados o auto–organizadas se diferencian de los supervisados en que éstos emplean en el aprendizaje parejas de patrones de entrada y objetivo del modelo, mientras que en los no supervisados no se emplearán las salidas objetivo que se desean asociar a los patrones de entrada. La red tratará de detectar la similitud existente entre los patrones de entrada, mediante un proceso de auto-organización.

Tras una breve introducción y visión de los modelos no supervisados, la atención se centrada en uno de los modelos más conocidos, y empleado en esta *Tesis Doctoral*, el modelo SOM, del cual se presentarán sus principales características, un análisis formal del mismo y ciertas nociones de la medida de similitud, que es la base de las asociaciones que emplea el modelo.



#### 4.5.6.1. Los modelos no supervisados

El modelo deberá detectar regularidades, correlaciones y rasgos comunes entre el *CE*, para tratar de incorporarlos a la estructura interna del modelo, por medio de los pesos sinápticos de las conexiones existentes. Es por esto que se dice que las neuronas se auto-organizan en base a los estímulos que percibe del exterior, en forma de datos de entrada.

Entre los principales tipos de procesamiento que los modelos no supervisados pueden realizar podemos encontrar:

- Mapa de rasgos: en estos modelos las neuronas se organizan geoméricamente, proyectando el espacio sensorial de entrada sobre la propia red. Esta proyección preserva la topología del espacio original.
- Similitud entre patrones: aparece cuando una neurona cuya salida es continua marca el grado de parecido entre un patrón presentado y la media del resto.
- Análisis de componentes principales: extensión del caso anterior pero a varias neuronas.
- Clustering: el modelo estará compuesto por neuronas de salida discreta, donde cada una de ellas representa una categoría y donde tan sólo una puede permanecer activada a la vez al presentar un patrón.
- Memoria asociativa: caso particular de la anterior, donde una neurona además de activarse, proporcionará el prototipo de la clase en cuestión.
- Codificación: similar anterior pero la neurona entrega una versión de codificación o etiqueta, normalmente con menos bits pero con la información relevante en la misma.

Los modelos no supervisados tienen su campo de aplicación propio, pero en ocasiones pueden sustituir a los supervisados o colaborar de forma híbrida con ellos. Si empleamos un modelo supervisado con *backpropagation*, pero su aprendizaje es excesivamente lento, podría resultar útil emplear un *modelo multicapa híbrido*, con una capa no supervisada y otra supervisada, ya que al romper el entrenamiento en dos fases se gana velocidad, como se muestra en (Hrycej, 1992).

Como norma general, los modelos no supervisados suelen ser monocapa, simples y basados en algoritmos rápidos y sencillos. Los dos grandes grupos de clasificación las redes no supervisadas son en *redes no supervisadas hebbianas* y *redes no supervisadas competitivas*.

Las *redes no supervisadas hebbianas* emplean el algoritmo de *Hebb*, en ellas, un alto número de neuronas de salida pueden ser activadas frente a un mismo patrón de entrada. La red Análisis de Componentes Principales (*Principal Component Analysis – PCA*), presentado en (Oja, 1982), emplean como regla de aprendizaje directamente la de *Hebb*. Otros modelos se derivan de ciertas propiedades a optimizar, véase el ejemplo de maximizar el contenido de la información de Linsker (1988).

Las *redes no supervisadas competitivas* son aquellas en las que tan sólo una neurona o grupo que comparte una cierta vecindad se activan. En estos modelos se produce una operación entre las neuronas, inhibiéndose o activándose de manera lateral, al final de la competición, la neurona más activada inhibirá a las demás, por lo que será la única que permanecerá activada. Por este motivo, este tipo de modelos se les conoce como *redes el ganador se lo lleva todo (winner-take-all)*. En estos modelos, en el proceso de la fase de aprendizaje, las neuronas vencedoras obtendrán como premio el reforzamiento de sus conexiones sinápticas, siendo esta competición un rasgo común en los modelos más conocidos, como *Neocognitron* de Fukushima (1980) o el modelo *SOM* de Kohonen (1982a; 1982b; 1989). La idea común de ellos es el agrupamiento de patrones, aunque pueden emplearse para la cuantificación vectorial, ajuste funcional, análisis estadístico y procesamiento de imagen, entre otros.

#### 4.5.6.2. El modelo de mapas auto-organizados

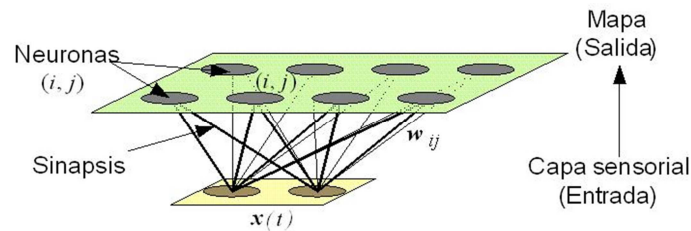
*SOM* es uno de los sistemas neuronales más conocidos y empleados, encontrándose encasillado dentro de las arquitecturas con operación no supervisada. Introducido por Kohonen, quien observó que en muchas regiones del córtex de los animales superiores existen zonas donde las neuronas que detectan de rasgos, se distribuyen topológicamente ordenadas, tal y como mostró en (Kohonen, 1990). En resumen, la representación de la información en la corteza cerebral aparece a menudo organizada espacialmente, circunstancia que el modelo neuronal *SOM* trata de reproducir, como se presentó en (Kohonen, 1989; 1990).

##### Funcionamiento del modelo

Como se ve en la Figura 4.27, en este modelo las neuronas se organizan en una arquitectura unidimensional de dos capas, la primera es la capa sensorial o de entrada, que consiste en  $m$  neuronas, una por cada variable de entrada, a modo de *buffers*, cuyo cometido es distribuir la información procedente del espacio de entrada a las neuronas de la segunda capa. La segunda capa forma un mapa de rasgos, y es donde se produce el procesamiento, siendo la capa normalmente rectangular (puede disponerse otra forma) de  $n_x \times n_y$  neuronas operando en paralelo. Se etiquetan las  $m$  neuronas de entrada con el índice  $k$  ( $1 \leq k \leq m$ ), y las  $n_x \times n_y$  neuronas del mapa con un par de índices  $i \equiv (i, j)$  ( $1 \leq i \leq n_x$ ,  $1 \leq j \leq n_y$ ), determinando de esta manera su localización espacial. Cada una de las neuronas de entrada ( $k$ ) estará conectada con la totalidad de las neuronas ( $i, j$ ) de la segunda capa mediante un peso sináptico ( $w_{ij}$ ).

En la operación normal de la red o *fase de ejecución*, los pesos permanecen fijos, cada neurona ( $i, j$ ) calcula la similitud entre el vector de entrada  $\mathbf{x}$ ,  $\{x_k / 1 \leq k \leq m\}$  y su propio vector de pesos sinápticos  $\mathbf{w}_{ij}$ , según una cierta medida de distancia. Posteriormente, se declara

vencedora la neurona cuyo vector de pesos se más parecido al de entradas, de esta manera, cada neurona se comporta como un detector de rangos específicos, y la neurona ganadora nos indica el tipo de rasgo o patrón detectado en el vector de entradas.



**Figura 4.27. Arquitectura SOM. Fuente Propia.**

En la *fase de aprendizaje*, cada neurona del mapa se identifica con diferentes rasgos del espacio de entrada. Una vez presentado y procesado de un vector de entradas  $x(t)$ , la neurona vencedora modifica sus pesos de manera que incorporen un poco más la información de similitud con  $x(t)$ . El proceso se repite para multitud de vectores de entrada, de forma que al final los diferentes vectores de referencia, se identifican con dominios específicos de las variables de entrada, llamados dominios de Voronoi, como presentó Kohonen (1988b).

Lo descrito hasta ahora responde a un esquema competitivo clásico, en el que cada neurona actúa de manera solitaria, sin embargo, el modelo de Kohonen aporta una importante novedad, ya que incorpora a este esquema relaciones entre las neuronas próximas en el mapa, por medio de la *función de vecindad*, la cual define un entorno alrededor de la neurona ganadora actual (vecindad), siendo su efecto que durante el aprendizaje, los pesos son actualizados tanto los de la neurona vencedora como los de las que pertenecen a su entorno. La *función de vecindad*, representa matemáticamente el efecto global de las interacciones laterales entre las neuronas del cerebro, modelamos esta situación mediante una sencilla función que define el tamaño de la vecindad en torno a la vencedora, dentro de la cual todas las neuronas son premiadas actualizando sus pesos, y fuera de ellas son castigadas al no hacerlo o hacerlo en sentido contrario.

#### Algoritmo de aprendizaje

La novedad de *SOM*, como ya se ha comentado, es que la modificación de los pesos afecta a varias neuronas, no sólo a la ganadora, estando aquéllas en su vecindad. Con el transcurso de las iteraciones, la función de vecindad irá reduciéndose, hasta que finalmente sólo se modifiquen los pesos de la neurona ganadora. Al comienzo, la función de vecindad abarca una amplia zona del mapa, permitiendo una ordenación global de los pesos sinápticos. Por tanto, la fase de aprendizaje comprende dos procesos: el primero consiste en una ordenación global para desplegar el mapa; y el segundo será un ajuste fino para especializar las neuronas.

A continuación se describirá brevemente el funcionamiento genérico del algoritmo de aprendizaje para *SOM* empleado en (Martín del Brío *et al.*, 1993):

1. Inicialización de los pesos sinápticos  $w_{ijk}$ , pudiéndose partir de pesos nulos, aleatorios que es lo más habitual o predeterminados.
2. Presentar un patrón en cada iteración, tomado de acuerdo con la función de distribución del espacio sensorial de entrada.
3. Cada neurona  $i \equiv (i, j)$ , y de forma paralela, calcula la similitud entre su vector de pesos sinápticos  $w_{ij}$  y el actual vector de entradas  $x$ . Un criterio de medida de similitud empleado es la distancia euclídea

$$d^2(w_{ij}, x) = \sum_{k=1}^n (w_{ijk} - x_k)^2 \quad (4.98)$$

4. Cálculo de la neurona ganadora  $g \equiv (g_1, g_2)$ , cuya distancia sea la menor de todas.
5. Se actualizan de los pesos sinápticos de la neurona ganadora y los de sus neuronas vecinas, siendo la regla más empleada para esta tarea la siguiente

$$w_{ijk}(t+1) = w_{ijk}(t) + \alpha(t) \cdot h(|i - g|, t) \cdot (x_k(t) - w_{ijk}(t)) \quad (4.99)$$

donde  $\alpha(t)$  es el parámetro llamado tasa de aprendizaje, la función  $h(\cdot)$  la función de vecindad, la cual valdrá 0 cuando la neurona  $i$  no pertenezca a la vecindad de  $g$ , y un número positivo cuando sí pertenezca. Tanto  $\alpha$  como el radio de vecindad (parámetro que se verá más adelante), disminuyen con  $t$ .

6. El proceso finaliza una vez alcanzado el número máximo de iteraciones, en caso contrario, se vuelve al paso 2.

Posteriormente, y en una segunda fase, se realizará un ajuste fino del mapa, de modo que la distribución de los pesos sinápticos ajuste mejor a los patrones presentados. El proceso es similar al descrito anteriormente, tomando  $\alpha(t)$  constante y con un valor pequeño (0.01 por ejemplo) y un radio de vecindad constante e igual a 1.

La tasa de aprendizaje  $\alpha(t)$ , es una función monótona decreciente con el tiempo. Para su actualización, se emplea habitualmente una función lineal

$$\alpha(t) = \alpha_0 + (\alpha_f - \alpha_0) \frac{t}{t_\alpha} \quad (4.100)$$

donde  $\alpha_0$  es la tasa de aprendizaje inicial ( $< 1.0$ ) y  $t_\alpha$  el máximo número de iteraciones hasta llegar a  $\alpha_f$ . Como alternativa es emplear una función que decrezca exponencialmente, aunque el empleo de una u otra no suele influir demasiado en el resultado final

$$\alpha(t) = \alpha_0 \left( \frac{\alpha_f}{\alpha_0} \right)^{t/t_\alpha} \quad (4.101)$$

La función de vecindad  $h(|\mathbf{i} - \mathbf{g}|, t)$ , define en cada iteración  $t$ , si una neurona  $\mathbf{i}$  pertenece o no a la vecindad de la ganadora  $\mathbf{g}$ , siendo la vecindad simétrica y centrada en  $\mathbf{g}$ , debido a esto se representa como uno de los argumentos la distancia entre la neurona genérica  $\mathbf{i} \equiv (i, j)$  y la vencedora  $\mathbf{g} \equiv (g_1, g_2)$ , ya que se cumple

$$|\mathbf{i} - \mathbf{g}| = \sqrt{(i - g_1)^2 + (j - g_2)^2} \quad (4.102)$$

Normalmente,  $h(\cdot)$  decrece con la diferencia respecto a la neurona vencedora, y depende del radio de vecindad  $R(t)$ , que viene a representar el tamaño de la vecindad actual. Como funciones de vecindad existen la escalón o rectangular, sombrero mejicano, pipa y *gaussiana*. En (Lo and Bavarian, 1991) se demuestra que con una función de vecindad decreciente se asegura un más rápido ordenamiento en el mapa y, además, la preservación del orden en cada iteración. El valor de  $R(t)$  debe decrecer más lentamente con una función tipo escalón, para así poder alcanzar mapas correctamente ordenados. En (Erwin *et al.*, 1992) se hace una distinción entre funciones de vecindad cóncavas y convexas, donde también se muestra que se logra una convergencia más rápida con las convexas.

La función de vecindad tiene una forma definida, pero su radio de vecindad  $R(t)$  varía con el tiempo, se parte de un valor inicial  $R_0$  grande para determinar vecindades amplias, con la finalidad de la ordenación global del mapa, después,  $R(t)$  disminuye monótonamente con el tiempo, hasta alcanzar un valor final de  $R_f = 1$ , para solamente actualizar la neurona ganadora y las adyacentes, siendo una posible función de actualización (linealmente decreciente) de  $R(t)$

$$R(t) = R_0 + (R_f - R_0) \frac{t}{t_R} \quad (4.103)$$

donde  $t$  corresponde a la iteración y  $t_R$  es el número de iteraciones para alcanzar  $R_f$ .

Otras recomendaciones adicionales, serían las que aporta el grupo de investigación dirigido por Kohonen, en la Universidad de Tecnología de Helsinki, como por ejemplo el prestar atención a la dimensión  $nx$  y  $ny$  del mapa, ya que ocasiones un mapa cuadrado, con  $nx = ny$ , puede causar inestabilidades en el desarrollo, por lo que se recomiendan mapas rectangulares.

#### Variantes de SOM

Una variación de *SOM*, consiste en comenzar el aprendizaje con un mapa de tamaño pequeño, e ir introduciendo posteriormente nuevas neuronas al mismo durante la fase de entrenamiento, y cuyos pesos serán la interpolación de sus neuronas vecinas ya presentes en el mapa. Algunas propuesta del esquema anterior, conocidos como *mapas auto-organizados crecientes*, se han propuesto en (Rodrigues and Almeida, 1990; Alahakoon *et al.*, 2000).

Otra aspecto a tratar tiene que ver con el efecto de los bordes del mapa, debido a que las neuronas de las esquinas del mapa poseen menos vecinas, presentando cierta asimetría respecto a las demás, una manera de resolver este problema consiste en establecer un ritmo de aprendizaje mayor para las neuronas de los bordes, tal y como se presenta en (Rodrigues and Almeida, 1990). Otra opción es cerrar la red, que consiste en considerar que una neurona del extremo del mapa es vecina de la neurona correspondiente del extremo opuesto.

Una variación pretende que todas las neuronas ganen con una frecuencia parecida, y de esta forma, no dejar que ningún grupo de neuronas acapare el proceso y no se desarrolle el mapa correctamente. Para alcanzar la equiprobabilidad en el proceso *winner-take-all*, se introduce una modificación en el algoritmo, conocido como *mecanismo de consistencia* y presentado en (DeSieno, 1988), que consiste en contabilizar durante el aprendizaje el número de veces que cada una de las neuronas gana, y penalizar las que ganan en exceso.

Otras variantes se presentan en (Kangas, 1994), como el conocido *hipermapa*, que opera sobre los propios patrones, permitiendo obtener secuencias temporales. En el mismo trabajo se presenta la asociación de cada neurona con cierto operador, en esta misma línea, Kohonen (1996) presentó el modelo Subespacio-Adaptable SOM (*Adaptive-Subspace SOM – ASSOM*). Otra variante propuesta por el propio Kohonen (1986) es *LVQ*, modelo supervisado que dispone de una capa simple de neuronas de Kohonen sin relaciones de vecindad, y se puede emplear como procedimiento supervisado de ajuste fino de *SOM*.

Resulta un error frecuente el emplear modelo *SOM* para labores de *clustering*, y como expone Kohonen (1990), su principal utilidad es para visualizar datos, no debiéndose emplear para *clustering*. Técnicas propias de *clustering*, como el algoritmo *k-means* entre otros, resultan más apropiados para esta labor, pudiéndose aprovechar el potencial de *SOM* de identificación y ordenación de rasgos, como paso previo al *clusterizado*.

#### 4.5.6.3. Neurona de Kohonen. Medida de similitud

El modelo de la neurona de Kohonen depende del cálculo de la similitud entre el vector de entrada y el de pesos. A continuación se detallarán los principales criterios de distancia, que refleja la similitud anteriormente mencionada.

##### Correlación o producto escalar

Dos vectores serán similares cuanto mayor sea su correlación, dada por la siguiente expresión

$$C_{ij} = \sum_{k=1}^n w_{ijk} x_k \quad (4.104)$$

sin embargo, la anterior medida es sensible a los tamaños de los vectores, la medida de la similitud se distorsionará si hay grandes diferencias en las longitudes de los vectores. Este problema se resuelve con el llamado criterio del coseno, que divide (4.104) por las normas de los vectores, tal y como vemos a continuación

$$\cos(w_{ij}, x) = \frac{\sum_{k=1}^n w_{ijk} x_k}{\|w_{ij}\| \cdot \|x\|} \quad (4.105)$$

al ser un ángulo, es independiente de los tamaños, ya que se basa en una característica relativa.

#### Distancia euclídea

En este caso, mayor similitud entre vectores, vendrá indicada por una distancia menor de la distancia euclídea, dada por la siguiente fórmula

$$d^2(w_{ij}, x) = \sum_{k=1}^n (w_{ijk} - x_k)^2 \quad (4.106)$$

#### Métrica de Minkowski

Es una generalización de la distancia euclídea, dada por la siguiente fórmula

$$d(w_{ij}, x) = \left( \sum_{k=1}^n |w_{ijk} - x_k|^\lambda \right)^{\frac{1}{\lambda}}, \quad \lambda \in \mathfrak{R} \quad (4.107)$$

coincidiendo con la distancia euclídea para  $\lambda = 2$ .

#### Distancia de Manhattan

El caso anterior, cuando se tiene  $\lambda = 2$ , se obtiene un criterio muy sencillo, ya que no precisa productos. Es conocido como distancia de Manhattan y viene expresado por

$$d(w_{ij}, x) = \sum_{k=1}^n |w_{ijk} - x_k| \quad (4.108)$$

#### **4.5.6.4. Modelos de aprendizaje en mapas auto-organizados**

Se plantearán diferentes reglas de aprendizaje para su aplicación en el modelo *SOM*. Se empleará la misma sistemática que en el caso de redes supervisadas, se propondrá una función error dependiente de los pesos de la red, y se obtendrá la regla de actualización por medio de la optimización mediante descenso por el gradiente.

La neurona ganadora  $g$  será la que cumpla, para una distancia genérica en el espacio de señales, la siguiente función

$$g = \arg \min_{ij} (d(w_{ij}, x)) \quad (4.109)$$

Para las redes no supervisadas, la función error no es tan trivial como para las supervisadas, ya que no existe un objetivo explícito al que deban tender las salidas de la red. Por tanto, el

problema radica en buscar una función de error, que aunque no sea una verdadera función potencial, permita obtener sistemáticamente reglas de aprendizaje. El objetivo de que los pesos se ajusten a las entradas presentadas, se puede obtener haciendo que dichos pesos tiendan a ellas, por tanto, se puede obtener una función objetivo global, según Kohonen (1993), de la siguiente forma

$$E = \int \sum_i h(|\mathbf{i} - \mathbf{g}|) f(d(\mathbf{x}, \mathbf{w}_i)) p(\mathbf{x}) d\mathbf{x} \quad (4.110)$$

donde  $p(\mathbf{x})$  es la función de distribución del espacio sensorial,  $h(\cdot)$  la función de vecindad y  $f(\cdot)$  la función de error de cuantificación. La función tiene cierta similitud a las planteadas en los modelos supervisados, pero no es una función potencial, ya que  $E$  depende de la neurona vencedora y, por lo tanto, de la entrada y pesos actuales (Kohonen *et al.*, 1991a). En los modelos supervisados, la función potencial posee cierta información global del entorno, que es recorrido durante el aprendizaje hasta encontrar un mínimo, mientras que en el modelo Kohonen, la función error cambia su visión del entorno con cada patrón, por tanto, como mucho se encontrará un punto cercano al óptimo, o dicho de otra forma, un óptimo aproximado (Kohonen *et al.*, 1991b).

Se definirá la siguiente función, para poder aplicar la aproximación estocástica y hacer frente a los retos comentados del modelo

$$E_1 = \sum_i h(|\mathbf{i} - \mathbf{g}|, t) f(d(\mathbf{x}(t), \mathbf{w}_i(t))) \quad (4.111)$$

siendo la función anterior una muestra de la función global para un instante determinado, donde la neurona ganadora será constante, por lo que el entorno no varía, y una solución empleando el descenso del gradiente será

$$\mathbf{w}_i(t+1) = \mathbf{w}_i(t) - \lambda(t) \nabla_{\mathbf{w}_i} E_1(t) \quad (4.112)$$

donde el ritmo o tasa de aprendizaje es  $\lambda(t)$ , que deberá cumplir las condiciones habituales

impuestas por:  $\sum_{t=0}^{\infty} \lambda(t) = \infty$  y  $\sum_{t=0}^{\infty} \lambda^2(t) < \infty$ .

Debe apuntarse que las soluciones obtenidas no son globalmente óptimas, ya que se estará consiguiendo soluciones locales al descender por gradientes locales en lugar de por la máxima pendiente que proporcionaría  $E$ .

#### Regla aprendizaje euclídea

Si se considera emplear como criterio (4.105), con función  $f(d) = d^2$ , la muestra de (4.111) quedaría de la siguiente forma

$$E_1 = \sum_i h(|\mathbf{i} - \mathbf{g}|, t) \left[ \sum_{k=1}^n (w_{ijk} - x_k)^2 \right] \quad (4.113)$$



si se calcula su gradiente, teniendo en cuenta que  $\mathbf{g}$  es constante, y llamando  $\alpha(t) = 2 \cdot \lambda(t)$ , se obtiene la regla de aprendizaje de Kohonen (4.98), dada en este caso por

$$\mathbf{w}_{ij}(t+1) = \mathbf{w}_i(t) + \alpha(t) h(\|\mathbf{i} - \mathbf{g}\|, t) (x_k - w_{ijk}) \quad (4.114)$$

Por tanto, la regla empleada en los modelos de Kohonen procede de la métrica euclídea, tal y como se ha demostrado anteriormente.

#### Regla aprendizaje Manhattan

A partir del criterio (4.108), tomando  $f(d) = d$ , la muestra de (4.111) quedaría de la siguiente forma

$$E_1 = \sum_i h(\|\mathbf{i} - \mathbf{g}\|, t) \left[ \sum_{k=1}^n |w_{ijk} - x_k| \right] \quad (4.115)$$

Calculando el gradiente, realizando varios pasos de adecuación de las funciones para los casos  $w_{ijk} > 0$  y  $w_{ijk} < 0$  y llamando  $\alpha(t) = \lambda(t)$ , la regla de aprendizaje queda de la siguiente forma

$$\mathbf{w}_{ij}(t+1) = \mathbf{w}_i(t) + \alpha(t) h(\|\mathbf{i} - \mathbf{g}\|, t) \text{sign}(x_k - w_{ijk}) \quad (4.116)$$

a pesar de la apariencia con la euclídea, (4.116) es mucho más simple de realizar.

Tras numerosas simulaciones comparativas entre un modelo basado en Manhattan y otro euclídeo, en (Onodera *et al.*, 1990) se concluye que ambos procesos son muy similares, presentando una mejoría del 2% los euclídeos.

#### Regla aprendizaje derivada del producto escalar

Partiendo del criterio del producto escalar, se definirá la función de error para un tiempo determinado de la forma siguiente

$$E_2 = \sum_i h(\|\mathbf{i} - \mathbf{g}\|, t) f(c(\mathbf{x}(t) \mathbf{w}_i(t))) \quad (4.117)$$

cuanto mayor sea el parecido entre  $\mathbf{x}$  y  $\mathbf{w}_i$  será mayor  $c(\cdot)$ , y  $f(\cdot)$  será una función introducida por generalidad. En este caso consistirá en maximizar  $E_2(t)$ , y se obtendrá una solución aproximada iterando de la forma conocida

$$\mathbf{w}_{ij}(t+1) = \mathbf{w}_i(t) + \lambda(t) \nabla_{\mathbf{w}_i} E_2(t) \quad (4.118)$$

siendo  $\lambda(t)$  el ritmo de aprendizaje. Una vez operada la anterior ecuación y teniendo en cuenta que  $f(c) = c$ , se obtiene la regla de actualización

$$\mathbf{w}_{ij}(t+1) = \mathbf{w}_{ijk}(t) + \alpha(t) h(\|\mathbf{i} - \mathbf{g}\|, t) x_k \quad (4.119)$$

Esta regla presenta el problema de que al presentar de manera sucesiva cada patrón, los pesos pueden crecer de manera indefinida, por lo cual se debe normalizar los pesos tras cada iteración.

Esto conlleva un coste computacional añadido a cada iteración, por lo que se presentan algoritmos que preservan la norma de los vectores de pesos. Tras varias operaciones y artificios matemáticos, se lleva a la siguiente fórmula para actualizar pesos

$$w_{ijk}(t+1) = w_{ijk}(t) + \alpha h(x_k - y_{ij}(t)) \cdot w_{ijk}(t) \quad (4.120)$$

donde  $y_{ij}(t)$  es el producto escalar del vector de entradas por los pesos de la neurona  $(i, j)$ , que se considera como salida de la neurona  $(i, j)$ .

#### 4.5.7. Otros modelos de redes neuronales

Se ha creído conveniente hacer una introducción a otros modelos de especial interés y muy empleados en la práctica, además de los presentados en anteriores secciones. Algunos de estos modelos pertenecen al grupo de *redes realimentadas*, como el modelo *Hopfield*, que cuando es interpretado desde el punto de vista probabilístico se convierte en la *máquina de Boltzmann*. También se tratará brevemente el modelo *Hopfield analógico*.

De la misma forma, y no del grupo de las *redes realimentadas*, se detallarán los modelos basados en *RBF* y los modelos *LVQ*, modelo supervisado introducido por Kohonen.

##### 4.5.7.1. Redes neuronales realimentadas

Los modelos vistos hasta ahora son *feedforward*, donde la información se propaga en un único sentido al no existir realimentación. Al añadir la realimentación, el proceso de la operación, ya por sí complicado, se dificulta aún más.

La respuesta de la red realimentada se estabilizará una vez transcurran un determinado número de iteraciones, convergiendo a un estado estable. La demostración de la estabilidad en la respuesta se hace por medio del método de *Lyapunov*, como alternativa a la integración de los sistemas de ecuaciones diferenciales o en diferencias. La función de *Lyapunov*, también se conoce como *función energía de Lyapunov*, ya que es una generalización del concepto físico de la energía.

##### 4.5.7.2. El modelo Hopfield

El mérito de Hopfield (1982) con su modelo, no consistió en la propuesta de una nueva arquitectura de red, sino en las nuevas ideas que sobre un viejo modelo propuso en su obra. Además, Hopfield subrayó la fácil implementación de su nuevo enfoque sobre *Hardware*.

En la Figura 4.28 se representa la arquitectura básica del modelo *Hopfield*, donde existe una única capa de neuronas conectada con todas las demás. El modelo de neurona empleado se detalla en la Figura 4.29.

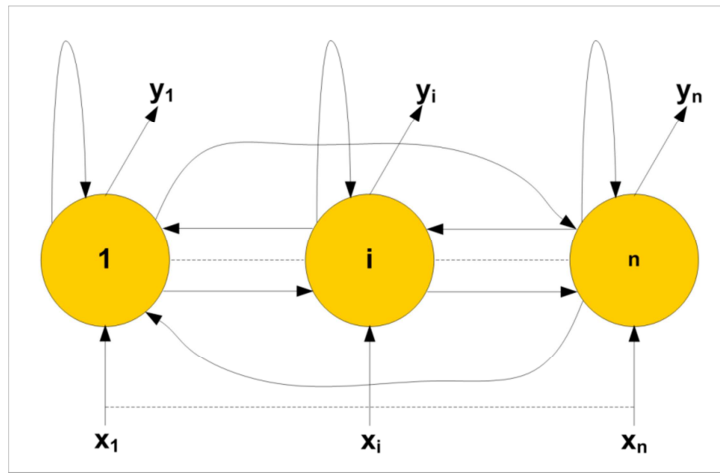


Figura 4.28. Esquema genérico del modelo Hopfield. Fuente Propia.

Como se ve en la Figura 4.29, es una neurona idéntica a *Perceptron*, un sencillo elemento de umbral, con entradas binarias ( $x_j(t) \equiv (\{0,1\} \text{ o } \{-1,+1\})$ ), y salidas igualmente binarias ( $y_i(t)$ ).

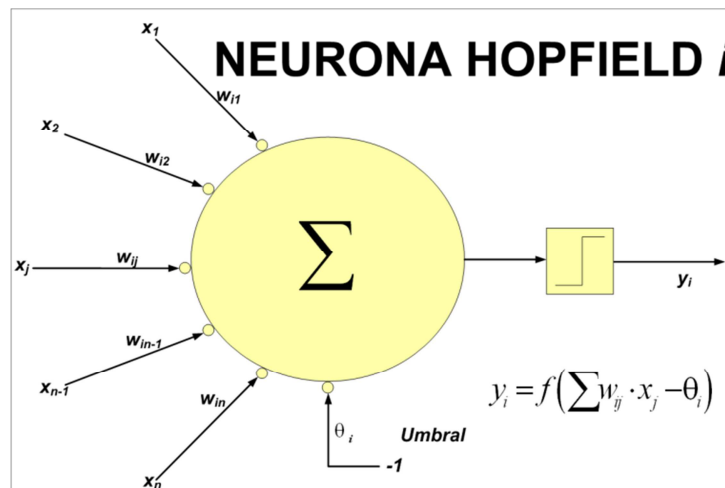


Figura 4.29. Neurona del modelo Hopfield. Fuente Propia.

Los pesos ( $w_{ij}$ ) son continuos, es decir, números reales; la neurona  $i$  calcula el potencial postsináptico ( $h_i$ ), como la suma de entradas ponderada con los pesos sinápticos, menos un umbral de disparo, como vemos a continuación

$$h_i(t) = \sum_j w_{ij} x_j(t) - \theta_i \tag{4.121}$$

Aplicando a (4.121) la función tipo escalón se obtiene la salida digital de la neurona por medio de

$$y_i(t) = f(h_i(t)) = f\left(\sum_j w_{ij} x_j(t) - \theta_i\right) \tag{4.122}$$

Como aparece en (Müller and Reinhardt, 1990), cuando las neuronas hacen uso de valores  $\{-1,+1\}$  se conoce como de tipo *Ising*, con función de activación  $f(x) = \text{sign}(x)$ , un +1 es la neurona activada y un -1 desactivada.

Frente a las entradas del exterior, las neuronas del modelo calculan las salidas asociadas, y debido a la realimentación, estas salidas se convierten de nuevo en entradas de la propia red, proporcionando ésta una nueva salida. Por tanto, el modelo *Hopfield* puede expresarse de la siguiente manera

$$x_i(t+1) = f\left(\sum_j w_{ij}x_j(t) - \theta_i\right) \quad (4.123)$$

En el instante  $t = 0$  las neuronas reciben del exterior un vector de entradas  $x_j(0)$ , con lo que cada neurona  $i$  queda en ese estado, a partir de entonces, la red deja de comunicar con el exterior, procesando las entradas recibidas; para tiempos posteriores  $t + 1$ , las entradas en cada neurona serán el estado en que se sitúan en el instante anterior  $t$ .

Posteriormente, Hopfield (1984) presentó su modelo pero para respuesta analógica, pareciéndose todavía más a la realidad biológica. El modelo de salida binaria hacía que el modelo se alejara de la operación biológica. Las neuronas en este caso presentan una respuesta sigmoidea, con salida en el intervalo  $[0, +1]$ .

#### Dinámica de la red

Existen diferentes dinámicas de actualización de la red, pudiendo distinguir dos tipos genéricos según Müller and Reinhardt (1990):

- *Modo serie o dinámica asíncrona*: en un instante dado, solamente una neurona actualiza su estado, la cual podrá ser seleccionada aleatoriamente o con cierto orden.
- *Modo paralelo o dinámica síncrona*: en un instante dado, varias neuronas actualizan su estado.

#### Memoria asociativa

La memoria de los computadores convencionales suele ser de *acceso directo*, donde la memoria se encuentra dividida en casillas, numeradas con cierto orden, y donde cada una de ellas registra una porción de la información total. Este número de orden se llama dirección de memoria, si se quiere seleccionar un elemento, simplemente se debe direccionar la posición donde se encuentra.

En el cerebro, la organización de la memoria es muy diferente, la información no se almacena en casillas definidas, parece que sigue cierta distribución dispersa. La operación de la memoria sigue un patrón muy similar a las *memorias asociativas*, que asocian ideas con imágenes, imágenes con etiquetas, ideas con ideas, etc.

Las ANNs operan siguiendo un esquema similar, por ejemplo una *MLP* implementa una *memoria asociativa de tipo heteroasociativo*, donde asocia un patrón de entrada con un patrón objetivo en la salida. Por el contrario, el modelo *Hopfield* toma el papel de *memoria autoasociativa*.

Una red *Hopfield*, en cada iteración, pasa de un estado al siguiente, el proceso finalizará al llegar a un estado estable, es decir, cuando se cumpla la siguiente condición

$$\mathbf{x}(t+1) = \mathbf{x}(t) \equiv \mathbf{x}^* \quad (4.124)$$

supone que la red ya no cambiará, que la red se ha estabilizado. En este momento podemos suponer que la red ha terminado de procesar el patrón procedente del exterior, siendo  $\mathbf{x}^*$  la respuesta proporcionada.

Un estado estable es un mínimo local de una *función de energía de Lyapunov* que será definida en la red, de manera que la forma comentada anteriormente definirá un estado estable.

El objetivo del entrenamiento de este modelo es almacenar un conjunto de patrones en forma de estados estables de la red, de tal manera, que ante cierto patrón de entrada, se espera que la red obtenga como salida alguno de los estados estables memorizados, por lo que la red actuará como *memoria asociativa*. Además, si el patrón de entrada presentado a la red es alguno de los estados estables ya memorizados, obviamente, la red proporcionará el mismo estado, por tanto, el modelo *Hopfield* actúa como *memoria autoasociativa*.

El interés de *Hopfield* radica en lo siguiente, si se presenta a la red un estado con cierto ruido, se espera que la ANN evolucione hasta alcanzar un nivel de estado estable, lo más próximo posible a la solución. Por tanto, esta red *autoasociativa*, puede eliminar cierto ruido en los patrones de entrada, como por ejemplo en las imágenes, y se puede hablar de una red tolerante al ruido.

#### La función de energía de la red

El incremento de energía de la red, para neurona con salida  $\{0,1\}$ , al actualizar cierta neurona. Este incremento viene definido por

$$\Delta E_i = - \left( \sum_{j=1}^n w_{ij} x_j(t) - \theta_i \right) \cdot \Delta x_i(t) \quad (4.125)$$

será siempre menor o igual a cero; a partir de (4.125), la energía de una neurona individual vendrá dada por la siguiente expresión

$$E_i = -x_i(t) \cdot \left( \sum_{j=1}^n w_{ij} x_j(t) - \theta_i \right) \quad (4.126)$$

A partir de (4.126), sumando la contribución de todas las neuronas y haciendo la suposición de que la matriz de pesos es de diagonal nula y, además, simétrica, la expresión de la *energía de la red* será

$$E = -\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n w_{ij} x_i x_j + \sum_{i=1}^n \theta_i \cdot x_i \quad (4.127)$$

Al estar la energía limitada, la red llegará a un estado de mínima energía siempre, que será un mínimo local de la energía de la red, correspondiéndose con un estado estable. Para que (4.127) sea una función de energía se debe cumplir  $w_{ij} = w_{ji}$  y  $w_{ii} > 0$ .

Lo anterior es válido para dinámicas síncrona o asíncrona, y ciertas condiciones de los pesos. Al existir más posibilidades, se remite al lector a contemplarlas en la obra de (Bruck, 1990).

La energía de red para neuronas tipo *Ising* ( $\{-1, +1\}$ ), y umbrales nulos, puede expresarse de la manera siguiente

$$E = -\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n w_{ij} x_i x_j \quad (4.128)$$

considerando  $w_{ij} = w_{ji}$  y  $w_{ii} = 0$ .

La *función de energía* para el caso continuo (Zurada, 1992), definida mediante una función de *Lyapunov*, vendrá dada por

$$E = -\frac{1}{2} \sum_{i,j} w_{ij} V_i V_j + \sum \left( \frac{1}{R_i} \right) \int_0^{V_i} g^{-1}(V) dV - \sum_i I_i V_i \quad (4.129)$$

siendo  $V_i$  una tensión continua positiva, dada por cierta función de transferencia  $g(\cdot)$ ;  $R_i$  la resistencia del circuito electrónico, ya que *Hopfield* planteó la semejanza entre éste y su modelo *ANN*. La variación con el tiempo de (4.129) es no positiva, si la función  $g^{-1}$  es monótona creciente y la matriz de pesos simétrica.

Todos los detalles del aprendizaje de la red Hopfield, con los detalles de los ajustes de patrones, así como la *regla de Hebb* y de la *pseudoinversa* aplicada a tal modelo pueden encontrarse en (Martín del Brío and Sanz, 2006).

#### 4.5.7.3. Funciones de base radial (RBF)

Otra red unidireccional para aproximación funcional, similar en cuanto a concepto a *MLP*, es la *RBF* introducida en (Moody and Darken, 1989; Poggio and Girosi, 1990), con la peculiaridad de que es de tipo *híbrido*, ya que introduce aprendizaje supervisado y no supervisado.

La arquitectura de un modelo *RBF* consta de tres capas de neurona, similar a *MLP* de una capa. Las neuronas de la capa de salida son lineales y calculan la suma ponderada de las salidas proporcionadas por la capa oculta. La diferencia fundamental frente a *MLP* está en la operación de la capa oculta, ya que en lugar de emplear la función sigmoidea, operan en base a la distancia del vector de entradas respecto al vector sináptico que cada una almacena, llamado *centroide*, cantidad a la que aplican una función radial con forma *gaussiana* (Figura 4.30), por lo que las neuronas son de *respuesta localizada*, ya que responderán con cierta intensidad, cuando el vector de entrada

presentado a la red y el centroide de la neurona estén en una zona próxima en el espacio de entradas.

Según Figura 4.30, cada neurona de la capa oculta almacena un vector  $c_{ji}$ , correspondiendo con el centroide. Cada neurona calcula la distancia euclídea  $r_j$  que separa vector de entradas y centroide, siendo esta distancia la siguiente

$$r_j^2 = \sum_i (x_i - c_{ji})^2 \quad (4.130)$$

Emplea la *función radial*  $\phi(r)$  de la salida de la neurona  $y_j$ , siendo una función típica la gaussiana  $\phi(r) = e^{-r^2/2\sigma^2}$ , o similares. Precisamente el término *función de base radial* procede de la simetría radial de ese tipo de funciones, donde el parámetro de normalización  $\sigma$  determina la anchura *gaussiana*, equivalente al radio de influencia de una neurona en el espacio de las entradas, cuando mayor sea dicho parámetro, mayor será la zona *dominada* por la neurona desde su centroide. Aglutinando expresiones, la salida de la neurona oculta, vendrá dada por la expresión siguiente

$$y_j = e^{-r_j^2/2\sigma_j^2} = e^{-\sum_i (x_i - c_{ji})^2 / 2\sigma_j^2} \quad (4.131)$$

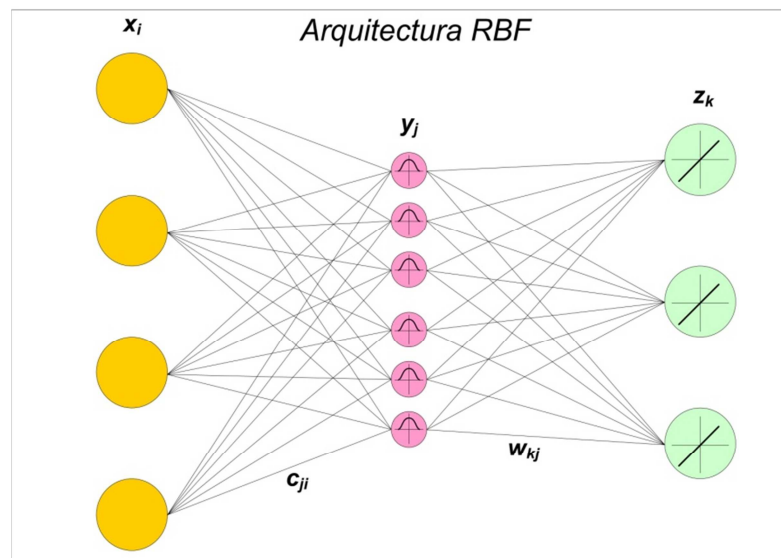


Figura 4.30. Arquitectura RBF. Fuente Propia.

La neurona responderá con la máxima salida, si el patrón que se presenta coincide con el centroide. Dependiendo de la zona del área donde el patrón caiga, con respecto al centroide y su radio, la neurona se activará más o menos.

La salida de las neuronas de la capa oculta serán las entradas de las neuronas de salida, que a su vez, calcularán la respuesta según la siguiente expresión

$$z_k = \sum_j w_{kj} y_j + \theta_k = \sum_j w_{kj} \phi(r_j) + \phi_k \quad (4.132)$$

siendo  $w_{kj}$  el peso de la neurona de la capa oculta con la de salida, y  $\theta_k$  un parámetro adicional de la capa oculta, que al igual que en *Perceptron*, se conoce como *bias*.

### Aprendizaje en RBF

Una cuestión importante en estas redes es la forma de elegir el número de nodos radiales, los de la capa oculta; como cada nodo cubre un espacio de entrada, se debe escoger su tamaño para que cubra su totalidad. Si el espacio de entradas tiene gran número de variables, los nodos necesarios crecerán exponencialmente. Por tanto, debe llegarse a un compromiso entre el número de nodos radiales, error en el ajuste de los patrones de aprendizaje, y capacidad de generalización. Los anteriores aspectos son similares a los tratados en *backpropagation*.

Para la selección del número de neuronas ocultas, se puede comenzar por un grupo reducido de nodos *gaussianos*, y si algún patrón de entrada no activa en medida suficiente ninguna de las neuronas ocultas, se introduce una nueva; el modelo *algoritmo autoorganizado jerárquico* (Lee, 1992) emplea un método similar al explicado. Una vez determinados los nodos radiales, se deberá encontrar los parámetros de la arquitectura, para lo cual se suele emplear un *aprendizaje por etapas*, donde primero se entrenan las neuronas ocultas para posteriormente entrenar las de salida.

Como muestran Tou and González (1974), se puede emplear de manera óptima el algoritmo de *k-means* para obtener el valor de los centroides; en *k-means*, *k* hace referencia al número de clústers que se tratan de localizar, que para este caso, corresponderán con el número de nodos *gaussianos* planteados en el comienzo. El algoritmo, una vez propuestos los grupos de partida, procederá de la siguiente forma:

1. Inicialmente, como centroides, se escogen los primeros *k* patrones de aprendizaje.
2. En cada una de las iteraciones *t* se reparten los patrones de aprendizaje entre las neuronas, seleccionando aquella neurona cuyo centroide diste en menos valor al patrón presentado.
3. Una vez distribuidos los patrones, para cada neurona se calcula el nuevo centroide como promedio de sus patrones asignados; llamando  $N_j$  al número de patrones de la neurona *j*, el valor del centroide será

$$c_{ji} = \frac{1}{N_j} \sum_{x \in \text{neurona } j} x \quad (4.133)$$

4. Si el valor de (4.133) no varía respecto a la iteración anterior, el valor ya ha convergido, si no es así, se vuelve a 2.

Una vez obtenidos los centroides, y como se muestra en (Wassermann, 1993), se calculan los parámetros  $\sigma_j$  para cada uno de las neuronas, calculándose de forma aproximada el radio de influencia de cada neurona con relación a las demás, para lo cual se seleccionarán los centroides de



las  $N$  neuronas más próximas al nodo  $j$ , para posteriormente calcular el promedio de las distancias cuadráticas entre ellas mediante

$$\sigma_j^2 = \frac{1}{N} \sum_{l=1}^N \sum_k (c_{lk} - c_{jk})^2 \quad (4.134)$$

Una alternativa a Wassermann es la indicada por Hush and Horne (1993), donde se calcula el promedio de la distancia de diversos patrones que representan al centroide

$$\sigma_j^2 = \frac{1}{N} \sum_{x \in \text{nodo } j} \|\mathbf{x} - \mathbf{c}_j\|^2 \quad (4.135)$$

donde  $N_j$  será el número de patrones para el cálculo del factor de escala del nodo  $j$ . Calculados los factores de escala mediante (4.134) o (4.135) finaliza el entrenamiento de la capa oculta.

En el entrenamiento de las neuronas de salida, para el cálculo de los pesos  $w_{kj}$  y umbrales  $\theta_k$  se calculan empleando la *regla LMS*, con lo que la salida de la capa final será

$$z_k = \sum_j w_{kj} \phi(r_j) + \theta_k \quad (4.136)$$

y los pesos vendrán dados por la siguiente expresión

$$w_{kj}(t+1) = w_{kj}(t) + \varepsilon(t_k - z_k) \phi(r_j) \quad (4.137)$$

siendo  $t_k$  los valores objetivo; las actualizaciones de los umbrales siguen un esquema similar, al considerarse de nuevo pesos con entradas de valor -1.

Con la ruptura del aprendizaje en dos etapas, se consigue acelerar el proceso de aprendizaje si lo comparamos con *backpropagation*. Como hemos visto, en la primera etapa, se hace uso de un algoritmo no supervisado (por ejemplo *k-means*), mientras que en la segunda se emplea uno supervisado (similar al de *Adaline*), por este motivo se considera a *RBF* como una *red híbrida*.

#### 4.5.7.4. Learning Vector Quantization (LVQ)

Modelo introducido por Kohonen (1990) para llevar a cabo la tarea de clasificación de patrones, formado por una capa de neuronas de Kohonen. Se supondrá inicialmente que cada una de las  $n$  neuronas representa una clase, y almacenará un vector de referencia que caracteriza el *prototipo de una clase*,  $\mathbf{w}_j$  ( $j=1, \dots, n$ ). Al presentar un patrón  $\mathbf{x}$  con  $m$  entradas, cada una de las neuronas calculará la distancia que separa su vector de referencia del vector presentado por medio de

$$d^2(\mathbf{w}_j, \mathbf{x}) = \sum_{k=1}^m (w_{jk} - x_k)^2 \quad (4.138)$$

la clase  $C$  donde el patrón  $\mathbf{x}$  queda clasificado vendrá dada por la neurona  $c$  de menor distancia con el vector de referencia  $\mathbf{w}_c$ .

El aprendizaje básicamente consistirá en premiar a la neurona que clasifica de manera correcta el patrón presentado, actualizando sus pesos según la regla presentada para *SOM*, y castigando a la neurona que hace una mala clasificación mediante la actualización de sus pesos de manera contraria a la premiada.

Supongamos  $p$  patrones para el aprendizaje  $\mathbf{x}^\mu$  ( $\mu = 1, \dots, p$ ), donde cada patrón pertenece a una clase  $C_{x\mu}$  conocida ( $n$  clases en total); cada neurona es etiquetada con una clase  $C_{wj}$ . Si al presentar cierto patrón de entrada  $\mathbf{x}^\mu$ , el vector de referencia más próximo viene dado por  $\mathbf{w}_c$ , con su clase de referencia  $C_{wc}$ , se actualizará la neurona  $c$  de la siguiente manera:

1. Si la clasificación es correcta ( $C_{x\mu} = C_{wc}$ ) el vector de referencia será modificado de la siguiente manera

$$\mathbf{w}_c(t+1) = \mathbf{w}_c(t) + \alpha_t [\mathbf{x}^\mu - \mathbf{w}_c] \quad (4.139)$$

donde  $0 < \alpha_t \leq 1$  y decreciente en el tiempo; en este caso los pesos rotarán hacia las entradas, en un intento de aproximarse.

2. Si la clasificación no es correcta ( $C_{x\mu} \neq C_{wc}$ ), la modificación se realizará en sentido contrario, siendo esta modificación la siguiente

$$\mathbf{w}_c(t+1) = \mathbf{w}_c(t) - \alpha_t [\mathbf{x}^\mu - \mathbf{w}_c] \quad (4.140)$$

como se aprecia, el castigo será que los pesos se alejarán de las entradas.

El proceso se repetirá una y otra vez para todos los patrones de entrenamiento. Para evitar el sobreentrenamiento es preciso realizar una parada temprana durante el proceso de aprendizaje, siendo normal hacerlo tras un número de iteraciones comprendido entre 50 y 200 veces el número de neuronas de la red.

Hasta el momento se ha supuesto que cada neurona representa una clase, pero es posible obtener unos resultados más satisfactorios si varias neuronas representan una misma clase, donde este número de neuronas perteneciente a cada clase, dependerá de cada tipo de problema.

#### **4.6. ANTECEDENTES SOBRE TRABAJOS DE LA PREDICCIÓN DE LA DEMANDA ELÉCTRICA CON MODELOS LINEALES Y NO LINEALES**

En el Capítulo III se ha presentado la clasificación de la predicción de la demanda, sin tener en cuenta el modelo empleado. Se ha creído conveniente tratar la problemática del modelo en una Sección independiente, ya que es la verdadera clave del modelado; aquélla es una decisión que atañe al objetivo del problema, pero la elección del modelo supone una apuesta del investigador.

A continuación, se presentarán los antecedentes de las dos vertientes principales en cuanto a modelos de predicción de la demanda. Estas dos vertientes agrupan los modelos desde dos

perspectivas: los que emplean técnicas lineales y los que emplean técnicas no lineales. Un tercer grupo podría ser los modelos que emplean combinaciones de ellos, los cuales también serán citados.

A medida que se presenten las diferentes técnicas, se observará como los modelos no lineales (basados en *ANN*) han ido ganando protagonismo a partir de la segunda mitad de la década de los ochenta del siglo pasado. Este progreso es debido a que determinados investigadores hicieron grandes progresos en *ANN*, tal es el caso de James Anderson, quien presentó un modelo basado en el principio de que las conexiones entre neuronas se refuerzan cada vez que se ven activadas, llamó al modelo *Brain-State-in-a-Box* (Anderson *et al.*, 1977). En ese mismo año, pero en la Universidad de Helsinki, Teuvo Kohonen presentó la evolución del de Anderson, obteniendo su modelo *SOM* (Kohonen, 1977). Tres años después y en Japón, Kunihiko Fukushima presentó un modelo de red para el reconocimiento visual de patrones (Fukushima, 1980). En 1982 volvió a resurgir el interés por las redes neuronales, por diferentes motivos, uno de ellos es el trabajo presentado por John Hopfield (Hopfield, 1982), donde describe claramente y de manera matemática, una red que precisamente lleva su nombre, además, mostró como dichas redes pueden trabajar. Además, en 1985 comenzaron a surgir diferentes conferencias internacionales a lo largo del planeta alrededor de la temática de *ANNs*.

#### **4.6.1. Trabajos con modelos lineales: Antecedentes**

Antes de comenzar, se debe recordar que los modelos lineales consisten en tratar de sintetizar en una o varias ecuación/es, más o menos compleja/s, la totalidad de las características del problema que se pretende resolver. Como ya es sabido, el problema de la predicción de la demanda eléctrica, independientemente del horizonte de predicción o su objetivo, se convierte en un dilema no trivial. La demanda presentará no linealidades con gran cantidad de variables y ocurrencias, que deberán ser reconocidas, para posteriormente ser trasladadas a la/s ecuación/es correspondiente/s en estos modelos.

En estas secciones no interesa el horizonte de la predicción ni el objetivo, pero se debe recordar que los modelos ya se clasificaban en los que predecían *carga pico* y los llamados Forma de la Carga (*load shape*), siendo estos segundos los que nos servirán como punto de partida del análisis para presentar los diferentes modelos lineales aplicados hasta la fecha.

*Load shape* se corresponden con los que procesan series temporales sobre un intervalo de tiempo concreto, pudiendo estar los que predicen la curva de carga y, por tanto, las predicciones *STLF*. Combinaciones de *load shape* y *carga pico* aparecieron combinados, como el presentado por Goh *et al.* (1986). *Load shape* puede dar origen a dos variantes, los llamados Tiempo de Día (*time-of-day*) y los Modelos Dinámicos (*dynamic models*).

#### 4.6.1.1. Modelos basados en *time-of-day*

Los modelos *time-of-day* definen la carga  $L(t)$  en cada tiempo discreto  $t$  de periodo de predicción de duración  $T$  por una serie de tiempo

$$\{L(t), t = 1, 2, \dots, T - 1, T\} \quad (4.141)$$

el modelo almacena  $T$  valores de carga basados en observaciones anteriores de dicha carga. Dependiendo el modelo, algunos almacenaban varias curvas de carga de semanas previas y, otros, tan sólo la semana anterior. Posteriormente, y a juicio del operador, se trata de predecir la demanda por medio de fórmulas, por tanto, es común el uso de sistemas expertos que emularan la función del operador, y la aplicación de reglas, como muestran Rahman and Bhatnagar (1987). Los sistemas expertos son una rama de la *AI*, por lo que se aproximarían a las *ANN*, pero se ha considerado el introducirlo en esta sección, ya que es precisa mucha labor humana para la creación de las reglas de la base de conocimiento.

Una forma común de un modelo *time-of-day* es

$$L(t) = \sum_{i=1}^N \alpha_i f_i(t) + v(t) \quad (4.142)$$

donde la carga en el tiempo  $t$ ,  $L(t)$ , es considerada como la suma de un número finito de funciones explícitas de tiempo  $f_i(t)$ , normalmente sinusoidales con un periodo de 24 a 168 horas, dependiendo de la previsión esperada, los coeficientes  $\alpha_i$  son tratados como constantes de variación de tiempo, mientras que  $v(t)$  representa el error del modelo. Cuando las funciones  $f(\cdot)$  se seleccionan a priori como funciones de tiempo explícitas, tales como sinusoides, los parámetros  $\alpha_i$  son estimados a través de regresión lineal simple o técnicas parecidas, siendo ejemplos de estos modelos los encontrados en (Quintana *et al.*, 1978; Fankhauser, 1984), caracterizados por su sencillez de modelo y la fácil actualización de sus parámetros.

Dentro de los modelos *time-of-day*, aparecen los basados en Descomposición espectral (*spectral decomposition*), que teniendo la forma de (4.142), sus funciones de tiempo  $f(\cdot)$  representan las correspondientes funciones propias para la función de correlación de la serie de tiempo, como vemos en (Belik *et al.*, 1978), teniendo mejores resultados que sus homólogas, anteriormente comentadas. Pese a su solidez respecto a otros modelos del tipo *time-of-day*, los modelos *spectral decomposition* se encuentran en pocas aplicaciones para *utilities*, algunos ejemplos del pasado se encuentran en las obras (Farmer and Potton, 1968; Anelli *et al.*, 1978).

Dentro de *time-of-day*, aparecen modelos basados en Mínimos Cuadrados Ordinarios (*Ordinary Least Squares – OLS*), Meng and Shang (2008) presentan un modelo Mínimos Cuadrados Parciales (*Partial Least-Square – PLS*) para hacer la predicción anual de China; su

característica principal es la necesidad de emplear 12 años para modelar su propuesta y la validan con 5 años.

#### 4.6.1.2. Modelos basados en dynamic models

Los autores que emplean *dynamic models*, reconocen que la carga depende no sólo de la hora del día, sino que depende de parámetros climáticos y entradas aleatorias. Los *dynamic models* son de dos tipos, *ARMA* y Espacio de Estado (*state-space*). *ARMA* toman la forma siguiente

$$L(t) = y_p(t) + y(t) \quad (4.143)$$

donde  $y_p(t)$  depende fundamentalmente de la hora del día y de patrones climáticos normales para el día en particular, pudiéndose expresar esta componente por (4.142). El término  $y(t)$  representa un término residual de carga, que representa la desviación de los patrones climáticos debidos a los efectos aleatorios de la correlación.

Abou-Hussien *et al.* (1981) plantean un *dynamic models* formado por cinco componentes de  $y_p(t)$ , siendo los principales componentes: la media de los patrones de carga diaria y la media del incremento de carga seminal; además, introduce un Sesgos (*bias*) error basado en horas inmediatamente anteriores y comportamientos climáticos.

Variaciones de (4.143) aparecen en la literatura, por medio de nombres como *Box-Jenkins*, Series de Tiempo (*time series*), Función de Transferencia (*transfer function*), Estocástico (*stochastic*), *ARMA* y Auto-Regresivo Integrada de Media Móvil (*Auto-Regressive Integrated Moving Average - ARIMA*), presentando ligeras diferencias entre ellos. Es posible englobar todos dentro de *ARMA* y, en concreto, dentro de *dynamic models*.

Algunos autores han optado por representar la componente de carga periódica tal y como aparece en (4.143), como es el caso de Rajurkar and Nissen (1985) y Schneider *et al.* (1985). Otros, pre-filtraron los datos de carga para así eliminar la componente periódica, como los trabajos de Abu-El-Magd and Sinha (1981), Ernoult and Mattatia (1984) y Paysti (1984). Como consecuencia del pre-filtrado, un nuevo proceso de carga resulta

$$L'(t) = L(t) - L(t - t_p) \quad (4.144)$$

donde  $t_p$  es el periodo de la hora del día de la componente, el resultado del proceso será  $L'(t)$  libre de término periódico y satisfaciendo ecuaciones similares a (4.144) de *ARMA*.

Hagan and Behr (1987) hacen una exposición de ciertas limitaciones existentes en los modelos *Box-Jenkins*, siendo la principal la propia linealidad y, por tanto, no siendo la mejor elección para el problema de predicción de la demanda que es, ciertamente, una cuestión no lineal. Plantean un *ARIMA*, al que se le introduce una variable de temperatura, pero modificada por una función de transformación no lineal.

Solo algunos trabajos presentan factores climáticos en las entradas de los modelos *ARMA*, ya que en momentos con cambios rápidos de las variables climáticas, no es posible la actualización automática de los parámetros del modelo. Algunos de los modelos *ARMA* que incluyen los efectos de las variables climáticas como entradas adicionales, se tratan en (Schneider *et al.*, 1985; Paysti, 1984; Keyhani and Miri, 1983; Vemuri *et al.*, 1986), mientras que otros confían en un enfoque heurístico y, la carga es corregida en base a la temperatura, antes de aplicar el modelo *ARMA*, como vemos en (Ernoul and Mattatia, 1983). Tal y como indican Gross and Galiana (1987), las entradas de los modelos son la temperatura y, como característica fundamental, se destaca su comportamiento no lineal.

La labor de identificación de los parámetros de los modelos lineales, resulta una labor bastante complicada y tediosa. Para la identificación de parámetros de los modelos *ARMA*, se puede emplear un esquema recursivo por medio de la solución de la ecuación de Yule–Walker, tal y como se aplica en (Vermudi *et al.*, 1986).

Como demuestran Ljung and Soderstrom (1983), los modelos *state–spaces* son modelos fácilmente convertibles en *ARMA* y viceversa; por esto su especial interés, ya que no existen especiales diferencias entre ellos. Sin embargo, se presentan numerosos trabajos de *state–space*, ya que añaden un grado de estructura que no aparece en los *ARMA*. En estos modelos la carga vendrá dada por

$$L(t) = \mathbf{c}^T \mathbf{x}(t) \quad (4.145)$$

donde

$$\mathbf{x}(t+1) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) + \mathbf{w}(t) \quad (4.146)$$

$\mathbf{x}(t)$  es el vector de estado en el instante  $t$ ,  $\mathbf{u}(t)$  vector de variables climáticas de entrada,  $\mathbf{w}(t)$  vector de entrada de ruido blanco, y las matrices  $\mathbf{A}$ ,  $\mathbf{B}$  y el vector  $\mathbf{c}$  se asumen como constantes. A partir de (4.145) y (4.146), existen una importante variación de modelos, pero en todos, la diferencia con *ARMA* es, que en los modelos *state–space* los parámetros que definen la componente periódica de carga son procesos aleatorios.

En algunos de los modelos las matrices  $\mathbf{A}$  y  $\mathbf{B}$  no tienen por qué ser conocidas, como los trabajos presentados en (Irisarri *et al.*, 1982; Campo and Ruíz, 1987). En cambio, en (Abu–El–Magd and Sinha, 1981), los autores afirman que las matrices deben estar completamente identificadas. No están claras las ventajas de *state–space* models sobre *ARMA*, pero como se indica en (Gross and Galiana, 1987) es posible que su principal aplicación esté en Predicción de la Carga de Bus (*bus load forecasting*), donde bus load presenta un alto grado de correlación.

En los primeros años del presente siglo, se continúan encontrando algunas obras relacionadas con *ARMA*, pero en menor proporción que *ANN*. Sin embargo, se debe indicar que hoy en día,

numerosas *TSOs* y *DSOs* operan con dichos modelos, como en las obras (Fan and McDonald, 1994; Yang *et al.*, 2005).

A partir del año 2000, se siguen encontrando modelos *ARMA*, pero tratando de presentar nuevos modelos en base a nuevas propuestas de parámetros o empleándolos en combinación con las ya empleadas *ANN*. Alguna obra también realiza alguna comparativa con los *ANN*, presentando las ventajas de los modelos *ARMA* frente a las desventajas de los *ANN*. A continuación algunos ejemplos.

Huang and Shin, (2003) presentan un modelo *ARMA* junto a un Proceso No–Gaussiano (*non–Gaussian process*) para una mejor identificación de parámetros, con resultados ligeramente mejores que un modelo *ANN* probado. Chen *et al.* (2006) presentan una generalización de *ARMA* llamada Heteroscedasticidad Condicional AutoRegresiva (*AutoRegressive Conditional Heteroscedasticity – ARCH*), mejorando a los modelos *ARMA* probados pero no frente a modelos basados en *ANN*. Baharudi and Kamel (2008) presentan el método Covarianza Modificada (*Modified COVariance – MCOV*), el cual es una variante mejorada del método *AR*; *MCOV* supera al método de *Durbin (ARMA)* y al de *Burg (AR)*, la predicción es validada para un número pequeño de días, por lo que no se verá la influencia de los meses de inviernos ni de los festivos. Wang *et al.* (2009) muestran la combinación de un modelo *ARMA* junto a un modelo *ANN* entrenado con *backpropagation*, mostrando mejoras con respecto al modelo *ARMA* de una sola etapa. Fadhila *et al.* (2009) comparan los mejores resultados de un modelo Sistema de Inferencia Neurofuzzy Adaptativa (*Adaptative NeuroFuzzy Inference System – ANFIS*) con uno *AR*, cuando se excluyen los fines de semana, obteniendo similares resultados que otro *ARMA* presentado, y concluyendo los autores de la necesidad de emplear modelos no lineales para detectar los picos de la serie, provocados por fines de semana y festivos, y así reducir algunos rasgos en la propia serie y obtener mejores resultados. Kamaev *et al.* (2012) presentan una comparativa entre Modelo de Media Móvil (*Moving Average Model – MAM*), Regresión Lineal (*Linear Regression – LR*) y *ANN*; presenta como desventaja del modelo *ANN* el no disponer del conocimiento interno del modelo.

#### **4.6.1.3. Agrupación de los modelos basados en modelos lineales**

Para finalizar, se resumirá en la Figura 4.31 los modelos presentados en esta sección. Tras el análisis, se puede concluir que el principal reto de los modelos lineales es el conseguir trasladar el conocimiento del problema tratado a las ecuaciones que definen el modelo.

La predicción de la demanda es un problema complejo, debido a sus características no lineales, por tanto, el esfuerzo planteado para su representación en estos modelos es grande. Debemos recordar de nuevo la necesidad imperiosa de trasladar el conocimiento del problema tratado (demanda eléctrica) a las ecuaciones que deben tratar de representar dichas particularidades.

El reto de construir estas ecuaciones se magnifica si pensamos en las no linealidades que la demanda eléctrica presenta con ciertas variables que la afectan.

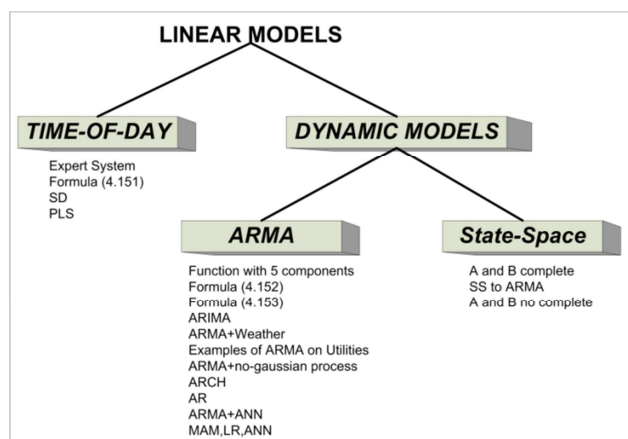


Figura 4.31. Resumen de los modelos lineales presentados en esta sección. *SD*: Spectral Decomposition; *PLS*: Partial Least-Square; *ARIMA*: Auto-Regressive Integrated Moving Average; *ARCH*: AutoRegressive Conditional Heteroscedasticity; *AR*: Auto-Regressive; *ARMA*: Auto-Regressive and Moving Average; *MAM*: Moving Average Model; *LR*: Linear Regression; *SS*: state-space. Fuente Propia.

#### 4.6.2. Trabajos con modelos no lineales: Antecedentes

En 1985, ya se comenzó a ver la manera en que los modelos no lineales describían de manera correcta las relaciones entre la componente periódica y la residual, como por ejemplo en (Bunn and Farmer, 1985). Además, recordemos las limitaciones observadas por Hagan and Behr (1987) en torno a los modelos lineales, frente a los no lineales, encabezados por las técnicas basadas en ANN.

##### 4.6.2.1. ANN y su clasificación

Las ANN, son capaces de aprender de la experiencia, generalizar de casos anteriores a nuevos casos, abstraer características esenciales a partir de entradas que representan información relevante, etc.; esto hace que presenten ciertas ventajas y su interés en aplicaciones aumente con el tiempo.

Precisamente la capacidad de generalización y sobre todo, la facilidad de descubrir relaciones no lineales en entornos complejos, las hace muy atractivas para emplearse en problemas como el de la predicción de la demanda, donde la limitación de las técnicas lineales ha quedado de manifiesto, requiriéndose un gran esfuerzo del investigador para suplir tales limitaciones.

Las ANN están formadas por neuronas dispuestas en diferentes capas (entrada, oculta/s y salida). La neurona llamada estándar, a partir de la neurona artificial presentada entre otros por McClelland y Rumelhart, mostrada en las obras (McClelland and Rumelhart, 1986; Rumelhart and McClelland, 1986), dispone de regla de propagación la suma ponderada de los valores de las entradas junto a los pesos asociados y, la identidad como función de salida de la neurona. A partir



de los valores de la entrada, y tras aplicar la regla de propagación, se obtendrá un resultado a la salida de la neurona.

Como ya se ha explicado, se distinguen dos maneras de operar con las ANN, el modo de aprendizaje o entrenamiento, y el modo de ejecución o recuerdo. El aprendizaje consistirá en actualizar los pesos sinápticos siguiendo una regla de aprendizaje o entrenamiento, construida a partir de la optimización de una función de error, que intenta medir la eficacia de la operación de la red.

Los dos tipos básicos de aprendizaje son el supervisado y el no supervisado, diferenciándose en la manera en que tratan los patrones. El entrenamiento supervisado proporciona cierta información de la función  $e-s$  a estimar, en forma de distribución de las clases o las etiquetas de los patrones de salida, mientras que en el no supervisado no existe tal información. Las características fundamentales de los modelos son su tipo de aprendizaje y la arquitectura de red, por tanto, en la Figura 4.17 se mostró una clasificación en base a estas características.

Ya desde los comienzos aparecen autores que tratan de verificar si los modelos ANN producen mejoras que los modelos de la familia ARMA y, por supuesto, que los primeros basados en medias históricas. Con Sharda and Patil (1990), se demuestra que un MLP iguala a modelos Box-Jenkins. Tang *et al.* (1991) demuestran que para series temporales largas, los tres modelos ANN analizados ofrecen similares resultados que el modelo Box-Jenkins, no obstante, con serie temporales cortas, los modelos ANN mejoran al modelo Box-Jenkins, por lo que los autores las posicionan como alternativa a los modelos ARMA y similares en el futuro. La misma conclusión con respecto a series largas y cortas se encuentra en (Badri, 1996), siendo extremadamente interesante cuando las series son cortas y los presentan como modelos de futuro.

De lo arriba indicado se pueden obtener varias conclusiones: la primera evidencia que a partir de la última década del siglo pasado, los autores ya comienzan a presentar trabajos basados en ANN que mejoran los modelos basados en técnicas lineales; la segunda es que para series de datos cortas, los modelos ANN son capaces de predecir más eficientemente, debido a que sus características de detección de no linealidades, hará que ante un número corto de patrones realice la predicción con un error más bajo. Esta última conclusión lleva a la reflexión de que a partir del despliegue de medida en las SG (y en el resto de entornos desagregados), y teniendo en cuenta que se está en una sociedad totalmente cambiante (social y económicamente), se posibilitará disponer de datos actualizados de los consumos de una manera más regular, por lo que los modelos basados en ANN y/o híbridos (se comentarán más adelante) se presentan como la solución a emplear.

Hippert *et al.* (2001) tras una revisión de ANN para predicción de la demanda de los últimos 10 años, apuntan como clave de estos modelos los siguientes temas: pre-procesamiento de datos; diseño de la ANN; implementación de la ANN; y validación de la ANN; y además, no se tiene aún

claro la forma en que la *overfitting* y *overparameterization* afectan al proceso. Pero añaden, que los trabajos presentados en los últimos años, posicionan a los modelos basados en ANN como de futuro.

Trabajos posteriores en torno a *overfitting* y su solución, habrán resuelto las dudas de los anteriores autores, como es el caso del algoritmo presentado por Wu and Liu (2009). En (Van Wyk and Engelbrecht, 2010) se presenta la prevención de *overfitting* empleando una Función de Activación sin Límites (*unbounded activation function*). Corrales and Auron (2000) estudian la compensación del fenómeno causado por la *overparameterization* en el algoritmo AutoRegresiva no Lineal con Entradas Exógenas (*Nonlinear AutoRegressive with eXogenous inputs – NARX*) cuando es aplicado para técnicas relacionadas con eventos.

Basándonos en la clasificación realizada en la Figura 4.17, y teniendo en cuenta la gran cantidad de trabajos presentados en los últimos 30 años, se procederá a presentar los trabajos más significativos. Se debe tener en cuenta que se han presentado trabajos basados en ANN y ANN junto a otros modelos los cuales llamaremos híbridos. Dentro de los híbridos, habrá trabajos que suponga la incorporación de otros modelos basados en ANN, o incluso modelos lineales o de otro estilo (lógica fuzzy).

#### 4.6.2.2. Modelos ANNs supervisados feedforward

En cuanto a los modelos supervisados unidireccionales, desde el comienzo de la aplicación de modelos basados en ANN para la predicción de la demanda eléctrica (independientemente del horizonte de predicción y objetivo), se vio el claro uso de modelos basados en el algoritmo de aprendizaje *backpropagation*. El motivo del uso de modelos basado en dicho algoritmo radica en sus excelentes resultados para el ajuste de funciones, junto a su facilidad para la generalización. Algunos de los trabajos ya mencionados han sido desarrollados bajo MLP, como son el caso de (Drezga and Rahman, 1988; Ho *et al.*, 1990; Park *et al.*, 1991; Ho *et al.*, 1992; Lee *et al.*, 1992; Lu *et al.*, 1993; Papalexopoulos *et al.*, 1994; Bakirtzis *et al.*, 1995; Mohammed *et al.*, 1995; Alfuhaid *et al.*, 1997; McMenemy and Monforte, 1998; Doveh *et al.*, 1999; Drezga and Rahman, 1999a; Charytoniuk and Chen, 2000; Daneshi *et al.*, 2008). El trabajo presentado por García–Ascanio and Mate (2009) presentan un modelo MLP para STLTF donde previamente los datos son ordenados según un criterio de análisis; posteriormente, el modelo es comparado con un Vector Auto–Regresivo Vector AutoRegressive (VAR) obteniendo resultados parecidos. En Hernández *et al.* (2013b) se presenta un modelo basado en MLP para STLTF en un entorno desagregado; los resultados son satisfactorios por meses, y además, se presenta un estudio que relaciona el error medio de la fase de operación frente al número de patrones en la fase de entrenamiento, planteando la posibilidad de actualizar los datos para el entrenamiento en base a un número de patrones determinados, ya que el modelo no mejorará si seguimos agregando patrones al histórico.

Se han empleado otros modelos como alternativa a *MLP*. Nair and Joshi (2010) hacen una previsión a muy corto plazo de tiempo (media hora y una hora) por medio de Red Neuronal Probabilística (*Probabilistic Neural Network – PNN*), obteniendo buenos resultados comparado con modelos similares, y destacando la habilidad de las *ANN* para aprender las no linealidades inherentes a la demanda. Mori and Iwashita (2005) se compara un modelo *Radial Basis Function Neural Network (RBFNN)* para *STLF* con modelo *MLP* y otros, resultando muy similar a *MLP*, la limitación de la prueba es que tan sólo se han empleado 79 días para la predicción, no pudiendo ver el efecto estacional y, además, los “*días especiales*” (festividades) fueron excluidos.

Otro de los grandes paradigmas de los modelos supervisados corresponde con Máquina de Vector de Apoyo (*Support Vector Machine – SVM*). Li *et al.* (2012) presentan una variante de *SVM* llamado Máquina de Vector de Apoyo de Mínimos Cuadrados (*Least Squares Support Vector Machine – LSSVM*), obteniendo resultados satisfactorios comparándolo con modelos similares basados en *ANN*. Hong *et al.* (2011a) exploran técnicas Regresión de Vector de Apoyo (*Support Vector Regression – SVR*), basadas en *SVM*, para tratar de modelar la estacionalidad y ciclicidad de la demanda.

#### 4.6.2.3. Modelos ANNs supervisados feedback

Existen algunas obras que se basan en modelos *ANN* pero realimentados (*feedback*). Aunque no se han empleado tanto como los de la sección anterior, merece la pena relatarlos. Se debe de tener en cuenta que la fase de aprendizaje de estos modelos es más lenta que por ejemplo un *MLP*, debido a su algoritmo de actualización de los pesos sinápticos, ya que debe de tener en cuenta la retroalimentación.

En cuanto a los modelos supervisados *feedback*, Sun *et al.* (2005) realizan un complejo estudio entrópico de las variables para quedarse con las más relevantes, para después hacer *STLF* mediante un modelo *Elman*, que es un modelo de *ANN feedback*; los resultados obtenidos son comparables con los de una *ANN* con *backpropagation*. Su and Zhang (2007) emplean un filtro para eliminar ruido de los datos y posteriormente un modelo *Elman* para *STLF*; la limitación de este trabajo es los pocos días empleados para la predicción, por lo que no se podrá apreciar efectos estacionales. Yongchun (2010) emplean un modelo *Elman*, cuya peculiaridad es que en lugar de emplear cada hora del día anterior como entrada, emplea la carga de 2 horas, por lo que en la entrada tiene 15 variables, 12 debido a la demanda y 3 variables climáticas.

#### 4.6.2.4. Modelos ANNs no supervisados

Existen trabajos del grupo de los modelos no supervisados como alternativa a los supervisados. Por ejemplo, se desarrollan trabajos donde se aplican en cascada varios modelos, para intentar obtener mejores resultados. Marín *et al.* (2002) aplican *SOM* para hacer la clasificación de

una zona extensa del centro de España, y posteriormente entrenar modelos *Elman* para cada grupo de clasificación, el inconveniente del trabajo es su agrupamiento manual, en super-grupos, tras el clasificado de *SOM*. No ocurre con Hernández *et al.* (2012b), quienes emplean *SOM* como reconocimiento de patrones, para posteriormente mediante el algoritmo *k-means* realizar clusters de curvas similares en un polígono industrial de manera desatendida, con el objetivo de hacer *STLF* posteriormente; además, emplean *PCA* para la detección de los outliers técnicos que hay en las curvas de carga del histórico (en bruto).

Chicco *et al.* (2004) presentan dos técnicas de reconocimiento de patrones aplicados a la clasificación de clientes eléctricos, por medio de *SOM* y un Algoritmo Modificado Sigue al Líder (*modified follow-the-leader algorithm*), cuya aplicación será la de diversificación de tarifas en base a los diferentes perfiles de clientes y la previsión de demanda eléctrica.

Carpinteiro and Reis (2005) emplean para la predicción de la demanda eléctrica de Brasil dos *SOM* en cascada, considerándose igualmente un sistema híbrido. Tsekouras *et al.* (2007) hacen una doble clasificación por medio de *SOM*, la primera servirá para la predicción de carga de los clientes y determinación de las tarifas, mientras que la segunda proporcionará valiosa información a la *utility*, empleándose en los mercados de energía.

Zhang *et al.* (2012a), en una propuesta similar a la de Hernández *et al.* (2012b), donde presentan los pasos de un algoritmo clasificador de perfiles de carga, tras una emplear *SOM* para un clasificado y clusterizado (*k-means*); presentan este tipo de clasificadores como la primera etapa de la predicción de la demanda.

#### 4.6.2.5. Modelos ANNs híbridos

Existen sistemas híbridos que se centran en estrategias fuzzy. Wang (2006) se hace predicción de la demanda eléctrica de China por medio de *SOM* y *fuzzy-rough*. Duan *et al.* (2011) combinan un clusterizado con *Fuzzy c-means (FCM)* y técnicas *SVR*. Che *et al.* (2012a) presentan un sistema híbrido llamado Modelo de Combinación Fuzzy Adaptativo (*Adaptive Fuzzy Combination Model – AFCM*) cuyo objetivo es combinar iterativamente la solución de diferentes subclases, tras aplicar *SOM* y *SVR*, y también establecer las funciones fuzzy para cada subclases homogéneas. Otros sistemas híbridos, que combinan fuzzy junto a variadas técnicas se encuentran en (Nadimi *et al.*, 2010; Lou and Dong, 2012).

Continuando con los sistemas híbridos que emplean *RBF*, Mori and Kanaoka (2006) predicen la temperatura para luego hacer *STLF*, para lo cual hace un clusterizado mediante el algoritmo Recocido Determinista (*Deterministic Annealing – DA*) y posteriormente para cada clúster, emplear un modelo *RBF*, obteniendo una alta eficiencia comparada con otros probados (*MLP* y *RBF*). Estos

autores se habían basado en un trabajo previo (Mori and Yuihara, 2001), donde el enfoque era similar pero basándose en *MLP* en lugar de *RBF*.

Los sistemas híbridos se han empleado y verificado frente a diferentes modelos *ARMA*, concluyendo que merece la pena la complejidad de la arquitectura del sistema híbrido frente a la dificultad de la conformación de las funciones lineales. Amjady (2007) emplea un método híbrido formado por un predictor lineal, llamado Pronóstico Asistido por Estimador de Estado (*Forecast–Aided State Estimator – FASE*) y *MLP*, comparando los resultados con modelos *MLP*, *FASE* y *AR*. Meng *et al.* (2011) presentan una combinación de Modelo Gris (*grey model*) para mitigar el efecto estocástico existente en la tendencia de la demanda eléctrica, para posteriormente hacer la predicción de la demanda mensual por medio de *RBF*.

Es momento de recordar que la propia complejidad de los sistemas híbridos supondrá un aumento en el tiempo de cómputo de la resolución del problema que deberá tenerse en cuenta. Che *et al.* (2012b) presentan el problema de la predicción de la demanda por medio de *SVR*, tratando de resolver el problema de tiempo de cómputo del orden de  $N^2$ . Lo anterior es replicable a sistemas con varios modelos en cascada, los cuales deben de ser valorados previamente para tener un orden computacional de la solución.

Los sistemas híbridos también se han empleado para modelar los propios parámetros del modelo que realizará la predicción. Por ejemplo, en las siguientes obras se han empleado algoritmos caóticos para la optimización de *SVR*, obteniendo unos resultados satisfactorios comparándolos con otros modelos de regresión, e incluso modelos *ARMA* (Hong, 2010; Hong, 2011b; Zhang *et al.*, 2012b; Hong *et al.*, 2013).

Continuando con los algoritmos caóticos y algoritmos genéticos en los sistemas híbridos, Deihimi and Showkati (2012) presentan una red recurrente Red de Estado Eco (*Echo State Network – ESN*) para hacer *STLF* donde en una primera etapa se emplean algoritmos genéticos para poder seleccionar las mejores variables de entrada al modelo (parámetros de carga y climáticas) obteniendo unos buenos resultados a costa de un mayor coste de cómputo en el modelado final. Li *et al.* (2013) presentan una variante de *RBF*, llamada Red Neuronal de Regresión Generalizada (*Generalized Regression Neural Network – GRNN*) para la predicción de la demanda anual; el sistema híbrido emplea Algoritmo de Optimización de Mosca (*Fly Optimization Algorithm – FOA*) para la selección apropiada de los parámetros de *GRNN*.

#### **4.6.2.6. Agrupación de los modelos basados en ANN**

Para finalizar esta sección, trataremos de agrupar los trabajos expuestos en una sencilla figura. Tal y como se ha visto, los modelos no lineales empleados en predicción de la demanda, se basan principalmente en técnicas *ANN* o combinación de éstas con otros modelos. Al igual que los

modelos lineales, se han empleado en horizontes de predicción de todo tipo, así como para la obtención de todo tipo de objetivos de la predicción. La principal diferencia de las técnicas no lineales frente a las lineales radica en que aquéllas no emplean apenas tiempo en disponer de un modelo matemático claro de las dependencias de los parámetros para explicar el problema de la predicción de la demanda, en cambio, los modelos lineales requieren de la definición precisa de las ecuaciones que lo definen. Lo que la totalidad de los modelos ANN requerirán es la adecuada parametrización de las variables de entrada al modelo, así como la definición topológica de la red (número de capas, neuronas en las mismas, funciones de entrenamiento, etc.); no obstante, esto es un trabajo sencillo y asumible (un sencillo *script*) si se compara con la parametrización de las funciones de los modelos lineales. No obstante, si el sistema de predicción va acumulando etapas en la predicción, con sucesivos modelos, estos modelos adicionales, supondrán un coste computacional adicional al modelo global, tanto para la fase de aprendizaje global, como para la fase de operación (predicción).

En resumidas cuentas, un modelo de predicción basado en una sola ANN, supondrá un coste de cómputo muy inferior que si disponemos de un sistema que requiere sucesivos refinamientos o el empleo de algoritmos genéticos por ejemplo. Es más, si se opta por un único modelo ANN, dependiendo el tipo de modelo escogido el coste computacional de la fase de aprendizaje puede ser elevado. En la Figura 4.32 se muestran los trabajos presentados en esta sección, bajo la clasificación mostrada en la Figura 4.17, y teniendo en cuenta que los sistemas de predicción pueden ser únicos o híbridos (en los sistemas híbridos se indicaran el modelo de base empleado; en este caso híbrido no tiene el mismo significado que en la Figura 4.17); además, se da una indicación del coste computacional que podría tener el sistema global de la predicción.

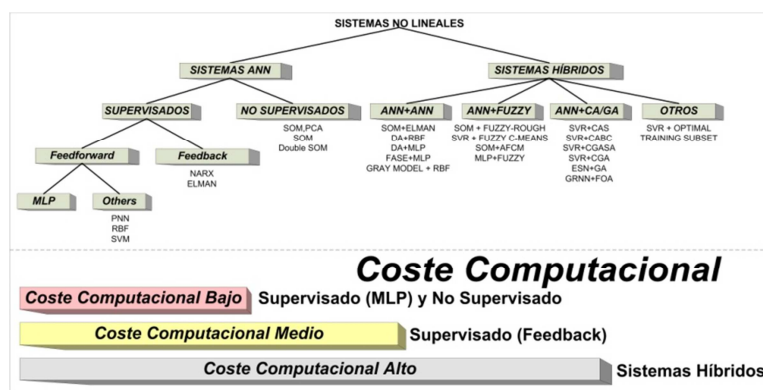


Figura 4.32. Resumen de los modelos no lineales presentados en esta sección. NARX: Nonlinear AutoRegressive with eXogenous; PNN: Probabilistic Neural Network; RBF: Radial Basis Function; MLP: Multi-Layer Perceptron; GRNN: General Regression Neural Network; SVM: Support Vector Machine; SVR: Support Vector Regression; PCA: Principal Component Analysis; SOM: Self-Organizing Map; DA: Deterministic Annealing; FASE: Forecast-Aided State Estimator; AFCM: Adaptive Fuzzy Combination Model; CAS: Chaotic Ant Swarm; CABC: Chaotic Artificial Bee Colony; CGASA: Chaotic Genetic Algorithm Simulated Annealing; CGA: Chaotic Genetic Algorithm; GA: Genetic Algorithm; CA: Chaotic Algorithm; FOA: Fly Optimization Algorithm. Fuente Propia.

### 4.6.3. Trabajos de predicción de la demanda en entornos desagregados:

#### *Antecedentes*

La aparición de nuevos espacios desagregados, con operación y control (subestaciones, *microgrids*, *Smart Buildings*, etc.), unido al despliegue de medida e inteligencia en dichos lugares, posibilitará el desarrollo de herramientas de planificación, que a su vez dependerán de estimadores de la demanda y de la generación.

En cuanto a la previsión de la demanda, con excepciones de obras del tipo a la presentada por Sargunraj *et al.* (1997), donde se habla de subestación desagregada, el resto de obras detalladas (con independencia de su modelo y objetivo de su predicción) tienen en común que tratan de predecir una extensa zona geográfica, como las citadas a continuación (Hsu and Yang, 1991; Taylor and Buizza, 2002; Chu *et al.*, 2011; Nose-Filho *et al.*, 2011; Rejc and Partos, 2011; Wang *et al.*, 2011).

En los últimos años, se está llevando la mirada a estos nuevos espacios de integración de generación y demanda, viendo la necesidad de aplicar modelos de predicción, la mayoría basados en *AI*, para la predicción de la demanda y generación, y así poder operar, tal y como muestran Chaouachi *et al.* (2013).

La *microgrid* vista como un ente único, como su interacción con la *SG*, deberá disponer para su operación y gestión, herramientas de predicción, tanto de sus elementos generadores como de su demanda, para poder actuar en base a ellas, junto a las señales de precios externas a la *microgrid* y la situación de su almacenamiento. Dimeas and Hatzargyriou (2005) mediante una operación con *MAS*, y que pese al avance en técnicas de predicción, la experiencia se tiene en grandes sistemas interconectados y para horizontes de predicción de varias horas, sugiriendo la necesidad de disponer de modelos para entornos desagregados y tiempos inferiores a las horas. Para la correcta operación de la *microgrid*, Khodayar *et al.* (2012) presentan un modelo económico de operación de la *microgrid*, para poder balancear la energía entre la *DG* disponible y el almacenamiento, una visión anticipada de la demanda es necesaria. Sugaya *et al.* (2012) indican que para poder disponer de un control estable en una *microgrid*, se precisa construir una eficiente programación de las plantas de energía existentes en la *microgrid* y, que a su vez, dependerá de la previsión de la demanda. Como ya se intuirá, los diferentes sistemas de gestión de estos entornos desagregados, deberán conocer de manera anticipada, tanto la generación como la demanda a su cargo, como se presenta en (Guan *et al.*, 2010; Chen *et al.*, 2011; Kanchev *et al.*, 2011).

Los trabajos de predicción y clasificación de la demanda en *microgrids* ponen de manifiesto la imperiosa necesidad de aplicar modelos de predicción ya empleados, pero para las

particularidades de estos entornos con demandas desagregadas y curvas de carga con desniveles que hacen la predicción más complicada, como se muestra en (Amjady *et al.*, 2010; Chan *et al.*, 2011).

Técnicas de clasificación y clusterizado de las curvas de carga de polígonos industriales por medio de *SOM+k-means*, para una posterior predicción de la demanda eléctrica, como lo expuesto en (Hernández *et al.*, 2012b), servirá a figuras como las del *agregador* para obtener mejores precios y poder realizar *DR*. Tal y como presentan Kamaev *et al.* (2012), la predicción de la demanda en centros comerciales (*microgrids*) se presenta como fundamental. El gestor de estos entornos deberá estimar el consumo de energía eléctrica con la mayor exactitud, para de esta manera poder hacer órdenes de compra de energía a las *utilities*. Los centros comerciales poseen una programación, que dependerá de las horas de apertura, la afluencia y la estacionalidad. Introduciendo las anteriores variables, se obtienen mejoras en las predicciones, al igual que si se emplea información estimada del día siguiente, tal y como emplearon también Hernández *et al.* (2013b). En el estudio se prueban modelos lineales y no lineales, pudiendo concluir que los modelos basados en *ANN* obtienen unos resultados óptimos, pero a cambio se pierde la posible interpretación del problema, ya que se pierde la referencia del objetivo de las variables seleccionadas con respecto a su relación con la predicción de la demanda. Lo anterior no ocurre si se emplean modelos lineales, donde el trabajo del investigador creando las funciones permitirá la creación de dicho conocimiento.

En estas *microgrids*, el disponer de estas herramientas de predicción de la demanda, no sólo permitirá un ahorro energético al poder realizar compras de energía más óptimas, tal y como se expone en (Kamaev *et al.*, 2012), sino que permitirá integrar fuentes de generación renovables y almacenamiento eléctrico en su medida justa, tal y como muestran Hida *et al.* (2010). La misma responsabilidad de la predicción de la demanda y de la generación, para un correcto funcionamiento de la *microgrid* y un óptimo despacho económico, es asumido por Controlador Central de Microrred (*MicroGrid Central Controller – MGCC*) (Lopes, 2009). Dimeas and Hatziargyriou (2005) y Sechilariu *et al.* (2013) presentan el despacho económico óptimo para una *microgrid*, donde la posibilidad de la máxima penetración de las fuentes de generación renovables y almacenamiento eléctrico pasa por una predicción de la demanda ajustada.

Si se atomiza aún más el problema, aparecen espacios más desagregados con inteligencia asociada, tal y como son los *Smart Building*, *Smart Home*, etc.; ya Skrzypczak (1987) exponía la casa con inteligencia para el 2010, donde el principal reto sería la integración de las áreas: mecánica, electricidad, electrónica, informática y ergonomía; donde la monitorización y control deberían tratar de ajustar parámetros eléctricos para el confort, en base a medidas ambientales y de comportamiento. Costanzo *et al.* (2012) presentan un Gestión del Lado de la Demanda (*Demand Side Management – DSM*) para un edificio, donde el predictor de la carga será un elemento primordial. En (Harris, 2012) se muestra que para automatizar y controlar la energía en un *Smart*



*Building* será preciso poder anticipar su consumo. Sechilariu *et al.* (2013) muestran la necesidad de comunicación entre *SG* y el control de los edificios, para la correcta operación de la *DG* y almacenamiento instalados en estos últimos, los cuales deberán disponer de predicciones de generación y demanda, para poder operar en sus decisiones. Abreu *et al.* (2012) emplean técnicas para el reconocimiento de patrones y así identificar patrones de comportamiento en consumos residenciales, permitiendo hacer predicción de la demanda en entornos desagregados, como por ejemplo en los *Smart Buildings*.

Además, estos entornos desagregados, permitirán realizar estudios más exhaustivos y detallados de la influencia de las variables climáticas sobre la demanda eléctrica, para su aplicación en modelos de predicción de la demanda en aplicaciones para *Smart World* de la energía, como el trabajo presentado por Hernández *et al.* (2012a).

En resumen, la aparición de estos nuevos entornos de futuro, consecuencia de la *SG*, necesitarán de modelos predictivos de la demanda eléctrica bajo su control. Se han presentado algunos trabajos que han abordado la temática, pero a medida que se siga investigando en ellos (*microgrid*, *Smart Building*, etc.), continuarán apareciendo más trabajos. No se debe olvidar que, conceptualmente, en estos entornos la integración de fuentes de generación renovable y/o almacenamiento eléctrico es esencial, por tanto, técnicas de predicción tanto de demanda como generación van a ser determinantes para posibilitar dicha penetración de renovables.

#### **4.6.4. Comparativa**

En esta sección se pretende hacer una comparativa entre los modelos lineales y no lineales para predicción de la demanda eléctrica. Primero se realizará una comparativa entre estos dos grandes grupos, tratando de analizar los rasgos característicos de cada uno de ellos y su aportación para el problema de la predicción de la demanda.

Posteriormente, se mostrará una clasificación de los trabajos presentados, tratando de agrupar las obras por sus rasgos comunes, y en base a la clasificación anteriormente expuesta.

Antes de presentar la comparativa global de los dos grandes grupos de modelos (lineales y no lineales) para la predicción de la demanda, algunas reflexiones de lo visto hasta el momento. Como se comentó en la sección anterior, los modelos de predicción basados en *ANN* posibilitan la reducción de coste intelectual del investigador, ya que la labor de detección de no linealidades entre las diferentes variables y la demanda es delegada sobre la propia red.

El investigador con un conocimiento medio sobre la demanda eléctrica, podrá plantear alternativas diferentes de predicción de la demanda, centrándose tan sólo en desarrollar una metodología completa de test a desarrollar y sus controles para una posterior comparativa. No obstante, como se presentó en (Kamaev *et al.*, 2012). Lo anterior es un arma de doble filo, ya que la

facilidad con que las ANN permiten acometer el problema de la predicción de la demanda, llevará implícito una pérdida de la base de conocimiento del problema.

Además, como apoyo al empleo de modelos de predicción basado en ANN, está el poder disponer de computadores con alto poder de cómputo y algoritmos mejorados. Adicionalmente, la minimización de la función objetivo de la fase de aprendizaje (Rying *et al.*, 2002), la mejora de los algoritmos de entrenamiento en las ANN (Han and Qiao, 2012), la reducción de la complejidad computacional de las ANN (Bragatto *et al.*, 2008), que ya se empezó a tratar Orponen (1994), y por último, la detección automática de arquitecturas de ANN (Islam *et al.*, 2009), postulan a las ANN como modelos a seguir empleando frente al enfoque clásico de los modelos lineales. Lo anteriormente explicado ha propiciado que en la primea década del presente siglo, los investigadores se decanten por los modelos de predicción basados en ANN, presentándose muchos más trabajos que los basados en modelos lineales.

**Tabla 4.5. Comparativa de los rasgos principales de los modelos lineales vs no lineales para la predicción de la demanda eléctrica. Fuente Propia.**

	<i>MODELOS LINEALES</i>	<i>MODELOS NO LINEALES</i>
<i>HORIZONTE DE PREDICCIÓN</i>	<i>VSTLF/STLF/MTLF/LTLF</i>	<i>VSTLF/STLF/MTLF/LTLF</i>
<i>OBJETIVO DE LA PREDICCIÓN</i>	<i>UN VALOR/VARIOS VALORES</i>	<i>UN VALOR/VARIOS VALORES</i>
<i>PÉRDIDA DEL CONOCIMIENTO BASE</i>	<i>NO</i>	<i>ES POSIBLE</i>
<i>FACILIDAD PARA LA DETECCIÓN DE NO LINEALIDADES INTRÍNSECA AL MODELO</i>	<i>NO</i>	<i>SÍ</i>
<i>FACILIDAD DE PARAMETRIZACIÓN</i>	<i>NO</i>	<i>RELATIVAMENTE FÁCIL</i>
<i>NECESIDAD DE AJUSTE INICIAL</i>	<i>SÍ</i>	<i>SÍ</i>
<i>COSTE COMPUTACIONAL</i>	<i>MEDIO/ALTO (DEPENDE COMPLEJIDAD DEL MODELO)</i>	<i>BAJO/MEDIO/ALTO (DEPENDE MODELO)</i>
<i>POSIBILIDAD DE SIST. HÍBRIDOS</i>	<i>SÍ</i>	<i>SÍ</i>
<i>RESPUESTA A GRAN CANTIDAD DE PATRONES HISTÓRICOS</i>	<i>ALTA</i>	<i>ALTA</i>
<i>RESPUESTA A POCA CANTIDAD DE PATRONES HISTÓRICOS</i>	<i>MEDIA</i>	<i>ALTA</i>
<i>TRABAJOS ANTERIORES A 1990</i>	<i>MUCHOS</i>	<i>APENAS</i>
<i>TRABAJOS 1990–2000</i>	<i>MUCHOS</i>	<i>BASTANTE</i>
<i>TRABAJOS POSTERIORES A 2000</i>	<i>APENAS</i>	<i>MUCHOS</i>

En la Tabla 4.5 se presenta un resumen de la comparativa entre los modelos lineales y los no lineales. Los rasgos analizados son mostrados y representan los más críticos para decidir entre un

grupo u otro. Los rasgos principales seleccionados son: horizonte de predicción; objetivo de la predicción; pérdida del conocimiento base; facilidad para la detección de no linealidades intrínseca al modelo; facilidad de parametrización; necesidad de ajuste inicial; coste computacional; posibilidad de sistemas híbridos; respuesta a gran cantidad de patrones históricos; respuesta a poca cantidad de patrones históricos; trabajos anteriores a 1990; trabajos 1990–2000; trabajos posteriores a 2000. La Tabla 4.5 puede servir para decidir entre modelos lineales y no lineales para la predicción de la demanda. Los principales puntos a tener en cuenta para la decisión se resumen en:

- La elección de modelos no lineales frente a los lineales puede llevar consigo una pérdida del conocimiento del problema, ya que al emplear modelos basados en *ANN*, la responsabilidad de la detección de rasgos que definan la complejidad del problema, queda delegada a la propia red, pudiendo el investigador no emplear un tiempo mínimo para la comprensión del objetivo a solucionar. No obstante, la necesidad de proponer unas variables de entrada óptimas al modelo *ANN*, requerirá un estudio mínimo/medio del problema de la demanda eléctrica.
- Para los modelos lineales, el coste computacional vendrá dado por la complejidad de las funciones planteadas, así como el uso de sistemas híbridos. En el caso de *ANN*, el coste computacional quedará definido por la fase de aprendizaje del modelo escogido e igualmente por la decisión de emplear sistemas híbridos o no. Cuestiones relacionadas con el coste computacional y rendimiento del modelo pueden encontrarse en Kaushik *et al.* (2010) para *ANN* y en Chow and Tan (1998) para *ARMA*.
- El despliegue de “*medida inteligente*” en los nuevos entornos de futuro (*SG*, *microgrid*, *Smart Building*, etc.) permitirá disponer de datos con mayor frecuencia para actualizar los modelos. La demanda eléctrica es claramente dependiente de ciertos parámetros exógenos a la misma (del tipo social, económico, etc.), por lo que poder disponer de datos recientes, que lleven implícitos posibles cambios en esos parámetros y, por tanto, en la demanda eléctrica, se antoja muy atractivo. Los modelos no lineales responde mejor a una cantidad pequeña de patrones para ser empleados en la fase de aprendizaje. Por tanto, estos dos motivos hacen pensar que los modelos más empleados de aquí en adelante, sobre todo en los entornos ya citados, serán los no lineales.
- Las conclusiones anteriormente comentadas quedan de manifiesto al ver que la gran cantidad de trabajos de predicción de la demanda desarrollados a partir de 2000, y en concreto en los últimos cinco años, son basados en modelos no lineales.

A continuación, en la Tabla 4.6 se procederá a agrupar y destacar las principales características de los trabajos presentados en esta obra. Los modelos lineales se han agrupado en dos grandes categorías, los que se basan en modelos *ARMA* (1–6) y los que dependen de otros que

no son de la anterior familia (7–9). Los modelos no lineales se ha creído conveniente agruparlos en modelos basado en *ANN* (A–K) y sistemas híbridos (L–Y).

Tabla 4.6. L: carga; L+W: carga + climáticas; L+W+O: carga + climáticas + otras; L+O: carga + otras. MG: microgrid, SB: Smart Building. Fuente Propia.

		MODELOS NO LINEALES										MODELOS LINEALES																											
		Y	X	W	V	U	T	S	R	Q	P	O	Ñ	N	M	L	K	J	I	H	G	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1			
AREA	PAÍS, REGIÓN, CIUDAD	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
	PEQUEÑA CIUDAD, MG, SB									<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>						<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
HORIZONTE	VSTLF	<input checked="" type="checkbox"/>								<input checked="" type="checkbox"/>					<input checked="" type="checkbox"/>					<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>						<input checked="" type="checkbox"/>												
	STLF	<input checked="" type="checkbox"/>								<input checked="" type="checkbox"/>					<input checked="" type="checkbox"/>					<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>						<input checked="" type="checkbox"/>												
	MTLF				<input checked="" type="checkbox"/>				<input checked="" type="checkbox"/>						<input checked="" type="checkbox"/>			<input checked="" type="checkbox"/>															<input checked="" type="checkbox"/>						
	LTLT						<input checked="" type="checkbox"/>																<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>															
OBJETIVO	UN VALOR						<input checked="" type="checkbox"/>								<input checked="" type="checkbox"/>			<input checked="" type="checkbox"/>									<input checked="" type="checkbox"/>												
	VARIOS VALORES	<input checked="" type="checkbox"/>			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>				<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>						<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>				<input checked="" type="checkbox"/>													
VARIABLES	L	<input checked="" type="checkbox"/>																																					
	L+W																																						
	L+W+O																																						
	L+O																																						

De la totalidad de los agrupamientos se ha mostrado sus principales características según lo visto anteriormente, siendo las siguientes: área de aplicación (país o región y *microgrid* o *Smart Building*); horizonte de predicción; objetivo de la predicción; y variables principales empleadas.

En la Tabla 4.7 se muestra cada uno de los grupos, indicando las referencias que pertenecen a cada uno de ellos. En las filas de la Tabla 4.7 se indica cada uno de los grupos y en las columnas las dos clasificaciones fundamentales de los modelos: lineales y no lineales.

**Tabla 4.7. Grupos formados, indicando su modelo de pertenencia así como las referencias que lo componen.**

**Fuente Propia. (Continúa).**

<b>GRUPO</b>	<b>MODELO LINEAL</b>	<b>MODELO NO LINEAL</b>
<b>1</b>	Belik <i>et al.</i> (1978); Abu–El–Magd and Sinha (1981); Schneider <i>et al.</i> (1985); Huang and Shin (2003); Chen <i>et al.</i> (2006); Baharudi and Kamel (2008); Wang <i>et al.</i> (2009).	–
<b>2</b>	Paysti (1984); Vemuri <i>et al.</i> (1986); Hagan and Behr (1987); Fan and McDonald (1994); Yang <i>et al.</i> (2005); Abou–Hussien <i>et al.</i> (1981);	–
<b>3</b>	Ernoul and Mattatia (1984); Gross and Galiana (1987)	–
<b>4</b>	Fadhilah <i>et al.</i> (2009)	–
<b>5</b>	Rajurkar and Nissen (1985)	–
<b>6</b>	Keyhani and Miri (1983); Kamaev <i>et al.</i> (2012)	–
<b>7</b>	Anelli <i>et al.</i> (1978); Quintana <i>et al.</i> (1978); Rahman and Bhatnagar (1987); Irisarri <i>et al.</i> (1982); Fankhauser (1984); Campo and Ruíz (1987)	–
<b>8</b>	Farmer and Potton (1968)	–
<b>9</b>	Meng and Shang (2008)	–
<b>A</b>	–	Charytoniuk and Chen (2000); Nair and Joshi (2010)
<b>B</b>	–	McMenamin and Monforte (1998)
<b>C</b>	–	Kamaev <i>et al.</i> (2012)
<b>D</b>	–	Zhang and Ye (2011)
<b>E</b>	–	Li <i>et al.</i> (2012)
<b>F</b>	–	Sharda and Patil (1990); Tang <i>et al.</i> (1991); Lee <i>et al.</i> (1992); Badri (1996); Chicco <i>et al.</i> (2004); Carpinteiro and Reis (2005); Tsekouras <i>et al.</i> (2007); García–Ascanio and Mate (2009); Zhang <i>et al.</i> (2012a)

Tabla 4.7. Fuente Propia.

<b>GRUPO</b>	<b>MODELO LINEAL</b>	<b>MODELO NO LINEAL</b>
<b>G</b>	–	Drezga and Rahman (1988); Hsu and Yang (1991); Park <i>et al.</i> (1991); Lu <i>et al.</i> (1993); Papalexopoulos <i>et al.</i> (1994); Bakirtzis <i>et al.</i> (1995); Drezga and Rahman (1999a); Mori and Iwashita (2005); Sun <i>et al.</i> (2005); Su and Zhang (2007); Chu <i>et al.</i> (2011); Yongchum (2010)
<b>H</b>	–	Hernández <i>et al.</i> (2012b); Hernández <i>et al.</i> (2013b)
<b>I</b>	–	Hong <i>et al.</i> (2011a)
<b>J</b>	–	Taylor and Buizza (2002)
<b>K</b>	–	Nose-Filho <i>et al.</i> (2011)
<b>L</b>	–	Guan <i>et al.</i> (2013)
<b>M</b>	–	Ho <i>et al.</i> (1990); Ho <i>et al.</i> (1992)
<b>N</b>	–	Alfuhaid <i>et al.</i> (1997)
<b>Ñ</b>	–	Abreu <i>et al.</i> (2012)
<b>O</b>	–	Mohammed <i>et al.</i> (1995); Piras <i>et al.</i> (1996); Mori and Yuihara (2001); Mori and Kanaoka (2006); Chan <i>et al.</i> (2011); Rejc and Partos (2011); Wang <i>et al.</i> (2011); Deihimi and Showkati (2012); Lou and Dong (2012)
<b>P</b>	–	Marin <i>et al.</i> (2002); Wang <i>et al.</i> (2009); Che <i>et al.</i> (2012b)
<b>Q</b>	–	Sargunraj <i>et al.</i> (1997)
<b>R</b>	–	Wang (2006); Hong (2011b); Hong <i>et al.</i> (2013)
<b>S</b>	–	Doveh <i>et al.</i> (1999); Zhang <i>et al.</i> (2012b)
<b>T</b>	–	Kandil <i>et al.</i> (2002)
<b>U</b>	–	Hong (2010)
<b>V</b>	–	Lamedica <i>et al.</i> (1996); Che <i>et al.</i> (2012a)
<b>W</b>	–	Nadimi <i>et al.</i> (2010)
<b>X</b>	–	Taylor and Buizza (2002)
<b>Y</b>	–	Daneshi <i>et al.</i> (2008); Li <i>et al.</i> (2013)

Se observa como existen pocos grupos que han dedicado sus esfuerzos a aplicar las técnicas de predicción de la demanda en entornos desagregados (*microgrid*, *Smart Building*, ciudad pequeña, pueblo, polígono industrial, etc.). En concreto, los grupos que modelan estos entornos son: 5, 6, C, H, N y Q. Dos de ellos se corresponden a modelos lineales y tres a no lineales. Los dos grupos de los modelos lineales están formados por 3 trabajos, que suponen tan sólo un 12% del total de los trabajos presentados. Con respecto a los grupos C, H, N y Q, agrupan un total de 6 trabajos, que suponen tan sólo un 10% del total de los trabajos realizados. Del total de los 9 trabajos, 3 de ellos

corresponden a finales de la década de 1990, y los 6 restantes se reparten entre 2012 y 2013. El despliegue de entornos desagregados está comenzando en los últimos 5 años, por tanto, números trabajos relacionados con la predicción de la demanda eléctrica y la generación asociada comenzarán a presentarse.

Con respecto al horizonte de predicción, el 6% de los grupos analizados abordan *VSTLF*, el 58% de los grupos analizados abordan *STLF*, el 20% de los grupos analizados abordan *MTLF* y el 16% de los grupos analizados abordan *LTLF*. De los grupos de los modelos lineales, 7 corresponden a *STLF*, 1 de *MTLF*, 1 de *LTLF*, mientras que de *VSTLF* no se han analizado.

Con respecto al objetivo de la predicción, 15 grupos corresponden a un valor, mientras que 19 predicen varios valores. Existe discrepancia en los modelos lineales, donde tan sólo 1 grupo se ha encargado de un valor y el resto predicen varios valores. Los grupos de los modelos no lineales se distribuyen entre ambas alternativas por igual.

Como ya se explicó, el decantarse por el horizonte de la predicción y su objetivo dependerá del problema al que el investigador se enfrente. Por tanto, la tendencia de los trabajos futuros seguirá pautas parecidas, apareciendo obras en todos y cada uno de las posibles alternativas en cuanto a horizonte y objetivo de predicción.

Con respecto a las variables empleadas, el 47% de los grupos emplean modelos que tan sólo emplean valores de L (junto a variables de calendario). El 32% de los grupos emplean variables de L junto a W (climáticas). El 9% de los grupos emplean variables de L y O (otras). El 12% de los grupos emplean variables de L, W y O.



## CAPÍTULO V. SISTEMAS MULTI-AGENTES

Lo presentado hasta el Capítulo IV debería haber dejado patente la importancia de modelos de predicción de la demanda eléctrica, los cuales necesitan comenzar a aplicarse en entornos desagregados, como por ejemplo una *microgrid*, *VPP* o *Smart Building*. Completando a lo anterior, en el Capítulo VI se propondrá un modelo teórico-conceptual basado en *MAS*, que intentará controlar un ambiente heterogéneo en una *VPP* para hacer predicción de la demanda eléctrica. Por este motivo, en este Capítulo V se darán los conceptos fundamentales alrededor de la teoría de agentes, tratando de destacar sus peculiaridades e idoneidad en estos entornos de futuro que son las *VPP* y sus retos como el control de la demanda eléctrica por medio de predicciones.

Se comenzará presentando la historia reciente en torno a los agentes. A continuación, se presentarán las diferentes arquitecturas que despliegan agentes. Seguidamente, se presentará el concepto de comunicación entre agentes, para continuar con las comunicaciones entre ellos. Para finalizar, se presentarán las herramientas y métodos que apoyan el despliegue de agentes.

### 5.1. LA TECNOLOGÍA DE AGENTES: ORIGEN Y VENTAJAS

El mundo de la informática y las comunicaciones suele acuñar un innumerable vocabulario, cuyo significado suele escaparse al conocimiento exacto de la población. A pesar de que el término agente evoca algo más rico que por ejemplo “*objeto*” (programación orientada a objetos), no deja de ser un vocablo abstracto y de difícil relación con la programación. Sin embargo, debe tenerse en cuenta, que al igual que sucedió con la programación orientada a objetos, se ha desarrollado una tecnología clara alrededor de los agentes, cuyo objetivo es consolidarlos y mejorar la producción *Software*.

Los agentes pueden funcionar en casi cualquier dispositivo (computador, teléfono móvil, etc.), y se presentan como entidades computacionales con comportamiento casi humano. De manera aislada o colaborando con otros agentes, emplean sus capacidades para tratar de resolver problemas de forma inteligente y de manera proactiva.

La tecnología de agentes recoge los frutos de tres décadas de trabajo, la mayoría científicos, y de desarrollo de ingeniería. Resulta una tecnología abierta y cuyos métodos y principios son compatibles con el resto de tecnologías. Se alimenta de diferentes tecnologías consolidadas, por un lado la Ingeniería del Software y por otro la *AI*.

El desarrollo de herramientas o aplicaciones mediante tecnología basada en agentes tiene algunas ventajas frente a otras tecnologías:

- Mayor funcionalidad y claridad, ya que los sistemas actuales suelen ser poco intuitivos, rígidos, inestables y caros.

- Reducido mantenimiento y facilitan su transformación y evolución.
- Su reusabilidad hace que el coste del producto disminuya.
- Integración con otras tecnologías.
- Simplicidad de ingeniería, ya que permite centrarse en el comportamiento del agente, dejando la labor de codificación en un segundo plano.

### **5.1.1. El concepto de Agente. Inicio y evolución de la tecnología**

Como bien expone Mas (2005), en todos los tratados de agentes se comienza diciendo que su definición es algo complicado y confuso. Nos recuerda su procedencia del latín “*agere*” cuyo significado es “*hacer*”, expresando de esta manera la capacidad de acción o actuación de cierta entidad.

Según la RAE, un agente es: “*persona o cosa que produce un efecto*”; “*persona que obra con poder de otra*”. Debería existir cierto complemento de la parte física (por ejemplo agente humano o biológico), contexto de actuación (por ejemplo agente social o comercial) o su comportamiento (por ejemplo agente patógeno o decisor), y así poder disponer de un significado más preciso.

Parece lógico tratar de encontrar características diferenciadoras de agente sobre el resto de entidades *Software*. Es posible encontrar definiciones por *Internet* que no prestan atención al detalle comentado anteriormente, pudiendo encontrar la siguiente definición: “*sistema software o hardware que funciona de forma autónoma y puede interactuar con su entorno y/o con otros agentes*”. Esta definición podría valer para una planta de producción con autómatas programables, o incluso, para un computador personal conectado a *Internet*.

Otras definiciones confusas (*ircbeginner*) indican que “*los agentes son herramientas de búsqueda que encuentran de manera automática información de la red según las especificaciones del usuario*”. Pero según esta definición, cualquier buscador de *Internet* sería un agente por definición.

Una de las definiciones más usadas es la de Wooldridge (1997): “*un agente es un sistema informático situado en un entorno y que es capaz de realizar acciones de forma autónoma para conseguir sus objetivos de diseño*”. Evidentemente la autonomía es una característica fundamental en los agentes, pero con tan sólo esto, no se permite diferenciarlo de un sistema distribuido convencional.

Otros autores, como Garijo (2002), se centran en destacar las características con respecto al comportamiento del agente y ciertos requisitos de arquitectura, resultando la siguiente definición: “*agente caracteriza a una entidad software con una arquitectura robusta y adaptable que puede funcionar en distintos entornos o plataformas computacionales y es capaz de realizar de forma inteligente y autónoma distintos objetivos intercambiando información con el entorno, o con otros*

*agentes humanos o computacionales*”. Siguiendo con el mismo autor, en cuanto a su comportamiento, las características deseables son:

- Operación continua y autónoma.
- Comunicaciones con su entorno y otros agentes.
- Adaptabilidad para la realización de objetivos y tareas en distintos escenarios, de forma flexible e incremental.
- Robustez.

Otros atributos importantes en los agentes, aunque no todos los poseen, son:

- Movilidad, los agentes móviles se pueden desplazar por los nodos de la red y ejecutarse en diferentes plataformas.
- Razonamiento y aprendizaje, fundamentales para que los agentes se comporten de manera inteligente.

Por tanto, y a pesar de que poco a poco las diferentes tendencias alrededor de los agentes comienzan a converger, se precisa una clara definición de los elementos diferenciadores de los agentes, principalmente requerido por la ingeniería. Para tratar de entender esta situación, se darán algunos datos históricos sobre agentes.

#### La primera etapa

Los primeros aportes aparecen a finales de la década de los setenta y comienzos de los ochenta, donde cooperan múltiples sistemas para la resolución, de manera inteligente, de problemas complejos (Erman and Lesser, 1975; Smith, 1980; Lesser and Corkill, 1981; Kornfeld and Hewitt, 1981).

Hasta esa fecha, los científicos se habían centrado en el estudio de los componentes de la inteligencia, tales como el razonamiento y la resolución de problemas. Es durante la década de los setenta y comienzos de los ochenta cuando se presentan los sistemas expertos, con cierta capacidad de razonamiento y resolución de problemas, de manera semejante a la de los humanos (Hayes–Roth *et al.*, 1983). La industria norteamericana potenció los sistemas expertos, financiando gran cantidad de proyectos basados en este enfoque, mientras tanto, los japoneses lanzan la quinta generación de computadores (Moto–oka, 1983), siendo máquinas con capacidad de procesamiento simbólico, con posibilidad de implementación de funciones inteligentes.

Mientras se aplica la *AI*, comienzan los primeros debates sobre las limitaciones de los sistemas expertos y de la tecnología del conocimiento. Los puntos más importantes serán:

- La resolución de problemas de los sistemas expertos se centra en dominios concretos y a tareas determinadas, por tanto, los sistemas expertos tiene una inteligencia limitada.
- Resulta complejo aumentar la inteligencia de los sistemas expertos, ya que dependerá del esfuerzo de ingeniería y análisis dedicado.

- En casi todos los sistemas expertos, es fundamental el intercambio de información con el usuario u otras aplicaciones, pero no son capaces del intercambio con otros sistemas expertos.

### Inteligencia Artificial Distribuida

Para tratar de paliar las limitaciones de los sistemas expertos surge el concepto de cooperación; cuando un sistema experto no dispone de conocimiento, tan sólo debe acudir a otro experto que se lo facilite, o que directamente le resuelva el problema. La anterior idea no está carente de ciertas complicaciones: es necesario saber a qué experto acudir; una vez localizado el experto, es posible que no quiera o pueda colaborar; aun colaborando, es posible que el experto consultado formule nuevos problemas o preguntas a contestar; y por último, es posible que durante la espera, el primer experto puede ser interrumpido por un tercero y no prestar atención a su cometido inicial.

Por tanto, es necesario un nuevo sistema experto con al menos, los siguientes elementos:

- Capacidad para resolver problemas en el dominio, para así poder cooperar.
- Conocimiento de las entidades con las que cooperar, saber buscarlas, y saber sus capacidades y limitaciones.
- Capacidad de comunicación para así poder intercambiar peticiones y respuestas.
- Un modelo global de cooperación, posibilitando coordinar sus capacidades con las del resto, de forma que se complementen las actividades.

Se planteaban tres grandes retos a los investigadores, de cara a la resolución de problemas complejos mediante sistemas inteligentes cooperativos, siendo estos retos los siguientes:

- Planteamiento de una nueva arquitectura para que estos sistemas de resolución de problemas cooperasen entre ellos.
- Conseguir modelos de cooperación que pudieran ser interpretados por los sistemas de resolución de problemas.
- Validar y probar los sistemas cooperantes frente a otras soluciones.

A raíz de lo anterior, surgen dos enfoques, uno concibe la cooperación desde el punto de vista de entidad colaboradora, incorporando la cooperación como un mecanismo adicional para aumentar la capacidad individual de resolución de los problemas. El segundo enfoque, presenta la cooperación desde el punto de vista del problema a resolver, dado un problema complejo y un conjunto de sistemas de resolución de problemas, se trata de determinar la mejor manera de estructurar dicho problema y tratar de descomponerlo para resolverlo de manera eficaz. Este último enfoque tiene similitudes con la problemática alrededor del procesamiento distribuido y el paralelismo, donde se pretende repartir la carga de cómputo entre los procesadores disponibles. Con

independencia del enfoque, esta nueva área de actividad de trabajo dentro de la *AI*, se comenzó a llamar *Inteligencia Artificial Distribuida (IAD)*.

El modelo presentado por Erman *et al.* (1980) permite la cooperación entre fuentes de conocimiento heterogéneas, mediante un mecanismo sencillo de comunicación, “*la pizarra*”. Cuando existen datos en la pizarra, la fuente de conocimiento capaz de resolver un problema, los toma, da una solución al mismo, y vuelve a colocar los datos en la pizarra.

Lesser and Corkill (1981) realizaron experimentos relacionados con modelos de cooperación y su aplicación al reconocimiento del habla continua, obteniéndose mejoras en el tiempo de reconocimiento del orden del diez por ciento y mayor robustez ante errores.

Hasta esa época no aparecía el término agente, en su lugar se identificaban *nodos de resolución de problemas*, pero con la arquitectura de pizarra, comienzan las primeras confusiones terminológicas, comenzando a llamar agentes a las fuentes de conocimiento, que en definitiva, son las encargadas en obtener soluciones a partir de datos.

Continuando con los trabajos de Lesser, se comenzaron a emplear estas técnicas en la fabricación, por medio de la llamada *red de contratos* (Parunak, 1990); para la detección, clasificación y seguimiento de vehículos desplazándose por una zona determinada (Lesser and Corkill, 1983); y para el control de tráfico aéreo, planificando entradas y salidas en el espacio aéreo (Cammarata *et al.*, 1988). A consecuencia de estos trabajos y algunos otros, se consolida el término agente, no importando tanto qué son, sino destacando el comportamiento colectivo para la resolución de problemas.

En contraposición a las capacidades cognitivas, Brooks (1991) presenta agentes totalmente reactivos, donde reciben eventos del entorno y realizan acciones en base a ellos y el estado del agente. Emplea autómatas de estados finitos como mecanismos de control del agente. Este nuevo enfoque tuvo muy buena acogida en simulación de sistemas biológicos, sociología y la gestión empresarial.

Pero los trabajos de Lesser seguían planteando retos, y surgieron los modelos que planteaban inspirarse en organizaciones humanas y biológicas, para tratar de aplicar las teorías de la organización, como se muestra en (Galbraith, 1973; Fox, 1979). En los modelos organizativos se detallan las relaciones entre los agentes, el tipo de comunicación que deben emplear, el momento preciso de hacerlo y qué tipo de información a usar. A partir de ese momento se precisan nuevos instrumentos de análisis que permita el desarrollo de modelos organizativos y la experimentación con dichos modelos organizacionales.

Se estudian los principales elementos del proceso de comunicación, como el emisor, receptor, protocolo, paradigma y la semántica. Los elementos anteriores se agrupan en tres categorías fundamentales: sintaxis, semántica y el paradigma de la comunicación.

Con lo expuesto hasta este momento, a finales de la década de los 80 nos encontramos con el siguiente panorama:

- No existe todavía una definición precisa de agente, siendo un sistema con ciertas habilidades, como inteligencia, resolución de problemas y cooperación. Los sistemas se comienzan a modelar como organizaciones de agentes o *MAS*.
- Existen problemas de cooperación, conocimiento, teorías y herramientas.
- Se mantienen los lazos con la *AI*.
- La industria da la espalda en cierta forma a la ingeniería del conocimiento, además, la quinta generación de computadores no verá la luz, sin embargo sí las estaciones de trabajo y la ingeniería del *Software* orientada a objetos.

#### Consolidación en la década de los noventa

La década de los 90 se caracteriza por el despegue de *Internet* y las empresas que realizan aplicaciones o proliferan a través de él. Los agentes, entre otras tecnologías, aparecen como fundamentales para poder hacer posible la visión del comienzo del siglo XXI.

La comunidad científica organiza congresos centrándose en agentes y *MAS*. En Europa, se forma un grupo reducido pero activo, tratando de unificar y diferenciar los términos agente y *MAS*, de la *AI* e incluso de los sistemas distribuidos, aunque se reconocen muchos puntos comunes.

Comienzan a aparecer prototipos de servicios apoyados por la industria, se mejoran las arquitecturas, modelos de razonamiento y la comunicación. Aparecen igualmente los agentes móviles y las primeras pruebas de la industria para resolver sistemas complejos del mundo real.

A finales de los noventa comienzan a surgir grupos de investigación que plantean el uso de agentes en nuevas áreas de aplicación. El grupo de Sycara and Zeng (1996) planteó el Asistente Visitante (*visitor assistant*), donde el sistema trata de organizar el trabajo por medio de la información proveniente del visitante, para conseguir esto, el agente deberá contactar con otros similares, para tratar de cuadrar las agendas comunes. Desde el *MIT*, Maes (1995) trabaja sobre agentes que dialogan con el usuario en lenguaje natural y aprenden del usuario; desde el mismo grupo, presentan agentes en comercio electrónico, que ayudan al comprador a obtener los productos que buscan de forma sencilla y flexible y, además, aprenden sus preferencias y ayudan a obtener mejores precios creando coaliciones con otros agentes para hacer labores de mayoristas (Maes *et al.*, 1999).

La movilidad tiene que ver con la capacidad de un agente para moverse entre los nodos para la realización de determinada tarea. Es justificable por la eficiencia, supongamos la necesidad de información y empleo de manera intensiva de recursos remotos, el agente puede desplazarse donde están los recursos y evitar tediosas comunicaciones. Otra justificación es la implementación dinámica de funcionalidades nuevas, si el agente precisa una nueva funcionalidad, puede pedírsela a

un agente de un nodo remoto. Para conseguir la movilidad, el agente debe desplazarse a otro nodo conservando su estado, esto es, partir de un nodo con ciertas tareas a resolver y datos sobre el problema, y deberá volver con el trabajo realizado y los resultados conseguidos. Lo anterior se consigue realizando los siguientes pasos: *desmaterialización* del agente; *transmisión* del agente entre nodos; *materialización* del agente en el nodo destino; y *activación* del agente en el nodo destino.

A comienzos de los 90 se comienza a hablar de programación orientada a agentes como un novedoso paradigma de desarrollo de *Software*. El apoyo de la industria hace que surja el lenguaje Consulta de Conocimiento y Lenguaje de Manipulación (*Knowledge Query and Manipulation Language – KQML*), proporcionando un formalismo de alto nivel para expresar actos comunicativos y el intercambio de conocimiento empleando Formato de Intercambio de Conocimiento (*Knowledge Interchange Format – KIF*). El anterior lenguaje será adoptado por los grupos americanos, mientras que los europeos apoyan un nuevo lenguaje, Lenguaje de Comunicación de Agente (*Agent Communication Language – ACL*) bajo la tutela de Fundación para Agentes Inteligentes Físicos (*Foundation for Intelligent Physical Agents – FIPA*), que aglutinaba grupos industriales y de investigación para labores de estandarización de modelos y tecnologías agentes.

Desde los grupos de investigación, y centrados en la fase de análisis, surgen las primeras propuestas metodológicas para construir *MAS*. Se desarrollan notaciones para tratar de modelar la arquitectura y la funcionalidad del sistema a construir, empleando algunos conceptos básicos de la tecnología: agente, rol, responsabilidad, iniciativa, actividad, objetivo, tarea, etc.; se darán métodos para encontrar los agentes adecuados e indicaciones genéricas para la implementación.

#### Agentes a los finales de los noventa

La clasificación de los agentes dependerá de la observación de los mismos. Para comenzar, hay que prestar atención a las características individuales; posteriormente observar su entorno y comportamiento; y por último, considerarse su utilidad.

Demazeau (2001) propuso una clasificación basado en las vocales *AEIOU*, donde la *A* se refiere a agente, y caracteriza los rasgos individuales (arquitectura, funcionamiento, complejidad, etc.); la *E* se refiere al entorno, y caracteriza los requisitos computacionales para el funcionamiento del agente; la *I* tiene que ver con la interacción del agente, con quién se comunica y su forma de comunicación; la *O* de organización, considera el papel del agente en el conjunto del sistema; la *U* de utilidad, la aplicación de la que forma parte el agente.

Los agentes según sus características individuales (*A*) pueden clasificarse en *reactivos* y *cognitivos*. Los agentes *reactivos* se encargan de tareas sencillas, siendo su modelo computacional un ciclo recepción de eventos *externos/reacción*. La reacción consistirá en la ejecución de

procedimientos según algún estado interno del agente, no realizando procesos de razonamiento, ni existe un mecanismo explícito de representación del conocimiento, el mecanismo de control serán autómatas de estados finitos extendidos. Por tanto, en estos agentes, el modelo computacional es la Hebra (*thread*).

Los agentes *cognitivos* suelen realizar tareas complejas, empleando algún tipo de representación del conocimiento. Las tareas requieren procesos de razonamiento y procesos cognitivos como la planificación y el aprendizaje. Las arquitecturas *cognitivas* tienen como núcleo algún tipo de procesador de conocimiento que integra la información procedente del entorno y controla el comportamiento interno y las acciones del agente de acuerdo con el conocimiento de éste. El modelo computacional se basa en un ciclo *percepción–asimilación–razonamiento–actuación*.

A pesar de las diferencias entre *reactivos* y *cognitivos*, ambos pueden ser representados por una arquitectura común, ya que reciben información del entorno y disponen de un mecanismo de control que dirige el ciclo de funcionamiento y decide la ejecución de acciones internas o externas. El elemento distintivo será el mecanismo de control, en los *reactivos* podrá ser un autómata, una ANN o ambos; en los *cognitivos* el control es un procesador de conocimiento que admite más variedad de modelos computacionales. El formalismo de representación del conocimiento puede ser la lógica de predicados, lógica modal, o incluso objetos, ontologías y procedimientos. En cuanto a la resolución de problemas, se puede basar en alguna de las aproximaciones siguientes: resolución basada en objetivos; resolución dirigida por creencias o eventos; y la planificación.

El modelo Creencia–Deseo–Intención (*Belief–Desire–Intention – BDI*) (Rao and Georgeff, 1991) representa una solución híbrida que combina formalismos de representación y resolución. Esta arquitectura dispone de diferentes módulos, entre ellos una base de conocimiento donde se modelarán las creencias del agente. A partir de estas creencias, el agente genera deseos, modelados como objetivos, siendo la resolución de estos objetivos llevada a cabo por planificación.

Según el entorno (*E*) donde funcionan los agentes, podemos encontrarnos con aquellos que emplean toda la infraestructura computacional que le rodea, el propio sistema operativo, protocolos de comunicación, entre otros. Desde este punto de vista se pueden encontrar los agentes que requieren un entorno específico (plataforma especial) y los que se ejecutan en las plataformas existentes.

Con respecto al modo de interacción (*I*), se distinguen tres tipos de interacciones: *agente–agente*; *agente–entorno*; y *agente–persona*. Para cada una de ellas se emplean diferentes tecnologías, protocolos y lenguajes, que tendrán un impacto con respecto al agente y la arquitectura. Las comunicaciones *agente–agente* pueden llevarse a cabo mediante lenguajes estándar de comunicación entre agentes, como *ACL* o *KQML*. También puede implicar el uso de sistemas



distribuidos y telecomunicaciones, como *CORBA*, *HTML* o *SOAP*. La interacción *agente-entorno* comprende la comunicación con distintos tipos de elementos, los cuales recibirán o emitirán información de manera espontánea (base de datos, servidores vocales, librerías, pilas, interfaces de aplicaciones, etc.). Las comunicaciones *agente-persona* se harán empleando los medios necesarios para el entendimiento entre las partes, por medio de sensores y efectores adecuados, o con lenguajes semiformales, gráficos, animaciones, entre otros.

Desde el punto de organización (*O*), se pueden considerar a los agentes como entes individuales o con un rol dentro de un *MAS* con un objetivo común. Por tanto, estaremos ante agentes *individuales* y agentes *cooperantes*, pudiendo realizar tareas de manera aislada o junto a otros agentes. Dentro de los *cooperantes*, podrán tener una representación del modelo de manera *implícita* (mediante reglas, en agentes *reactivos*) o *explícita* (definición de roles, responsabilidades, etc.; agentes *cognitivos*); además, el rol del agente podrá ser fijo o variable, siendo los primeros *estáticos* y los segundos *evolutivos*; y por último, podrán tener conflictos *internos* o *externos*.

Con respecto a la utilidad (*U*) se utilizará el criterio del *dominio de aplicación* y el del *tipo de tarea* dentro de él. Los dominios de aplicación han crecido en los últimos años, siendo algunos de ellos: el comercio electrónico, telecomunicaciones, comercio, fabricación, entretenimiento, energía, entre otros. Las tareas posibles a realizar por los agentes, podrán ser especializadas como por ejemplo la monitorización, control, búsqueda de información, asesoría; tareas de mediación como por ejemplo en las bases de datos o interfaces de usuarios; y las tareas de coordinación entre agentes y de resolución de conflictos.

### ***5.1.2. Y a partir del siglo XXI...¿Qué?***

El nuevo siglo comienza con cierta crisis alrededor del estallido de la burbuja tecnológica, desapareciendo gran número de empresas que trataban de crecer alrededor de las nuevas tecnologías. La inversión en innovación se hace cada vez más selectiva y se endurecen las exigencias de resultados a corto plazo.

Con este panorama, los trabajos alrededor de los agentes intentan presentar las ventajas de los mismos en cuanto a costes, calidad, integración e interoperabilidad con otros sistemas, así como la escalabilidad de los mismos y el mantenimiento. Estos aspectos son estudiados en Garijo *et al.* (2004), entre otros trabajos, demostrando que pueden obtenerse ganancias en torno al sesenta y cinco por ciento respecto a los desarrollos anteriores, reduciéndose los ciclos de prueba y corrección de fallos.

A pesar de los anteriores resultados, el empleo de tecnología es muy limitado, el salto de prototipos a sistemas utilizables es muy limitado, presentando numerosos retos:

- Las definiciones de agentes deberán materializarse en términos de ingeniería. El agente debe identificar a una entidad computacional con un forma y un comportamiento bien definido, y con mayor nivel de abstracción que los otros elementos usuales en programación, como los objetos, librerías, etc.; por tanto, se pretende asociar al concepto de agente un patrón arquitectónico que especifique de manera clara: forma externa; estructura interna; comportamiento básico y específico; y su entorno computacional.
- El concepto agente no puede depender de una plataforma propietaria, deberá poderse ejecutar en cualquier entorno computacional, incluido dispositivos de usuario.
- El desarrollo con agentes deberá apoyarse en los modelos y buenas prácticas de ingeniería del *Software*. Deberán cubrir todos los aspectos del ciclo de desarrollo, desde los requisitos hasta la implantación y las pruebas, incluso los aspectos no funcionales como prestaciones, instalaciones, configuración, actualizaciones, seguridad y su gestión.
- Se debe poder pasar del modelo conceptual teórico a modelos de ingeniería con la finalidad de apreciar las limitaciones del sistema y detectar fallos.
- Se deberá intentar conseguir aplicaciones baratas, útiles, robustas e inteligentes.

## 5.2. ARQUITECTURA DE AGENTES

Al surgir la necesidad de la organización entre agentes, aparece el concepto de *MAS*, cuya problemática se detallará en la sección 5.2.3; a modo de resumen, la arquitectura determinará los mecanismos que emplea un agente para poder reaccionar a estímulos, poder actuar y realizar comunicaciones. Como exponen Wooldridge and Jennings (1995) existirán tantas propuestas de arquitecturas como equipos de investigación centrados en la temática quieran plantear.

La arquitectura seleccionada para estos sistemas marcará la descomposición de los agentes en un conjunto de módulos que interactuarán entre sí para la obtención de las funcionalidades especificadas. La principal diferencia entre arquitecturas radicará en el método de descomposición del trabajo en tareas particulares. Se debe recordar que la planificación es un área estrechamente ligada al mundo de la agencia, donde en ésta área se centra en la organización la ejecución de las acciones, siendo un agente un sistema ejecutor de acciones en un determinado entorno.

Los sistemas de planificación, para la consecución de sus objetivos básicos, elaboran un plan, empleando para ello modelos de representación del conocimiento y razonamiento de tipo simbólico. La desventaja es cuando se emplean en problemas de tiempo real, ya que requieren gran cantidad de tiempo. Esta crítica está enfocada principalmente al modelo simbólico, por lo que las búsquedas se orientaron a los modelos de representación o razonamiento, como los modelos *reactivos* o *híbridos*.

Previo a la introducción de los tres tipos de arquitecturas de la presente arquitectura, y de presentar *MAS*, se ha creído conveniente en la siguiente sección aclarar los agentes como sistemas intencionales.

### **5.2.1. Agentes como sistemas intencionales**

Los humanos a menudo empleamos afirmaciones del tipo: “*Pepe lleva una chaqueta en el brazo porque considera que puede hacer frío más tarde*”; esta manera de estructurar las ideas es empleada por la psicología para tratar de estimar el comportamiento humano y darle una explicación por medio de actitudes mentales, como las creencias, deseos, anhelos, etc., como encontramos en las obras siguientes (Grosz and Sidner, 1990). Según Dennett (1987) se entiende por sistema intencional a aquellas entidades “*cuyo comportamiento se puede predecir por el método de atribuirles creencias, deseos y perspicacia racional*”.

Según Seel (1989), es posible describir cualquier cosa desde una postura intencional. No tanto en sistemas sencillos, sí en sistemas complejos las nociones intencionales proporcionan herramientas de abstracción que proporcionan una manera correcta y utilizable de estimar y explicar el comportamiento de los sistemas.

Un agente puede verse como un sistema que contiene actitudes que proveen de información y pro-actitudes (Bratman, 1990). El conocimiento del agente con respecto su entorno tendrá que ver con la actitud de información, mientras que las pro-actitudes son las que guiarán a las acciones del agente. No existe una combinación única para las actitudes de información y pro-actitudes, pudiendo encontrar una lista variada de actitudes y pro-actitudes en (Rao and Georgeff, 1991; Singh, 1994)

### **5.2.2. Arquitecturas para construir agentes**

A continuación, en esta sección se detallarán las principales características de las arquitecturas *deliberativas*, así como las *reactivas* y las *híbridas*, siendo estas últimas combinación de las dos anteriores.

#### **5.2.2.1. Deliberativas**

Son las arquitecturas que emplean modelos de representación simbólica del conocimiento, basadas normalmente en la teoría de planificación. Tal y como indica Maes (1990), estos agentes parten de un estado inicial y van generando planes para alcanzar sus objetivos.

Un agente deliberativo tendrá un modelo simbólico del mundo, con representación explícita, donde las decisiones se consiguen por medio de razonamiento lógico, basados en correspondencia de patrones y manipulación simbólica, con el propósito de alcanzar los objetivos. Por tanto, para la

implementación de estos agentes, se precisa una descripción simbólica adecuada para el problema e integrarla en el agente.

Los agentes intencionales pueden implementarse empleando este tipo de arquitectura, y como muestra Jennings (1993), estos agentes están dotados de modelos de planificación capaces de generar planes a partir de creencias e intenciones.

### Arquitectura BDI

Los agentes están dotados de estados mentales de *Creencias*, *Deseos* e *Intenciones* (Haddady and Sundermeyer, 1996). Los agentes en general, y en particular los agentes *BDI*, incorporan componentes que permiten el desarrollo de sistemas que se integren adecuadamente en el mundo real (Georgeff and Rao, 1998). Algunas aplicaciones requieren sistemas complejos que se relacionen con el entorno cambiante y con cierto grado de incertidumbre, el modelo *BDI* se ha desarrollado para proporcionar soluciones en entornos dinámicos e inciertos, donde el agente sólo tendrá una visión parcial del problema y manejará un limitado número de recursos.

Las *creencias* representan el conocimiento que se tiene del entorno, por ejemplo por medio de variables, valores de bases de datos o expresiones simbólicas del cálculo de predicados. Será necesario mantener un histórico de los eventos al mismo tiempo que se permita su adaptación y evolución, por lo que es fundamental la gestión de dichos eventos.

Los *deseos* podrían ser el valor de una variable, un registro o una expresión simbólica en determinada lógica, donde el *deseo* representaría un estado final que se desea alcanzar. El desarrollo de *Software* clásico está basado en tareas, no en objetivos, de forma que una tarea se ejecuta sin ningún recuerdo del porqué de su comienzo, y no podrá recuperarse ante fallos ni aprovechar oportunidades que vayan surgiendo.

Las *intenciones* deberán ser identificadas por medio de la planificación, para que de esta manera se puedan conseguir los *deseos* a partir de las creencias existentes. El sistema deberá llegar a un compromiso entre los planes y sub-objetivos, pero reconsiderando éstos en determinados momentos, de esta manera, estos planes vinculados a la consecución de los objetivos constituyen las intenciones del agente. Por tanto, las *intenciones* son un conjunto de caminos de ejecución que podrán ser interrumpidos de forma apropiada al recibir cierta información cambiante del entorno (Kinny and Georgeff, 1991).

Las actitudes se pueden representar por un formalismo basado en la lógica modal y el concepto de *mundos posibles*. La lógica modal es la que razona sobre lo que podría ser, o se cree que es, en lugar de lo que realmente es, de esta manera es posible representar conocimiento de tipo “*Es probable que Juan tenga el brazo roto*”, dicho de otra forma, desde la situación actual, se puede llegar a que Juan tenga un brazo roto. Para que esta lógica incluya significados como los anteriores, se maneja una semántica de *mundos posibles* con relación de accesibilidad entre ellos. Esto hará

necesario nuevos operadores unitarios, como el operador de necesidad y el operador de posibilidad (Chellas, 1980; Hughes and Cresswell, 1996).

Los agentes *BDI* pueden modelarse en base a la lógica de *mundos posibles* denominada árbol temporal con múltiples futuros y un único pasado (Rao and Georgeff, 1991), tal y como se muestra en la Figura 5.1, donde cada nodo del árbol es una situación posible y las ramas serían las opciones disponibles para el agente en cada momento del tiempo. Para cada situación se definen una serie de mundos accesibles desde el punto de vista de *creencias*, *deseos* e *interacciones* (Rao and Georgeff, 1995), pudiendo ser incompatibles los *deseos*, exigiendo el modelo que al menos sean consistentes y, además, que el agente crea que son alcanzables.

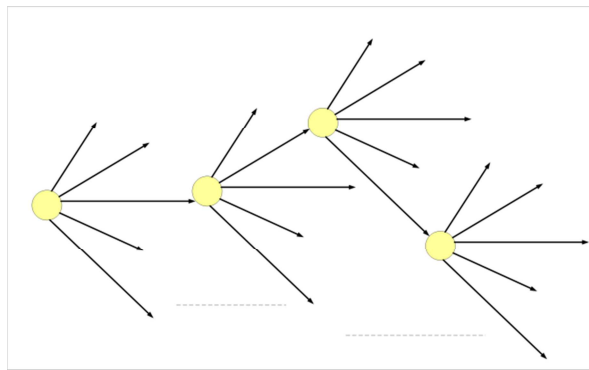


Figura 5.1. Árbol de “mundos posibles”. Fuente Propia.

Un concepto importante es la noción de compromiso, un compromiso tiene dos partes: una es la condición que el agente está comprometido a mantener, llamada condición comprometida; y la otra es la condición bajo la cual el agente renunciará al mismo, llamada de terminación (Rao and Georgeff, 1995). Un agente puede elegir qué hacer con sus *intenciones*, comprometiéndose con una en base a que el objetivo es satisfecho en un camino futuro o incluso en todos, provocando por tanto diferentes condiciones de comportamiento y, por tanto, diferentes comportamientos dinámicos.

Dependiendo de la manera en que afectan las *intenciones* pasadas a las futuras, se identifican diferentes tipos de agentes:

- Ciego: el agente mantiene sus *intenciones* hasta saber si las alcanzará o no.
- Firme: el agente mantiene sus *intenciones* mientras crea que las opciones son alcanzables.
- Imparcial: el agente mantiene sus *intenciones* mientras éstas se correspondan con sus *deseos*, dicho de otra forma, mientras que los *deseos* que dieron lugar a la *intención* no cambien.

Lo planteado hasta aquí se ve ampliado con librerías de planes, que definen métodos estándar para la consecución de ciertos objetivos (Cavendon and Rao, 1996). Los planes están compuestos por los objetivos que permite conseguir la *intención* y la secuencia de acciones para conseguir un objetivo o sub-objetivos, pudiendo asociarse a nuevos planes.

Se evidencia la necesidad de disponer una interrelación entre las *intenciones* de un agente y los planes que dispone, siendo las más importantes las siguientes:

- Las *intenciones* de un agente están restringidas por sus planes.
- Si un agente pretende alcanzar un objetivo, deberá ejecutar el cuerpo del plan que permitirá su consecución.
- Si un agente pretende alcanzar cierto objetivo, dentro de las *creencias* del agente estará el hecho de que el plan logrará dicho objetivo.

### Abstracción de la arquitectura

La Figura 5.2 muestra la arquitectura propuesta por Rao and Georgeff (1995), con cada uno de los componentes *BDI*, donde cada uno de los componentes se guarda de manera aislada, por tanto, existirán tres listas diferentes. De la misma forma, se guarda una secuencia de eventos ocurridos en el entorno, implementándose como una cola.

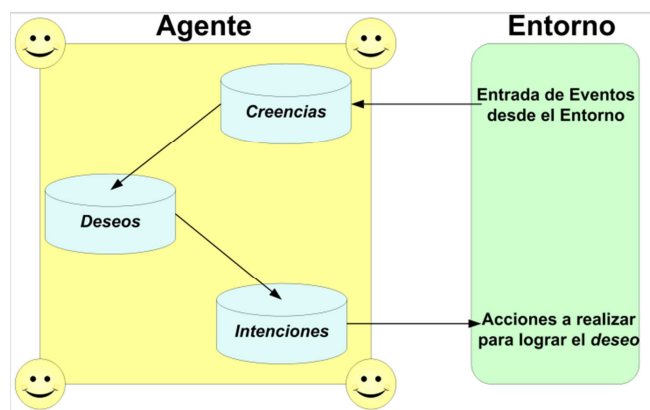


Figura 5.2. Arquitectura *BDI*. Fuente Propia.

La arquitectura propuesta realiza iterativamente una serie de pasos, cada uno con cierta duración, por lo que se posibilita la reacción ante el entorno en tiempo suficiente. Un esquema del ciclo de ejecución, que por su sencillez no necesita explicación, sería:

*Interprete-BDI*

*Inicializar\_estado();*

*Repetir*

*Opciones:=generador\_opciones (cola\_eventos);*

*Opciones\_seleccionadas:=decidir (opciones);*

*Actualizar\_intenciones (opciones\_seleccionadas);*

*Ejecutar();*

*Leer\_nuevos\_eventos\_externos();*

*Eliminar\_intenciones\_alcanzadas();*

*Eliminar\_intenciones\_imposibles();*

*Fin repetir*

Con la arquitectura presentada, y empleando el formalismo teórico descrito, se han implementado algunas aplicaciones interesantes. Una de ellas es el gestor de tráfico aéreo *OASIS* de

Georgeff and Rao (1998), formado con un gran número de agentes que seleccionan rutas de vuelo entre óptimas posibles, minimizando el consume de carburante, mejor la productividad y garantizar todos los tiempos. Rao and Georgeff (1992) presentan un sistema de modelado de combate aéreo, llamado *SWARMM*, que permite codificar las tácticas adoptadas por los pilotos en escenarios de combate y crear agentes que puedan participar con los pilotos reales en estudios de simulación. Otro ejemplo es el presentado por Carbó *et al.* (2003) donde se incluye lógica borrosa en el razonamiento interno del agente, para gestionar el problema de compra y venta de bienes y servicios, el sistema es conocido como *AFRAS*.

#### 5.2.2.2. *Reactivas*

La ventaja de las arquitecturas reactivas es que no disponen como elemento central de razonamiento un modelo simbólico, y por no emplear razonamiento simbólico complejo (Keith *et al.*, 1997). Un ejemplo de estas arquitecturas es la propuesta por Brooks (1991) llamada de *subsunción*, que se basa en la hipótesis de que la inteligencia es una propiedad emergente de ciertos sistemas complejos y que a partir de esto, se pueden generar comportamientos inteligentes sin modelo simbólico.

La mayor aplicación de la arquitectura está en robótica, donde los robots pueden considerarse como agentes *Hardware* reales que actúan en un entorno cambiante. La necesidad de una actuación en entorno impredecible dificultará la adopción de la arquitectura deliberativa, ya que la re-planificación y adaptación del plan a la realidad hace difícil que la arquitectura sea suficientemente ágil.

La arquitectura reactiva tan sólo necesita de un conjunto de acciones muy básicas (Brooks, 1986), pensando de nuevo en el robot, tan sólo se necesitarán acciones básicas de bajo nivel y simplemente será preciso decidir qué acción ejecutar en el momento actual. El problema surge cuando hay problemas complejos, donde el espacio de búsqueda es tan extenso que hace imposible una solución.

A veces se define una relación entre las acciones básicas, de modo que puedan existir bloqueos entre ellas. Otras veces se ponderas las salidas de las acciones por medio de la fusión (Goodridge and Luo, 1994).

La arquitectura idónea para Brooks (1996), es donde el entorno es el propio entorno, y el sistema de control debe razonar sobre los datos adquiridos por el robot en el propio instante. La información recibida en ese instante debe servir para definir la acción siguiente, convirtiéndose lo anterior en el concepto de reactividad, tal y como vemos en la Figura 5.3.

El concepto de acción básica ha servido para el desarrollo de sistemas de aprendizaje empleados para adecuar estas acciones tanto en arquitecturas reactivas como deliberativas (Molina *et al.*, 2000; Berlanga *et al.*, 2002).

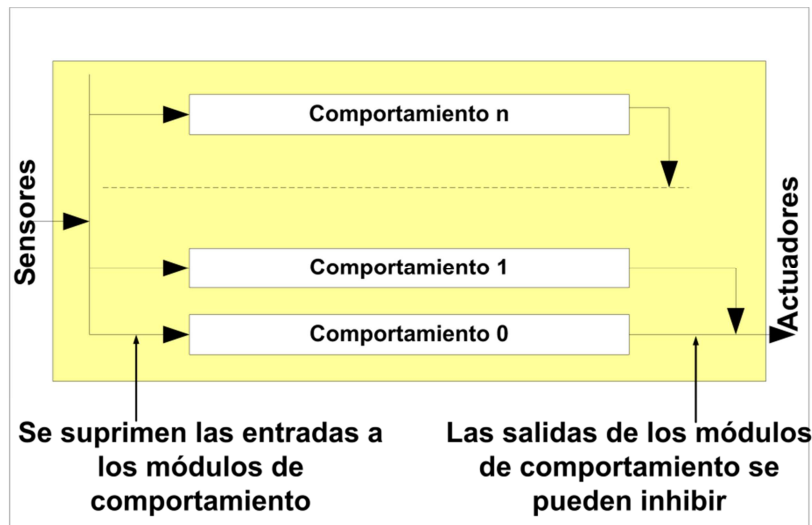


Figura 5.3. Arquitectura subsunción. Fuente Propia.

### 5.2.2.3. Híbridas

Se ha planteado la arquitectura híbrida, para tratar de paliar ciertas limitaciones que presentan el modelo deliberativo y el reactivo. Un primer planteamiento consiste en construir un agente con un subsistema deliberativo con un modelo simbólico para generar planes, y otro reactivo que no precise mecanismos de razonamiento. Estas arquitecturas son propicias para una implementación por capas, obteniéndose dos enfoques:

- Vertical: sólo una capa accede a los sensores y actuadores.
- Horizontal: todas las capas accederán a los sensores y actuadores.

Las capas se organizarán por jerarquías con información del entorno a diferentes niveles de abstracción, encontrándose normalmente tres niveles:

- Reactivo: se toman decisiones sobre qué hacer en base a los estímulos recibidos en tiempo real. Arquitectura de *subsunción*.
- Conocimiento: basado en el conocimiento que el agente tiene sobre el medio, ayudándose de la representación simbólica.
- Social: maneja los aspectos sociales del entorno, como la relación con otros agentes.

Ferguson (1992) presentó un ejemplo de esta arquitectura, conocida como *TOURINGMACHINES*, cada nivel solicita continuamente qué acción realizar y existe un entorno de control que garantiza el correcto funcionamiento del agente, tal y como se aprecia en la Figura 5.4.

Es una arquitectura con tres capas horizontales:

- Reactiva: respuesta inmediata a cambios del entorno.
- Planificadora: pro-actividad en base a esquemas de planes.



- Modeladora: trata de modelar el mundo anticipándose a los conflictos y creando nuevos objetos para la capa planificadora.

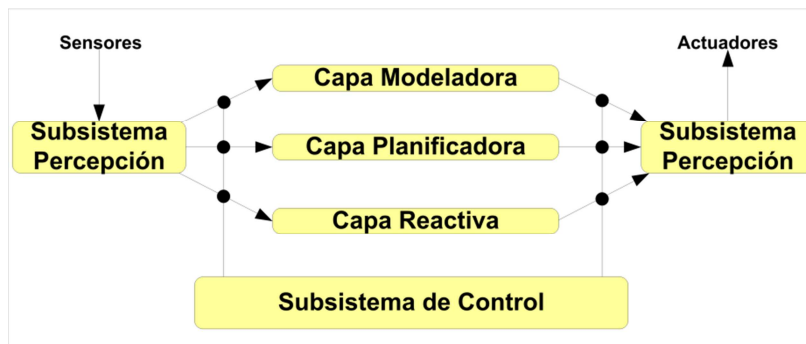


Figura 5.4. Arquitectura *TOURINGMACHINE*. Fuente Propia.

Müller (1997) plantea la arquitectura *INTERRAP*, similar a la anterior pero las capas están dispuestas verticalmente y añade una base de conocimiento organizado en capas, tal y como se ve en la Figura 5.4. Las capas de conocimiento son:

- Capa de comportamiento: reactiva, controlando la información del entorno.
- Capa de planificación: pro-activa, seleccionando planes y acciones del agente.
- Capa de cooperación: interacción social con otros agentes e incluso el entorno.

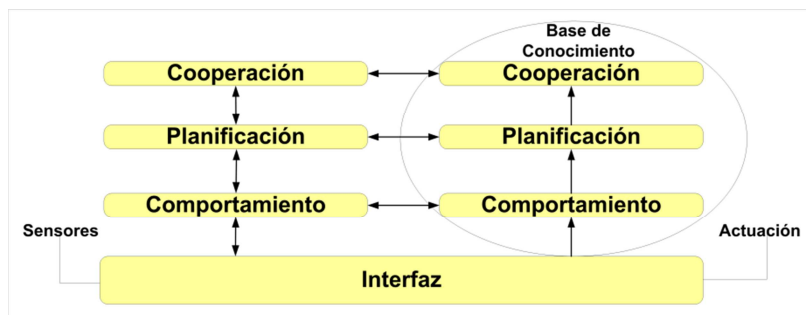


Figura 5.5. Arquitectura *INTERRAP*. Fuente Propia.

La anterior arquitectura es aplicable en problemas donde es necesaria la integración del razonamiento deliberativo para la negociación conducente a una modificación de la ejecución, pero donde dicha ejecución es independiente del razonamiento. Son problemas típicos de robótica, donde el bajo nivel es reactivo (ejecución de acciones continuas) y a la vez, se integran los robots en un *MAS* para la consecución de un objetivo común. Un problema similar surge en los sistemas de vigilancia, y en concreto en las aplicaciones de radares multifunción de vigilancia aérea (Molina *et al.*, 2003).

### 5.2.3. Arquitecturas Multi-Agente

La complejidad de las aplicaciones, compuestas por multitud de subsistemas que interaccionan entre sí, obliga a disponer inteligencia entre diversos agentes y a construir *MAS*, que permitirá la gestión inteligente de dichos sistemas complejos. Como indica Ferber (1999), la

complejidad de la mayoría de los problemas es de tal medida, que se precisa una solución distribuida, capaz de adaptarse a cambios estructurales y del entorno. De la misma manera, se precisará una metodología capaz de construir un sistema a partir de las unidades autónomas.

Si los diferentes subsistemas son dotados con decisión local, la gestión se puede abordar definiendo políticas de cooperación, coordinación y negociación de agentes. Como norma general, un MAS presentará las siguientes características (Wesson *et al.*, 1981):

- Cada agente del sistema deberá mantener sus propias habilidades.
- El MAS tiene una versión única.
- Cada agente del sistema tiene un conocimiento limitado.
- Cada agente del sistema tiene una especialización alrededor de ciertas tareas.

Para la coordinación entre agentes, es tan importante el razonamiento interno como el proceso de comunicación. El razonamiento interno consistirá en decidir e identificar la información a comunicar, mientras que el proceso de comunicación, el cómo y el qué, entre los diferentes agentes.

Se precisa identificar los elementos fundamentales de los agentes, distinguiéndose tres métodos de análisis:

- Descripción de las funcionalidades del sistema por medio de un *análisis funcional*.
- Distinguir las diferentes formas de organización e identificar los parámetros estructurales por medio de un *análisis estructural*.
- *Análisis de parámetros*, para solventar cuestiones que surgen al pasar de una estructura abstracta a una organización concreta.

Las relaciones entre los agentes de una organización, los modos de enlace entre agentes y las posibles subordinaciones, se podrán obtener una vez descrito el MAS. Las relaciones abstractas describirán las formas de interacción entre las clases de agentes (roles), siendo las relaciones las siguientes:

- La relación mínima entre dos agentes supone el conocimiento de su direccionamiento, conocida como *relación de conocidos*.
- La relación que indica que un agente puede enviar comunicaciones a otros se conoce como *relación de comunicación*.
- La transferencia de una tarea entre agentes la describe la *relación de subordinación*.
- La relación de dependencias entre tareas se conoce como *relación operativa*.
- A veces el conocimiento de un agente depende del conocimiento de otro, la relación que establece esta validez se conoce como *relación de información*.
- A veces los recursos son requeridos por varios agentes, necesitando coordinar las tareas mediante negociación, esto se conoce como *relación de conflicto*.

- Cuando los objetivos de los agentes son incompatibles, se plantea una competición entre ellos, llamándose *relación competitiva*.

Las relaciones abstractas pueden ser *estáticas* o *dinámicas*. Las primeras se determinan al definir la organización y no cambian durante la ejecución, mientras que las segundas cambian en tiempo de ejecución.

La unión entre agentes y la capacidad del agente en cuanto a su modificación, vendrá determinada por el *modo de enlace*. Los diferentes tipos serán: *fijos*, impiden la reorganización y adaptación al entorno de la propia organización y del agente; *variables*, permiten la adaptación de los agentes en ciertos parámetros; y *evolutivos*, no tienen límites en cuanto a evolución de agentes y relaciones.

La subordinación puede definirse mediante *estructura jerárquica* o *estructura igualitaria*. En la primera, ciertos agentes pueden ordenar de manera controlada al resto e incluso coordinar todas las actividades. En la segunda, todos los agentes participan de la misma forma en la toma de decisión.

Con respecto a las habilidades de los agentes, que dependerán de los problemas que deban resolver, se pueden encontrar: la totalidad de los agentes pueden tener la totalidad de las habilidades posibles; y que cada agente tan sólo tenga una habilidad, convirtiéndose en especialista. A partir de lo anterior, pueden encontrarse cuatro situaciones:

- Cada agente sólo tiene una habilidad y no es de ningún otro, se está ante una *organización no redundante e hiperespecializada*.
- Cada agente sólo tiene una habilidad y todos los agentes la poseen, se está ante una *organización redundante y especializada*.
- Cada agente tiene varias habilidades y varios agentes las poseen, estando ante una *organización redundante y general*.
- Cada agente tiene todas las habilidades y nadie más que ese agente las posee, estando ante una *organización no redundante y general*.

#### 5.2.4. La arquitectura FIPA

Para posibilitar el despegue de toda tecnología se necesita conseguir la interoperabilidad de estos nuevos sistemas basados en dicha tecnología con lo existente, y la apertura de la misma posibilitando su extensión. Para lo anterior se precisan estándares, y para los agentes la organización que más ha trabajado es *FIPA*, definiendo entre otras cosas, las características que deben de cumplir las plataformas de gestión *MAS*. La mayoría de los entornos de desarrollo y ejecución de agentes, están concebidas a partir de los estándares *FIPA*, como es el caso de *Java Agent DEvelopment framework (JADE)*, que es una plataforma *Software* que proporciona

funcionalidades básicas de capa *middleware*, que son independientes de la aplicación específica y la cual simplifica la realización de aplicaciones distribuidas (Wooldridge and Jennings, 1995).

*FIPA* define el comportamiento externo de la plataforma de agentes, dejando el diseño a los equipos de desarrollo. Otro de los principios que guía todo el estándar es obtener un sistema abierto, con lo que sistemas heterogéneos puedan interactuar a nivel de sociedades de agentes. El modelo *FIPA* establece el marco lógico en cuanto a creación, destrucción, registro, localización y comunicación de agentes.

El núcleo del modelo de referencia de *FIPA* lo proporciona la Plataforma de Agentes (*Agent Platform*), la cual provee de la infraestructura para el desarrollo y empleo de los agentes, teniendo los recursos *Hardware* y *Software* necesarios para ello. De esta manera, en cada implementación concreta se deberá decidir sobre componentes a emplear para el desarrollo, pero teniendo un conjunto de interfaces externas para la interoperabilidad entre diferentes plataformas, como pueda ser la comunicación y gestión de agentes.

Los servicios que debe proporcionar la plataforma de agentes, según marca el estándar *FIPA* son: Plataforma Interna de Transporte de Mensajes (*Internal Platform Message Transport*), Sistema de Gestión de Agentes (*Agent Management System*), Facilitador de Directorio (*Directory Facilitator*) y Canal de Comunicación de Agentes (*Agent Communication Chanel*). A excepción del servicio de transporte, el resto son suministrados por agentes especializados, por lo que la comunicación con ellos se realizará con mensajes *ACL*, según la ontología definida para ese servicio. En la Figura 5.6 se representa el modelo *FIPA* según lo indicado anteriormente.

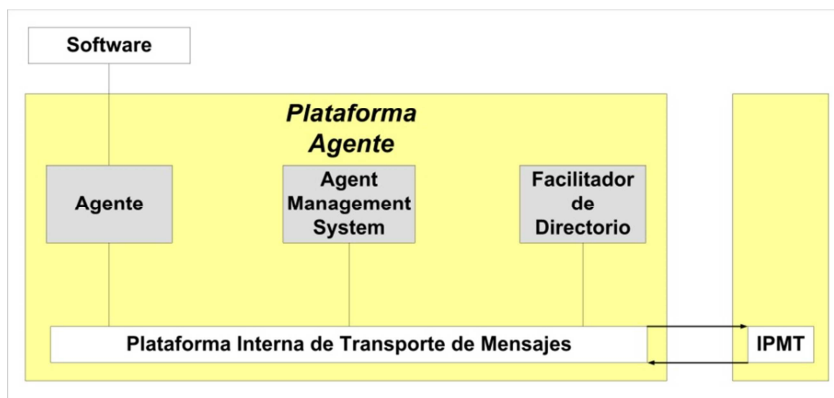


Figura 5.6. Modelo *FIPA*. Fuente Propia

Según lo representado en la figura anterior, los principales componentes de una plataforma *FIPA* son:

- *Agent Management System*: elemento gestor principal, que conoce el estado de la plataforma y de todos sus agentes. Creará, destruirá y controlará el cambio de estado de todos los agentes, control de registro de nuevos agentes, movilidad de agentes, recursos compartidos y gestión de canal de comunicación. Dispone de un Servicio de Nombres de Agente (*Agent*

*Name Service*), que asocia el nombre del agente en la sociedad con su dirección de transporte real. El gestor controlará el ciclo de vida de los agentes, siendo los estados posibles: *Iniciado*, *Activo*, *Suspendido* y *Esperando*.

- *Directory Facilitator*: los agentes se registrarán a través de este componente, indicando los servicios que ofrecen, de esta manera, el resto de agentes podrán conocer los servicios ofrecidos por el resto de agentes.
- *Internal Platform Message Transport*: es el encargado de gestionar mensajes entre agentes de una plataforma y agentes de distintas plataformas.

#### 5.2.4.1. Seguridad en MAS

Debido a la autonomía, colaboración y cooperación, los problemas de seguridad en MAS son inherentes a su propia concepción. FIPA ha estudiado soluciones de seguridad para MAS, de manera que su despliegue en sistemas abiertos sea seguro.

FIPA trata de emplear los estándares existentes en seguridad, por ejemplo, se plantea emplear X.509 basado en infraestructuras de seguridad de clave pública (PKI), que está basado en un modelo asimétrico de certificados de clave privada y pública.

El anterior modelo se presenta a nivel de agente, siendo posibles modelos a nivel más bajo, como por ejemplo en la capa de transporte, empleando Enchufes (*sockets*) seguros como Capa de Enchufes Seguros (*Secure Socket Layer – SSL*).

Otros problemas asociados a la comunicación y el despliegue, y que son contemplados por FIPA son: copia de códigos y datos; alteración de código y datos; ataques en grupo; copia y reenvío; denegación de servicio; suplantación de identidad entre agentes; y rechazo de responsabilidades al reasignarlas cuando ocurre problemas con agentes.

#### 5.2.4.2. Movilidad de agentes

Los agentes móviles seguían una línea de investigación paralela a la de los agentes inteligentes, ya que resolvían problemas distintos. Ahora parece posible la integración de ambas partes de la investigación para disponer de un agente con todas las capacidades deseables.

La motivación de la existencia de este tipo de agentes es muy sencilla, la idea es que este agente móvil reemplace a Llamada a Procedimiento Remoto (*Remote Procedure Call – RPC*), para procesos de comunicación de la red, como se aprecia en la Figura 5.7. Con *RPC*, la idea es que un proceso puede invocar a un método o procedimiento en otro proceso, el cual está remotamente localizado. Para entenderlo, un proceso *P* invoca un método *m* en un proceso *R* con ciertos argumentos, el valor devuelto por este último proceso se asignará a determinada variable *i*. esta comunicación entre procesos es asíncrona, o dicho de otra forma, el proceso *P* se bloquea desde que se ejecuta la instrucción hasta que el proceso *R* devuelve un valor. Algún error en la red, con la

consecuente no devolución del valor por parte de  $R$ , hará que  $P$  permanezca interrumpido indefinidamente. Además, la conexión de red entre procesos permanecerá abierta, independientemente o no del envío de información.

La idea del agente móvil es enviar un agente para realizar la computación, de esta forma, en lugar de invocar un método, el proceso  $P$  envía un agente al proceso  $R$ . Debido a que el agente y  $R$  comparten el mismo espacio de direccionamiento, la interacción se consigue de una manera más eficiente. Cuando el agente finaliza sus tareas, regresa a  $P$  con los resultados, y de esta forma, la ocupación de la red tan sólo se ha realizado al enviar el agente en ambos sentidos.

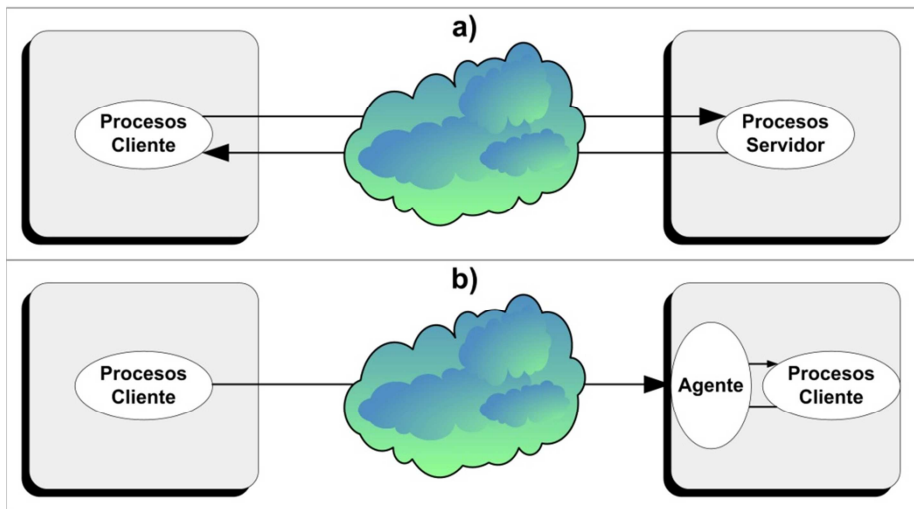


Figura 5.7. a) Modelo RPC; b) Modelo agente móvil. Fuente Propia.

En el caso de agentes móviles, *FIPA* propone dos soluciones a llevar a cabo:

- *Migración*: requiere la transferencia de código y datos del agente hasta la plataforma remota, aumentando la dificultad. Durante el envío, el agente queda *suspendido*, para ser *despertado* al llegar al otro extremo, momento en que podrá ejecutar sus tareas.
- *Clonación*: se crea copia del agente en el nodo remoto que se encargará de hacer el trabajo solicitado por el clon de la máquina origen. Sólo habrá transferencias de órdenes que el clon deberá realizar, no habiendo código ni datos transferidos.

Dependiendo quién hace la gestión de la transferencia, *FIPA* define dos modelos de movilidad:

- *Modelo simple*: el agente solicita a su *Agent Management System* la transferencia, y éste se encarga de todo.
- *Modelo complejo*: toda la responsabilidad recae sobre el agente.

En la actualidad, y bajo el beneplácito de *FIPA*, existen dos plataformas de agentes móviles, siendo *JADE* y *Grasshopper* (Bäumer and Magedanz, 1999).

### 5.2.4.3. Servicio de Ontología

La plataforma *FIPA* se basa en la comunicación entre agentes, la cual se realiza mediante mensajes *ACL*. El mensaje será una expresión en un determinado lenguaje con ciertos términos de la ontología empleada. Una ontología puede entenderse como un conjunto de términos (pueden ser símbolos) junto a su significado.

Ya que los agentes se registran en *Directory Facilitator*, cuando un agente quiere comunicar con otro, consulta en el directorio la ontología de las que tiene conocimiento, para emplear una común. Si el dominio es pequeño, las ontologías pueden integrarse en el agente, limitándose las mismas al contexto de utilización.

Los sistemas de gran envergadura, recurren al servicio Agente Ontología (*Ontology Agent*), que proporcionará la información precisa a los agentes para entender los términos del mensaje. Las labores básicas del servicio son:

- Los agentes disponen del conjunto de ontologías públicamente.
- Hace de traductor entre ontologías diferentes.
- Responde a preguntas en torno a las ontologías que están bajo su gestión.
- Descubre nuevas ontologías y las pone a disposición pública entre los agentes.

### 5.2.4.4. Transporte de mensajes

*FIPA* también incluye dentro de su estándar la labor de diseño de la componente de bajo nivel *Internal Platform Message Transport*, debido a su gran importancia. Este componente maneja la siguiente terminología:

- Servicio de Transporte de Mensajes de Agente (*Agent Message Transport Service*): es el objeto que implementa el servicio de transporte dentro de la plataforma de agentes *FIPA*.
- Fábrica de Conversación (*Conversation Factory*): interfaz necesaria cuando un agente quiere dialogar con otro, necesita comenzar un nuevo contexto (Contexto de Conversación – *Conversation Context*).
- *Conversation Context*: extremo lógico de un canal de comunicación que se establece para el intercambio de mensajes.
- Fábrica de Transporte (*Transport Factory*): interfaz que permite a un agente saber el conjunto de mecanismos de transporte soportados (*http*, *Wap*, etc.).
- Transporte (*Transport*): mecanismo de transporte concreto, que identifica el mecanismo de transporte a emplear al establecer cada conversación.
- Dirección (*Address*): representa un extremo del canal de transporte empleado. Las direcciones dependerán del transporte escogido (*IP*, *URL*, etc.).
- Destino (*DestinationK*): identificación de la entidad remota con la que se comunicará.

- Calidad de Servicio (*Quality of Service*): encapsula las propiedades del canal de transporte, como integridad de datos, privacidad, ancho de banda, etc.; esto permite contactar calidad de servicio al inicio del contrato.
- Mensaje (*Message*): unidad básica de información a transmitir.

#### 5.2.4.5. Nombres

*FIPA* ha definido un sistema de nombres de los agentes estáticos, donde un agente es identificado por un único *GUID*, de la forma siguiente:

$$\langle name \rangle @ \langle hostname \rangle : \langle port \rangle / \langle target \rangle \quad (5.1)$$

donde *name* es el identificador del agente en su plataforma; *hostmane* es dirección *IP* o *DNS* del host en el que se ejecuta; *port* es el puerto de escucha; y *target* es el identificador del agente que debería recibir el mensaje para encaminarlo al agente destino.

Más tarde se dotó a la estructura más flexibilidad, con la intención de acometer necesidades futuras, definiéndose de la siguiente manera

$$\begin{aligned} \text{AgetName} &= && (":guid" \text{ Word NameFields}*) \\ \text{NameFields} &= && ":addresses" (" CommAddress+ ") \\ &&& /":resolver" (" AgetName+ ") \\ &&& /":authenticator" (" Agentname+ ") \end{aligned} \quad (5.2)$$

### 5.3. COMUNICACIÓN ENTRE AGENTES

La coordinación de las actividades de los agentes en un entorno distribuido y, en determinados casos, su cooperación entre algunos de ellos, se hace fundamental. Será preciso por tanto, mecanismos y modos de comunicación entre los agentes. Los primeros *MAS* que se desarrollaron hacían uso de sus propios sistemas de comunicaciones, aunque las necesidades de interacción y comunicación aparecían en todos los agentes, por ser la esencia del *MAS*. El problema sucedía si se pretendían comunicar agentes desarrollados por diferentes empresas, ya que aparecía la falta de entendimiento.

*FIPA* trató nuevamente de estandarizar la interacción real y homogénea entre sistemas distintos. En esta sección se darán ciertas nociones de los estándares *FIPA* con respecto a la comunicación entre agentes.

#### 5.3.1. El acto de comunicarse

##### 5.3.1.1. Filosofía

A partir de mediados del siglo XX surge una línea de investigación filosófica analítica, que pretende responder preguntas del tipo “¿cuál es el motivo de la comunicación entre seres



humanos?” y “¿cómo lo consiguen e influyen en otros seres y el mundo exterior?”. Esta línea seguida por Austin (1962; 1970) y Searle (1979), concretó en el estudio y la clasificación de los llamados *actos del habla*.

Austin apuntó que los filósofos y lógicos habían caído en la *falacia descriptiva*, donde para ellos, el lenguaje servía únicamente para realizadas descripciones del mundo, siendo éstas verdaderas o falsas. Sin embargo, numerosas declaraciones del día a día no están encaminadas a constatar ni describir una opinión o hecho, al emitirse, no se *describe* un acto, sino lo que se *realiza*. Austin las bautizó como *performatives*, siendo un ejemplo, quien al pisar a otro dice “*le pido disculpas*”, no enuncia la veracidad o falsedad de la declaración, sino que realiza la acción de disculparse. Por esto, Austin distingue entre *locución* (contenido proposicional), *ilocución* (orden, promesa, etc.) y *perlocución* (efecto en el receptor).

Continuando con el trabajo de Austin, Searle representó un acto *ilocucionario* como  $F(p)$ , donde  $F$  es la fuerza *ilocucionaria*,  $p$  el contenido proposicional, y se distinguen cinco tipos de fuerzas:

- *Asertiva*: el emisor informa de la veracidad o falsedad de la proposición.
- *Compromisoria*: el hablante toma compromiso de futuro con la acción que depende de él.
- *Declarativa*: el hablante enuncia una acción directamente con el propio enunciado (disculpa, perdón, etc.).
- *Directiva*: el hablante insta de alguna manera, para que se realice la acción futura (petición, orden, etc.).
- *Expresiva*: el hablante expresa estados de ánimo (felicitación, alegría, etc.).

### 5.3.1.2. Actos comunicativos

Las reflexiones abstractas anteriores han servido para fundamentar los actos comunicativos básicos que aparecen en los estándares, como por ejemplo en *FIPA*, contenido en *FIPA CALL* (*FIPA*, SC00037J). Es una librería que contiene 22 acciones de comunicación, proporcionando la siguiente información para cada una de ellas:

- Lenguaje natural de la acción y consecuencias.
- Semántica de la acción, precondiciones formales y sus efectos.
- Ejemplos del acto.

Las acciones se pueden agrupar en cuatro clases, en base a su actividad social a la que se encuentran destinadas:

- *Información*: para estas acciones intervienen dos agentes, uno será el emisor y el otro el receptor. El emisor puede representar dos papeles, si ofrece información se conoce como *ROL1* y si la solicita se conoce como *ROL2*.

- *Acciones de realización:* si el emisor está interesado en realizar la acción desempeña *ROL1*, en cambio, si realiza la acción desempeñará *ROL2*.
- *Acciones de negociación:* el emisor desempeña *ROL1* cuando es el agente interesado en la realización de la acción, no conociendo las condiciones bajo las cuales el cooperante realizará la acción solicitada. Desempeñará *ROL2* cuando es el agente cooperante para que la acción e cumpla.
- *Acciones de intermediación:* el emisor desempeña *ROL1* cuando solicita la intermediación, desempeñando *ROL2* cuando hace de puente entre el primer agente y el resto.

### 5.3.1.3. Más estándares

Una organización que trabaja en estándares, paralelamente a *FIPA*, es el grupo Intercambio de Conocimientos (*Knowledge Sharing*), perteneciente a Agencia de Proyectos de Investigación Avanzados de Defensa (*Defense Advanced Research Projects Agency – DARPA*). El logro más importante, fue la creación de un lenguaje del que *FIPA ACL* tomó su filosofía, llamado *KMQL* (Finin *et al.*, 1993; Labrou and Finin, 1997). *KMQL* fue creado como lenguaje de formato de mensajes y protocolo de tratamiento de mensajes, para compartir conocimiento entre agentes.

*KQML* emplea 34 *performatives*, clasificadas en tres categorías:

- *De discurso:* empleadas en el contexto de interacciones de intercambio de información y conocimiento entre dos agentes.
- *De intervención y mecánica de la conversación:* es para intervenir en la conversación entre dos agentes o para finalizarla de manera prematura o bien para ignorar el protocolo de comunicación empleado por defecto.
- *De comunicación en red y facilitación:* los agentes podrán buscar otros agentes que puedan ayudarles a responder ciertas preguntas. La búsqueda se hace por medio de los *facilitadores*, agentes especiales únicos en transmitir información entre toda la comunidad de agentes.

### 5.3.2. Estructura de los mensajes (estándares)

#### 5.3.2.1. Estándar FIPA: ACL

*ACL* define la estructura de los diferentes mensajes para el envío entre agentes, donde cada mensaje tendrá uno o más elementos, como se ve en la Tabla 5.1, no siendo todos obligatorios. Se presenta el elemento del mensaje, así como la categoría al que pertenece cada uno de los elementos detallados.

El contenido del mensaje será uno u otro, dependiendo del tipo de acto comunicativo. Estará incluido en el elemento *content* en los mensajes.

**Tabla 5.1. Componentes del mensaje FIPA ACL. Fuente Propia.**

<i>Elemento del mensaje</i>	<i>Categoría de elementos</i>
<i>Performative</i>	Acto comunicativo
<i>Sender</i>	Participante de la comunicación
<i>Receiver</i>	Participante de la comunicación
<i>Reply-to</i>	Participante de la comunicación
<i>Content</i>	Contenido del mensaje
<i>Language</i>	Descripción del contenido
<i>Encoding</i>	Descripción del contenido
<i>Ontology</i>	Descripción del contenido
<i>Protocol</i>	Control de la conversación
<i>Conversation-id</i>	Control de la conversación
<i>Reply-with</i>	Control de la conversación
<i>in-reply-to</i>	Control de la conversación
<i>Reply-by</i>	Control de la conversación

### 5.3.2.2. Estándar KQML

*KQML* tiene una sintaxis parecida a *ACL*, los elementos reservados se muestran en la tabla 5.2. Tanto *ACL* como *KQML* pueden hacer uso de cualquier lenguaje en su elemento *content*, aunque *KQML* está ligado a un lenguaje de representación creado por el mismo grupo, y conocido como *KIF* (Patil *et al.*, 1992). *KIF* implementa la lógica de predicados de primer orden, e incluye mecanismos para traducir desde y hacia los principales lenguajes de representación de contenido.

**Tabla 5.2. Componentes reservados para un mensaje KQML. Fuente Propia.**

<i>Elemento del mensaje</i>	<i>Categoría de elementos</i>
<i>:sender</i>	Nombre del emisor del mensaje
<i>:receiver</i>	Receptor del mensaje
<i>:in-reply-to</i>	Etiqueta para la respuesta de un mensaje anterior
<i>:reply-with</i>	Etiqueta para una respuesta que el emisor espera
<i>:language</i>	Nombre del mensaje de representación del parámetro <i>:content</i>
<i>:ontology</i>	Ontología empleada en <i>:content</i>
<i>:content</i>	Información de la actitud

### 5.3.2.3. Protocolos de comunicación

La conversación entre agentes se suele corresponder con parámetros de secuencia de mensajes, donde estas secuencias se conocen como protocolos de comunicación. En *FIPA* existen ya unos protocolos definidos (*FIPA PC00025C*), siendo una lista abierta, pudiendo añadir nuevos protocolos los diseñadores.

Para la definición de los protocolos, *FIPA* ha desarrollado un lenguaje de representación de protocolos llamado Lenguaje de Modelado Unificado de Agente (*Agent Unified Modeling Language – AUML*) (Baclawski *et al.*, 2002), que está basado en los diagramas de secuencias Lenguaje de Modelado Unificado (*Unified Modeling Language – UML*) (Booch *et al.*, 1998), cuyos resultados se llamarán *diagramas de protocolo*. Dentro de los diagramas *AUML* se pueden distinguir los siguientes elementos:

- *Rol del agente*: indica el papel de cada agente dentro de la comunidad, cuya representación será un rectángulo con el nombre del rol de cada uno de los agentes.
- *Línea de vida*: indica el tiempo en que el agente participará en la comunicación, representándose por una línea discontinua vertical. La línea de vida puede dividirse en varias para mostrar paralelismo, como se aprecia en la Figura 5.8.
- *Hilos de interacción*: representa el periodo de tiempo en que el agente está realizando una tarea como consecuencia de la recepción de un mensaje. Su representación es por medio de un rectángulo sobre la línea de vida del agente.
- *Mensajes*: se representa como una fecha continua horizontal, parte del hilo de interacción de un agente a otro de un segundo agente. Llevará una etiqueta del acto comunicativo, junto a los argumentos en el caso que sean precisos. También pueden llevar asociados cierto paralelismo en los mensajes.

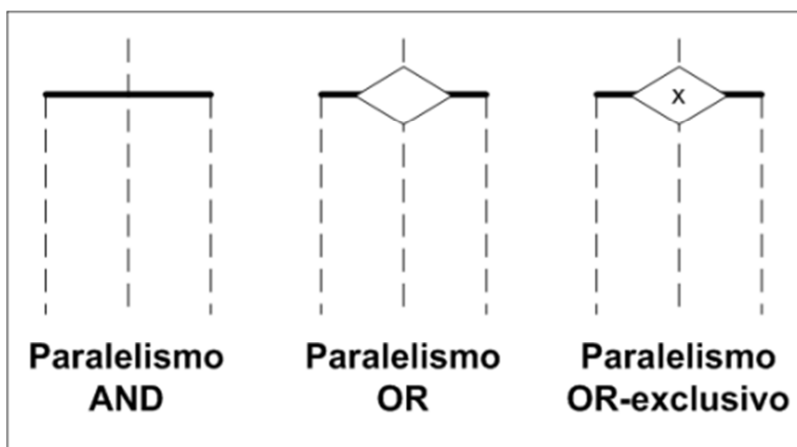


Figura 5.8. Tipos de paralelismo en la línea de vida. Fuente Propia.

El resumen de los protocolos estándar se puede ver en la Tabla 5.3, todos los detalles se encuentran en (Mas, 2005). En la Figura 5.9 se representa el protocolo *query*, donde un agente inicia una conversación para que un receptor le informe de algo en concreto, el iniciador en el primer mensaje puede mandar una *performative* de tipo *query-if* o *query-ref*, a lo que el receptor, tras un periodo de proceso representado por el hilo de interacción (rectángulo), responderá con algún mensaje alternativo: no entiende la pregunta (*not-understood*); no quiere contestar (*refuse*); error al procesar la pregunta (*failure*); o envía el resultado esperado por el iniciador (*inform*).

Tabla 5.3. Protocolos de comunicación estándar *FIPA*. Fuente Propia.

Nombre del protocolo	Descripción del protocolo
<i>Request</i>	Se le pide a un agente que realice determinada acción
<i>Request when</i>	Se le pide a un agente que realice determinada acción si se cumple la precondición
<i>Query</i>	Se le pide a un agente que informe sobre algo
<i>Contract net</i>	Se le pide a un conjunto de agentes que realicen determinada tarea por parte de un agente. Aquéllos dan respuesta en función de ciertos costes y el agente selecciona de entre el grupo quien la realiza finalmente
<i>Brokering</i>	Un agente ofrece las funcionalidades de otros agentes
<i>English auction</i>	Varios agentes participan en una subasta que comienza con el precio más bajo y se va subiendo progresivamente
<i>Dutch auction</i>	Varios agentes participan en una subasta que comienza con el precio más alto y se va bajando progresivamente
<i>Recruiting</i>	Similar a <i>brokering</i> , pero las respuestas sobre el servicio van directamente al agente que lo precisa
<i>Propose</i>	El iniciador propone hacer cierta tarea a un grupo de agentes, y estos aceptan o no
<i>Subscribe</i>	Un agente solicita ser informado si una condición se vuelve verdadera

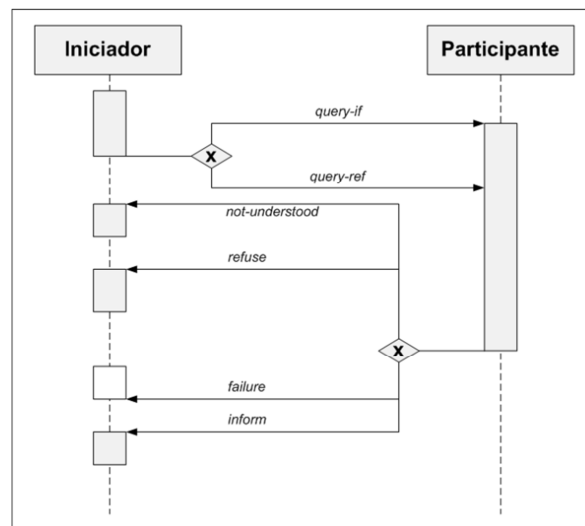


Figura 5.9. Representación del diagrama de protocolo *query*. Fuente Propia.

### 5.3.3. Semántica de los mensajes

Supongamos dos agentes comunicándose por medio de un mismo lenguaje (*FIPA ACL*), e incluso ambos están de acuerdo con las intenciones del otro, podría suceder que no manejaran el mismo contexto del mensaje, con lo que no llegarían a entenderse.

Para paliar el problema arriba indicado, surge el concepto de *compartición de conocimiento*, de manera que el contenido intercambiado en el mensaje sea entendido por todos los agentes. La semántica debe de ser común entre todos los agentes que comparten información, para de esta

manera interpretar lo mismo dentro de un dominio común, siendo este papel clave reflejado en la *ontología*.

No resulta sencillo identificar una definición de *ontología*, por ejemplo Neches *et al.* (1991) la definen como “*la que define los términos y relaciones básicas que componen en vocabulario de un dominio, así como las reglas que permiten combinar dichos términos y relaciones para definir extensiones del vocabulario*” o la de Studer *et al.* (1998) “*es una especificación formal y explícita de una conceptualización compartida*”, refiriéndose con conceptualización a la construcción de un modelo abstracto de un dominio, identificando los conceptos relevantes.

Según Gómez-Pérez and Benjamins (1999), los elementos de una *ontología* son:

- *Concepto*: entidad susceptible de poner nombre, además de la descripción de una tarea, función, acción, etc.
- *Relación*: indica la interacción entre conceptos del dominio, como por ejemplo *parte-de*.
- *Función*: caso especial de las anteriores, donde un elemento será único para los restantes.
- *Axioma*: sentencias verdaderas para ese dominio, como por ejemplo “*el mes febrero va siempre después de enero*”.
- *Instancias*: conceptos concretos, como por ejemplo “*Juan*” o “*Caballo*”.

No es imprescindible que estén todos los anteriores elementos, cuando una *ontología* tiene sólo conceptos y relaciones, se conoce como *taxonomía* u *ontología ligera*. La ventaja de las *ontologías* es que son utilizables en un dominio tanto por las personas como por las computadoras.

Es complicado emplear *KIF* para la implementación directa de *ontologías*, sin embargo por medio de *Ontología Marco (Frame Ontology)* se puede representar el conocimiento que proporcionan primitivas de representación basadas en formalismos de los marcos (clase, instancia, etc.). Gruber (1993) creó *Ontolingua*, que se basa en *KIF* y *Frame Ontology*, que permite crear *ontologías* de tres maneras: por medio de expresiones *KIF*, empleando *ontología* de marcos predefinida, o las anteriores a la vez. Otras posibilidades de *ontologías* las tenemos en *LOOM* (MacGregor, 1991), *FLogic* (Kifer *et al.*, 1995) y *OntoBroker* (Decker *et al.*, 1999).

La influencia de Lenguaje de Marca eXtensible (*eXtensible Markup Language – XML*) sobre los *MAS* es evidente, tanto *FIPA ACL* como *KQML* tienen una especificación en *XML* de su estructura, por lo que es posible emplear analizadores sintácticos para generar e interpretar de manera automática documentos *XML* con mensajes en algunos de los dos lenguajes comentados y con contenido en algún lenguaje de representación (*KIF*, *Prolog*). *XML* ha facilitado la definición de lenguajes de representación de conocimiento y *ontologías* para la construcción de la *Web Semántica*, siendo algunos de los lenguajes más conocidos *SHOE*, *OIL*, *DAML+OIL*, *RDFS* u *OWL*.

### 5.3.4. Tecnologías de interacción

En la actualidad, diferentes tecnologías compiten por ser la referencia en *MAS*, pudiendo destacar *XML*, *Java*, *RMI* y *CORBA*. Cada una tiene ciertas ventajas con respecto a las otras, y a continuación se detallarán brevemente, enfatizando en lo relacionado con agentes y su comunicación.

#### 5.3.4.1. XML

Como ya se sabe, es un lenguaje de marcado, cuyo objetivo es representar documentos que contienen información estructurada, de manera que dicha información se pueda almacenar, transmitir o procesar por algún programa. Por tanto, *XML* es un metalenguaje que posibilita definir diferentes lenguajes de marcado. *XML* define la forma en que se manejan las marcas, pero nunca especifica qué significan en el contexto de la especificación. Las principales características de este lenguaje son:

- Permite el envío de documentos estructurados por la red.
- Es un lenguaje basado en texto, por tanto, legible para computadoras como para personas.
- Su código es extensible, permitiendo añadir nuevas estructuras y etiquetas sin modificar el texto ya construido.

*XML* es una tecnología completa, no sólo un lenguaje, surgiendo alrededor suyo ciertos módulos de especial interés, los cuales se detallarán a continuación.

#### Lenguaje eXtensible de Hojas de Estilo (eXtensible Stylesheet Language – XSL)

*XML* separa de manera estricta la estructura del documento y la forma en que éste se presenta al usuario a través de las aplicaciones. Con *XSL* se puede asociar a una etiqueta un aspecto concreto, convirtiéndose en el sustituto de Hojas de Estilo en Cascada (*Cascading Style Sheets – CSS*). Es un estilo libre y puede adaptarse a diferentes salidas, por ejemplo, se pueden definir estilos diferentes para diferentes formatos de documentos.

#### Marco de descripción de Recursos (Resource Description Framework – RDF)

Es la plataforma que posibilita la codificación, intercambio y reutilización de la meta-información de manera estructurada. Además, proporciona una manera de publicar vocablos entendibles por humanos y procesadores, potenciando su reutilización y permitiendo la ampliación de las semánticas entre comunidades.

#### Simple API para XML (Simple API for XML – SAX) y Modelo de Objeto de Documento (Document Object Model – DOM)

*SAX* y *DOM* representan los dos enfoques en procesadores de *XML*, para crear una estructura de objetos con la información que contiene el documento *XML*.

Los *parsers SAX* leen los documentos linealmente, generando eventos al detectar símbolos de marcado. La ventaja de este protocolo es que no supone la construcción de una representación del documento, como sí ocurre en *DOM*.

*DOM* es un protocolo independiente de plataformas y lenguajes, que permite acceder dinámicamente a los contenidos, a la estructura y al estilo del documento *XML*. *DOM* convierte un documento *XML* en un conjunto de objetos que pueden ser modificados, proporcionando además un conjunto de objetos estándar y una interfaz para acceder a ellos y modificarlos.

### 5.3.4.2. Java y RMI

El principal atractivo de *Java* es la independencia de la plataforma, por lo que no es necesaria su re-compilación cuando se ejecuta un mismo código en lugares distintos, consiguiendo algo importante en los agentes autónomos, la *movilidad*. De esta manera, un agente puede cambiar de plataforma (máquina) para ejecutar sus tareas.

Otra ventaja de *Java* para el desarrollo de agentes es su orientación a la ejecución distribuida y por *Internet*, por tanto, se puede emplear protocolos de comunicaciones tipo *TCP/IP* directamente desde el código. Además, cuando se compila un programa en este lenguaje en una plataforma *Windows/Intel*, se obtiene el mismo código que en un sistema *Macintosh* o *UNIX*. La ejecución del programa pasa por emplear un intérprete de código de bytes, que procesa el programa compilado en forma de *bytecode*. En la Figura 5.10 se esquematiza el proceso y su independencia de plataforma.

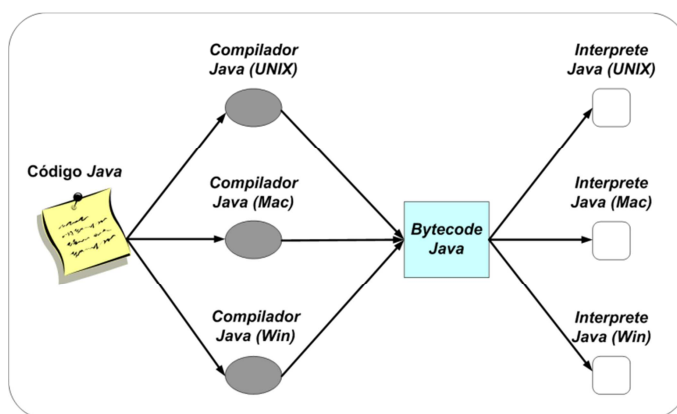


Figura 5.10. Java y la independencia de la plataforma. Fuente Propia.

*RMI* es un *RPC* para *Java*, que presenta algunas ventajas respecto a los métodos clásicos de *RPC*, ya que es parte del enfoque orientado a objetos que *Java* dispone. Las características y ventajas de *Java* son:

- Permite emplear objetos como argumentos de las llamadas.
- Las aplicaciones que emplean pueden cambiar su comportamiento de cliente a servidor y viceversa.
- Emplea los mecanismos y política de seguridad de *Java*.
- Sencillo de usar.



- Puede hacer uso de hilos para hacer procesamiento concurrente de las peticiones clientes.

La arquitectura de las aplicaciones *RMI* es siempre igual, está basada en dos programas diferentes: un servidor y un cliente. El servidor será una clase que creará objetos remotos, los hace accesibles y espera a que los clientes hagan peticiones a los objetos remotos. El cliente obtiene la información de los objetos remotos y realiza llamadas a métodos de los objetos. De manera resumida, las aplicaciones deben localizar los objetos remotos mediante el servicio *rmiregistry*, y poder comunicarse con los objetos remotos.

#### 5.3.4.3. *CORBA*

La ventaja de *CORBA* para los *MAS*, en concreto para sus comunicaciones, es su independencia de la plataforma o lenguaje que se emplee en las aplicaciones que necesiten sus servicios. Anteriormente se ha explicado la ventaja de emplear *RMI*, pero debe añadirse que a ambos lados de la comunicación deben existir procesos *Java*. En cambio, *CORBA* posibilita el intercambio de mensajes *XML* entre agentes que hayan sido desarrollados en cualquier tipo de lenguaje, y para el que se haya definido una correspondencia con el lenguaje de especificación de Lenguaje de Descripción de Interface (*Interface Description Language – IDL*).

*CORBA* plantea un modelo de distribución de objetos donde la plataforma soluciona, de forma transparente al programador, la manera de localizar los objetos y la forma de realizar el paso por mensajes. A los desarrolladores se les facilita las aplicaciones distribuidas, haciendo que no deban preocuparse de los detalles de implementación asociados a la distribución. Los objetos *CORBA* estarán accesibles mediante sus interfaces ya definidas. En un *MAS*, *CORBA* se pueden emplear para definir directamente interfaces de los agentes, y que éstos se puedan ver como objetos *CORBA*, o como mecanismo para implementar un sistema de transporte de mensajes *FIPA*.

#### 5.3.5. *Lenguaje natural: interacción persona–Agente*

El objetivo de un sistema de procesamiento de lenguaje natural entre un usuario (persona) y un agente o asistente virtual son, la identificación y, el saber transmitir el contenido y la intencionalidad de las intenciones. Estos sistemas pueden clasificarse en:

- *Catálogos on–line*: incorporan algún tipo de soporte al usuario.
- *Simuladores de conversaciones*: analiza la intervención en el lenguaje natural mediante patrones predefinidos.
- *Asistentes virtuales interactivos*: interfaces multimedia con un modelo simple de gestión de diálogo para tareas concretas a realizar en aplicaciones complejas, como en venta de billetes, traducción, tiendas en página *Web*.

Los problemas que se tienen en estos sistemas ya se han visto para los agentes, primero, dado que el asistente virtual o agente debe de tratar con el sub-lenguaje correspondiente a un dominio concreto, se precisa identificar la terminología del dominio y las relaciones existentes, para obtener un modelo del dominio, dicho de otra forma, una *ontología* y sus palabras asociadas. A partir de esto, se puede hacer una fase de estudio de frases, identificando la intención del usuario y poder extraer los datos semánticos incluidos en las frases, es decir, los detalles de la conversación necesarios para la gestión de la misma.

Para codificar la información de manera conjunta se recurre a la teoría de actos comunicativos de Searle (1979). En el ejemplo de tienda electrónica de herramientas de bricolaje de la autora, se definieron seis actos de comunicación: aportar información, solicitar información, cambiar la iniciativa, dar órdenes, actos de cortesía y nada que decir.

Además, se precisa una generalización de las frases más habituales con patrones y de una manera de anotar el contenido intencional, para poder asociarlas con los actos comunicativos, empleando para tal relación una *gramática semántica*. Como exponen Rodrigo *et al.* (2001) es un análisis intuitivo pero con baja cobertura.

Por último, el asistente virtual deberá tratar de responder de manera humana, por lo que se deberá dotar de cierto modelo de gestión de diálogo, para poder recordar ciertos detalles dados por el usuario, con la finalidad de no volverlos a requerir, incluso anticiparse a ciertas preguntas o aclarar con explicaciones en ciertos momentos de la conversación.

#### **5.4. COORDINACIÓN EN SISTEMAS MULTI-AGENTES**

Como ya se explicó, la comunidad científica tiene opiniones diversas en torno a cuándo poner la etiqueta de agente a ciertos módulos, objetos, componentes, etc.; y hasta su grado de comportamiento de manera inteligente. Pero lo que no se discute, es que en los MAS es clave la coordinación y, por tanto, se convierte en propiedad clave para cualquier agente. Lo que sí es más complicado es dar una definición exacta y concreta de coordinación en el ámbito de MAS.

Esto no debe sorprender, ya que el término coordinación se emplea en muchas disciplinas. Por ejemplo en el ámbito organizacional, se trata de predecir el comportamiento futuro de la organización. Los economistas se centran en la dinámica y estructura de los mercados y, por tanto, se precisará de una coordinación entre las partes. El interés del mundo informático radica en diseñar mecanismos de coordinación eficientes entre entidades computacionales encapsuladas, y dentro de esta área, se aplica con connotaciones diferentes si se habla de robótica, sistemas distribuidos o ingeniería del *Software*.

### 5.4.1. Modelos para la coordinación

En la teoría de la organización, según Malone and Crowston (1994), se define la coordinación como “*la gestión de dependencias entre actividades organizativas*”. La interrelación entre las diferentes actividades organizativas se modelan con una dependencia productor/consumidor, la cual podrá ser modificada con acciones de *notificación* y *transporte* adicionales en el flujo de trabajo.

La extensión de la organización a los *MAS* es muy sencilla, los sujetos cuyas actividades deben coordinarse con los agentes, los objetos de la coordinación suele conocerse por metas, acciones y planes. Normalmente, es posible establecer una *taxonomía* de las dependencias y un conjunto de acciones potenciales de coordinación asignadas a cada una de ellas.

D’Inverno and Luck (2004) plantea una formalización más reciente de las relaciones para *MAS*, el proceso de coordinación radica en la consecución de dos tareas: primero, *detección* de las dependencias; y segundo, tomar una *decisión* acerca de la acción de coordinación que deberá emplearse. Un *mecanismo* coordinador decide el modo en que uno o más agentes realizan las tareas (Ossowski, 1999).

Tanto la *calidad* como el resultado de la *coordinación*, son de diferente nivel de granularidad dependiendo los autores. Para Von Martial (1992) la coordinación es una *forma de adaptación al contorno*, al integrarse nuevos agentes al entorno de uno ya existente, será éste quien deba coordinarse y replantear planes, metas y acciones, función de las nuevas interrelaciones con los nuevos agentes. Según Nilsson (1998), al introducir un agente en un *MAS*, deberá adecuar sus planes individuales a las nuevas dependencias entre sus acciones y las posibles acciones potenciales del resto, a este nivel, para el agente, el resultado de la coordinación es mejor cuanto más contribuya a la consecución de sus metas en el nuevo entorno.

Desde el punto de vista del *MAS*, la coordinación y su consecuencia puede concebirse como un plan global. El plan se compartirá (Rosenschein and Zlotkin, 1994) si los agentes llegan a un acuerdo explícito durante la coordinación, o puede ser la suma de los planes individuales de cada agente como si *MAS* fuera un observador externo (Ossowski, 1999). La calidad de los resultados de la coordinación a nivel global podrá evaluarse según las metas conjuntas de los agentes, o por medio de la funcionalidad deseada global de *MAS*.

El modelo de dependencia de coordinación puede ser adecuado para *representar* características relevantes de un problema de coordinación en *MAS*. La detección de las dependencias deberá basarse en grandes cantidades de conocimiento sobre el dominio, siendo difícil adquirir y mantener la información ya que las vistas locales de los agentes resultarán inconsistentes e incompletas.

Como exponen Wegner (1997) y Busi *et al.* (2001), desde un punto de vista práctico es mejor concebir la coordinación como el esfuerzo de *administrar el espacio de interacción*. Si se hace que los agentes converjan hacia los patrones de interacción que resuelvan la relación de dependencia y tareas de decisión, se estará abordando la coordinación desde una perspectiva de *diseño*, pudiéndose encontrar diversas aproximaciones para abordar este problema, siendo algunas de estas aproximaciones: gestión de la reputación, estructuras organizativas, negociación, planificación, convenciones, etc.; la instrumentación de algunas de las aproximaciones puede hacerse por medio de *middleware* de coordinación (Omicini *et al.*, 2004).

En base a las características de clasificación de los modelos y mecanismo de coordinación anteriormente expuestos, se refleja el hecho de que cuanto más abierto sea un entorno más compleja será su coordinación. En un entorno cerrado, el comportamiento del agente podrá controlarse en *tiempo de diseño*, pudiendo implementar el mecanismo de coordinación implícitamente dentro de cada agente, por lo que el espacio de interacciones se determinará totalmente a la hora de construir el MAS.

En los entornos abiertos, será casi imposible prever e influir en el espacio de interacciones del agente, comportándose los agentes de una manera auto-interesada, siendo esta característica capaz de anticipar ciertas acciones de los agentes, y sirviendo como pistas para el diseño de estrategias de coordinación a nivel micro.

Según Omicini and Ossowski (2003), una aproximación para implementar la coordinación en los sistemas abiertos es ejerciendo un *control en tiempo de ejecución* sobre el agente. El control no puede imponerse, pero sí entregar diferentes servicios de coordinación junto a la descripción declarativa de su naturaleza. Los agentes podrán escoger un servicio en base a sus propios intereses, pero una vez registrados en un servicio de coordinación, la infraestructura del servicio hará que cumplan las restricciones conductuales.

Debe distinguirse entre la coordinación *directa* y la *mediada*. La primera involucra tan sólo a los agentes que deben coordinarse, mediante interacciones Punto a Punto (*peer-to-peer*), implementándose de forma descentralizada y apta para entornos abiertos, donde la coordinación puede hacerse en ejecución. La segunda se basa en la participación de agentes intermedios que ofrezcan servicios, como mínimo, de *páginas amarillas*, implementándose de manera centralizada y en entornos abiertos. Además, se implementa una coordinación mediana de modo *multi-céntrico*, en base a agentes intermedios múltiples y heterogéneos, atendiendo a las necesidades de ingeniería (eficiencia) como a las restricciones del entorno cada vez más abierto.

Continuando con lo anterior, Tolksdorf (2000) plantea un nuevo concepto, donde un mecanismo de coordinación se dice *acoplado*, si la eficacia del comportamiento de coordinación de un agente está basada en suposiciones sobre la actuación del resto de agentes. Cuando no se

imponen restricciones al comportamiento de los demás, estaremos ante un mecanismo *desacoplado*. La descentralización implica un alto grado de acoplamiento, con lo que se añadirá carga de comprobación del comportamiento de los agentes según lo esperado.

A nivel micro lo que entra en juego son los modelos *cualitativos* y *cuantitativos* (Ossowski and García–Serrano, 1998). La aproximación cualitativa sigue el modelo de dependencias ya expuesto, el comportamiento de coordinación local de un agente se basará en si sus acciones dependen positiva o negativamente de las acciones de los demás. En los modelos cuantitativos el problema de coordinación se esconde bajo la forma de una función de utilidad multi–atributo, el problema de decisión sobre su coordinación local equivaldrá entonces a un tipo de problema de optimización, determinando una acción local y tratar de inducir sobre el resto acciones locales para maximizar su utilidad (Ossowski, 1999).

La dualidad entre micro y macro fue abordada por Schumacher (2001), distinguiendo entre coordinación *subjetiva* y *objetiva*. La perspectiva *subjetiva* se centra en las interacciones desde dentro del sistema, por lo que cada agente monitorizará las interacciones interesantes para el agente y estudiará su evolución en el tiempo, y determinará las acciones a realizar para llegar a un estado global del sistema hacia un estado con las metas del agente. La perspectiva *objetiva* concibe la interacción del MAS desde fuera del agente, pretendiendo influir en las decisiones de los agentes, para que ciertas características del MAS estén en concordancia con los objetivos del diseñador. La coordinación *objetiva*, bajo ningún concepto trata de determinar el comportamiento de los agentes, violando el principio de *autonomía*, tal y como recogen Russell and Norvig (2003).

#### **5.4.2. Espacio de coordinación**

Un mecanismo de coordinación deberá poder identificar la necesidad de coordinarse así como las diferentes alternativas para hacerlo, es lo que se conoce como establecimiento del *espacio de coordinación*. Como ya se ha argumentado, la coordinación equivale a determinar y clasificar las interrelaciones entre los agentes implicados en el MAS.

En los entornos abiertos las dependencias son positivas, ya que los agentes pueden tener la posibilidad de ayudar a otros a alcanzar sus objetivos. Cuando un agente no sepa alcanzar la consecución de una tarea, precisará de un mecanismo para localizar otros que puedan realizarla por él. El mecanismo mostrará los *servicios* disponibles por otros agentes que podrían ser adecuados para solucionar el problema. Seguidamente, deberá seleccionar alguno de ellos, llegar a un acuerdo con los agentes que lo ofrecen, y ejecutarlo.

#### 5.4.2.1. Coordinación mediante agentes intermediarios

Los agentes intermediarios son los encargados de facilitar la coordinación entre agentes en entornos abiertos, mediante servicios de localización y comunicación entre agentes que proveen de servicios y agentes que los demandan. Por tanto, en el contexto presentado aparecen tres tipos de agentes, los que proporcionan servicios, los que lo solicitan y los agentes intermediarios. La mediación requiere el siguiente proceso:

- Los servicios son anunciados a los agentes intermediarios. Esta tarea la realizan los agentes proveedores.
- Los anteriores anuncios son almacenados por los agentes intermediarios.
- Un agente solicitante demanda a un intermediario la localización y comunicación con agentes que facilitan determinado servicio, pudiera incluir una transacción completa u otros servicios añadidos.
- Tras un proceso de decisión, y en base a las capacidades de los agentes registrados, el agente intermediario devuelve el resultado, pudiendo ser un subconjunto de anuncios junto a los nombres de los proveedores, o la transacción completa empleando el servicio más adecuado.

La idea es sencilla pero difícil de llevar a cabo en sistemas abiertos, donde existen agentes heterogéneos, desarrollados de manera diferente y con cambios frecuentes. Por tanto, la funcionalidad de los agentes intermediarios es facilitar la interoperabilidad de servicios y agentes, siempre de manera segura, flexible y fiable.

Como indican Decker *et al.* (1997), los agentes intermediarios ayudan a otros a localizar y conectar con proveedores de servicios. Además, un agente intermediario se debe caracterizar por habilidades relativas a:

- Prestación de servicios básicos de mediación a la sociedad de agentes. Deben de ser capaces de entender y procesar de manera automática peticiones y descripciones de servicios. Disponen de mecanismos para el almacenamiento y gestión, de manera eficiente, no sólo de descripciones de capacidades y servicios de los agentes registrados, sino también peticiones anteriores de agentes, conocimiento compartido entre agentes, entre otras cosas, pudiéndose emplear dicha información para mejorar la eficiencia y calidad de los servicios del agente intermediario. En ocasiones, el agente intermediario consulta a diferentes bases de datos u otras fuentes externas de información para satisfacer a los solicitantes del servicio.
- Coordinación de los servicios de mediación. Requiere capacidad de comunicación con los agentes, sistemas y usuarios implicados en el proceso de mediación conforme a protocolos, convenciones y políticas existentes. El agente intermediario debe de ser capaz de localizar a proveedores que presten el servicio, incluso cuando pertenecen a otros MAS, siendo este hecho transparente para el agente solicitante del servicio, esto se consigue mediante la

colaboración de los agentes intermediarios (Giampapa *et al.*, 2000). Además de la capacidad de comunicación con sus clientes, puede proporcionar servicios de interfaz a usuarios humanos para ver la información sobre servicios disponibles en la sociedad de agentes.

- Fiabilidad de la mediación. Proporcionarán calidad de servicio y confianza a sus clientes cuando soliciten servicios. Además de la calidad de servicio, deberá inspirar confianza a sus clientes, garantizando privacidad de datos, anonimato y verificación de las capacidades de los agentes registrados (Wong and Sycara, 2000). Los agentes intermediario deben de ser robustos frente a ataques de agentes o manipulaciones externas, además, los clientes de un agente intermediario no deben hacer mal uso de los datos facilitados por el agente intermediario durante la mediación.

Según Klusch and Sycara (2001) se distinguen tres tipos de agentes intermediarios:

- Mediador (Wiederhold, 1992): función de interfaz a los usuarios que solicitan el servicio, frente a los datos y recursos del conocimiento.
- Broker: funciones de interfaz entre los agentes que proporcionan los servicios y los que los emplean, haciendo de intermediarios entre las acciones. La totalidad de las comunicaciones entre pares se realiza a través del agente.
- Matchmaker: agente especial que empareja solicitantes de servicios con proveedores mediante equiparación de solicitudes con servicios anunciados por agentes.

#### 5.4.2.2. *Lenguajes de descripción de los servicios*

Dos de los lenguajes de descripción de servicios más empleados son Lenguaje de Descripción de Servicios Web (*Web Services Description Language – WSDL*) y Lenguaje Ontológico Basado en Web Semántica (*Ontology Web Language base on Semantic Web – OWL-S*).

*WSDL* es un lenguaje basado en *XML* para describir servicios *Web*. Estos servicios son aplicaciones que pueden ser descritas, publicadas, localizadas e invocadas en una red. *WSDL* intenta separar los servicios de los formatos de datos y protocolos empleados para la implementación. Un documento *WSDL* define los servicios como colecciones de puntos finales o puertos, estando separada su definición abstracta de su formato de datos.

*OWL-S* es una ontología escrita en *OWL* para describir servicios *Web*. El objetivo es que estos servicios sean interpretables por un programa, dicho de otra manera, descritos con suficiente información para permitir de manera automatizada, el descubrimiento, composición, monitorización e invocación de la ejecución de servicios. Desde el punto de vista de *ontología*, tiene la semántica bien definida, permite la definición de vocabulario de servicios en términos de objetos y las relaciones complejas entre ellos. El elemento principal de la *ontología* es la clase Servicio (*Service*), que se componen de tres *sub-ontologías*:

- *ServiceProfile*: describe qué hará el servicio, describiendo las propiedades necesarias para descubrimiento automático y demás precondiciones.
- *ServiceModel*: describe el flujo de control y de datos involucrados en el uso del servicio, dicho de otra manera, la forma en que se ejecuta el servicio.
- *ServiceGrounding*: detalles de la manera en que se accede al servicio, especificando el protocolo de comunicación, formatos de los mensajes, puerto del mismo, etc.

### **5.4.3. Toma de decisiones para la coordinación**

Una vez que los agentes tienen claro su modificación de planes, al interactuar con el resto de agentes, se tendrá que decidir la forma de modificar su actuación en vista a las diferentes dependencias detectadas. Las dependencias serán positivas cuando algún agente pueda ofrecer algún servicio que necesita el primero, mientras que se consideran negativas cuando un agente compite con el primero por el uso de un recurso. En un entorno abierto con agentes autónomas, la coordinación no es labor sencilla, ya que un agente no puede obligar a otro a prestar cierto servicio y, además, el modelo del mundo y del resto de los agentes es incompleto.

Las acciones de coordinación pueden basarse en *razonamiento social*, en base a la información del otro agente, pueden tratar de inferir sobre las acciones que el otro agente tomará, y actuar de forma correspondiente sobre sus objetivos, preferencias y capacidades. El mecanismo para que los agentes lleguen a una decisión conjunta se llama *negociación*. Finalmente, pueden existir *conversaciones* o *normas sociales* que podrán limitar las acciones de coordinación de los agentes, siendo la *ingeniería social* la que trata de diseñar las normas, intentando sesgar el comportamiento de los agentes autónomos interesados en su propio beneficio.

#### **5.4.3.1. Razonamiento social**

En (Sichman, 1995; Sichman and Demazeau, 1995) se presenta un ejemplo intuitivo de mecanismo simple de razonamiento. Plantean un mundo donde cada agente sigue un conjunto de objetivos locales, expresando las diferentes maneras de obtener cada objetivo por medio de planes, que supondrá una secuencia de acciones, y la necesidad de recursos.

Existe una descripción externa para cada agente, siendo esta parte del agente  $i$  que se refiere al agente  $j$ , la siguiente información:

- Conjunto de objetos del agente  $j$ .
- Conjunto de planes del agente  $j$  a partir de los cuales tratará de acometer sus objetivos.
- Conjunto de acciones y recursos del agente  $j$  para ejecutar sus planes.

Existe dependencia mutua entre dos agentes, respecto a cierto objetivo y plan concreto, si ninguno de los dos agentes puede alcanzar el objetivo común por separado, pero uniendo



capacidades si pueden ejecutar el plan y alcanzar el objetivo común. La dependencia recíproca es una relación débil, ya que no presupone que ambos agentes compartan el objetivo. Existe creencia mutua si el plan que consigue el primer agente se encuentra en la parte del segundo agente, en su descripción externa. Si las creencias se encuentran en el primer agente, se estaría hablando de creencias locales.

Por tanto, se definen cuatro situaciones de dependencia con respecto a dos agentes: Creencia Mutua de Dependencia Mutua (*CMDM*), Creencia Local de Dependencia Mutua (*CLDM*), Creencia Mutua de Dependencia Recíproca (*CMDR*) y Creencia Local de Dependencia Recíproca (*CLDR*). Se completa con la situación de Independencia (*IND*) cuando el agente es autónomo de su objetivo, y de Dependencia Unilateral (*DU*) cuando el primer agente depende del segundo, pero no viceversa. Cuanto más dependan dos agentes uno del otro, más dispuestos estarán a ayudarse entre ellos, por lo que se define un orden parcial sobre las situaciones de dependencia, que expresará el *grado* de dependencia de los agentes. Un agente estará más dispuesto a pedir ayuda a un agente con el que tiene situación de *CLDR* que con otro *DU*, de igual forma, interactuará con uno en situación *CMDM*, antes de negociar con *CMDR* o *CLDM*.

A veces se definen las interacciones entre agentes de manera *cuantitativa*, para lo cual se definen funciones de utilidad multi-atributo, siendo los atributos de la función las acciones que los agentes pueden tomar y sus valores representan de manera cuantitativa el grado de preferencia del agente con respecto a los resultados esperados de las acciones.

No obstante, a pesar de lo intuitivo de los anteriores métodos, si se quiere afrontar con rigurosidad la toma de decisiones, se debe acudir a la *Teoría de Juegos* de Owen (1995). Los agentes tendrán información completa del juego, conociendo la totalidad de las acciones y funciones de utilidad de todos los agentes.

#### 5.4.3.2. Negociación

Los mecanismos de negociación intentarán modelar la coordinación entre agentes con interés propio e intentaran llegar a acuerdos de vinculación. Según Müller (1996), los mecanismos permiten la convergencia hacia decisiones de coordinación conjuntas por medio de comunicación explícita.

Los mecanismos de negociación en *MAS*, se inspiran en modelos sociales y en especial de la economía (Sandholm, 1999). Además, muchos de los enfoques se fundamentan en la filosofía cooperativa de la *teoría de juegos* (Auman and Hart, 1994). Los mecanismos más importantes son: mecanismos de mercado basado en la teoría del equilibrio general (Varian, 1992); teoría del regateo, donde los agentes podrán llegar a un acuerdo beneficioso para todos, pero tendrán el problema de seleccionar qué acuerdo escoger (Thomson, 1997); la formación de coaliciones, donde

el objetivo de los participantes es maximizar su propia utilidad sabiendo que la misma actitud será adoptada por el resto (Friedman, 1991); y las reglas de votaciones, que es un mecanismo que deriva una regla de elección social que pueda clasificar los resultados sociales posibles, basándose en las clasificaciones individuales de esos resultados (Sandholm *et al.*, 1999).

#### 5.4.3.3. *Ingeniería Social*

La ingeniería social aborda el problema de toma de decisiones como un observador externo al sistema, siendo lo normal que sea uno de los diseñadores. La pregunta es, “¿cómo modificamos el espacio de coordinación para producir características deseadas en el comportamiento del sistema o agentes?”. Puede ocurrir, como en el dilema del prisionero (Simmons, 1983), donde la acción coherente sea no cooperar entre los agentes.

La ingeniería social plantea diferentes opciones para superar el anterior problema, tratando de modificar las reglas de juego de MAS. Tuomola (2000) modifica las funciones de utilidad, cambiando las penalizaciones de los diferentes casos. Más sorprendente fue el torneo realizado por Axelrod (1994), donde varios agentes diseñados por diferentes investigadores jugaron al Dilema del Prisionero Iterado, concluyendo que modificando las reglas del juego y permitiendo ciertas iteraciones repetidas, se podría lograr una opción anteriormente descartada.

#### 5.4.4. *Colaboración y cooperación*

Cuando los agentes han llegado a un acuerdo sobre el objetivo, se precisará especificar los detalles de la actuación, debiendo definir tareas concretas a realizar por cada agente, tiempo en el que deberían estar terminadas, qué hacer si fallan, etc.; de este problema se ocupa la Resolución Distribuida de Problemas (*RDP*), introducido por Durfee (2001), donde se supone que los agentes son *benévolos* (Durfee and Rosenschein, 1994), dicho de otro modo, que no persiguen su propio interés. Desde la perspectiva de ingeniería, la *RDP* aporta ciertas ventajas:

- Permite hacer uso concurrente de los recursos del sistema, teniendo repercusión en el rendimiento del sistema.
- La experiencia u otras habilidades para la resolución de un problema pueden ser distribuidas, permitiendo agentes especializados.
- Al distribuir datos y creencias, se puede tratar de forma local sin necesidad de enviar cantidades grandes de datos.
- Facilita la distribución física de agentes.

#### 5.4.4.1. *Dividir tareas*

Si un agente debe realizar una cantidad determinada de tareas, puede localizar a otros que le presten ayuda, por tanto, la división de tareas sigue los siguientes pasos:

- Encontrar las tareas que pueden ser realizadas por otros agentes, descomponiendo una tarea compleja en distintas tareas.
- Asignar las tareas anteriormente seleccionadas a diferentes agentes.
- Los agentes realizarán las tareas que les han encomendado, pudiendo a su vez descomponer y repartir dichas tareas de manera recursiva.
- Notificar los resultados de las sub-tareas y composición de los mismos para dar la solución definitiva.

Cuando las características de los agentes son muy distintas, y las sub-tareas emplean diferentes capacidades para resolverlas, la asignación de las mismas entre los agentes no es trivial. Una manera de descubrir la disponibilidad de los agentes es mediante el protocolo *Contract Net*, que anuncia el problema a un agente específico o a un conjunto de ellos, solicitando una respuesta en cuanto a su disponibilidad, para posteriormente asignar el problema al que responda de manera afirmativa. Si ninguno responde, se pueden acometer algunas de las siguientes opciones:

- Se puede actualizar el conocimiento de agentes objetivo emitiendo un *broadcast*, para que llegue a todos los agentes.
- Repetir la solicitud hasta que algún agente responda.
- Es posible que el primer anuncio sea muy restrictivo, por lo que es posible bajar el nivel de exigencia y volver a lanzar el mensaje.
- Es posible descomponer de nuevo el problema en base a los agentes disponibles.

#### 5.4.4.2. *Distribuir resultados*

Una misma tarea resuelta por diferentes agentes puede dar soluciones alternativas, por tanto, si se distribuyen los resultados entre los agentes que resuelven la misma tarea, se podría aumentar el rendimiento del grupo. La distribución de resultados presenta ciertas ventajas:

- Al tener un Retorno (*feedback*) de la solución de la tarea por medio de varios agentes, es posible aumentar la confianza de la solución, al poderla contrastar.
- Cada agente formula resultados de las tareas que se le han asignado, y todos los resultados juntos cubren una porción más completa de la tarea global.
- Un agente podrá emplear la solución de otros para refinar la suya propia.
- Con resolución paralela, se pueden emplear las soluciones de otros agentes para hacer que otro entregue la suya de manera más rápida.

Pero debe de tenerse cuidado con un par de cuestiones, primero, los agentes deben saber asimilar los resultados de otros agentes (qué hacer y cómo integrarlos), y en segundo lugar, ya que la asimilación puede no ser sencilla y el volumen de información importante, se debe ser selectivo con la información intercambiada.

La distribución de resultados es útil cuando un agente necesita hacer suposiciones acerca de situaciones que pueden influir en sus resultados. En otros casos puede suponer una sobrecarga intolerable de la red, cuestión que se evita empleando repositorios de información compartida, del tipo pizarra compartida o ciertas estrategias de búsqueda de resultados.

Una alternativa a la pizarra y a la búsqueda puede ser emplear una *estructura organizativa*. Esta estructura define roles, ciertas responsabilidades y preferencias entre los agentes que cooperan y, por tanto, los patrones de comunicación y control entre todos los participantes. En definitiva, estas estructuras muestran los agentes que estarán interesados en recibir resultados parciales.

#### **5.4.4.3. Planificación distribuida**

Previo a la colaboración entre agentes por medio de resolución distribuida, se precisa saber la manera en que se debe colaborar para resolver el problema, en otras palabras, se debe establecer para cada agente un plan de acciones a realizar que tenga en cuenta los planes de otros agentes.

Es posible que la ejecución de las tareas que forman el plan se distribuya entre varios agentes, estando el plan centralizado. También es posible que lo que se distribuya sea la obtención del plan, siendo centralizada la ejecución del mismo. Y por último, podría ocurrir que ejecución y plan fueran distribuidos.

Para la distribución total, se precisa que no existan conflictos entre los agentes durante su ejecución. Supóngase que varios agentes formulan planes individuales para ellos mismos y se debe asegurar que su ejecución se hará sin conflictos. Esta resolución de conflictos se puede hacer desde un enfoque centralizado, un agente recoge todos los planes individuales, analiza todos para ver que secuencias generan conflictos, y los modifica si es necesario. Se trata por tanto de un problema de posibilidad de éxito, donde partiendo de unos estados iniciales y una secuencia de acciones, ejecutables asíncronamente, se trata de enumerar todos los estados alcanzables, de entre ellos, se buscan los estados no deseados y se ponen restricciones para evitarlos. En ocasiones, enumerar el espacio de estados alcanzables resulta inabordable, por lo que se pueden emplear técnicas que reduzcan el espacio de búsqueda, como la propuesta por Georgeff (1983).

En ciertas ocasiones, las decisiones locales dependerán del resto de agentes, por tanto, resulta de interés estudiar hasta qué punto interesa que los planes locales atiendan a cuestiones de coordinación o que el agente se considerase de manera aislada. Una forma de aplicar las restricciones globales a los planes locales es que el agente construya todos los posibles planes

locales en lugar de tan sólo uno. Ephrati and Rosenschein (1994) plantean que los agentes refinan el conjunto de planes proponiendo ciertas modificaciones locales. Otro enfoque emplea la estructura jerárquica de un espacio de planes para llevar a cabo una planificación jerárquica distribuida.

Si al ejecutar un plan por varios agentes no se consigue lo esperado, se debe rehacer el plan, pudiendo existir diferentes maneras de afrontar el problema. Mediante la planificación de contingencias, cada agente formula su plan esperado y planes alternativos para responder a determinadas contingencias, teniendo los planes bifurcaciones condicionales. Otra alternativa consistirá en que cada agente supervise su plan y si se desvía, detenga la ejecución del resto de agentes y se re-planifica.

Durfee and Lesser (1991) presentaron un marco para la planificación llamado Planificación Parcial Global (*PPG*), caracterizado de la siguiente forma:

- *PPG* asume que los agentes pueden no tener consciencia de qué tareas podrían planificar otros agentes y las relaciones entre ellas.
- Antes de cualquier coordinación, un agente establecerá un plan local con alternativas, recogiendo todos los objetivos.
- Para conseguir una correcta coordinación, un agente deberá tener un modelo de sí mismo como del resto de agentes.
- Existe un conocimiento de meta nivel que especifica quién necesita conocer los planes de un agente concreto y quién impondrá nuevos planes.
- Los planes locales se pueden agrupar en un plan parcial global con las actividades concurrentes planificadas.
- Tras la reordenación de los planes locales, se comunican los resultados de tareas.
- Mientras se ejecuta, se decidirá el momento en que los cambios en los planes locales son suficientemente grandes para la re-ordenación.
- Es posible que ciertos agentes estén sobrecargados de tareas, y localizar quiénes están menos cargados, proponiendo modificaciones al plan para distribuir la carga.

## 5.5. HERRAMIENTAS Y MÉTODOS

Al comienzo de los *MAS*, y hasta no hace mucho, el desarrollo de los mismos era casi un arte más que un método estructurado. La evolución en el desarrollo de agentes y los problemas planteados desde el mundo de la ingeniería, ha supuesto que los agentes se consideren una nueva abstracción, pudiéndose emplear a lo largo de todo el ciclo de vida. A continuación, se mostrará qué proceso seguir para concebir, diseñar e implementar *MAS*, así como algunas de las herramientas disponibles.

### 5.5.1. Necesidad de herramientas y métodos para el desarrollo con Sistemas Multi-

#### *Agentes*

Un *MAS* lleva implícito la aplicación de diferentes tecnologías: la estructuración del desarrollo mediante la *Ingeniería del Software*; manejo de situaciones delicadas y toma de decisiones por medio de la *AI*; y la coordinación de la ejecución de tareas y su planificación en diferentes máquinas mediante *Programación Concurrente y Distribuida*.

Plataformas con soluciones parciales al modelado del comportamiento y coordinación de agentes pueden encontrarse en las que proporcionan servicios básicos como *JADE* (Bellifemine *et al.*, 2001; Garijo *et al.*, 2001), o entornos de desarrollo de *frameworks* como *ZEUS* (Nwana *et al.*, 1999).

Las plataformas de desarrollo quedan incompletas sin procesos de desarrollo de *Software* para agentes, para acercarlos a la producción *Software* convencional. Por tanto, se han planteado diferentes *metodologías* centradas en agentes, las cuales parten de un modelo que defina el *MAS* y describen guías y métodos para su construcción.

### 5.5.2. Cuándo desarrollar con Agentes

La primera pregunta que debe hacerse es si realmente es necesario abordar un problema mediante desarrollo *MAS*, o se pueden aplicar técnicas más sencillas, como por ejemplo las desarrolladas en objetos.

Lo primero a tener en cuenta es la distribución de entidades que componen el sistema en entornos abiertos, por ejemplo piénsese en sistemas de servicios *Web*, donde hay infinidad de variables a controlar y difíciles de gestionar con un enfoque tradicional, además, estas entidades pueden evolucionar constantemente y actuar de manera concurrente. En estos entornos, una acción en el entorno no permitirá saber de antemano el efecto que producirá, tan sólo la intención de lo que pretende. Lo anterior supone una indeterminación en la dinámica del sistema, por lo que los agentes, con su adaptación y planificación, resultan idóneos.

Cada día existen más dispositivos distribuidos, de diferentes tecnologías y fabricantes, con diferentes capacidades de cómputo y funcionalidad, conectados a redes fijas y móviles. Por este motivo, la heterogeneidad en estos entornos es una característica a tener en cuenta, debiendo disponer de mayores grados de distribución en la gestión, la localización del control y en las interacciones. Los agentes encajan en esta filosofía desde su concepción, ya que son entidades autónomas que residen en diferentes nodos de la red, o incluso, como en el caso de los agentes móviles, pueden migrar entre dichos nodos.

La interacción entre los diferentes componentes podría necesitar grados de abstracción mayores, con más soporte en el tratamiento semántico de la información. El uso de *ontologías* y lenguajes de comunicación entre agentes supone un avance con respecto a los métodos de comunicación tradicionales, como los empleados en los procesos de objetos distribuidos, que están limitados por la interoperabilidad semántica.

Un último detalle es la usabilidad, ya que hoy en día se requieren sistemas altamente configurables, donde cada usuario requiere un tratamiento especial. Se puede abordar teniendo un agente especial, a modo de asistente, para cada usuario, con capacidad de aprender y adaptarse a los cambios en el perfil del usuario.

La primera pregunta que deberá hacerse es si el sistema a abordar tiene peculiaridades para hacer un planteamiento mediante agentes en lugar de por paradigmas tradicionales. Algunas preguntas a contestar, planteadas en (Mas, 2005; Wooldridge, 2009) son:

- ¿Es un sistema abierto?, ¿se pueden incorporar nuevas entidades? o ¿pueden cambiar las existentes?
- ¿Los componentes, o al menos alguna parte significativa, pueden cambiar su comportamiento de manera independiente?
- ¿Existe incertidumbre en los procesos o planteamientos?
- ¿Es necesaria la personalización de los servicios?
- ¿Se precisan definir organizaciones de entidades para la resolución problemas de manera conjunta?

Si algunas de estas preguntas se pueden responder de manera positiva, es posible abordar el problema con un planteamiento basado en agentes.

### **5.5.3. Modelado con Agentes**

En el desarrollo *Software* clásico, los modelos secuenciales presuponen que con suficiente estudio y análisis, se posibilita la identificación de los requisitos del sistema antes de comenzar con el diseño y su implementación. Se ha demostrado que esta suposición es falsa y que incluso los requisitos del sistema evolucionan con el tiempo, por lo tanto, el desarrollo del *Software* deberá asumir esta nueva dificultad. Se han ideado nuevos modelos evolutivos, donde el sistema se construye en base a unas interacciones, proporcionando nuevas funcionalidades al sistema de manera incremental. Es por tanto que surge la necesidad de un nuevo paradigma que facilite la reutilización, como es el caso de la orientación a objetos, mediante herencia de clases y composición de objetos.

El planteamiento para los agentes *Software* no cambia con respecto a los modelos orientados a objetos, es más, casi todas las metodologías adoptan un modelo orientado a objetos, definiendo

ciertas actividades específicas para la concepción del sistema de agentes. Por tanto, en las siguientes secciones, la visión se centrará en aspectos metodológicos.

### **5.5.3.1. Diferentes puntos de vista de MAS**

Las diferentes metodologías MAS existentes, proponen hacerlo desde puntos de vista complementarios, para así gestionar mejor la complejidad de MAS.

#### Metodología AAI/BDI

Kinny *et al.* (1997) presentaron una de las primeras metodologías desde el Instituto de Inteligencia Artificial Australiano (*Australian Artificial Intelligence Institute – AAI*), considerando un punto de vista externo, donde se identifican agentes y sus interacciones, y un punto de vista interno, donde se describen los comportamientos de los agentes.

La metodología consiste en un proceso de refinamiento de los dos puntos de vista comentados, considerando primero el externo, posteriormente el interno, que a su vez realimentará a los modelos externos, y así sucesivamente hasta conseguir un nivel de detalle suficiente para la implementación.

La finalidad del punto de vista externo será plantear una jerarquía de clases de agentes y sus relaciones. Para conseguir tal fin se plantean una serie de pasos:

- Definición de los Papeles (*roles*) en el dominio de aplicación, y a partir de ellos se definirán las primeras clases de agentes.
- Identificar las responsabilidades y servicios asociadas a cada *rol*.
- Identificar interacciones, *performative* y contenido de información para cada servicio.
- Refinar la jerarquía de clases y las relaciones de control.

Con el proceso anterior se consigue asignar la funcionalidad a los agentes, por medio de servicios, y determinar sus relaciones (interacciones).

El punto de vista interno sigue el modelo BDI, donde las creencias son la representación de información sobre el entorno del agente (estado interno), y las acciones que es capaz de realizar (*Modelo de Creencias*). El agente persigue la consecución de sus objetivos, los cuales definen el estado ideal del mundo que espera encontrar el agente (*Modelo de Objetivos*). El control del agente se define por un conjunto de planes (*Modelo de Planes*). La metodología para obtener estos modelos se resume en dos pasos:

- Análisis de medios para alcanzar los objetivos, descomponiendo los objetivos en acciones más sencillas.
- A partir del análisis del contexto y las condiciones de control de las actividades, construir las creencias.

#### Ingeniería de las vocales

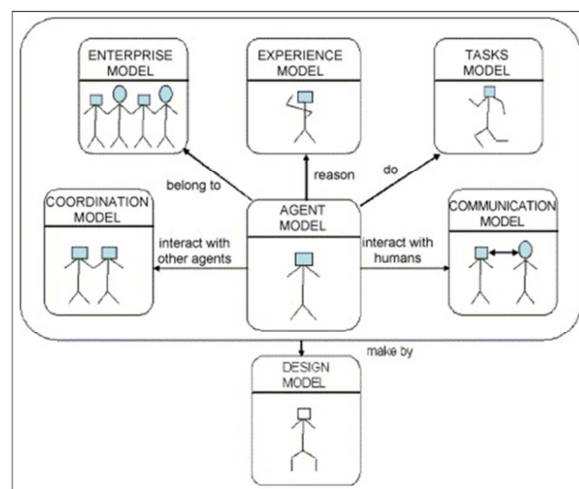


Ricordel (2001) propuso cinco puntos de vista diferentes para su metodología *Ingeniería de las Vocales*. Los puntos de vista corresponden a las vocales: Agente, Entorno, Interacción, Organización y Usuario.

El orden de aplicación de las vocales dependerá de la clase de sistema a desarrollar, por ejemplo, si lo más importante son las relaciones sociales, se deberá comenzar por la organización. Si el proceso comienza con los agentes, es posible que la organización resulte de la interacción de esos agentes individuales.

### MAS-CommonKADS

Este modelo, cuyo ejemplo se puede ver en la Figura 5.11, fue propuesto por Schreiber *et al.* (1994), y ya tenía la idea de diferentes puntos de vista. En concreto se consideran siete modelos: modelo de organización, modelo de tareas, modelo de experiencia, modelo de coordinación, modelo de agente, modelo de comunicación y modelo de diseño.



**Figura 5.11. Modelo MAS-CommonKADS. Fuente (Iglesias, 1998; Cerrada *et al.*, 2007).**

Iglesias (1998) incorporó la notación orientada a objetos para estructurar MAS, tal y como se ve en la Figura 5.11, definiendo casos de uso para la captura de requisitos, y técnicas de ingeniería de protocolos para especificar las interacciones entre los agentes.

### MESSAGE e INGENIAS

*MESSAGE* (Caire *et al.*, 2001) integra varias tecnologías y resultados alrededor de agentes, considerando cinco puntos de vista, de manera similar a *Ingeniería de las Vocales*. La novedad reside en describir los elementos de modelado del MAS, por medio de los diferentes puntos de vista empleando meta-modelos (Gómez-Sanz *et al.*, 2002).

La metodología *INGENIAS* (Pavón and Gómez-Sanz, 2003) es una extensión de los anteriores meta-modelos, apoyándose en un conjunto de herramientas llamadas *INGENIAS Development Kit (IDK)*, construidas sobre dichos meta-modelos. La metodología *INGENIAS* considera cinco puntos de vista para el modelado de MAS (Figura 5.12):

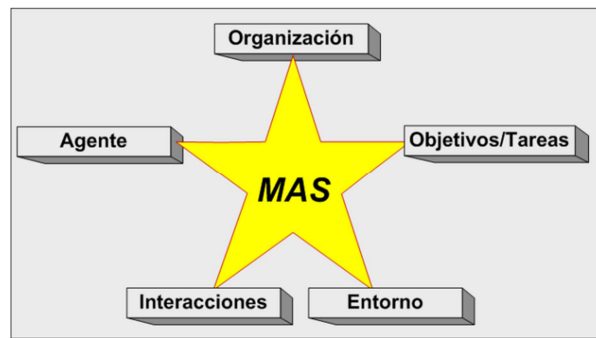


Figura 5.12. Metodología INGENIAS, puntos de vista de un MAS. Fuente Propia.

- Agente: las responsabilidades se describen con tareas y roles, y el control del agente se define con objetivos y estados mentales durante su ejecución.
- Organización: describe el propósito del sistema, tareas, recursos y el marco de existencia de los agentes.
- Entorno: define los sensores y actuadores de los agentes, así como el resto de recursos a interactuar con los agentes (bases de datos, servicios *Web*, etc.).
- Objetivos y tareas: se basa en el principio de racionalidad (Newell, 1982), siendo su propósito justificar la ejecución de tareas en función de sus objetivos. También facilita la descomposición de tareas y objetivos.
- Interacciones: describe la coordinación entre agentes. Incluye información del estado mental de los agentes en la ejecución, así como las tareas de la misma.

#### Otras metodologías

*MASSIVE* (Lind, 2000) es una metodología que considera el modelado de *MAS* desde siete puntos de vista: sistema, arquitectura, sociedad, interacciones, roles, tareas y entorno. *ODAC* (Gervais, 2003) presenta cinco puntos de vista: ingeniería, tecnología, computacional, información y empresa.

#### **5.5.3.2. Roles y servicios**

Al afrontar la organización dentro de un *MAS*, surge de manera espontánea el concepto de *rol*. Un *rol* identifica las funcionalidades cuando nos referimos a servicios, y las características de las partes en las interacciones. Si un agente presenta cierto *rol*, dispone de las capacidades necesarias para realizar los servicios asociados al mismo.

#### Metodología Gaia

La metodología *Gaia* (Wooldridge *et al.*, 2000) considera un conjunto de agentes como una organización o sociedad. La organización está formada por *roles* y sus relaciones, así como otros *roles* institucionales. El análisis *Gaia* tiene tres pasos:

1. Identificar la totalidad de los *roles* del sistema, pudiendo ser individuales u organizacionales.

2. Identificar y documentar los protocolos de cada *rol*.
3. Con todo lo anterior, elaborar un modelo de *roles*.
4. Iterar 1 a 3.

Tres son los modelos que *Gaia* produce, a saber:

1. El *Modelo de Agente* que identifica los diferentes tipos de agente y sus instancias en el sistema.
2. El *Modelo de Servicios* que define las funciones de cada agente.
3. El *Modelo de Conocidos* que es un grafo dirigido con los enlaces de comunicación e los agentes.

La metodología está algo limitada, ya que simplemente describe la manera en que coopera la sociedad de agentes, pero no la forma de definir el modelo computacional del sistema, por lo que los desarrolladores tendrán mucho trabajo hasta conseguir un sistema que pueda implementarse.

### Metodología MaSE

Como alternativa a *Gaia*, y teniendo todo el ciclo de desarrollo, DeLoach (2001) presentó *MaSE*. Sigue el paradigma orientado a objetos, mediante notación *UML*, considerando a los agentes como objetos especializados. La especialización será la coordinación entre los agentes a través de diferentes conversaciones y en la naturaleza proactiva de los agentes.

Los pasos de que consta *MaSE* se muestran en la Figura 5.13, pudiéndose realizar con la herramienta *agentTool*. Como puede observarse, existe una separación entre el análisis y el diseño. Un aspecto incidente de *MaSE* es la conversación entre agentes, definida como un protocolo de coordinación entre dos agentes y se describe con dos autómatas, uno para el iniciador y otro para el respondedor. Por esto, la mayoría de las implementaciones con esta metodología se limitan a la definición de agentes reactivos.

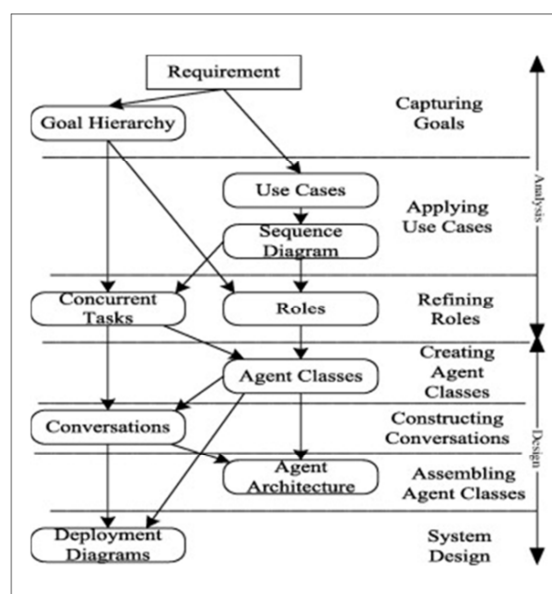


Figura 5.13. Fases de *MaSE*. Fuente Juan *et al.* (2009).

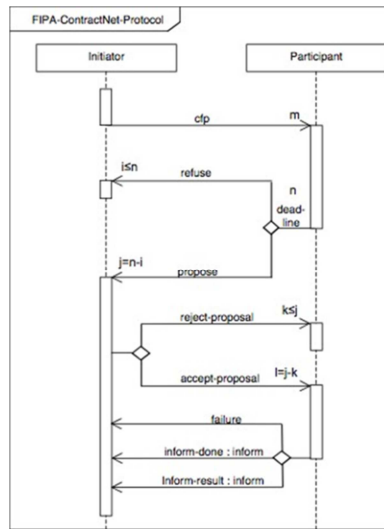
Como se ve en la Figura 5.13, la metodología *MaSE* considera importante el análisis basado en roles, a partir del cual estudiar las intenciones y las tareas que determinarán los tipos de agentes y la estructura de *MAS*.

Metodología AALAADIN

La metodología *AALAADIN* (Ferber and Gutknecht, 1998) se basa en que *roles* y *servicios* ayudan a definir la estructura de funcionalidad asociada a un agente o grupo de éstos. Se empleó un meta-modelo genérico de *MAS* con tres conceptos básicos: agentes, grupos y roles.

**5.5.3.3. Lenguaje de descripción de agentes**

*FIPA* ha planteado la creación de un lenguaje de notación común para agentes, extendiendo *UML* y cuyo nombre es *AUML*. En la Figura 5.14 se representa el protocolo de interacción *FIPA Contract Net*, según la notación *AUML*.



**Figura 5.14. Especificación AUML: protocolo de interacción FIPA Contract Net (FIPA, SC00029H). Fuente Propia.**

Bauer *et al.* (2001) extendieron *UML* para especificar los protocolos de interacción de agentes, empleando mecanismos para definir los *roles* de los agentes, líneas de vida de los agentes, protocolos y una semántica extendida de los mensajes *UML*. Se puede modelar clases de agentes, empleando rasgos especiales para las características de autonomía, pro-actividad, reactividad y comunicación basados en el acto del habla.

Parunak and Odell (2002) presentan otra propuesta donde consideran conceptos relacionados con la organización tales como *roles*, dependencias, actos del habla, planteando tres mecanismos:

- *Grupo*: conjunto de agentes asociado por algún interés o tarea. Un grupo es modelado como un diagrama de clases con líneas de partición para organizar *roles* dentro del grupo.

- *Rol*: representa la función del agente, pudiendo jugar varios papeles un mismo agente, incluso en organizaciones distintas. Cualquier tipo de relación (entre organizaciones, agentes y *roles*) se puede representar mediante diagrama de clases.
- *Entorno*: encargado de la distribución de información entre agentes.

### 5.5.4. Implementación

#### 5.5.4.1. Agentes: lenguaje de programación

La mayoría de los *MAS* se desarrollan bajo *Java*, ya que aporta muchas bibliotecas de clases de interfaz para usuarios, comunicaciones, etc.; por lo que se pueden añadir métodos y clases de manera sencilla. Siguiendo con esto, se han presentado implementaciones basadas en *Java* inmersas en las plataformas *JADE* y Construcción de Agente y Entorno de Aprendizaje (*Agent Building and Learning Environment – ABLE*). De la misma forma, *java* permite la integración con otros paradigmas de programación lógica, como *Ciao–Prolog*, o sistemas basados en reglas de producción como *Joint Expert Speciation System (JESS)* (May and Murray, 1991), lo que permite la implementación de los agentes en estos lenguajes para una posterior integración en alguna plataforma común. Tanto *Ciao–Prolog* como *JESS* permite integrar capacidad de razonamiento a los programas desarrollados en *Java*, incorporando conocimiento declarativo mediante reglas y hechos, e incorporar un motor de inferencias eficiente. Esto permitirá el desacople del control del agente de la codificación de sus actuadores y, además, se gana reutilización y se facilita la depuración. No obstante, el trabajo con *JESS* requiere mucha disciplina, primero porque *JESS* no se compila, limitando la detección de errores semánticos, y segundo por el gran número de reglas que pueden existir en un momento dado.

Esta última situación puede presentarse incluso en la codificación de un agente reactivo, en donde un conjunto de reglas de producción con eventos relacionados con la percepción del lado izquierdo y su correspondiente invocación de actuadores en la parte derecha. Este tipo de codificación no se aconseja, debido al gran número de reglas que pueden aparecer de manera rápida. Al disponer de un gran número de reglas, el desarrollador debe asignar prioridades a las reglas y codificar meta-reglas que regulen la ejecución de otras reglas controlando los hechos que las disparan.

Para los agentes *BDI* es el propio modelo quien estructurará las reglas a disparar, pudiéndose realizar la estructuración de dos maneras, por ejemplo atendiendo a la forma de gestionar los objetivos del agente, de tal manera que se crean los objetivos, se refinan, se tratan de resolver y quedan resueltos o fallidos. Por tanto un objetivo puede recibir la siguiente información

$$\text{Activar} \rightarrow \text{Refinar} \rightarrow \text{Resolver} \quad (5.3)$$

o bien

$$\text{Activar} \rightarrow \text{Refinar} \rightarrow \text{Resolver} \rightarrow \text{Gestionar Fallos} \quad (5.4)$$

Por tanto, se necesitan crear reglas para ejecutar dichas acciones, pudiendo imaginar que el número de reglas puede dispararse en el momento de tener unos pocos objetivos. Todo se estructura alrededor de los objetivos y su ciclo de vida, lo que facilita niveles adicionales de control, como por ejemplo, sobre qué objetivo se debe centrar el agente, creación de objetivos durante la marcha, abortado de objetivo, entre otros.

#### 5.5.4.2. Generar código de manera automática

Debido a que los agentes se pueden desarrollar en diferentes lenguajes y ejecutar en una variedad distinta de plataformas, es natural que se busque el desacoplo entre especificación de *MAS* y su implementación. Por tanto, se está investigando en mecanismos que transformen la especificación en código ejecutable, para no añadir esfuerzo al diseñador/analista y que no comprometa el desarrollo con la tecnología.

Algunas herramientas de desarrollo de agentes permiten generar código ejecutable que implemente la especificación *MAS*, como las ya comentadas *agentTool* y *ZEUS*. Lyytinen and Rossi (1999) presentan *METAEDIT+*, que define un lenguaje de exploración de modelos de una especificación y que permite imprimir esta información junto a esqueletos de código, documentos, o lo que precise el desarrollador. Pero a esta solución le ocurre algo similar que a *agentTool*, se mezcla el código que extrae la información junto al esqueleto del código a parametrizar, no siendo aconsejables si se pretende personalizar el proceso de generación del código, ya que las modificaciones afectarían al código de la especificación y al código responsable de volcar la información en los esqueletos de código.

La metodología *INGENIAS* integra como proceso del *MAS* la propia configuración y desarrollo de los procesos de generación de código, proporcionando herramientas que permiten el paso de especificación a implementación. Estas herramientas forman parte de *IDK* (Grasia, 2004), incluyendo un editor de especificaciones y ejemplos de generación de código para diferentes plataformas. Los esqueletos de código, llamados *plantillas*, serán ficheros externos que el desarrollador podrá modificar, a partir de ellos actuará el procesador de especificaciones, creando copias de plantillas y rellenándola con información de la especificación para obtener el código definitivo.

El objetivo de *INGENIAS* es producir un artefacto *Software* llamado módulo, capaz de generar código de forma automática para una plataforma de agentes concreta. La implementación de la especificación se realiza de manera iterativa, tal y como muestra la Figura 5.15. Inicialmente se elabora un prototipo reducido de una parte de la funcionalidad a implementar, se traduce dicho

prototipo al sistema de plantillas de *INGENIAS* y se programa un proceso de extracción de información sobre los diagramas definidos, con el fin de determinar con qué información rellenar las plantillas. En cada iteración se añaden más partes de la especificación hasta que el desarrollador lo considera oportuno.

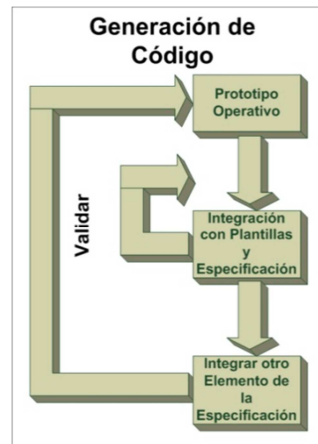


Figura 5.15. *INGENIAS*: desarrollo iterativo de un módulo. Fuente Propia.

El módulo forma parte del *IDK*, y su construcción consistirá en la definición de un algoritmo que recorrerá los diagramas, extraerá la información de entre ellos, y volcarla a un conjunto de plantillas. La descripción del recorrido se realiza mediante una *API* de *Java* que proporciona visión de la especificación en forma de grafos con nodos y aristas. Se emplea el mismo *API* para extraer la información y obtener los datos deseados, y finalmente, la instalación de plantillas se realiza automáticamente mediante librerías propias de *IDK*.

La construcción de un módulo, implementando agentes *JADE*, comienza desarrollando un prototipo inicial de agente, y para facilitar su desarrollo, se comienza de la siguiente manera:

- Un agente tan sólo conoce del protocolo las unidades de interacción en las que participa.
- En el instante inicial, no se conocen los agentes que participarán, sólo los *roles* necesarios.
- El agente dispondrá de mecanismos redundantes para controlar que la interacción transcurre como debe.

La arquitectura base sobre la que se soporta es *JADE* (sección 5.5.4.3), añadiendo sobre ella un nivel adicional que es el que proporciona lo requerido funcionalmente. En la Figura 5.16 se representa una arquitectura de agente prototipo sobre plataforma *JADE*.

La capacidad de gestión de diálogos la introduce *JADE Agent*, de la clase de *JADE* *jade.core.Agent*, cuya función será revisar el estado mental y decidir si se aborta o no la comunicación, si se disparan nuevas intenciones o si se ejecuta una tarea.

El estado mental del agente se implementa dentro de la clase Estado Mental (*Mental State*), tal y como se ve en la Figura 5.17. *Mental State* se implementa como una *pizarra*, donde acceden diferentes actuadores del agente, y esta clase es empleada por *DefaultCommControl* a lo largo de la comunicación para decidir cuál será el siguiente estado al que transitar. Al enviar o recibir un

mensaje, se debe decidir si se transita o no al estado, siendo la decisión dependiente del estado mental actual, decidiendo las entidades que deben estar presentes y los valores a observar.

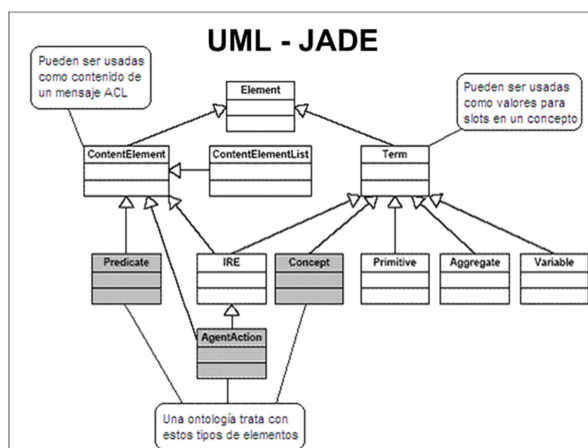


Figura 5.16. Especificación UML de arquitectura de agente ejemplo expresado según plataforma JADE. Fuente Propia.

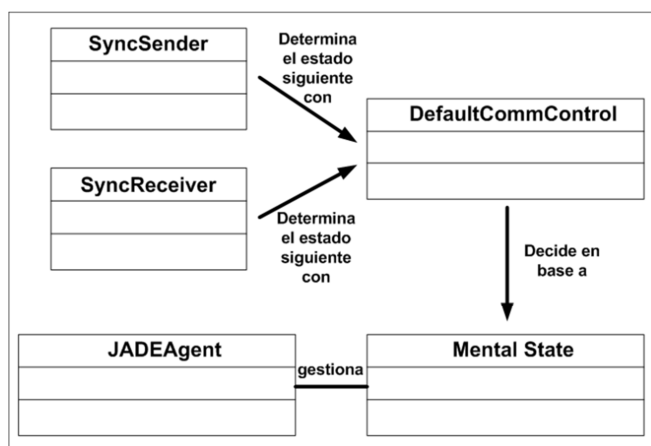


Figura 5.17. Relación entre Mental State y las comunicaciones del agente. Fuente Propia.

### 5.5.4.3. Plataformas de interconexión de agentes

Existen numerosas plataformas de agentes, algunas siguen el estándar FIPA y otras se basan en arquitecturas de agentes propios.

Java Agent Template Lite (JatLite) es una plataforma orientada a las comunicaciones entre agentes. Algunas soluciones para agentes móviles, basadas en el estándar MASIF son Aglets de IBM y Grasshopper de IKV++. AgentBuilder de Reticular Systems es un entorno que permite construir agentes BDI en Java. IBM desarrolló la plataforma ABLE para la definición composicional de agentes propios.

Siguiendo el estándar FIPA, las más conocidas son FIPA Open Source de Nortel Networks, que implementa el lenguaje de comunicación entre agentes y su plataforma para gestión, y JADE desarrollado por CSELT S.p.A. y con un uso altamente extendido.



A continuación se explicarán las plataformas más características y extendidas, *JADE* y *ABLE*, la primera por su uso generalizado y la segunda por permitir la reutilización de componentes.

### La plataforma JADE

*JADE* proporciona un conjunto de librerías para la creación de agentes y su posterior comunicación mediante lenguaje *FIPA ACL*, y acceder a los servicios estándar de gestión de agentes. Además, *JADE* entrega varios agentes implementados con diferentes propósitos y unidos a la plataforma:

- *Directory facilitator*: proporciona los servicios de directorio y es posible conectarlo a agentes similares para gestionar una posible jerarquía de dominios.
- *Agent Management System*: supervisor de uso y acceso a la plataforma, y existe una por plataforma de ejecución *JADE*. Mantiene una lista de identificadores de agentes y su estado, un agente debe registrarse inicialmente para conocimiento del sistema.
- Escuchador (*Sniffer*): se emplea para la depuración de las interacciones entre agentes. Intercepta mensajes entre agentes y muestra de manera gráfica la información, para de esta manera poder depurar sociedades de agentes.
- Introspector (*Introspector*): controla el ciclo de vida del agente en ejecución y los mensajes de intercambio, tanto los de la cola de entrada como los de salida.

Dentro de cada plataforma *JADE* se pueden definir diferentes contenedores, uno para los agentes del sistema, y varios para los de las diferentes aplicaciones. En la Figura 5.18 se presenta un contenedor llamado *Front-End*, donde se puede apreciar la existencia de diferentes agentes.

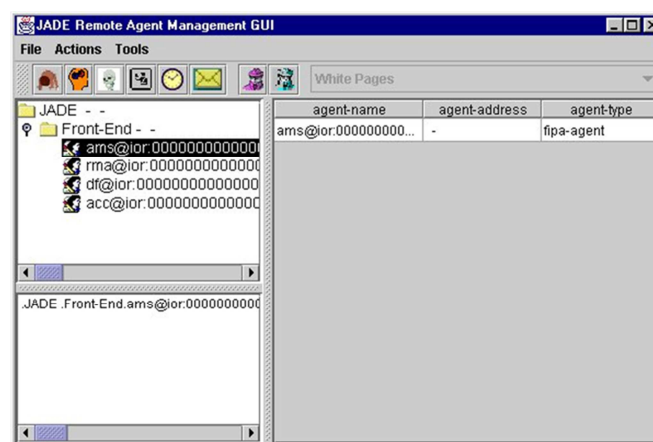


Figura 5.18. Pantalla de control principal de plataforma de ejecución *JADE*. Fuente Mas (2005).

Para la implementación de un agente, lo primero que debe es heredar de la clase *jade.core.Agent*; dicha clase proporciona métodos para iniciar el agente, así como posibilitarle para el envío y recepción de mensajes con el resto de agentes, a través de la misma plataforma u otras.

El envío de mensajes pretende causar cierto efecto sobre el receptor, en función de tal efecto, se elabora una lista de *actos del habla*, como informar, requerir, preguntar, etc.; el mensaje, además

de clasificarse dentro de un acto del habla, incluirá información para mantener una conversación (nombre del protocolo, ontología, lenguaje del contenido y contenido del mensaje).

El contenido se codifica en el lenguaje *SL-0*, pudiendo además definir la ontología particular. Al crear la ontología se crea de manera automática un codificador/decodificador del contenido del mensaje, de manera que el implementador no debe preocuparse de la forma de convertir una información a texto o la manera de analizar el mensaje. *JADE* crea las clases *Java* necesarias para todo este proceso, creando la plataforma los codificadores/decodificadores apropiados.

*JADE* hace lo mismo a nivel de protocolos de comunicaciones, implementando los estándares *FIPA*, de manera que los agentes pueden emplearlos para interrogar al resto y obtener información.

La integración de todo lo anterior se realiza por medio de una arquitectura agente básica que *JADE* proporciona. El acceso a servicios y recursos de la plataforma se hace por medio de código encapsulado en clases ya predefinidas, *jade.core.Behaviours*, que proporciona el método *action()* para describir la tarea que se asocia a cada uno de los comportamientos. Los comportamientos permiten emitir y recibir mensajes, y organizar las actividades de los agentes en bloques de código que se ejecutarán de forma secuencial o no determinista. De esta manera, todo el comportamiento del agente se codifica en estas clases, aunque algunos investigadores deciden construir una capa sobre los comportamientos, para permitir expresar de manera más clara la implementación de la conducta del agente.

### La plataforma ABLE

La plataforma permite la definición de modelo de componentes y el desarrollo de la programación por medio de herramientas, pero además, proporciona ciertos componentes para razonamiento automático y el aprendizaje (Bigus *et al.*, 2002).

*ABLE* entrega un conjunto de interfaces y clases que permite crear componentes *JavaBeans* llamados *AbleBeans*. A partir de éstos, los programadores podrán extenderlos o adecuarlos a sus necesidades, quedando definido por:

- *BeanInfo*: entrega una descripción del componente, consultable en tiempo de ejecución.
- *Customizer*: posibilita la parametrización de ciertas características de *AbleBeans*.
- *Inspector*: monitorización de las entradas y salidas de los *AbleBeans* concretos.

Los agentes se llaman *AbleAgents*, y están constituidos por varios *AbleBeans* de funcionalidad genérica. La formación de los agentes, a partir de los *AbleBeans*, se pueden conectar de tres maneras diferentes: con flujos de datos, por eventos y mediante propiedad. Las conexiones con flujos de datos se emplean para aplicaciones del tipo *ANN*, que precisan procesado de datos. En las conexiones por eventos, los objetos se registran como interesados en un modelo

suscriptor/publicador, para recibir información de los *AbleBeans*. El uso de propiedades permitirá sincronizar propiedades en *AbleBeans* diferentes.

Los diferentes *AbleBeans* heredan de la clase *AbleObject* que proporciona los métodos necesarios, *init()* para inicializar el *AbleBeans*, *process()* para tratamiento de los *buffers* de entrada y *processTimerEvent()* para se quiere que el *bean* sea un agente autónomo.

Para el desarrollo de los agentes, se entrega un editor a través del cual se podrán conectar diferentes *AbleBeans* y definir el comportamiento de los agentes, tal y como se ve en la Figura 5.19.

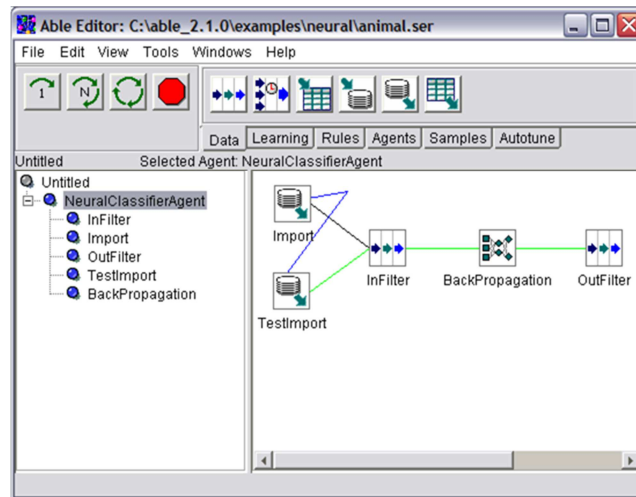


Figura 5.19. Editor *ABLE* para implementar agentes. Fuente Mas (2005).

La plataforma *ABLE* proporciona la clase *AbleDefaultFIPAAgent* para el desarrollo de agentes con requerimientos *FIPA*. La arquitectura distribuida de *ABLE* se corresponde con las especificaciones *FIPA97* y adaptado a *Java Agent Services (JAS)*, que proporciona un conjunto de interface *Java* e implementación de referencia de servicios que requiere una plataforma de agentes distribuida conforme a la arquitectura abstracta *FIPA*.

### 5.5.5. Validación y pruebas

Según Wooldridge (2009), existen dos enfoques diferentes para la verificación de *MAS*, el primero se basa en la demostración axiomática, que se centra en los métodos automáticos de demostración de teoremas, el segundo es la demostración semántica, con técnicas de demostración de modelos que buscan demostrar la corrección de cierta propiedad, y no tanto la completitud y corrección de *MAS*.

Un ejemplo de la demostración axiomática es el de De Giacomo *et al.* (2000) y su *ConGolog*, donde se aplica de manera intensa la demostración axiomática de teoremas para demostrar que tras la aplicación de una tarea no se alcanza un estado indeseable.

La aplicación de la comprobación de modelos es más extendida, por ejemplo Holzmann (1999) emplea una herramienta Chequeo del Modelo (*model checking*) llamada *SPIN*. Brazier *et al.*

(2004) presentan una variante de demostración de modelos llamada *verificación composicional*, que consiste en una negociación entre agentes, presentando diferentes propiedades que el sistema debe satisfacer.

Una línea en paralelo es la aplicada en *INGENIAS* para validación y verificación de *MAS* (Fuentes *et al.*, 2003; Fuentes *et al.*, 2009), basado en la *Teoría de la Actividad* (Leontiev, 1978). Aplicando dicha teoría a *MAS*, se logra identificar diferentes tipos de contradicciones en una especificación *MAS*. Al margen de la detección de contradicciones, dicha teoría propone herramientas conceptuales para la captura de requisitos e identificar actividades y otros conceptos.

## CAPÍTULO VI. APLICACIÓN DE TÉCNICAS NO LINEALES Y OTROS PARADIGMAS EN SMART GRID/MICROGRID/VPP

En este Capítulo se presentarán las aportaciones que enmarcan el trabajo de la *Tesis Doctoral*. Para comenzar, en la sección 6.1 se presentará un estudio de la relación existente entre las variables climáticas y la demanda eléctrica en entornos desagregados, así como su importancia en aplicaciones de futuro en el marco de la *SG/Smart World*.

Continuando con las aplicaciones de futuro en entornos desagregados, en la sección 6.2 se presentará un sistema de procesamiento de datos para el análisis de los patrones de consumo de energía en un polígono industrial (*microgrid* potencial). El sistema estará basado en una aplicación en cascada de *SOM* junto al algoritmo de clusterizado *k-means*. El sistema ha sido validado con datos de un entorno real, encontrando claras diferencias entre los patrones de consumo existentes en el emplazamiento, además, el sistema es capaz de operar sin supervisión y sin conocimiento previo sobre los datos. La posibilidad de hacer *DR* es entorno de *microgrids* o similares, pasará por disponer de herramientas que ayuden a conocer el comportamiento de la demanda y posible generación bajo su control. Un sistema como el presentado, además de permitir el avanzar en el conocimiento ya comentado, posibilitará realizar predicciones de la demanda más ajustadas, ya que se dispondrá de un selector del clúster correspondiente al que pertenece el día a estimar, y poder emplear modelos de predicción especializados en dichas curvas.

Posteriormente, y continuando con las necesidades de predicción de la demanda en entornos desagregados, en la sección 6.3 se presentará un modelo de arquitectura para *STLF* basado en *ANN*. Para validar el modelo se emplearán datos de un entorno desagregado, que se corresponde con un localización geográfica de consumos moderados, pudiéndose considerar como un entorno de una zona rural, una ciudad de pequeñas dimensiones o una potencial *microgrid*. Se presentan las variables de entrada seleccionadas para el modelo, tratando de minimizar el error obtenido en la fase de operación/validación. Adicionalmente, se presenta un estudio numérico entre el error cometido en las predicciones durante la fase de operación/validación y el número de patrones existentes en la fase de aprendizaje.

En base a los resultados obtenidos con el modelo arquitectural anterior y la experiencia del parque industrial, en la sección 6.4 se planteará una arquitectura basada en tres etapas para hacer *STLF*. La primera etapa consistirá en un reconocimiento de patrones por medio de *SOM*, para en una segunda etapa clusterizar empleando el algoritmo *k-means* los resultados entregados por *SOM*, y finalmente hacer *STLF* para cada uno de los clúster por medio de un modelo *ANN* especializado en las curvas de cada clúster. Al igual que se hizo en la sección 6.3, se mostrará un estudio de la evolución del error para cada uno de los modelos de *STLF* especializado en cada clúster, pudiendo

apreciar la mejora existente en cada uno de ellos. Se finalizará el trabajo con una comparativa con respecto a los resultados obtenidos en la sección 6.3, donde no se clusterizado los patrones de la curva de carga.

Los resultados de la sección de 6.1 se centran en la necesidad de nuevas aplicaciones de futuro para entornos *SG/Smart World*. Se presenta la relación existente entre variables climáticas y la demanda eléctrica en entornos desagregados (*microgrid* potencial), y por tanto, la posibilidad de cruzar información procedente de diferentes redes físicas, para poderla emplear en aplicaciones necesarias, como es el caso de las predicciones de la demanda eléctrica. En base a esto, en la sección 6.5 se presenta un estudio completo de las variables climáticas con respecto a la demanda eléctrica agregada, para plantear un modelo de estimación de la misma basada en *ANN*. Este modelo va evolucionando al ir incorporando nuevas variables al mismo, con la finalidad de ir comparando la mejora que producen en la estimación de la demanda agregada la incorporación de las variables incorporadas.

Como se mostrará en las secciones 6.3 y 6.4, las zonas de la curva de carga donde más error se comete al realizar *STLF*, se corresponden con las inmediaciones de ciertos puntos característicos de la curva de carga. En concreto, estos puntos característicos son los dos picos de carga y los dos valles de carga de la curva. Por tanto, en la sección 6.6 se planteará una arquitectura de modelo completa para hacer *STLF*, basada en estimaciones previas del día a predecir, de los puntos críticos comentados anteriormente. Los resultados serán comparados con los obtenidos en la sección 6.3, al tratarse del mismo entorno desagregado y el mismo conjunto de datos.

Para finalizar, en la sección 6.7 se presenta el diseño de una arquitectura conceptual basada en *MAS*. El objetivo será la gestión de una *SG*, en concreto, para la coordinación y control de la predicción de la demanda eléctrica en una *VPP* de futuro. Una vez analizadas las necesidades de la *VPP*, y presentados los procesos intra-*VPP* e inter-*VPP*, se plantea una arquitectura basada en agentes para el subsistema llamado Predicción de la Demanda (*Demand Forecasting*). Además, se muestra una posible implementación *STLF* para dicho subsistema.

## **6.1. ESTUDIO DE LA RELACIÓN ENTRE LAS VARIABLES CLIMÁTICAS Y LA DEMANDA ELÉCTRICA EN EL MARCO DE SMART GRID/SMART WORLD**

### **6.1.1. Introducción al problema**

Estamos viviendo una época de cambios, marcados principalmente por el intento por parte de las diferentes administraciones de los gobiernos, de ir hacia la consecución de los siguientes objetivos: disminución de gases efecto invernadero, ahorro energético a través de la eficiencia

energética y la promoción clara del uso de fuentes de generación por medio de energías renovables. A lo anterior, debemos unir la cada vez mayor demanda de necesidades energéticas por parte de la sociedad, para poder satisfacer los hábitos presentes y que parece ser, serán los futuros. Desde ya hace algunos años, se están potenciando nuevos entornos que tratarán de cambiar los actuales modelos clásicos encargados de abastecer las necesidades energéticas, teniendo todos ellos en común, el despliegue de dispositivos de inteligencia y sensores en sus capas físicas, orientados entre otras cosas, a la consecución de los objetivos arriba indicados.

Se nos presenta un escenario donde por un lado, existen unas necesidades energéticas por parte de la sociedad y que además serán cambiantes con los años, información desplegada en diferentes capas a través de los nuevos sensores e inteligencia distribuida, y la necesidad de una creciente conciencia medio ambiental y social que requerirá tener conocimiento de la relación entre las necesidades energéticas y ciertos parámetros del clima, y que de una manera u otra servirán para la toma de decisiones en el sistema donde se aplique.

No podemos obviar, que el cambio climático y todo lo que rodea al comportamiento de la sociedad frente al medio ambiente, está incrementando de manera más que notable la preocupación popular y política. El crecimiento de la población y el frenético ritmo de vida de la sociedad, junto con nuevos estándares de confort que hemos impuesto, incrementarán el consumo de energía eléctrica, gas, agua, transporte, construcción de edificios, etc.; lo cual propiciará el aumento en las emisiones de gases de efecto invernadero, dióxido de carbono junto a otras perturbaciones, produciendo cambios sustanciales en parámetros fundamentales para la vida. Debe ser tenido en cuenta el impacto futuro del cambio climático en la demanda de electricidad, con variaciones notables en el clima local. Para tratar este problema, Mirasgedis *et al.* (2007) utilizan un modelo climático regional, *PRECIS*, para predecir las futuras condiciones climáticas en diferentes escenarios de emisiones, mientras que Pili-Sihvola *et al.* (2010) examinan el impacto del calentamiento y cambio del clima, con la necesidad de calefacción y refrigeración, utilizando un modelo de regresión multivariada para cinco países de Europa a lo largo de una línea de norte a sur.

Messaoud *et al.* (2011) analizan la forma en que el cambio climático afecta en la actualidad al crecimiento de diferentes especies arbóreas en la zona de British Columbia, viendo como claramente las variables climáticas modificadas debido al cambio climático influyen en el crecimiento de algunos vegetales. Parkpoom and Harrison (2008) presentan como el cambio climático afectará sobre la demanda eléctrica a largo plazo en Tailandia, empleando para el estudio modelos de regresión para obtener la relación entre patrones de comportamiento eléctrico y la temperatura. Entre los muchos factores que afectan a la demanda de energía, numerosos estudios han puesto de manifiesto la influencia de las variables meteorológicas en el consumo de electricidad. Considine (2000) analiza el impacto de las variaciones del clima en las emisiones de

carbón y necesidad de energía. Hyndman and Fan (2010) presentan un modelo aditivo semi-paramétrico para estimar la relación entre la demanda eléctrica, temperatura, efectos del calendario y variables demográficas y sociales, para la predicción de la densidad de demanda de pico de la electricidad a largo plazo. La respuesta entre la demanda eléctrica y la temperatura no es lineal, en (Henley and Peirson, 1997) se investigan estas no linealidades empíricamente, usando tanto métodos paramétricos como no paramétricos para la identificación de no linealidad. La reacción de la demanda está causada por las diferencias entre la temperatura exterior y la temperatura interior, cuando a partir de un umbral de discrepancia se requiere la puesta en marcha de la calefacción o el correspondiente equipo de refrigeración, los resultados de los análisis de (Teräsvirta and Anderson, 1992) sugieren modelos de auto-correlación de transición suavizada para la evolución desde niveles fríos hasta umbrales calor. Hay que tener en cuenta que el confort es un parámetro subjetivo, ya que depende de las preferencias y nivel económico del usuario, así como de los precios de la energía. Los factores geográficos son determinantes en las predicciones meteorológicas, Psiloglou *et al.* (2009) realizan un estudio comparativo de consumos eléctricos en Atenas y Londres.

Arriba se han presentado trabajos que demostraban como las condiciones climáticas y medio ambientales afectaban a ciertos comportamientos de la vida de algunos seres vivos, así como la clara relación entre variables climáticas y ciertos comportamientos de consumo humano, como son el caso de la electricidad y del gas natural. Es importante tratar de predecir la demanda eléctrica con un error lo menor posible, sin embargo, el consumo de energía eléctrica depende de muchos factores como el día de la semana, mes del año, etc.; esto hace que la predicción sea un problema muy complejo que requiere de métodos que se alejan de los puramente estadísticos.

La conclusión es que resulta preciso prestar atención a los indicadores medio ambientales y las variables climáticas, unas veces porque representan una fuente de información para aplicaciones concretas, como es el caso de la demanda eléctrica, y otras porque afectan de manera directa sobre el devenir de ciertas especies vegetales y animales, o incluso del propio ser humano. Asumiendo además que nos encontramos inmersos en la era digital, parece más que evidente que se precisa emplear los avances de las *TICs* y los sensores existentes, para lograr lo anteriormente comentado y hacer frente a otras necesidades relacionadas con la operación y control directo de los sistemas donde todo es desplegado e implantado.

En (Abe *et al.*, 2011) se presenta a las *SG* como facilitadoras de la cada vez mayor integración de renovables en red y su operación, así como el almacenamiento y su impacto en la regularización de los mercados nacionales e internacionales. En (Zaballos *et al.*, 2011; Amin and Giacomoni, 2012) se define la arquitectura de red de *SG* basada en sensores ubicuos para posibilitar, entre otras cosas, la interoperabilidad, y evidenciando como fundamental una operación fiable y segura del sistema eléctrico. Destacar en (Barlett *et al.*, 2011; Yuan *et al.*, 2011) como se



plantea a partir de los datos obtenidos de los sensores en las *SC*, la creación de servicios bajo demanda y la monitorización en tiempo real a partir del citado despliegue. En (Harrison *et al.*, 2010; Lee *et al.*, 2011) se muestra la forma en que se potencia la gestión y la cooperación de los sensores heterogéneos de las *SC* para la monitorización de espacios públicos, y para asegurar el negocio y el funcionamiento de aplicaciones por parte de las *utilities*. Gann *et al.* (2011) consideran que las ciudades y espacios se convertirán en inteligentes cuando hagan uso de la disponibilidad de los datos y técnicas analíticas para mejorar la eficacia y la eficiencia de sus funcionamientos, pudiendo contribuir a este objetivo la integración de la infraestructura física y digital. Naphade *et al.* (2011) exponen que el éxito de las *SC* y la proliferación de los sensores y dispositivos, pasa por lograr y demostrar la interoperabilidad del sistema, garantizando la seguridad y la privacidad.

Como se ha indicado, cada vez existe una mayor conciencia social y política en torno a temas relacionados con el cuidado del medio ambiente, y con los compromisos mundiales adoptados de reducción de emisiones de contaminantes y de consumo de energía. Igualmente, se ha presentado la influencia de ciertas variables climáticas con respecto a determinados comportamientos de consumo energético, y como algunos de éstos pueden contribuir en mayor o menor medida al cambio climático y sus indeseables consecuencias. Lo que parece evidente, es la posibilidad de aprovechar el despliegue de sensores y de inteligencia distribuida anteriormente comentado, actualmente para la operación y funcionamiento del sistema donde se instalan, y así poder cruzar información proveniente de diferentes modelos para emplearla en nuevas aplicaciones que tengan dicha necesidad de información, o para el seguimiento y control de parámetros climáticos y medio ambientales que evalúen la manera en que los hábitos humanos y los cambios en los modelos influyen en el ecosistema o viceversa.

La aportación presentada en esta sección es doble:

- Por un lado se define concepto *Smart World*, ya introducido en el Capítulo II, como un marco global para la colaboración entre sistemas “*Smart*”.
- Estudio experimental entre un determinado número de variables climáticas y los patrones de demanda eléctrica, presentando además un sencillo caso de estudio basado en *MAS*, y enfatizando la necesidad de considerar su relación con *Smart World* (especialmente en *SG* y *microgrid*).

### **6.1.2. Sinergias entre modelos: Un nuevo concepto**

Como ya se explicó en la sección 2.6 del Capítulo II, tanto *SG*, *microgrids*, etc., tienen en común que son espacios con infraestructuras físicas desplegadas junto a elementos *TIC*. Una visión de futuro de estos espacios, independientemente de sus funciones y escenario físico, son lugares con sensores y *TIC* proporcionando inteligencia a las infraestructuras, y además, varias infraestructuras

inteligentes colaborando mutuamente para proporcionar mejor calidad de vida de los habitantes y usuarios. En la Figura 2.8 se presentó *Smart Place*, donde la infraestructura de energía eléctrica, se podría corresponder además con *SG* o una *microgrid*, junto a otras infraestructuras para cooperar conjuntamente.

En el Capítulo II también se definió un nuevo concepto para tratar de englobar al conjunto de las “*Smart*”. Este nuevo concepto recibirá el nombre de *Smart World*, y sus aplicaciones o servicios resultantes, independientemente del origen de su necesidad, cumplirán algunos o todos los objetivos siguientes: operación, mantenimiento, optimización y automatización de infraestructuras; Safety and security; Movilidad y transporte; urbanismo; ahorro energético y eficiencia energética; sostenibilidad; evolución y control medio ambiental; y calidad de vida.

En la Figura 2.8 se mostró este nuevo concepto *Smart World*, donde apreciamos como la información fluye desde los sensores e inteligencia distribuida desplegada en cualquier infraestructura del planeta, bien sea en las terrestres ya desplegadas o futuras, en las oceánicas o en las proporcionadas a través de los globos, aviones o satélites que monitorizan La Tierra. Vemos como el concepto *Smart World* se nutre de información proveniente de: la superficie terrestre; ríos, lagos, mares y océanos; observación de La Tierra desde la atmósfera o fuera de ella. A partir de la información de *Smart World*, se plantea la creación de servicios, aplicaciones, nuevos protocolos de inter-comunicación, tecnología y transferencia de información que puede ser explotada por los propietarios de la información o terceros, para comunidades interesadas.

### **6.1.3. Análisis de datos, procesamiento de información y metodología**

En esta sección se presentarán los datos que serán empleados para poder hacer el estudio entre las variables climáticas y la demanda eléctrica. En la sección 6.1.4 se presentarán los resultados del estudio. Los datos meteorológicos son recolectados de una estación climática existente en una red diferente, de manera que aplicaciones nuevas disponibles en *Smart World* emplearán datos de diferentes infraestructuras.

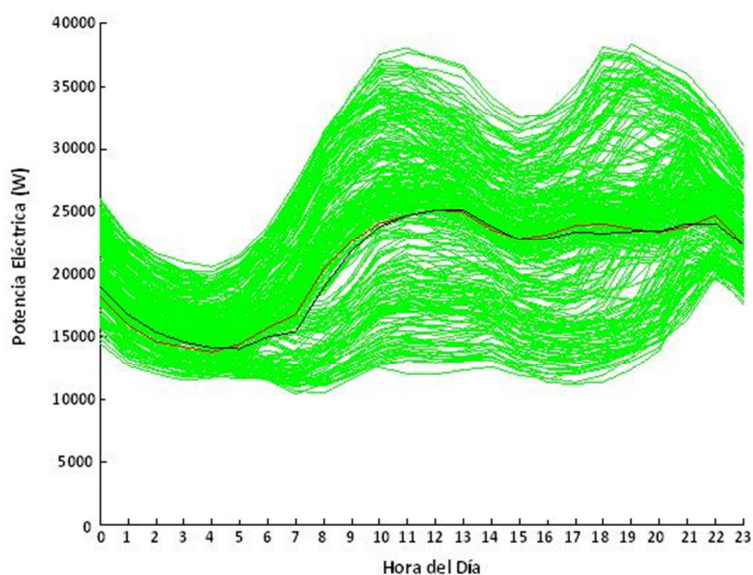
#### **6.1.3.1. Datos de consumo eléctrico y variables climáticas**

La localización del emplazamiento sujeto a estudio se sitúa en Soria, una capital de provincia de España, con 40.000 habitantes y 271 km<sup>2</sup> aproximadamente. Dispondremos por lo tanto de unos datos de comportamiento de consumo eléctrico y ciertas variables climáticas de una localización de dimensiones reducidas que hace pensar que puede existir cierta correlación, como más adelante demostraremos que así sucede. Además, el rango de consumos de Soria varía entre los 7–39 MW, por lo que los datos a experimentar no se corresponden con una gran ciudad, o una zona extensa ni tan siquiera un país, se aproximan más al consumo de lo que podría ser una *microgrid*, nuevo

escenario donde tanto el control de la posible demanda y generación se hace imprescindible para su correcto funcionamiento, de la misma forma que se hace necesaria la supervisión de variables climáticas que influyen en el comportamiento de aquellas y viceversa, monitorizando la forma en que los hábitos de demanda y generación pueden influir en ciertos índices medio ambientales en la *microgrid*.

La compañía española distribuidora de energía eléctrica Iberdrola ha facilitado el conjunto de datos para poder hacer el presente estudio. En concreto se cuenta con un histórico de datos desde el 1 de enero de 2008 hasta 31 de diciembre de 2010. Los datos facilitados por la compañía para ese rango de fechas son: día del mes, mes, año y los 24 consumos eléctricos que conforman la curva de carga de cada día. Al considerarse la curva de carga horaria como un elemento de vital importancia para las compañías distribuidoras o los gestores de las *microgrids*, se establece la unidad hora como el indicador a tener en cuenta en esta experiencia.

A modo de ejemplo, en la Figura 6.1 podemos ver las curvas de carga de los miércoles no festivos (en verde). Se muestra una variedad grande de curvas de carga diferentes, tanto en tamaño como e forma.



**Figura 6.1.** Curvas de carga para los miércoles no festivos. Los miércoles son representados en verde. La curva de carga del día 30/06/2010 es representada en rojo, y la predicción para ese mismo día se representa en negro.

**Fuente Propia.**

Las variables climáticas han sido proporcionadas por *AEMET*, a través de los datos registrados en su estación meteorológica situada en Soria. La radiación solar global horaria ha sido proporcionada en un fichero .csv, mientras que el resto de variables climáticas ha sido proporcionada en otro fichero pero en periodos diezminutales, ambos en un periodo desde el 1 de enero de 2008 hasta 31 de diciembre de 2010.

Cada fila del fichero de radiación solar global horaria está formada por la siguiente información: indicativo de la estación meteorológica, año del dato registrado, mes del dato registrado, día del dato registrado, nombre de la provincia, longitud, latitud y  $rad\_global_i$ ; con  $i = 5, \dots, 20$  las horas del día desde las 5:00 AM hasta las 20:00 PM.

Las variables climáticas que se encuentran en el fichero diezminutal son: precipitación (mm), temperatura del aire (°C), velocidad media del viento (m/seg), dirección media del viento (grados sexagesimales), humedad relativa (%) y presión (hPa). Cada fila del fichero está formada por la siguiente información: año del dato registrado, mes del dato registrado, día del dato registrado, hora del dato registrado, minuto del dato registrado, nombre de la estación meteorológica, altitud, nombre de la provincia, longitud, latitud, precipitación, temperatura ambiente, velocidad media del viento, dirección media del viento, humedad relativa y presión.

### 6.1.3.2. Procesamiento de información

En la sección 6.1.3.1 se han presentado los datos sin tratar disponibles, pero como se verá a continuación, se hace necesario el tratarlos como paso previo a la experiencia a realizar. En la Figura 6.2 se presenta el esquema entero del proceso para posteriormente explicarlo.

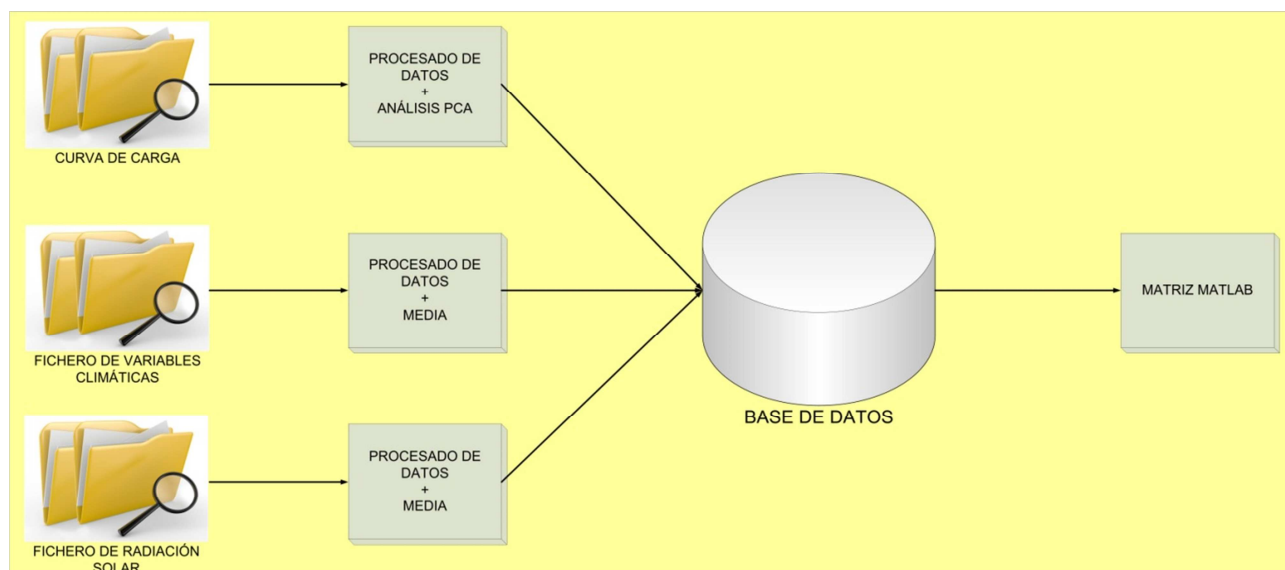


Figura 6.2. Esquema de procesamiento de la información. Fuente Propia.

A continuación indicaremos lo que ha habido que hacer en los 4 procesos indicados en la Figura 6.2. Los tres primeros procesos son acciones realizadas sobre los datos sin tratar (los recibidos) y como paso previo a su inserción en la base de datos. El cuarto proceso es la preparación de los mismos para su posterior envío a MatLab para realizar la experiencia.

El fichero con las curva de carga tienen registrado el valor de consumo eléctrico por horas, por lo que en principio la información viene adecuada a nuestras necesidades, ya que como se explicó, es una medida más que importante para las empresas distribuidoras de energía o gestoras

de las *microgrids*. Lo que si se hace necesario en el proceso 1, es detectar si en las curvas de carga falta algún dato, en cuyo caso se ha optado por mediar según la siguiente fórmula

$$L_{(d,i)} = \frac{|L_{(d,i-1)} - L_{(d,i+1)}|}{2}; i = 1, 2, \dots, 21, 22$$

$$L_{(d,23)} = \frac{|L_{(d,22)} - L_{(d+1,0)}|}{2}; i = 23$$

$$L_{(d,0)} = \frac{|L_{(d-1,23)} - L_{(d,1)}|}{2}; i = 0$$
(6.1)

donde  $i$  representa la hora del día comenzando en 0 y terminando en 23 y  $d$  se corresponde con el día de la curva de carga que se está tratando.

Otra labor del proceso 1 (a realizar tras aplicar (6.1)), es detectar curvas de carga erróneas. Para la detección de estos outliers se ha empleado *PCA*, que es un procedimiento matemático que emplea una transformación ortogonal para convertir un conjunto de observaciones de posibles variables correlacionadas dentro de un conjunto de variables no correlacionadas, llamadas componentes principales.

En la Figura 6.3 se muestran los resultados de *PCA*, viendo en a) los patrones (se introducen todos los datos en forma de curva de carga) que podrían considerarse como outliers tras la comparación de la pareja de dimensiones 8 y 9, mientras que en b) se muestran dos curvas de carga outliers (665 y 897) frente a una curva buena de un día especial.

En la Tabla C.1 del Anexo C se muestra las curvas de carga detectadas como outliers técnicos, las cuales serán marcadas con un indicador en base de datos, con la finalidad de no tenerlos en cuenta para el proceso 4, y así no enrarecer los resultados finales, al tratarse curvas de carga con fallos técnicos y por lo tanto, erróneas. Este análisis será empleado en las secciones 6.2, 6.3, 6.4, 6.5 y 6.6; por lo que no se volverá a explicar, ya que los patrones retirados serán los mismos.

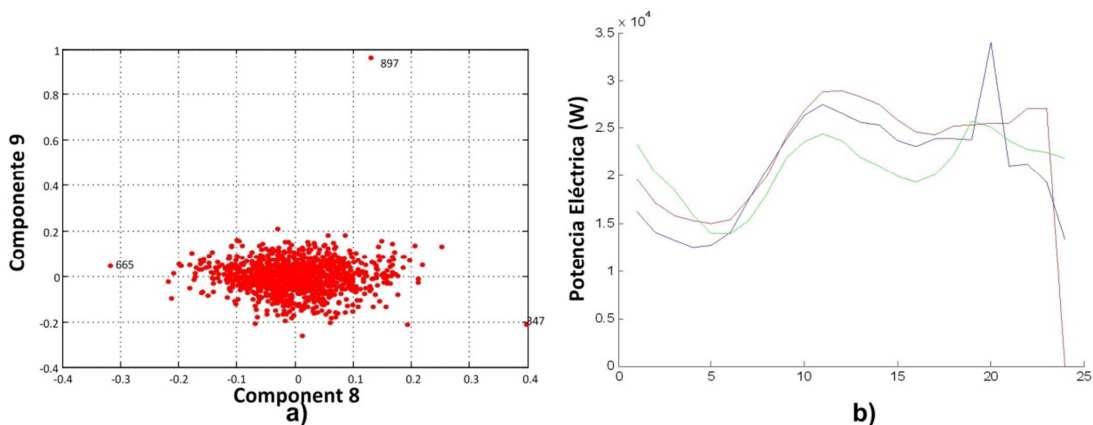


Figura 6.3. a) *PCA*: componentes 8 y 9; b) Azul: 665, rojo: 897 y verde: 347. El número indica el número de patrón en el conjunto de datos. Fuente Propia.

Las variables climáticas, exceptuando la radiación solar global, se encuentran en un fichero con valores diezminutales, y como ya se ha comentado el interés se centra en valores horarios. Lo primero que se hará en el proceso 2 será detectar si en cada fila del fichero falta algún dato correspondiente a algún valor diezminutal, en cuyo caso habrá que aplicar una fórmula de corrección análoga a (6.1) para cada una de las seis variables climáticas que lo forman, pero esta vez teniendo en cuenta que el dato anterior y posterior se corresponde con el valor diezminutal en lugar de horario. Tras la corrección, emplearemos un algoritmo que obtenga el valor de la variable pero horaria, empleando la fórmula

$$\text{var}_{i,d} = \sum_{j=1}^6 \frac{\text{var}_{j,d}}{6} \quad (6.2)$$

siendo *var* cada una de las variables: temperatura del aire, velocidad media del viento, dirección media del viento, humedad relativa y presión;  $i = 0, 1, \dots, 22, 23$  las horas del día  $d$  y  $j = 1, 2, \dots, 5, 6$  cada uno de los seis tramos diezminutales (0–10, 10–20, 20–30, 30–40, 40–50 y 50–60). Y empleando la fórmula

$$\text{prec}_{i,d} = \sum_{j=1}^6 \text{prec}_{j,d} \quad (6.3)$$

para la variable precipitación, que es la única que no se obtiene como media sino como sumatorio de los seis tramos diezminutales, siendo el resto de subíndices los mismos que para (6.2). Una vez obtenidos a través de (6.2) y (6.3) todos los valores horarios de las seis variables, se obtiene también su valor medio de cada día con la fórmula

$$\text{var\_mean}_d = \sum_{i=0}^{23} \frac{\text{var}_{i,d}}{24} \quad (6.4)$$

*var\_mean* representa a cada una de las seis variables una vez mediada,  $d$  el día tratado e  $i = 0, 1, \dots, 22, 23$  los veinticuatro valores de las horas del día para la variable en concreto representada por *var*.

Ya que el fichero de la radiación solar global viene con datos de cada hora, desde las 5:00 AM hasta las 20:00 PM, el proceso 3 sólo deberá detectar si en cada fila del fichero falta algún dato correspondiente a algún valor horario, en cuyo caso habrá que aplicar una fórmula de corrección análoga a (6.1) pero con 16 tramos horarios en lugar de 24. Al tener los valores horarios, tan sólo resta obtener el valor medio de cada día a través de la fórmula

$$\text{rad\_mean}_d = \sum_{i=5:00AM}^{20:00PM} \frac{\text{rad}_{i,d}}{16} \quad (6.5)$$

donde  $rad\_mean$  representa la radiación solar global una vez mediada,  $d$  el día tratado e  $i = 5:00 a.m., \dots, 8:00 p.m.$  los quince valores de las horas del día para la variable radiación solar global representada por  $var$ .

El fichero de radiación solar global y el que engloba el resto de variables climáticas han resultado sin apenas errores, no habiéndose detectado outliers técnicos, por este motivo no se han añadido análisis *PCA* ni al proceso 2 ni al 3, pero teniendo claro que en futuros análisis que lo requieran, será tan sencillo como añadirlo en el código.

Disponemos ya en base de datos las curvas de carga con los outliers técnicos identificados, y la totalidad de variables climáticas con datos horarios y la media de las mismas para cada uno de los patrones que componen cada uno de los días disponibles. El proceso 4 tiene como objetivo elaborar una matriz de datos, para posteriormente realizar el análisis de correlaciones por medio de MatLab. Lo primero que debe hacer dicho proceso será recorrer todos los patrones (días completos) y excluir los siguientes:

- Patrones cuyas curvas de carga estén marcadas como outliers técnicos, aunque tengan datos de variables climáticas.
- Patrones donde falten alguna de las variables climáticas. Es cierto que faltaba algún día entero de variables climáticas, seguramente por motivos de suministro eléctrico en la estación meteorológica de *AEMET*.

Una vez finalizada la tarea anterior, el proceso 4 elabora la matriz de datos donde en cada una de las filas estarán los patrones seleccionados, que constarán en las columnas todos los valores horarios y media del día de la curva de carga así como de todas las variables climáticas. En la Tabla 6.1 se puede ver la disposición de las columnas de la matriz formada.

**Tabla 6.1. Columnas de la matriz de entrada a MatLab. Fuente Propia.**

COLUMNAS	VALORES DE LA CURVA DE CARGA Y CONSUMO MEDIO DEL DÍA
Columnas: 1-25	24 valores de consumo y consumo medio del día
Columnas: 26-50	24 valores de precipitación y precipitación media del día
Columnas: 51-75	24 valores de temperatura y temperatura media del día
Columnas: 76-100	24 valores de velocidad de viento y velocidad de viento media del día
Columnas: 101-125	24 valores de dirección de viento y dirección de viento media del día
Columnas: 126-150	24 valores de humedad relativa y humedad relativa media del día
Columnas: 151-175	24 valores de presión y presión media del día
Columnas: 176-192	16 valores de radiación solar global y radiación solar global media del día

### 6.1.3.3. Metodología

En estadística y probabilidad, correlación es la fuerza y la dirección de una relación lineal entre dos variables aleatorias. Dos variables cuantitativas están correlacionadas, si los valores de

una de ellas varían sistemáticamente con respecto a los valores homónimos de la otra, por ejemplo, si tenemos dos variables  $X$  e  $Y$ , existe correlación si al aumentar los valores de  $X$  lo hacen también los de  $Y$ , y viceversa.

Existen diversos coeficientes que miden el grado de correlación, el más conocido es el coeficiente de correlación de Pearson y será el empleado en este estudio, que se obtiene dividiendo la covarianza de dos variables por el producto de sus desviaciones estándar, otros coeficientes son: coeficiente de correlación de Spearman, correlación canónica y coeficiente de correlación intra-clase. El coeficiente de correlación de Pearson, es un índice que mide la relación lineal entre dos variables aleatorias cuantitativas. A diferencia de la covarianza, la correlación de Pearson es independiente de la escala de medida de las variables. De manera menos formal, podemos definir el coeficiente de correlación de Pearson como un índice que puede utilizarse para medir el grado de relación de dos variables siempre y cuando ambas sean cuantitativas. El valor del índice de correlación varía en el intervalo  $[-1, +1]$ :

- Si  $r = +1$ , hay correlación positiva perfecta. El índice indica una dependencia total entre las dos variables denominada relación directa, cuando una de ellas aumenta, la otra también lo hace en proporción constante.
- Si  $0 < |r| < 0.009$  representa no correlación.
- Si  $0.1 < |r| < 0.25$  representa una correlación lineal pequeña.
- Si  $0.26 < |r| < 0.40$  representa una correlación lineal media.
- Si  $0.41 < |r| < 1$  representa una correlación lineal fuerte.
- Si  $r = 0$  indica que ambas variables no están relacionadas linealmente.
- Si  $r = -1$ , hay una correlación negativa perfecta. El índice indica una dependencia total entre las dos variables llamada relación inversa, cuando una de ellas aumenta, la otra disminuye en proporción constante.

La matriz obtenida tras el proceso 4 sirve para comenzar el estudio, el cual no es otro que tratar de determinar la existencia de una relación directa entre la demanda eléctrica y algunas variables climáticas. Para poder realizar este estudio haremos uso de una función potente existente en MatLab llamada *corrcoef()* (coeficientes de correlación; ver Anexo C) que nos devolverá una matriz  $192 \times 192$ , donde se indica el grado de correlación entre las columnas de la matriz de la Tabla 6.1. Los elementos de la diagonal son los coeficientes de auto-correlación de las variables, el elemento de la fila 1 y columna 2 es la correlación entre la variable 1 y la variable 2 (coincidente con el elemento de la fila 2 y columna 1, al ser la matriz simétrica), y así sucesivamente.



### 6.1.4. Resultados

Una vez realizado con MatLab la experiencia a través de *corrcoef()*, se obtiene la matriz solución, que deberá ser analizada para obtener las conclusiones que vamos a presentar. El resultado de esta matriz se lleva a una hoja de cálculo para su posterior análisis, a modo de resumen se presenta en la Tabla 6.2, las correlaciones medias con respecto a las 24 horas de consumo de las diferentes variables.

**Tabla 6.2. Correlación entre variables climáticas y el consumo de energía eléctrica. Fuente Propia.**

VARIABLE CLIMÁTICA MEDIA	CORRELACIÓN MEDIA CON LAS 24 HORAS DE CONSUMO
Precipitación Media	0.075
Temperatura Media	-0.458
Velocidad media del viento Media	0.199
Dirección media del viento Media	0.186
Humedad relativa Media	0.308
Presión Media	0.010
Radiación solar Media	-0.393

A partir de la tabla anterior, podemos observar como en media, las variables que más relación tienen sobre el consumo eléctrico son la temperatura, radiación solar global y la humedad relativa, presentando las dos primeras una correlación negativa y la última correlación positiva. La velocidad media del viento y dirección media del viento apenas tiene correlación con el consumo eléctrico, no correlacionando en nada ni la precipitación ni la presión.

A continuación, y de modo visual, se van a representar las correlaciones de todas las variables climáticas frente al consumo eléctrico, seleccionando la hora 12 como momento del día a mostrar. En la Figura 6.4 se muestran las variables climáticas sin correlación, que como se comentó son la precipitación y presión, donde apreciamos una total dispersión de los puntos frente a la recta de ajuste. En la Figura 6.5 las variables climáticas con ligera correlación, que son la velocidad media del viento y dirección media del viento, donde existe mayor agrupación alrededor de la recta con respecto a las dos anteriores variables climáticas, siendo un poco más evidente esta agrupación con la velocidad media de viento, pero manifestando ambas una dispersión considerable. En la Figura 6.6 las variables climáticas con una correlación mayor y más clara con el consumo eléctrico, siendo éstas la temperatura, radiación solar global y humedad relativa, donde se aprecia la clara correlación negativa de las dos primeras y la positiva de la tercera, no habiendo casi dispersión si se compara con el resto de variables de las anteriores figuras.

Como ha quedado demostrado, son la temperatura, radiación solar media y humedad relativa las variables con una correlación más que evidente sobre el consumo eléctrico. A continuación, para

cada una de estas tres variables en cuestión, se van a representar de nuevo gráficamente su correlación frente al consumo eléctrico, pero en momentos diferentes del día, a las 0h, 5h, 12h, 16h y 2h, siendo el primer instante el cambio de día, y el resto de cuatro momentos seleccionados los que más se aproximan a los cuatro puntos característicos de la curva de carga (dos picos y dos valles), y así poder apreciar el cambio de la relación a lo largo del día.

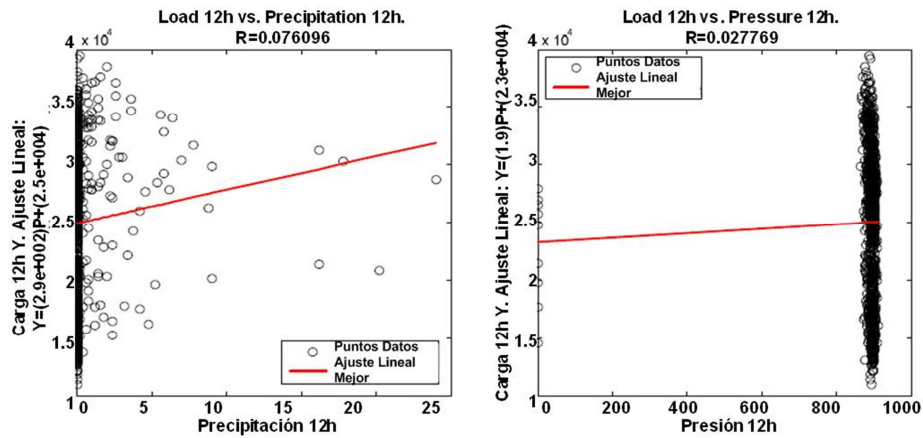


Figura 6.4. Correlación a las 12h entre precipitación y presión, con el consumo eléctrico. Fuente Propia.

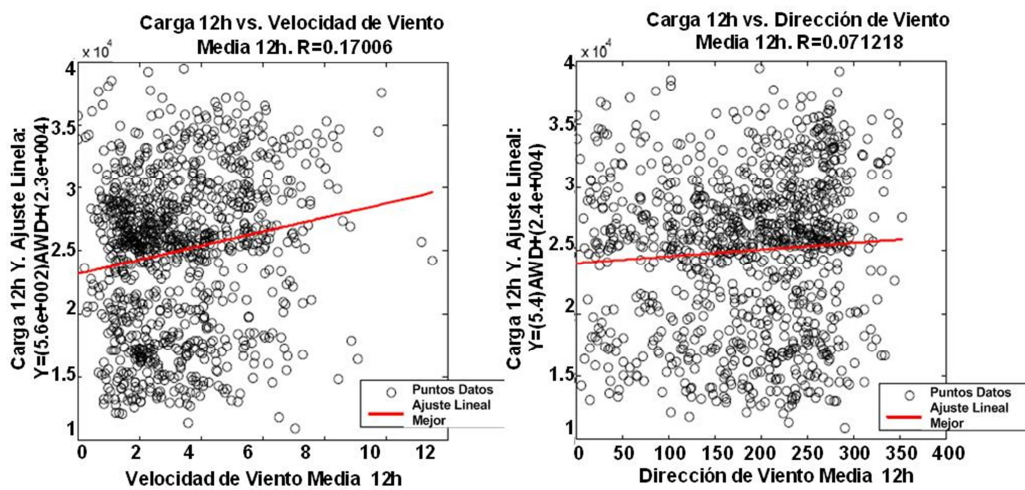


Figura 6.5. Correlación a las 12h entre velocidad media de viento y dirección media de viento, con el consumo eléctrico. Fuente Propia.

En la Figura 6.7 se muestran los cinco instantes de tiempo antes mencionados y la media total de la correlación de la temperatura frente al consumo eléctrico, siendo las 12h y las 16h los momentos donde la correlación baja un poco, debido principalmente a que durante estos instantes del día, los hábitos humanos hacen que el consumo aumente de manera independiente de la situación de la temperatura, pensemos por ejemplo en los hogares domésticos, habitualmente en esos momentos es cuando más actividad en el hogar se produce, principalmente con el incremento del uso de las cocinas eléctricas.

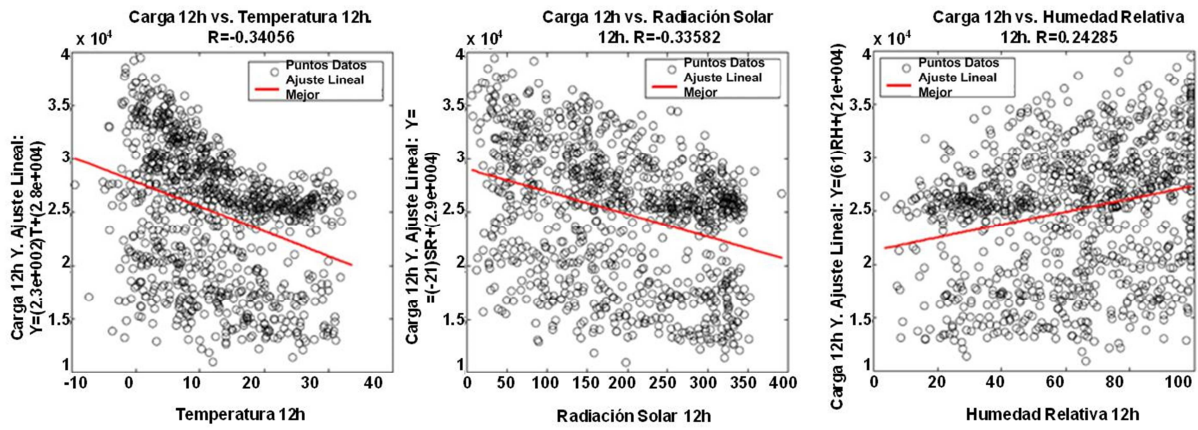


Figura 6.6. Correlación a las 12h de temperatura, radiación solar global y humedad relativa, con el consumo eléctrico. Fuente Propia.

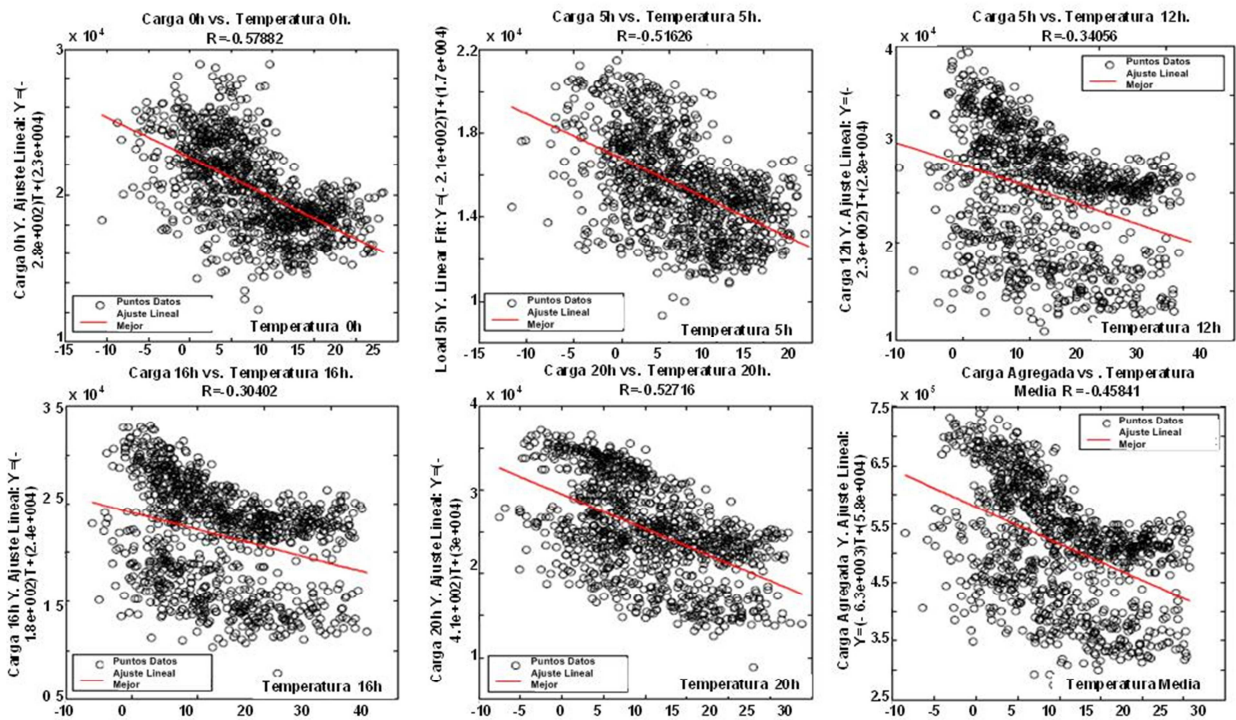


Figura 6.7. Correlación a las 0h, 5h, 12h, 16h y 20h entre temperatura y consumo eléctrico. Fuente Propia.

En la Figura 6.8 se muestran los cinco instantes de tiempo antes mencionados, excepto el cambio de día (0h) ya que no tiene sentido con esta variable climática, y la media total de la correlación de la radiación solar global frente al consumo eléctrico, pudiendo ver como los momentos centrales del día, que coincide con momentos de máxima radiación, la correlación es más evidente que en los instantes de las 5h y 20h, cuando la radiación comienza y decae respectivamente.

En la Figura 6.9 se muestran los cinco instantes de tiempo antes mencionados y la media total de la correlación de la humedad relativa frente al consumo eléctrico, pudiendo ver como en los instantes 0h, 5h y 20h, la mayoría de los patrones se corresponden con medidas de humedad relativa

mayores del 60%, mientras que en el resto aparecen mucho más dispersos. El instante de las 5h es el de peor correlación, seguramente debido a que los grandes procesos industriales comienzan a requerir energía eléctrica y el consumo eléctrico se ve influenciado por dichos hábitos en mayor medida que la humedad relativa.

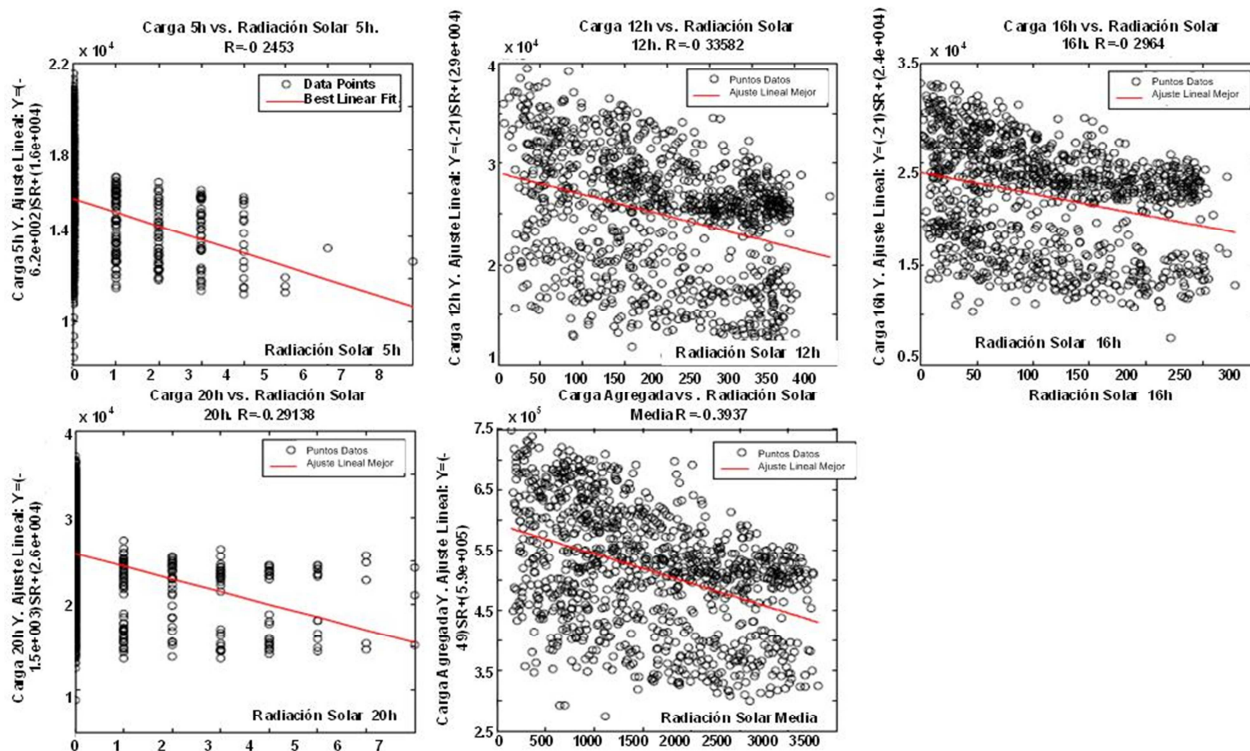


Figura 6.8. Correlación a las 0h, 5h, 12h, 16h y 20h entre radiación solar global y consumo eléctrico. Fuente Propia.

En los resultados anteriores se ha mostrado la clara relación de ciertas variables climáticas con la demanda eléctrica. La relación muestra una clara dependencia con la estacionalidad, pero el objetivo presentado en este trabajo es estudiar el comportamiento agregado de la demanda eléctrica con la demanda a lo largo del tiempo. No obstante, en la Tabla 6.3 se muestran los resultados de los coeficientes de correlación de la demanda con respecto a las principales variables, agrupado por estaciones.

Es fácil comprobar la influencia estacional en los coeficientes de correlación. Por ejemplo, la correlación para la temperatura media es alta en primavera y otoño, pero no en verano e invierno. Este fenómeno puede ser explicado porque en invierno y verano, la temperatura es siempre percibida por lo humanos como “fría” o “caliente” respectivamente, y como tal, el comportamiento relativo a la temperatura es normalmente fijo, mostrando una mayor dependencia con otros parámetros.

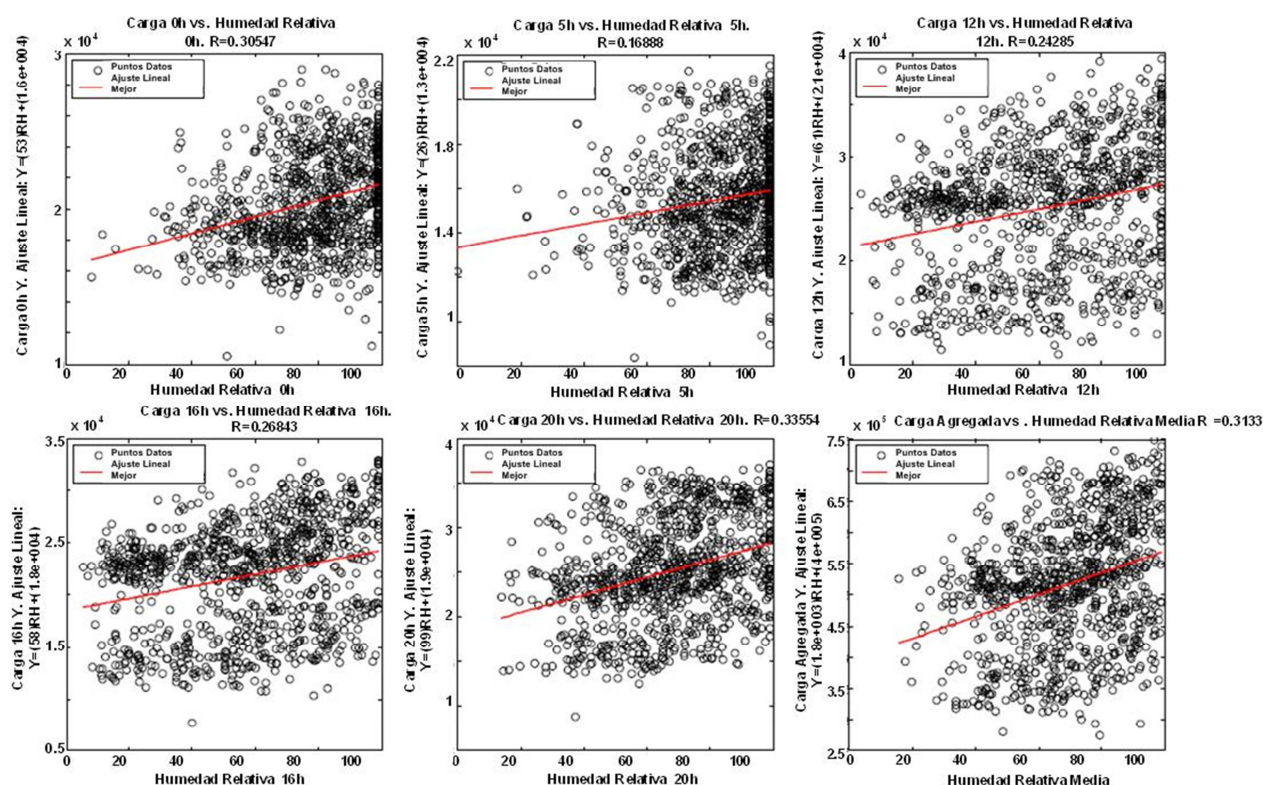


Figura 6.9. Correlación a las 0h, 5h, 12h, 16h y 20h entre humedad relativa y consumo eléctrico. Fuente Propia.

Tabla 6.3. Coeficientes de correlación estacionales entre las variables climáticas y demanda eléctrica. Fuente Propia.

	PRIMAVERA	VERANO	OTOÑO	INVIERNO
Temperatura Media	-0.326	-0.077	-0.396	0.065
Humedad relativa Media	0.228	0.053	0.418	0.027
Precipitación agregada	0.085	0.131	0.235	0.180
Presión Media	-0.138	-0.023	-0.022	-0.303

### 6.1.5. Aplicaciones futuras de variables climáticas

Como el concepto *Smart World* se extiende a todo el mundo, nuevos requerimientos sociales, políticos y de investigación surgirán y, con ellos, nuevas aplicaciones con interacción de modelos diferentes aparecerán. A continuación se presentarán algunos entornos (*SC* o *microgrid*) y sus futuras aplicaciones que requerirán la medida de las variables climáticas. Se finalizará la sección presentando un caso de estudio donde la relación entre climatología y demanda eléctrica es examinada mediante *MAS*.

#### 6.1.5.1. Algunas aplicaciones que requieren la medida de variables climáticas

Demostrada la relación existente entre ciertas variables climáticas y la demanda eléctrica en entornos particulares (entornos desagregados, *microgrid*). Con respecto a la demanda eléctrica, la correlación entre variables de diferentes modelos debe de ser entendida, como es el caso de las

variables climáticas requeridas en los modelos de predicción de la demanda basados en ANN. Adicionalmente, el empleo de variables climáticas como entrada de los modelos de predicción de la demanda, puede resultar interesante para evaluar la manera en que la relación entre las variables climáticas y la demanda eléctrica puede variar con el tiempo; esto podría ayudar a la identificación de nuevas variables climáticas requeridas para los modelos de predicción, o al menos la forma en que afectan a sus pesos en el modelo. Por tanto, los usuarios de por ejemplo una *microgrid*, podrían estar informados de los cambios en la relación entre las variables climáticas y la demanda para que entiendan la forma en que las condiciones climáticas influyen en su comportamiento energético.

Pensemos ahora en una SC, espacios medio ambientalmente saludables y sostenibles, con un alto grado de penetración de fuentes de generación renovable distribuida, y de manera habitual y necesaria. Estas nuevas ciudades van a contar a partir de ahora con algo hasta la fecha no disponible y de gran interés social, información en base a históricos y en tiempo cuasi real, que servirá para ver por ejemplo, la evolución de la SC medio ambientalmente, a través de ratios de control, como por ejemplo, por medio la evolución de la demanda eléctrica de la ciudad frente a las fuentes de generación renovable distribuidas, y la energía que con éstas se deja de solicitar a fuentes clásicas; o por ejemplo, ratios que tengan en cuenta el impacto sobre el grado de contaminación de la ciudad y la demanda eléctrica si no se hubiera generado con renovables disponibles en la SC. La información servirá también para la planificación y ordenación de espacios urbanos, y por tanto, de uso para SC o *microgrids*.

#### **6.1.5.2. Una propuesta de implementación**

Un sistema capaz de darse cuenta de las dependencias anteriormente comprobadas (relación variables climáticas y demanda eléctrica) no es sencillo de realizar. Existen factores desconocidos, como por ejemplo la forma de interactuar con todos los elementos, o por ejemplo el saber si existen otras variables climáticas de interés. Por tanto, se planteará una propuesta de solución basada en MAS, ya que ha sido ya empleada en SG (Pipattanasomporn *et al.*, 2009).

Tal y como expone Farhangi (2010), las redes de energía se han mejorado continuamente gracias al aumento del control, la flexibilidad y la eficiencia. El actual enfoque de infraestructura de medición avanzada es un paso inicial necesario para tener en todo el mundo las redes inteligentes. En esta transición, la tecnología capaz de tratar con *plug-and-play* los componentes, con inteligencia integrada, es muy necesario, y convirtiéndose en entornos candidatos para la aplicación de MAS. La necesidad de aumentar la inteligencia en los nodos locales a través de MAS es defendida también por Massoud and Wollenberg (2005), los autores comparan un enfoque puro centralizado con un enfoque MAS, señalando las capacidades *plug-and-play* proporcionados por la solución MAS y un mejor uso de las comunicaciones. La tecnología MAS tiene el rendimiento y la

escalabilidad necesaria para tratar con *microgrid*, como aparece en Oyarzabal *et al.* (2005), quienes han construido una red de micro y asociado a un agente para cada elemento. Estos agentes son capaces de negociar las cargas de trabajo, lo que condujo a una mayor eficiencia global y la adecuación a la demanda.

Debido a las razones antes mencionadas, se decidió experimentar con un diseño orientado al agente y su implementación. Como metodología, se optó *INGENIAS*, debido a su sencilla implementación por medio del diseño.

El resultado de aplicar *INGENIAS* es un proyecto de arquitectura de *MAS* para el despliegue de un sistema de conocimiento del entorno climático y la red eléctrica. La Figura 6.10 presenta la manera en que los elementos cooperan, destacando dos interacciones fundamentales: una para entregar datos meteorológicos y otro para proporcionar un pronóstico. Varios sensores, representados por el rol de *WeatherObserver*, se han planteado para medir diferentes variables meteorológicas. Los valores de variables se entregan a una o varias agencias de modelos climáticos, representado como rol *Forecaster*, que saben la forma en que la información del clima se asocia con el modelo climático que manejan. Posteriormente, se asocia la evolución de las variables climáticas con la demanda de energía. A continuación, el pronóstico de la demanda se notifica a cada una de las redes inteligentes (rol *PowerGrid Manager*) y, a cambio, se obtiene información sobre la demanda real, lo que ayuda a mejorar el modelo climático. Los roles mencionados (*Weather Observer*, *PowerGrid Manager* y *Forecaster*) son interpretados por tipos de agentes concretos que representan diferentes tipos de redes eléctricas. Sería de esperar que cada agente tenga cero o más casos, dependiendo del escenario considerado.

En este trabajo, se presenta una instancia de la implementación, para demostrar que la arquitectura se ha simulado. Se trata de la definición de un conjunto de agentes que cumplen la función de los agentes especificados en la Figura 6.10. Este esquema de agente se muestra en la Figura 6.11. Se asemeja a la estructura de algunas ciudades españolas, concretamente en Madrid y Soria donde los datos meteorológicos reales se suministran a los agentes de observación del clima. Madrid y Soria proporcionar sensores de humedad y temperatura (*Temperature Observer* y *Humidity Observer*). Hay agentes responsables de la administración de energía (*Smart City/Microgrid*) para ambos entornos. Habría un ejemplo de *Energy Demand Forecaster*, responsable de análisis de datos y las previsiones de demanda.

El escenario de esta simulación se presenta en la Figura 6.12. Modela una simulación basada en agentes de 100 unidades de tiempo simulado. En las 20 primeras unidades, un evento que indica un aumento de temperatura se genera cada 2 unidades de tiempo en los *Temperature Observers* de Madrid. Con esta arquitectura, podemos acomodar los algoritmos que producen el pronóstico dentro

del *Energy Demand Forecaster* y reproducir su fin a las condiciones reales de implementación de software, en lugar de trabajar con conjuntos de datos brutos.

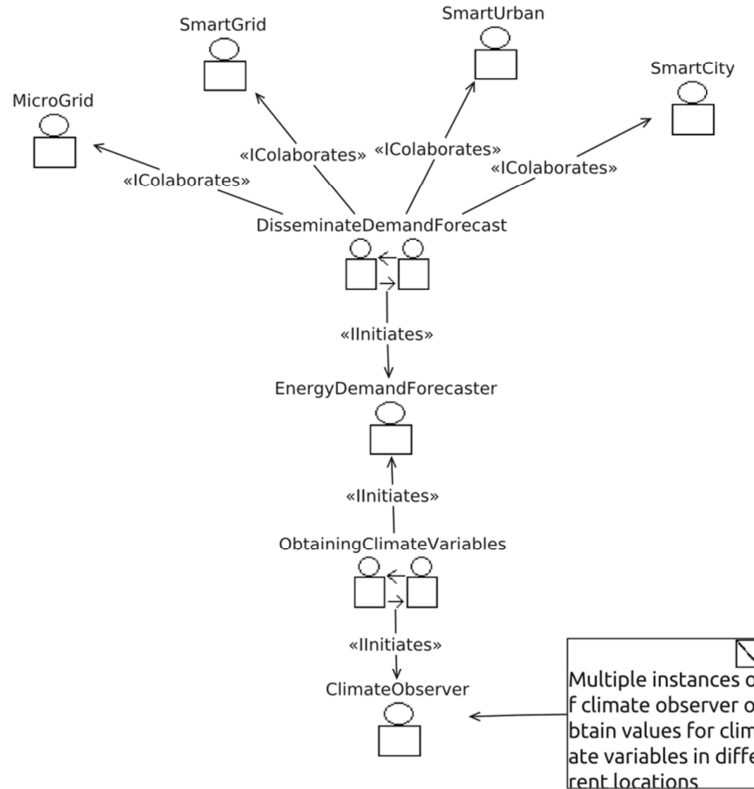


Figura 6.10. Diseño MAS de una arquitectura para *Smart World* aprovechando datos climáticos. Fuente Propia.

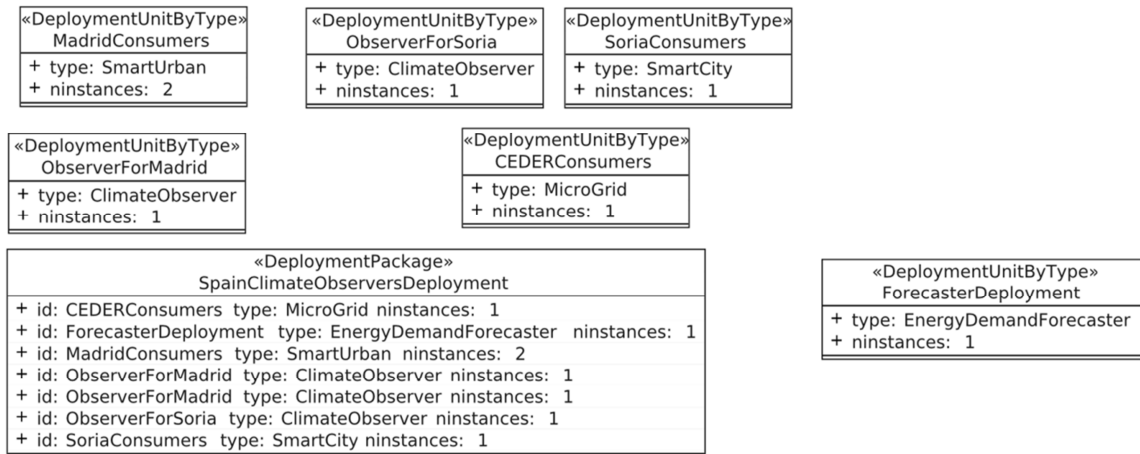


Figura 6.11. Instancia de esquema de agentes en la simulación. Fuente Propia.

La arquitectura resultante es bastante flexible y proporciona las características *plug-and-play* identificadas en (Pavón and Gómez, 2003). Sensores adicionales se pueden implementar incluso en tiempo de ejecución para participar de forma transparente en la ejecución. Del mismo modo, los nuevos agentes que representan a otros actores, se incorporarían automáticamente en la notificación de los informes del pronóstico. Esto es posible porque los agentes hacen un amplio uso de los servicios de páginas amarillas cada vez que se acopla una interacción. Por lo tanto, si se incorporan



nuevos agentes deben notificar al marco de gestión de agente y registrarse en los servicios de páginas amarillas.

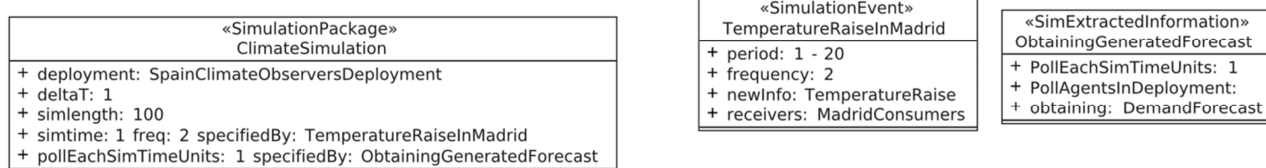


Figura 6.12. Configuración de la simulación para el escenario de Madrid. Fuente Propia.

### 6.1.6. Conclusiones y estudios futuros

Del análisis realizado sobre los actuales entornos *Smart*, y teniendo presente que en un futuro se irán incrementando con otros nuevos, se ha visto como todos tienen objetivos finales en común, por lo que se ha planteado un concepto, que será el que aglutine a todos los existentes, llamado *Smart World*, y que servirá a comunidades diferentes para obtener información variada, la cual procederá de diferentes tecnologías e infraestructuras. Los objetivos globales de *Smart World*, los cuales son compartidos por los distintos *Smart* en mayor o menor medida, han sido presentados. El diseño de aplicaciones, servicios y protocolos, que cumplan uno, algunos o todos los objetivos arriba indicados, independientemente que sean para *SG*, *microgrid*, *SC*, *SE*, etc.; formarán parte de un concepto de nivel superior, que será *Smart World*, formando así una jerarquía de entornos agrupados, y teniendo como origen a *Smart World*.

Se ha demostrado la relación existente entre ciertas variables climáticas y el comportamiento de la demanda eléctrica en un entorno semejante a una *microgrid*. La relación entre las variables climáticas y la demanda eléctrica será cambiante en el tiempo (la dependencia no fue la misma seguramente hace veinte años ni será parecida dentro de diez). De la misma forma, la demanda será muy dependiente del entorno físico donde se haga la prueba; no es lo mismo un entorno con una climatología muy adversa, que una zona con una estabilidad climática mayor y más suave. Estos dos factores comentados, tiempo y espacio físico, hacen necesaria la existencia de aplicaciones que evolucionen con el tiempo y se centren en la localización objeto del estudio, para tener una información precisa de los comportamientos cambiantes entre las variables climáticas y en este caso, la demanda eléctrica, precisando para su correcto funcionamiento, comunicación y captura de información procedentes de modelos diferentes, sin importar al objetivo de la aplicación si ciertos datos vienen de la infraestructura eléctrica o de la red nacional de observatorios meteorológicos. Continuando con el ejemplo presentado en esta publicación y para finalizar, indicar que la aparición de inteligencia distribuida en entornos urbanos, interurbanos y clásicos del sistema eléctrico, así como información procedente de otras plataformas, va a permitir nuevas aplicaciones en tiempo

cuasi-real que en muchas ocasiones tendrá como objetivo el seguimiento y control de las variables climáticas y condiciones medio ambientales de los espacios controlados.

Los resultados obtenidos entre variables climáticas y demanda eléctrica, permitirán el diseño para validar nuevos modelos basados en *ANN*, para la predicción de la demanda eléctrica de entornos desagregados (subestación, *microgrid*, etc.).

## **6.2. CLASIFICACIÓN Y CLUSTERIZADO DE PATRONES DE DEMANDA ELÉCTRICA EN POLÍGONOS INDUSTRIALES**

La comprensión de los patrones de consumo de energía eléctrica se convierte en una tarea extremadamente importante para la optimización de los recursos y la aplicación de tendencias “*verdes*”. Tradicionalmente, este tipo de análisis se han venido realizando en entornos de gran tamaño, como regiones extensas o países enteros. Sin embargo, con el advenimiento de las *SGs*, el estudio del comportamiento en entornos más pequeños se ha convertido en una necesidad para permitir una *microgestión* más profunda de las redes inteligentes.

En esta sección se presenta un sistema de procesamiento de datos para tratar de analizar los patrones de consumo eléctrico en polígonos industriales; el planteamiento se basa en la aplicación en cascada de *SOM* y el algoritmo de clusterizado *k-means*. El sistema será validado con datos de un entorno real, procedentes de un polígono industrial de España. Los resultados de la validación muestran que el sistema encuentra claras diferencias en los patrones de consumo (diferente comportamiento energético), y es capaz de hacerlo sin supervisión, y sin previa información sobre los datos.

### **6.2.1. Introducción sobre el estudio**

Como se ha mencionado, los polígonos industriales (*microgrids* desde el punto de vista de la red de energía) son actores muy importantes en el mercado energético de un país. Es necesario que los *agregadores* controlen y gestionen estos espacios, tratando al mismo tiempo, de optimizar el consumo de energía y las ofertas de energía emitidas. Existen varias herramientas para realizar esta optimización, como la demanda *DR*, lo que significa acoger parte de la demanda (por medio de la emisión de ofertas o el control directo de las cargas inteligentes con restricciones de tiempo) para el horario más adecuado de los generadores (por ejemplo, cambios en la demanda de los picos de consumo a los valles para obtener una curva más plana, con un pico máximo de potencia inferior, o para dar cabida a algunas operaciones durante períodos de viento a aerogeneradores de tamaño).

La importancia de la gestión de energía en polígonos industriales ha sido ampliamente demostrada en la literatura, como por ejemplo en (Hingorani, 1995; Halpin *et al.*, 2001; Liserre *et al.*, 2010; Santacana *et al.*, 2010; Xinghuo *et al.*, 2011). Park (2010) presenta la simulación de la

operación en una planta industrial con medición distribuida. Zareipoor *et al.* (2010) presentan un escenario de operación de una planta industrial basada en precios, para lo cual, la gestión y el control del polígono industrial es esencial. Un modelado de laboratorio, con supervisión y control de una *microgrid*, extrapolado a polígonos industriales, se muestra en (Vaccaro *et al.*, 2011).

*DR* y otros métodos de optimización no son posibles sin un conocimiento detallado del comportamiento del polígono industrial. En un polígono industrial, coexisten diferentes tipos de industrias, con hábitos de consumo eléctrico muy dispar; la previsión tradicional y los métodos de análisis de datos, se aplican en grandes regiones, por tanto, ya no son aplicables directamente.

Por lo tanto, el objetivo de este trabajo es presentar un sistema de análisis de datos para clusterizar las curvas de carga en los polígonos industriales; el sistema estará basado en *SOM+k-means* y se validará con datos reales. Algunas iniciativas con el mismo enfoque han sido ya planteadas en la literatura, pero con diferentes propósitos. Por ejemplo, Chicco *et al.* (2004) y George *et al.* (2007) aportan las herramientas de análisis de patrones de carga a diferentes tipos de clúster de clientes. La principal diferencia con el trabajo que aquí se presenta es que, mientras que los estudios fueron hechos para discriminar y agrupar las curvas de carga en diferentes *utilities*, aquí se trata de discriminar patrones de consumo diferentes de una sola *utility*, y en un entorno característico como el de los polígonos industriales.

### 6.2.2. Arquitectura del sistema

Como se mencionó en la introducción, el objetivo del sistema consistirá en agrupar las curvas de carga diarias de un polígono industrial en diferentes grupos significativos. Esto permitirá la clasificación de los días en diferentes grupos, donde cada uno de ellos presentará similares patrones de curvas de carga y un conjunto de rasgos característicos dentro de la misma agrupación. El sistema está compuesto por una *SOM* para clasificar las curvas de carga, seguido de una agrupación/clusterizado por medio del algoritmo *k-means*. La arquitectura del sistema se presenta en la Figura 6.13, y comprende los siguientes módulos:



Figura 6.13. Arquitectura del sistema. Fuente propia.

- Datos Históricos (*Historical Data*): la base de datos tendrá los datos de consumo cuarto–horarios, incluyendo información del calendario para cada muestra como día del mes, mes, año, laborabilidad y día de la semana.
- Procesado de Datos (*Data Processing*): limpiar la base de datos (extracción de muestras con error o interpolación cuando sea posible) y dar formato para la entrada de las muestras (por ejemplo agregar los valores cuarto–horarios a horarios).
- Detección Outliers (*Outlier Detection*): como método de detección de outliers se empleará *PCA*. Se empleará la misma estrategia que en 6.1.3.2.
- *SOM+CLUSTERING*: representa la aplicación combinada de *SOM* y *k-means*.
- Salida Gráfica (*Graphic Output*): su principal cometido es enviar la información de los resultados dados por la etapa previa, que serán mostrados gráficamente.

### 6.2.2.1. Self-Organizing Map

Descrita por Kohonen, *SOM* es una arquitectura *ANN* designada para el reconocimiento de patrones, y siendo algunas de sus aplicaciones la clasificación en grupos (Kohonen, 1982a; 1982b; 1990). La primera es la capa sensorial o de entrada, que consiste en  $m$  neuronas (tantas neuronas como variables de entrada), cuya función es distribuir la información desde el espacio de entrada a la segunda capa. La segunda capa forma un mapa de rasgos conformando una red de neuronas  $n_x \times n_y$  que operan en paralelo. Las neuronas de entrada se etiqueta con el índice  $k$  ( $1 \leq k \leq m$ ), y  $n_x \times n_y$  neuronas en el mapa con un par de índices  $I \equiv (i, j)$  ( $1 \leq i \leq n_x$ ,  $1 \leq j \leq n_y$ ), que determina su ubicación espacial. Cada neurona de entrada ( $k$ ) está conectada a todas las neuronas ( $i, j$ ) en el mapa con un peso  $w_{ij}$  sináptica. La representación de esta arquitectura se muestra en la Figura 6.14.

*SOM* presenta un enfoque de aprendizaje competitivo: cuando el vector de entrada  $[x(t)]$  se presenta a la red, la similitud entre este vector y el peso sináptico cada neurona ( $w_{ij}$ ) se calcula. La neurona cuyo vector de peso es más parecido a la entrada es considerada la ganadora. Entonces, el peso sináptico se modifica para estar más cerca de  $x(t)$ , por lo que cuando la red se enfrente a patrones similares en el futuro, esa responderá de manera más fuerte que las demás.

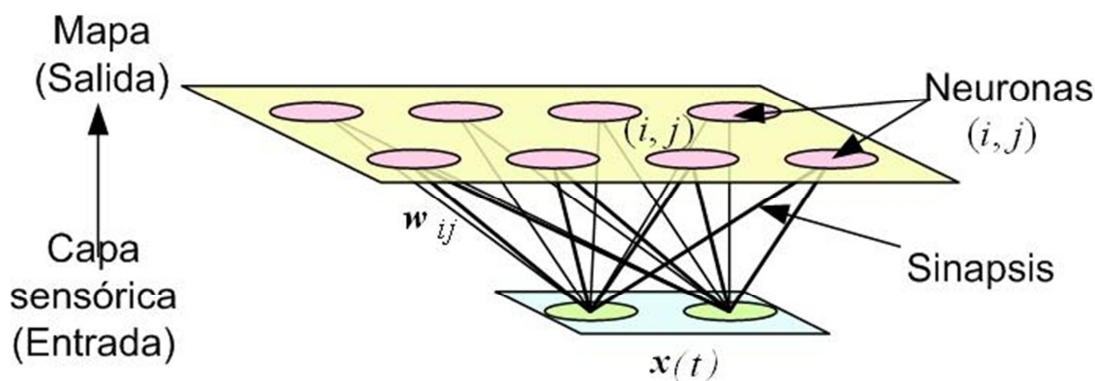


Figura 6.14. Arquitectura SOM. Fuente Propia.

### 6.2.2.2. Clustering

Clustering significa particionar un conjunto de datos en un conjunto de  $C$  clústers  $Q_i$ ,  $i=1, \dots, C$ ; implica que esta división es no supervisada, es decir, se hace sin ningún conocimiento previo acerca de la estructura de datos, como el número de grupos, su definición, o cualquier parecido de las muestras a su clústers correspondiente. Una definición ampliamente adoptada de clustering óptimo es una partición que minimiza las distancias dentro y maximiza las distancias entre los clústers.

El algoritmo *k-means* (MacQueen, 1967) es uno de los más simples y más ampliamente utilizado algoritmos de aprendizaje no supervisado que resuelve el problema de clustering. El procedimiento sigue una manera simple y fácil para clasificar un determinado conjunto de datos a través de un cierto número de grupos fijados a priori. La idea principal es definir  $k$  centroides, uno para cada grupo. El siguiente paso es tomar cada punto perteneciente a un determinado conjunto de datos y asociarlo al centroide más cercano. Cuando no hay un punto pendiente, el primer paso se ha completado y un clustering temprano se ha conseguido. En este punto, es necesario volver a calcular los nuevos centroides  $k$  como baricentros de los clústers resultantes de la etapa anterior. Después de tener estos nuevos centroides  $k$ , una nueva iteración de la asignación del registro de datos apunta al nuevo centroide más cercano. Los centroides  $k$  cambian su ubicación paso a paso hasta que ya no se mueven significativamente.

### 6.2.2.3. Combinación de algoritmos

Vesanto and Alhoniemi (2000) muestran que un clustering de *SOM* obtiene mejores resultados que clustering de los datos directamente. La principal ventaja del enfoque de dos niveles (*SOM+Clustering*) es la reducción del coste computacional: incluso con un número relativamente pequeño de muestras, muchos de los algoritmos de clustering (especialmente los jerárquicos) se hacen excesivamente intensivos en cuanto a recursos. Por esta razón, es conveniente agrupar un conjunto de prototipos en lugar de proceder directamente con los datos en bruto. Otro beneficio es la reducción de ruido: después de *SOM*, los prototipos son promedios locales de los datos y, por tanto, menos sensible a las variaciones aleatorias que los datos originales.

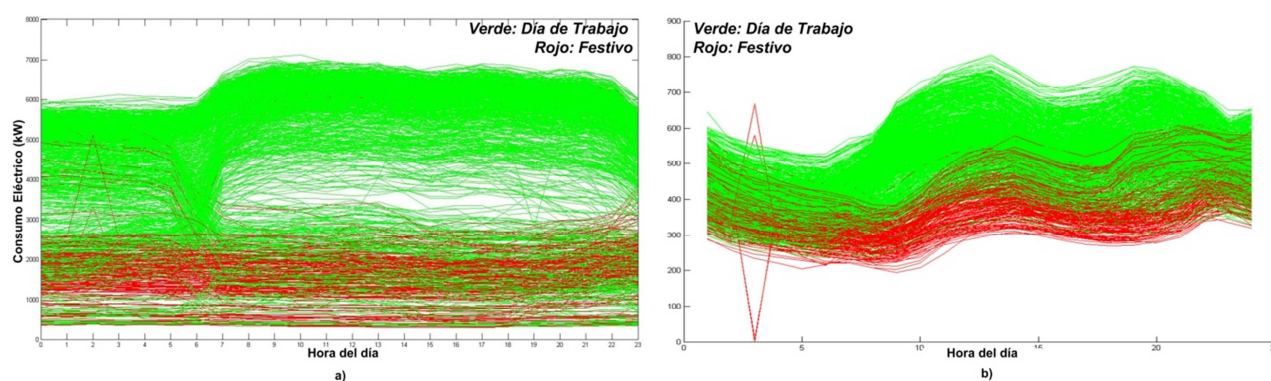
## 6.2.3. Caso de estudio

### 6.2.3.1. Datos de investigación y escenario de test

Iberdrola, compañía española distribuidora de energía eléctrica, ha facilitado el conjunto de datos para poder hacer el presente estudio, contando datos desde el 1 de enero de 2008 hasta 31 de diciembre de 2010, procedentes de un polígono industrial en Soria (Castilla y León – España). Los

datos facilitados por la compañía son: día de la semana, día del mes, mes, año y los veinticuatro valores de consumo eléctrico de cada día (curva de carga). El rango de consumos varía entre los 0.3–7 MW, no correspondiéndose con valores de una gran ciudad, o una zona extensa, y ni tan siquiera un país, por lo que se podría asimilar al consumo de una zona desagregada o *microgrid*. Por tanto, los datos anteriores podrían fácilmente modelar una zona desagregada o *microgrid*.

Como se ve en la Figura 6.15, las curvas de carga del polígono industrial no presentan un patrón fácilmente reconocible, si se compara con entornos con agregación de consumos (ciudades, países, etc.). Sirva de ejemplo las curvas comparativas del polígono industrial “Las Casas” y las curvas de carga de la totalidad de Soria, de la cual decir que además es una ciudad pequeña, por lo que si se compara con entornos de mayor tamaño, la diferencia sería aún mayor (en cuanto a magnitud del consumo como a la forma de las curvas). El polígono industrial escogido, está compuesto por un pequeño número de empresas, que pertenece a diferentes tipos de industrias, las se comportan de forma diferente en términos de consumo de energía.



**Figura 6.15. a) Curvas de carga del polígono industrial “Las Casas”; b) Curvas de carga de un entorno con mayor agregación que a) (Soria capital). Las curvas verdes representan días laborables y las curvas rojas representan días no laborables (domingos y festivos entre semana). Fuente propia.**

### 6.2.3.2. Configuración de SOM

Hay dos maneras diferentes para realizar la inicialización SOM:

- Aleatorio (*Random*): para cada componente  $x_i$ , los valores están distribuidos uniformemente en el intervalo de  $[\min(x_i), \max(x_i)]$ .
- Lineal (*Linear*): los valores propios y los vectores propios de la formación de datos se calculan y entonces el mapa comienza a lo largo de los mayores vectores propios de  $mdim$ , donde  $mdim$  es la dimensión de la asignación de red.

El método de inicialización lineal se emplea debido a que mejora el tiempo de formación. Después de realizar varios ensayos con diferentes tamaños, las dimensiones de la SOM se fija como una matriz de  $4 \times 4$  de neuronas hexagonales, siguiendo un enfoque heurístico similar al que se puede encontrar en (Hsu and Yang, 1991; García *et al.*, 1995; López *et al.*, 2011). Este tamaño permite

una buena dispersión de los datos, capaz de discriminar los factores importantes para este estudio, como la estacionalidad, laborabilidad o el día de la semana.

La función de vecindad empleado es *gaussiana*. Para la optimización de los parámetros de configuración de la *SOM*, se ha empleado un *script* y así poder probar el rendimiento de las diferentes combinaciones de la red; después de un centenar de iteraciones con cada combinación, la configuración óptima es escogida. Los parámetros elegidos para *SOM* son: inicialización del vector de referencia *linear*; algoritmo de entrenamiento tipo *batch*; tamaño del mapa  $4 \times 4$  (neuronas), función de vecindad *gaussiana*, entradas de red 27.

Las variables de entrada a la *SOM* (para cada vector de entrada) y su codificación, son los siguientes:

- Mes (*Enero=1, Febrero=2,..., Noviembre=11, Diciembre=12*).
- Día de la semana (*Domingo=0, Lunes=1,..., Viernes=5, Sábado=6*).
- Laborabilidad (*festivo 1 y laborable 2*).
- 24 valores de consumo de electricidad por hora, lo que representa la curva de carga diaria.

### 6.2.3.3. Configuración del algoritmo *k-means*

El algoritmo *k-means* requiere que el usuario defina el parámetro *k*, el número de clústers a construir, antes de su aplicación. Cuando no existe un conocimiento a priori sobre los datos, escoger el valor correcto del parámetro *k* es a veces difícil. Hay muchos métodos documentados para estimar el número óptimo de las agrupaciones; en este trabajo se han realizado diferentes pruebas, evaluándose por medio del índice de validez de Davies–Bouldin (Davies and Bouldin, 1979), de acuerdo con el mejor clustering que minimice

$$\frac{1}{C} \sum_{k=1}^C \max_{l \neq k} \left\{ \frac{S_c(Q_k) + S_c(Q_l)}{d_{ce}(Q_k, Q_l)} \right\} \quad (6.6.)$$

donde *C* es el número de clústers;  $S_c$  la distancia intra-clúster; y el  $d_{ce}$  la distancia inter-clúster. Para cada valor de *k* entre 2 y 8, el algoritmo *k-means* se ha ejecutado cinco veces y se ha calculado el índice de validez medio.

## 6.2.4. Resultados

A lo largo de esta sección, los resultados de la aplicación de la arquitectura propuesta se presentan, habiendo empleado los datos de validación.

### 6.2.4.1. Reconocimiento de patrones con *SOM*

Se ha empleado una *SOM* con arquitectura  $4 \times 4$ . Las figuras de esta sección (Figuras 6.16–6.19) representan los enlaces entre las neuronas, así que una cuadrícula de  $7 \times 7$  representa la

neurona en concreto más los enlaces con el resto de neuronas vecinas. Por ejemplo, en la fila superior, hay cuatro neuronas enlazadas por tres enlaces inter-neuronales, y la segunda fila comenzará con enlaces inter-neuronales seguido de neuronas (habrá cuatro enlaces y tres neuronas), conectando los enlaces a las neuronas de la primera fila con las de la segunda y tercera fila. Los hexágonos de colores representan las neuronas que albergarán a los patrones de entrada que han sido clasificadas en dicha neurona: cuanto más grande es el hexágono de color, mayor es el número de patrones asignados a dicha neurona.

Después de procesar los datos con *SOM*, el resultado de los clústers se presenta en las siguientes figuras, y se analizarán según cuatro criterios diferentes para simplicidad de su interpretación y comprensión:

- En primer lugar, los clústers se muestran en la Figura 6.16 según la laborabilidad de los días. La parte izquierda de la imagen muestra la manera en que los días no laborables se agrupan en neuronas próximas, y la parte derecha presenta la agrupación de días laborables.

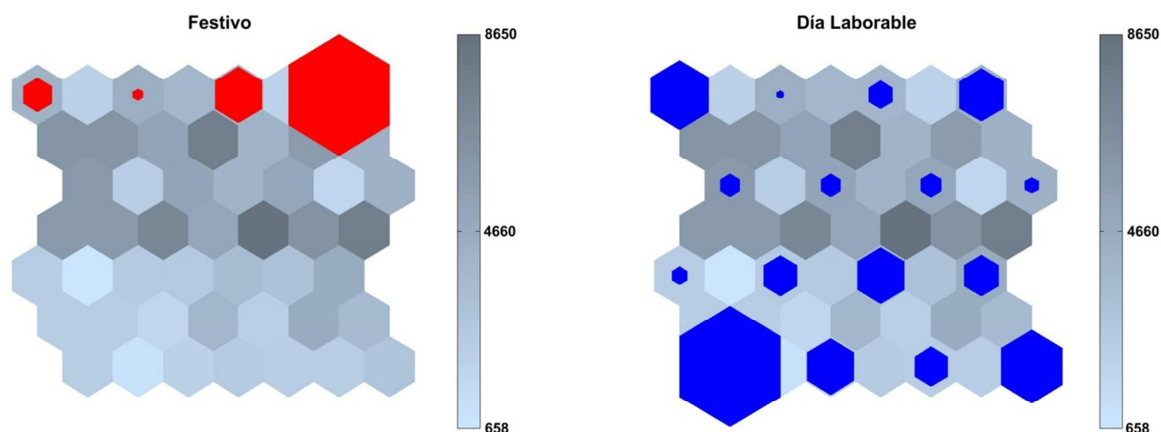


Figura 6.16. Mapa de activación según laborabilidad: a) días festivos; b) días laborables. Fuente Propia.

- La Figura 6.17 muestra la forma en que los días de los distintos meses se agruparon. Es fácil ver que enero, febrero y marzo activan neuronas similares; abril, mayo, junio y julio también activan grupos de neuronas muy próximas; así como septiembre, octubre y noviembre; agosto y diciembre, ambos, se muestran aislados. Esto tiene sentido, ya que las estaciones son más o menos separadas, junto con las vacaciones de verano (agosto) y Navidad (diciembre). La demanda de electricidad es estacional, tal y como muestran Hernández *et al.* (2012a).
- La figura 6.18 muestra la agrupación de los diferentes días de la semana, resultando cuatro diferentes patrones de activación: los lunes tienen su propio patrón de activación; martes, miércoles, jueves y viernes tienen patrones similares de activación; los sábados; y los domingos.



- Finalmente, la Figura 6.19 presenta un análisis combinado de las agrupaciones por día de la semana y laborabilidad. Todos los días festivos se agrupan alrededor de las neuronas similares a las del domingo, excepto los miércoles y los jueves.

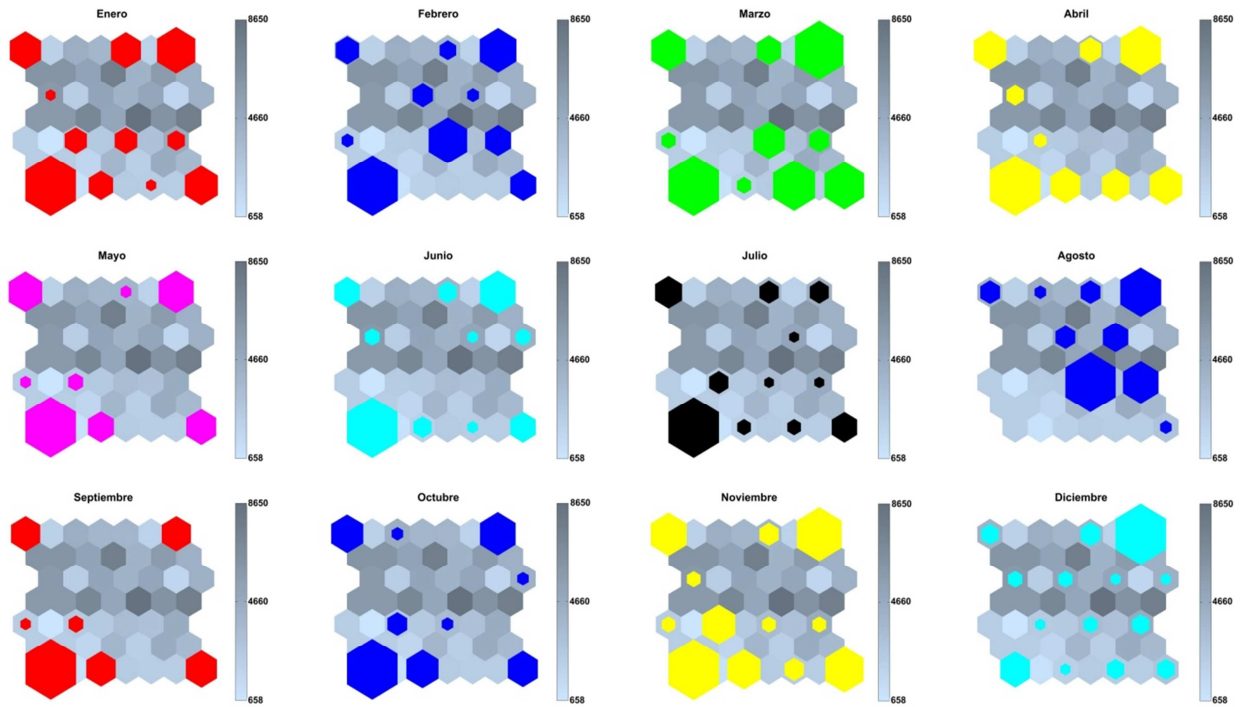


Figura 6.17. Mapa de activación según meses. Fuente Propia.

#### 6.2.4.2. Clustering con *k*-means

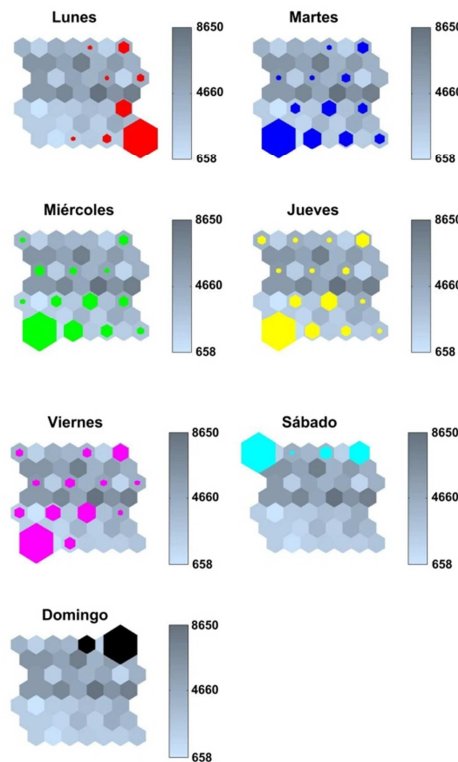


Figura 6.18. Mapa de activación según días de la semana. Fuente propia.

Los resultados de las diferentes evaluaciones de los clústers muestran como resultado óptimo cinco clústers para *k-means*. La Figura 6.21 presenta todas las curvas de carga pertenecientes a cada uno de los cinco clústers. Es fácil ver la gran similitud entre las curvas de carga del mismo clúster, y la diferencia apreciable cuando se comparan con las curvas de carga de los otros clústers.

En el anexo D se entregan las figuras con la representación de los patrones de carga que pertenecen a cada una de las neuronas, tras aplicar *SOM*. Tras esta etapa, y realizado el clusterizado mediante el algoritmo *k-means*, el agrupamiento de las neuronas en cada clúster se muestra a continuación, indicando con el número, la posición de la neurona en el mapa *SOM*: clúster 1: 1 y 5; clúster 2: 2 y 6; clúster 3: 3, 4, 7 y 8; clúster 4: 11, 12, 15 y 16; clúster 5: 9, 10, 13 y 14.

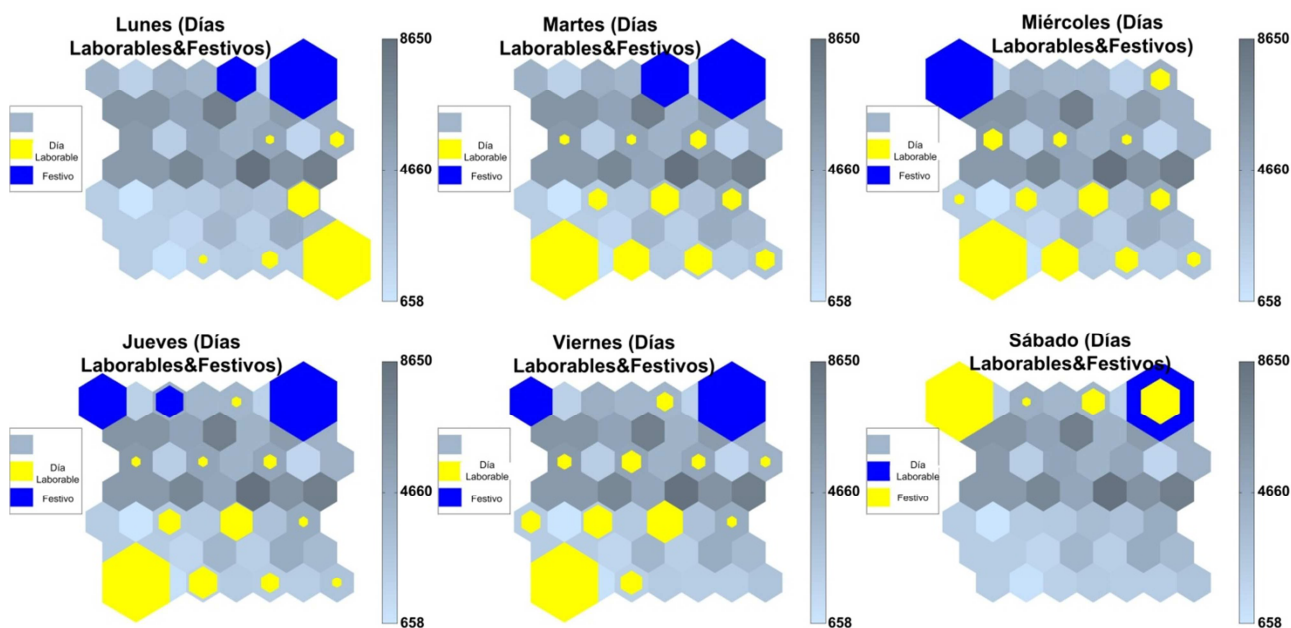


Figura 6.19. Mapa de activación según laborabilidad y día de la semana. Fuente Propia.

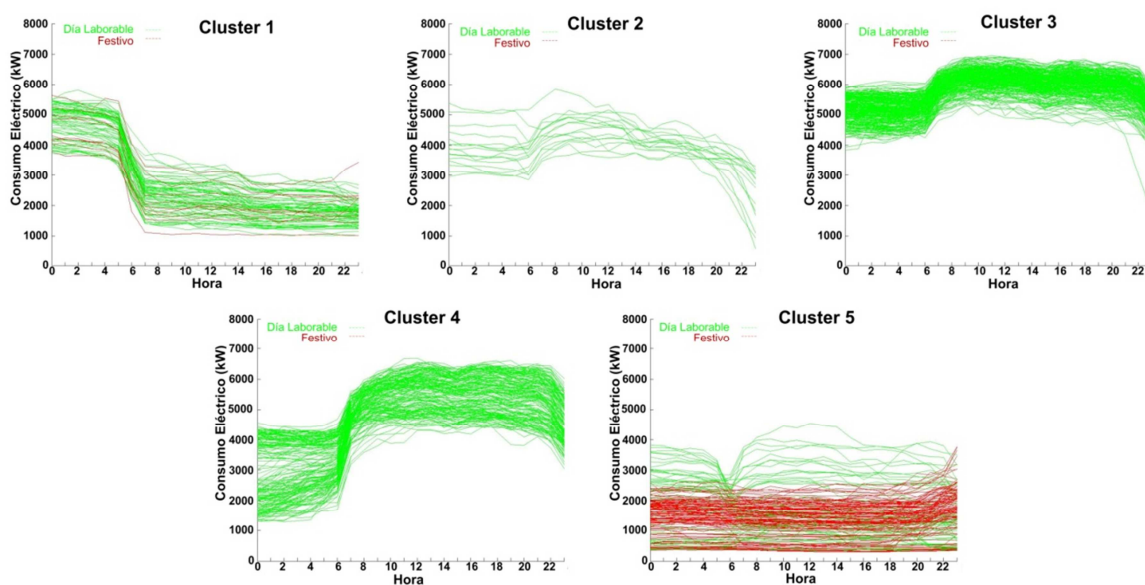


Figura 6.20. Curvas de carga de los cinco clústers. Las curvas rojas son días festivos, y las curvas verdes son días laborables. Fuente Propia.

6.2.4.3. Algoritmo de decisión

Después de los resultados anteriores, la Tabla 6.4 y la Figura 6.21 presentan un análisis del consumo diario acumulado para cada uno de los clústers.

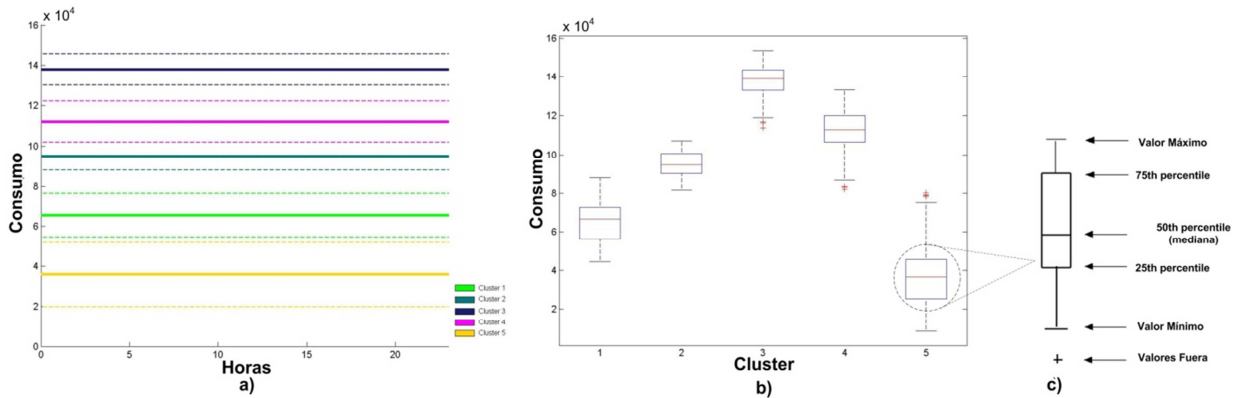


Figura 6.21. a) Carga agregada media±desviación estándar (y-eje muestra los valores de consumo de energía en W y x-eje las 24 horas del día); b) box plot (y-eje muestra los valores de consumo de energía en W a través de los cinco clústers en x-eje). Fuente Propia.

Tabla 6.4. Análisis de los resultados por clúster. Fuente Propia.

Número de clúster	Día de la semana	Mes	Laborabilidad	Consumo medio (kW)
Clúster 1	Sábado	Indiferente	Laborable	4000 < Consumo
	Sábado		Festivo	
Clúster 2	Indiferente	Indiferente	Laborable	4000
Clúster 3	Martes	No Agosto	Laborable	4000 < Consumo < 6000
	Miércoles			
	Jueves			
	Viernes			
Clúster 4	Lunes	Indiferente	Laborable	1500 < Consumo < 4500
	Martes			
	Miércoles	Agosto		
	Jueves			
	Viernes			
Clúster 5	Lunes	Indiferente	Festivo	Consumo < 4000
	Martes			
	Miércoles			
	Jueves			
	Viernes			
	Sábado			
	Sábado	Indiferente	Laborable	Consumo < 4000

La figura 6.22 muestra un algoritmo de decisión diseñado para controlar cuando los sistemas tienen que ser reentrenados debido a cambios en el comportamiento de la curva de carga.

Básicamente, el reentrenamiento se activa cuando la demanda agregada para un día está fuera de los márgenes de la demanda agregada del clúster al que pertenece dicho día.

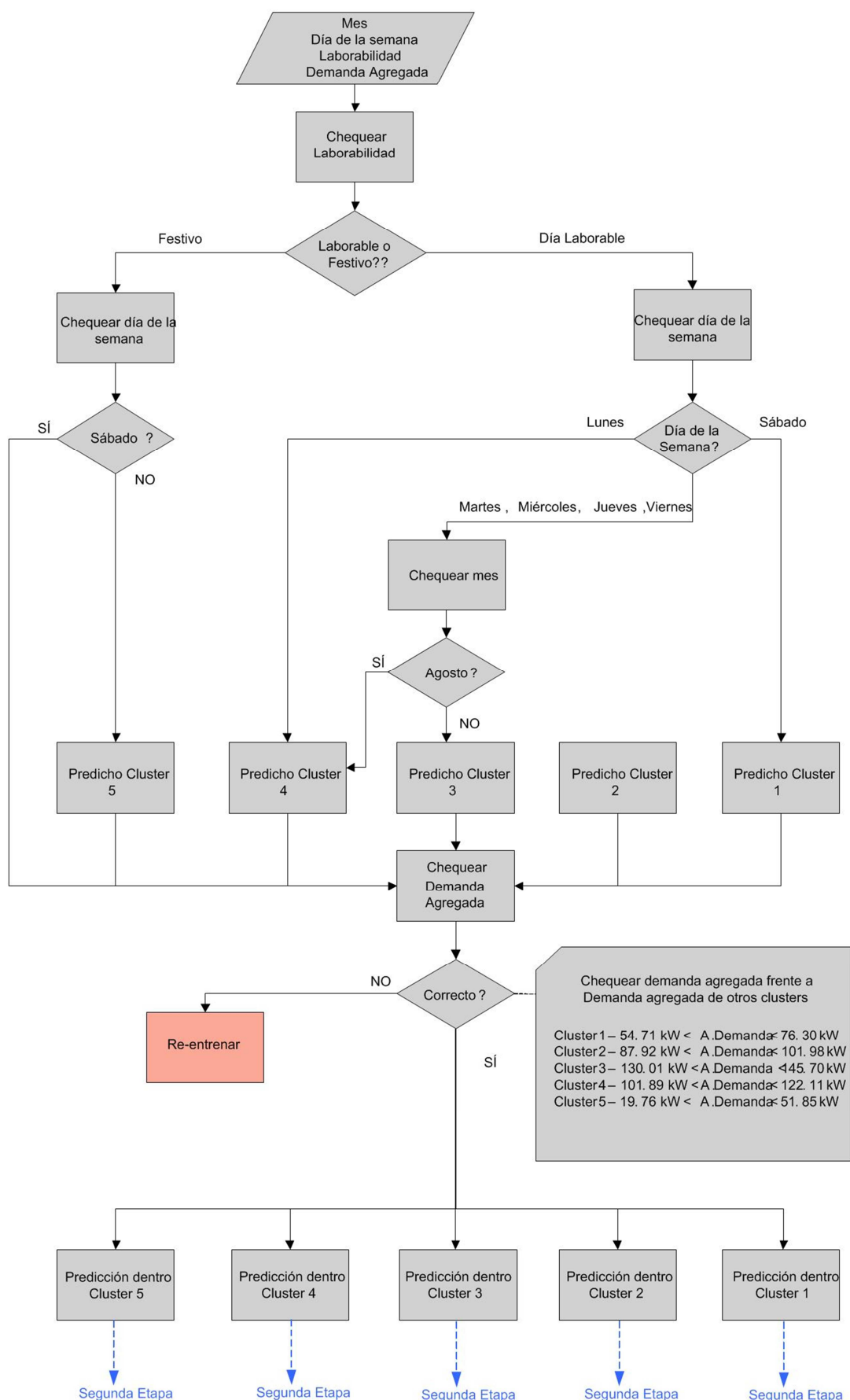


Figura 6.22. Algoritmo de decisión. Fuente propia.

### **6.2.5. Conclusiones y estudios futuros**

El sistema de análisis de los datos presentados en este trabajo se ha probado y validado con datos del mundo real. En los resultados, el comportamiento del consumo diario de un polígono industrial real ha sido analizado mediante el clustering de días diferentes, de acuerdo con sus curvas de carga, y los patrones de comportamiento han sido correctamente identificados por el sistema y, además, de una manera completamente no supervisado.

Esto demuestra que el sistema es realmente capaz de proporcionar información muy útil sobre los patrones de consumo en entornos desagregados y con comportamiento de consumo eléctrico a pequeña escala. Como se ha discutido, el sistema descrito representa un paso importante en el estado actual de la técnica de identificación de patrones de curvas de demanda, que se centraba en aplicaciones sobre grandes áreas (tales como regiones o naciones), pero no en entornos desagregados (entornos rurales, pequeñas ciudades, polígonos industriales o *microgrid*). Estos entornos no sólo son un escenario de aplicación diferente del caso típico, sino que además representan entornos más difíciles de controlar debido a la mayor variabilidad de las curvas de demanda debido a su desagregación, la cual será la causante de disponer de curvas de carga con perfiles más abruptos.

## **6.3. SHORT-TERM LOAD FORECASTING PARA ENTORNOS DESAGREGADOS BASADO EN ARTIFICIAL NEURAL NETWORKS**

En esta sección se presentará un modelo arquitectural basado en *ANN* para hacer *STLF*, justificando el porqué de las variables de entrada seleccionadas para la predicción, así como la presentación de los buenos resultados obtenidos y destacando la sencillez del modelo planteado. La predicción se ha centrado en un entorno desagregado, localización geográfica de dimensiones y consumos que podrían asemejarse a una *microgrid*. El modelo se validó con datos reales de dicho emplazamiento.

### **6.3.1. Introducción sobre el estudio**

Una de las características más notables de la producción de energía y el sistema de distribución (sistema tradicional) es que la mayoría de la energía se genera en grandes plantas ubicadas lejos de los puntos finales de consumo. Esto hace que aparezcan pérdidas debidas al transporte y, además, dificulta la posibilidad de descentralizar la generación de energía, resultando una alta dependencia de unas cuantas plantas de generación de gran tamaño. En los últimos años, se ha propuesto un cambio conceptual con la finalidad de que el sistema de suministro de energía sea

más sostenible, en términos económicos y ambientales, como por ejemplo queda reflejado en el Tratado de Lisboa (2009).

De acuerdo con estos nuevos conceptos, y con el fin de incrementar la sostenibilidad y optimizar el consumo de recursos, las *utilities* están constantemente tratando de ajustar el suministro de energía a la demanda. Teniendo en cuenta que es extremadamente difícil de almacenar energía a gran escala, la generación de energía eléctrica tiene que intentar ajustarse a la demanda en tiempo real. Por consiguiente, es importante que la predicción de la demanda eléctrica sea lo más exacta posible.

Sin embargo, la demanda de energía eléctrica depende de muchos factores, como el día de la semana, el mes del año, etc.; lo que hace de la demanda a pronosticar un proceso complejo, que sobrepasa los métodos estadísticos que se empleaban en los comienzos. En los últimos años, la previsión de la demanda se ha realizado empleando diferentes algoritmos de predicción, y entre ellos, las *ANNs* son una de las opciones más populares debido a su capacidad de aprender de forma automática a partir de la experiencia y adaptarse a sí mismos (Hernández *et al.*, 2013a).

Por otro lado, la necesidad de lograr un equilibrio entre la generación de energía eléctrica y la demanda se ha extendido a los entornos desagregados con menor generación de energía eléctrica y demanda (como subestaciones, pequeñas ciudades o *microgrids*), en las cuales la adaptación de la producción a la demandase puede realizar de forma más dinámica, debido a su distribución de elementos más pequeños (generadores y cargas) y la proximidad geográfica de todos ellos (que ayudará a reducir el transporte pierde). La curva de carga para un entorno desagregado (*microgrid* o similar), desagrega los datos de consumo de energía eléctrica, haciendo que los métodos tradicionales (diseñado para la previsión nacional o en toda una extensa región) no sean aptos para su aplicación directa. En estos entornos, no sólo la cifra global de consumo es varias veces menor que la que podría tener un país o una región, sino además, la curva de carga presenta una variabilidad mucho mayor y no siempre fácil de estimar. Algunos ejemplos de curvas de carga típicas para diferentes entornos se mostraron en la Figura 3.3, con el fin de ilustrar las diferencias. Es fácil darse cuenta de que la curva de carga típica presenta más ruido y presenta cambios abruptos, debido a la comentada desagregación de los consumos.

Sin embargo, las publicaciones examinadas hasta la fecha, sin importar el modelo de pronóstico y horizonte de predicción, tienen en común que la predicción se calcula para una amplia zona geográfica en la que la carga de energía eléctrica es agregada y muy alta.

Debe de tenerse en cuenta que en entornos desagregados (por ejemplo *microgrids*), como se mostró en la Figura 3.3, las características de la curva de carga son muy diferentes de la curva de carga agregada de un gran zona (regional o nacional) debido al efecto *agregador*; la carga eléctrica es mucho menor que en las grandes zonas, donde se suministra electricidad a países, regiones o

áreas muy grandes, en donde el valor de la carga es muy elevado, y por tanto, la aplicación directa de estas soluciones no es posible. Sus resultados no pueden extrapolarse directamente a entornos desagregados, como las *microgrids* o entornos similares.

Mientras que las soluciones estudiadas en la literatura presentar buenas cifras de predicción (normalmente con *MAPE* alrededor de 2%), se ocupan casi exclusivamente de grandes áreas o países enteros, y nunca se aplican a entornos más pequeños del tamaño de pequeñas ciudades, pueblos o *microgrids*. Por tanto, no dan ninguna evidencia del comportamiento cuando se aplica a las curvas de carga tan variables, como las que presentarán estos entornos desagregados.

Trabajos con respecto al procesamiento de datos en *microgrids* han comenzado a aparecer recientemente, tales como la clusterización de las curvas de carga en polígonos industriales (Hernández *et al.*, 2012a), lo que ayuda a extraer información significativa mediante la búsqueda de clústers de patrones similares.

Centrándose en la previsión de carga, Chan *et al.* (2011) presentan un modelo para *STLF* en una *microgrid* basado en Sistema Clasificador Múltiple (*Multiple Classifier Systems – MCS*), utilizando datos de una *microgrid*. *MCSs* son sistemas que combinan un conjunto de clasificadores básicos que ofrecen un mejor rendimiento cuando trabajan juntos que por separado. Los clasificadores básicos pueden incluir enfoques diferentes de clasificación, con diferentes algoritmos y conjuntos de datos, y luego combinarse con un método de fusión. En este trabajo específico emplea cuatro clasificadores básicos (emplean *MLP* o *RBF* debido a su capacidad de generalización bueno), dividiendo el conjunto de entrenamiento en varias partes: 24 horas, 3 días, 1 semana y 1 mes antes de las horas de predicción. El método de fusión empleado es la ponderación dinámica. Con un conjunto de datos recogidos de la demanda desde septiembre 2008 hasta agosto de 2010, el *MAPE* encontrado para este modelo es de 15.66% con Red Neuronal de Regresión Generalizada (*GRNN-MLP*) y 15.12% para una *RBFNN*, red neuronal de función base radial). Estos errores son sensiblemente superiores a los reportados en trabajos aplicados a entornos nacionales/regionales.

Por tanto, en esta sección se presenta un modelo de arquitectura basado en *ANN* para *STLF* y de aplicación para entornos desagregados, tales como *microgrids*. Se presentará la propuesta de arquitectura *ANN* ya comentada, junto a las variables de entrada que empleará el modelo. La validación se realizará con los mismos datos disponibles para el estudio de la sección 6.1, en concreto los datos han sido explicados en la sección 6.1.3.1. Se presentarán los resultados obtenidos y, por último, las conclusiones y trabajos de futuro.

### 6.3.2. Un modelo arquitectural para la predicción de carga en entornos desagregados

Una *microgrid* es capaz de controlar las cargas eléctricas que oscilan entre cientos de *kW* hasta cientos de *MW*. Por lo tanto, mientras que las redes tradicionales de suministro de energía eléctrica abastecen a todo un país, *microgrids* suministra energía eléctrica a las ciudades pequeñas y pueblos. La desagregación de datos produce picos de carga y valles que son más difíciles de predecir, y por lo tanto los métodos tradicionales no son directamente aplicables si se precisa una exactitud aceptable. Esta sección presenta la arquitectura de sistema completo para *STLF* en un entorno desagregado de características similares a las que se encontrarían en una *microgrid*. Para la implementación y las pruebas de este sistema, se han empleado los mismos datos de la sección 6.1, una pequeña ciudad española con un tamaño que podría considerarse similar a la de una *microgrid*.

#### 6.3.2.1. Arquitectura de alto nivel para el sistema de predicción

El objetivo del sistema es operar en tiempo real dentro de un entorno desagregado, recibiendo datos de los *DCs* conectados a los *SMS*, así como de otras fuentes de datos inteligentes presentes en la red. La arquitectura del predictor es similar a la presentada en la Figura 6.13, pudiéndose ver en la Figura 6.23. Los diferentes componentes son:



Figura 6.23. Arquitectura del sistema. Fuente Propia.

Los diferentes módulos de la arquitectura fueron presentados en la sección 6.2.2, ya que en este caso el sistema es idéntico conceptualmente. La única variante está en el módulo 4, donde en la presente arquitectura recibirá los datos del módulo 1 para poder realizar la predicción (*STLF*); una vez realizada ésta, se enviará la información al módulo 5 para su representación visual, y almacenándola igualmente de nuevo en la base de datos a través de 1.

La Figura 6.24 muestra el esquema de funcionamiento *on-line* del predictor. Los procesos internos son aquellos realizados durante la operación de predicción. Los procesos externos son aquellos que dependen de eventos externos o con la interacción con dispositivos externos. En esta figura se puede observar que con el fin de realizar una predicción para el día *d*, la información



almacenada en Datos Históricos (*Historical Data*) se presentan a ANN, una vez ésta está ya entrenada; tras esta presentación, se hará *STLF* del día  $d$ .

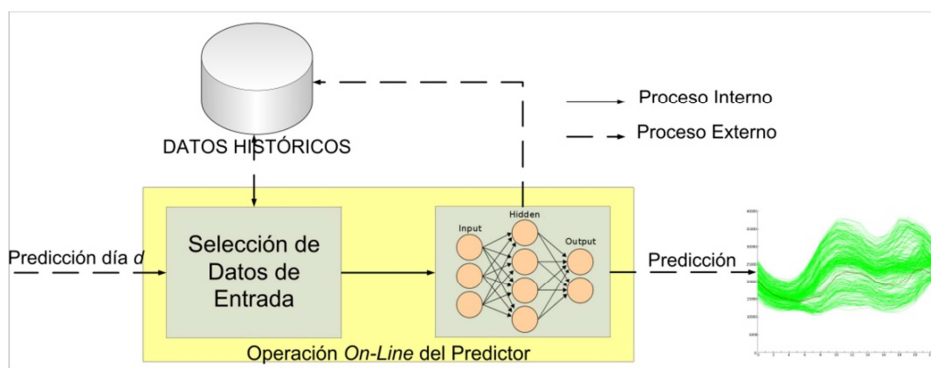


Figura 6.24. Operación *On-Line* del predictor. Fuente Propia.

### 6.3.2.2. Diseño de ANN

El sistema ANN presentado funciona sobre la hipótesis de que el patrón diario de carga eléctrica está relacionado con el patrón del día anterior y otros datos de calendario. Más específicamente:

- El consumo eléctrico es altamente dependiente de la hora del día, y de la curva de carga del día anterior. Esta curva de carga de día anterior en realidad reúne una gran cantidad de información acerca de otras condiciones (estacionalidad y climatología) como muestran Hernández *et al.* (2012a), y que no será necesaria introducir de manera explícita.
- Hay muchos modelos de predicción de la carga total del día siguiente, este es un dato de entrada muy valiosos para la ANN, ya que reúne una gran cantidad de información de manera implícita.
- Por lo tanto, la previsión de la curva carga se lleva a cabo sobre la base de la curva de carga del día anterior al día a predecir, y las variables de calendario (día de la semana, mes, etc.).
- Las variables periódicas se suministra a la red en forma de valores de senos y cosenos, ya que se ha demostrado que esta transformación mejora significativamente el rendimiento de la ANN, como muestran Drezga *et al.* (1999b). El día de la semana y el mes (esenciales para detectar patrones semanales, mensuales y estacionales) se introducen como seno y coseno, ya que las variables cíclicas son procesadas con mayor eficiencia por ANN, tal y como se muestra en (Ramezani *et al.*, 2005; Razavi and Tolson, 2011).
- Mientras que los estudios anteriores sobre los patrones de carga *-REE* (Rodríguez *et al.*, 1998)– han demostrado que el tipo de día (laborable o festivo) tiene un claro efecto sobre la demanda eléctrica, durante la fase de pruebas, se encontró que la precisión de la previsión no mejoró con la información proporcionada por el tipo de día. La razón de lo anterior podría ser que las variables de entrada utilizadas pertenecientes a la curva de carga del día anterior,

junto la previsión de carga agregada del día a predecir, son suficientes para que la red infiera las características principales del día a estimar y, por tanto, no sean necesarias entradas al modelo con la información del tipo de día.

- La carga eléctrica varía entre días laborables y fines de semana. Además, la demanda eléctrica en un día festivo presenta similitudes a la de los domingos.
- La estacionalidad de la demanda eléctrica es evidente, ya que varía considerablemente a lo largo de todo el año.

La arquitectura empleada en este estudio sigue el siguiente modelo: realizar una previsión de carga para el día  $d$ , cuando el día  $d-1$  ha terminado y los datos de ese día están disponibles, el sistema puede realizar la previsión de carga para el día  $d$ . La arquitectura implementada se muestra en la Figura 6.25, una *MLP* de tres capas: una capa de entrada, una capa oculta, y una capa de salida.

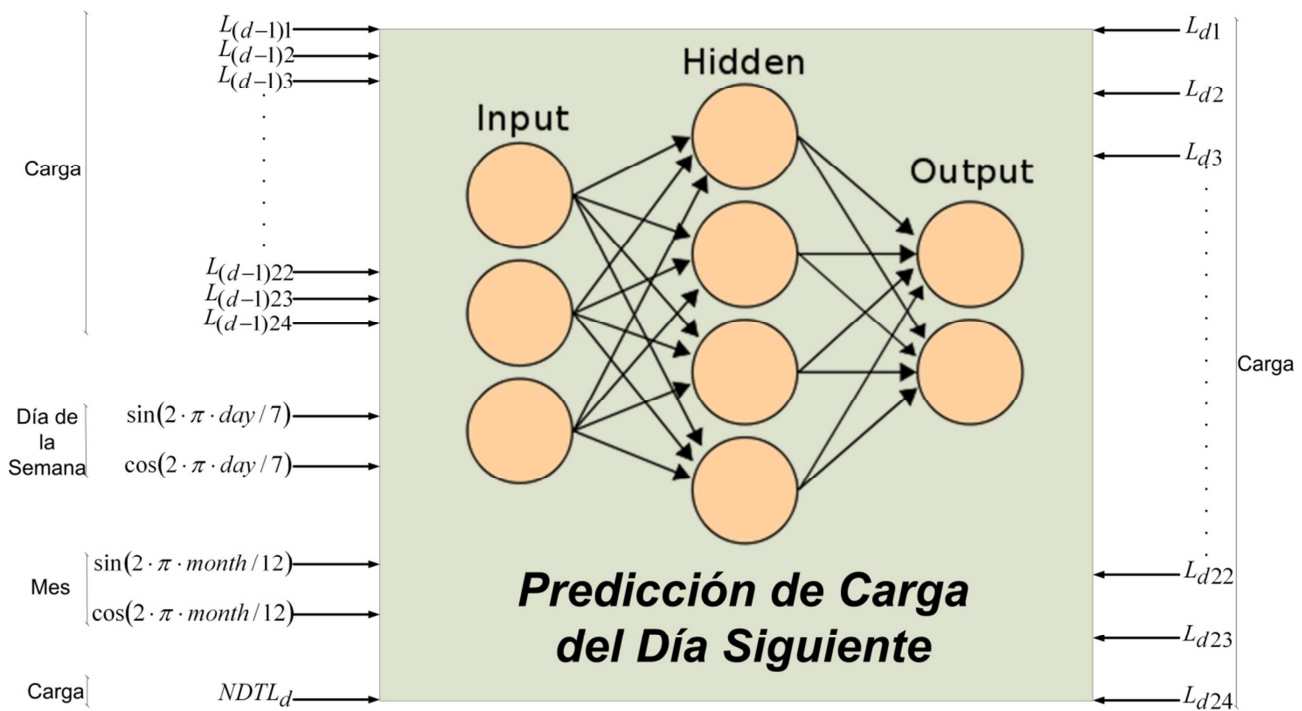


Figura 6.25. Arquitectura MLP. Se muestran las variables de entrada y salida en la figura. Fuente Propia.

Entrada:

- $L_{(d-1)1}, L_{(d-1)2}, L_{(d-1)3}, \dots, L_{(d-1)24}$ : representan los 24 valores de la curva de carga del día anterior al que se pretende predecir.
- Día de la semana  $d-1$ : esta variable se introduce como dos variables expresadas en forma de seno y coseno dado por  $\sin[(2 \cdot \pi \cdot \text{day}) / 7]_{(d-1)}$  y  $\cos[(2 \cdot \pi \cdot \text{day}) / 7]_{(d-1)}$ , con los días de 0 a 6 (domingo=0, Lunes=1, martes=2, miércoles=3, ..., sábado=6).

- Mes  $d-1$ : esta variable se introduce como dos variables expresadas en forma de *seno* y *coseno* dado por  $\sin[(2 \cdot \pi \cdot \text{day})/12]_{(d-1)}$  y  $\cos[(2 \cdot \pi \cdot \text{day})/12]_{(d-1)}$ , mes 1 a 12 (*enero=1, febrero=2, ..., noviembre=11, diciembre=12*).
- Carga Total del Día Siguiente (*Next Day's Total Load – NDTLd*): puede ser fácilmente estimada con un error de  $\pm 2\%$  empleando por ejemplo, el modelo propuesto por Hsu and Chen (2003).

*Salida:*

- $L_{d1}, L_{d2}, L_{d3}, \dots, L_{d24}$  representan los 24 valores de la curva de carga del día a predecir.

*Oculto:*

- Las neuronas de la capa oculta están completamente conectadas con las neuronas de la entrada y de la salida.
- Hay 16 neuronas en la capa oculta.

Antes de la operación, la ANN tiene que ser entrenada (con el 70% de los datos). Durante esta etapa de aprendizaje, la ANN se enfrenta a una serie de entradas acopladas con la salida esperada, es decir, un conjunto de entradas están asociadas a la curva de carga real, la cual el sistema deberá predecir. Durante este aprendizaje, los pesos internos de la ANN se ajustan para producir las salidas apropiadas.

Para la optimización de ANN, tanto para determinar el número de neuronas en la capa oculta como para establecer el mejor algoritmo de formación, normalmente se emplea un método heurístico. En nuestro caso, hemos decidido utilizar un *script* automatizado, en el cual todos los parámetros posibles se han modificado (número de neuronas en la capa oculta, función de entrenamiento, función de rendimiento de la red durante el entrenamiento, etc.), calculando el error de estimación para la prueba en varias ejecuciones, y para cada combinación de valores de los diferentes parámetros. Los mejores resultados se obtuvieron con un total de 16 neuronas en la capa oculta, la función de entrenamiento *Bayesian Regulation Backpropagation* y *Sum Squared Error* para la función de rendimiento de la red.

### 6.3.2.3. Cálculo del error

Los modelos y la precisión de la predicción fueron validados con *MAPE*, expresándose

$$MAPE = 100 \cdot \frac{\sum_{i=1}^n \left| \frac{L(i) - \hat{L}(i)}{L(i)} \right|}{n} \quad (6.7)$$

donde  $L(i)$  representa el valor medido para  $t=i$ ,  $\hat{L}(i)$  representa el valor estimado y  $n$  representa el tamaño de la muestra de prueba.

Una vez calculado (6.7) para cada uno de los días del conjunto de pruebas, se obtiene el error medio para todos los días de la fase de operación (con el 30% de los datos) por medio de

$$ERROR_{OP} = \frac{\sum_{i=1}^n MAPE_{di}}{n} \quad (6.8)$$

Para examinar la forma en que el error de la predicción se plasma en la curva de carga, el error se muestra en un gráfico, incluyendo todos los días estimados de la fase de operación, pero por horas; utilizando este método, el error medio de predicción para cada una de las 24 horas se obtiene por medio de

$$MAPE_{-h_i} = \frac{\sum_{k=1}^n MAPE_{i,k}}{n} \quad (6.9)$$

con  $i=1,2,\dots,23,24$ ;  $n$  representa el tamaño de la muestra de la fase de operación y  $MAPE_{i,k}$  el error horario  $i$  para el día  $k$ .

### 6.3.3. Resultados

Esta sección proporciona los errores por día, errores de curva en forma Función de Densidad de Probabilidad (*Probability Density Function – PDF*), los errores por hora, errores de curva por hora en forma *PDF*, y las predicciones de varios días con error promedio bajo.

#### 6.3.3.1. Resultados generales

Una vez que la red está entrenada, la predicción se lleva a cabo para el conjunto de validación (fase de operación); una predicción de curva de carga se genera para cada uno de los datos y el error promedio diario es calculado; los errores promedio se muestran en la Figura 6.26 junto con el valor medio del error de toda la fase de operación, media  $\pm$  desviación estándar y la media  $\pm 2x$  desviación estándar. El error medio de la fase de operación ha resultado con un valor de 2.4037%. La figura utiliza una nomenclatura específica con el formato “AB / C – DE”. A representa el tipo de día de la jornada anterior (2: día laborable, 1: día de fiesta), B es el número del mes del día a predecir; C es el día del mes del día a predecir, D es el día de la semana (lunes, ..., domingo) del día a predecir, y E el día tipo (2 día laborable, 1 día de fiesta) del día a predecir.

En la Figura 6.27 los errores se expresan como *PDF*, donde los intervalos entre la media y la media  $\pm$  desviación estándar, y la media y la media  $\pm 2x$  desviación estándar son mostrados. Como muestra la figura, la mayoría de los errores  $-72\%$  corresponden al primer intervalo, como se muestra en los porcentajes de la Tabla 6.5. Por encima y por debajo de ese intervalo, los errores tienen una distribución muy similar.

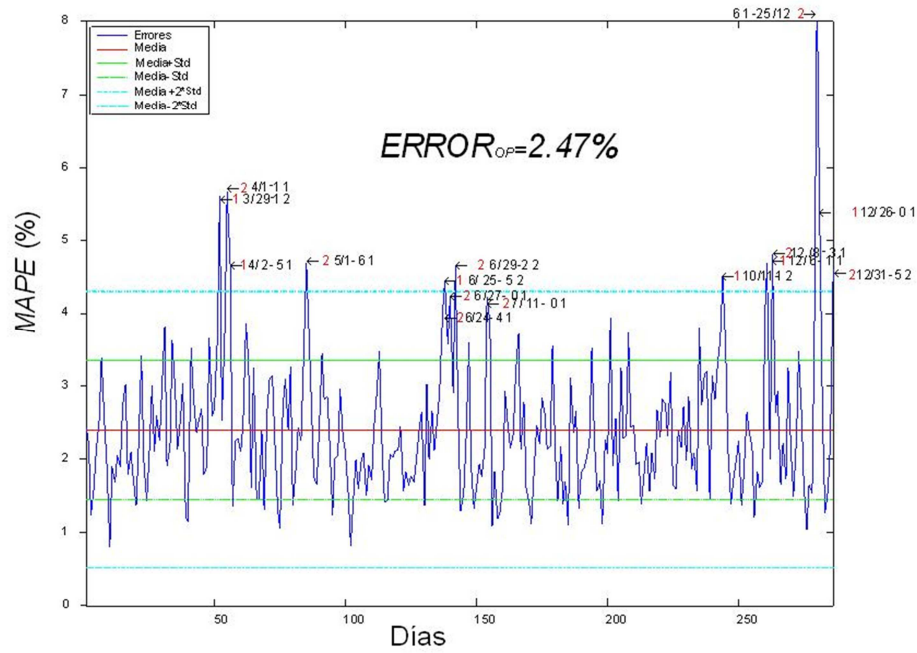


Figura 6.26. Errores por día. En x-eje son los días. En y-eje son los errores dados por (6.7). Fuente Propia.

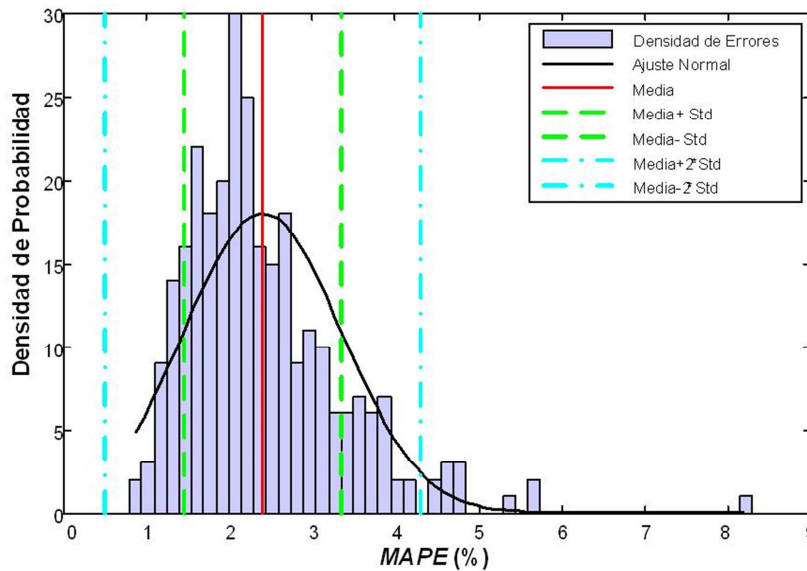
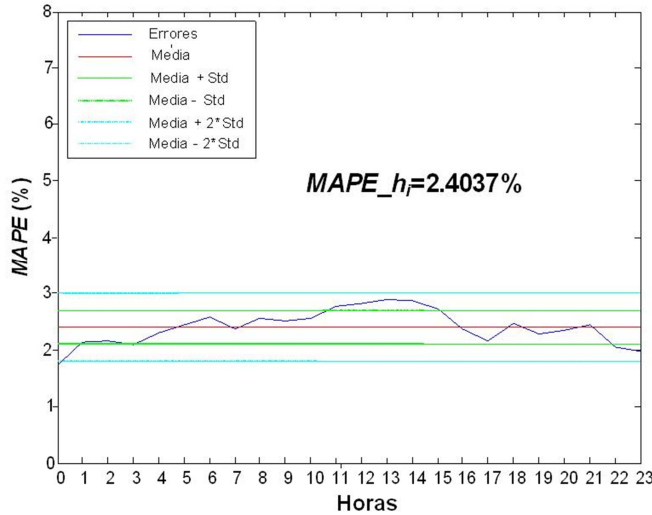


Figura 6.27. Errores de la predicción de los datos. En x-eje son los errores dados por (6.7). En y-eje son las densidades de probabilidad. Fuente Propia.

Tabla 6.5. Distribución de errores por día. Fuente Propia.

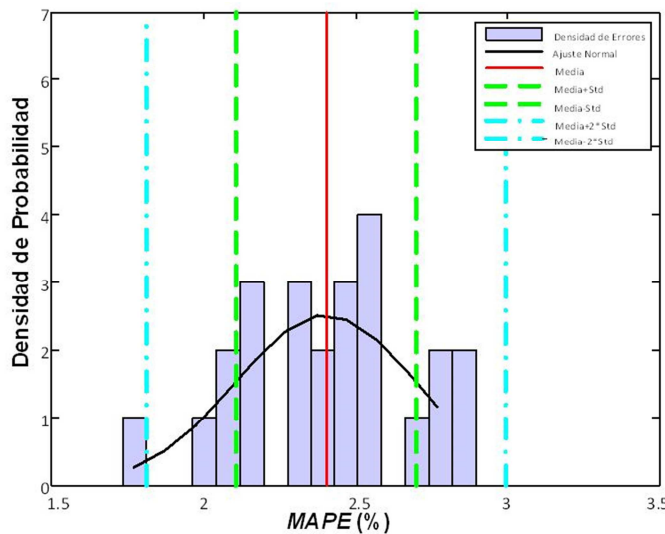
Variable	Valor	Porcentaje
Media	0.024	2.40%
Desviación estándar (Std.)	0.0095	0.95%
Nº errores por encima x1 Std.	42	14.73
Nº errores entre x1 Std.	206	72.28%
Nº errores por debajo x1 Std.	37	12.99%
Nº errores por encima x2 Std.	12	4.21%
Nº errores entre x2 Std.	273	95.79%
Nº errores por debajo x2 Std.	0	0.00%

La Figura 6.28 muestra los errores por hora obtenidos durante la fase de operación. La mayoría de los errores ocurren en regiones específicas de la curva de carga, siguiendo el mismo patrón: de 4 horas a 7 horas, donde la curva empieza a aumentar desde el primer valle de carga; desde 10h a 15h, donde la curva se eleva hasta alcanzar el primer pico de carga; la curva comienza a caer en el segundo valle de carga; desde 18h a 21h, la curva empieza a subir de nuevo hacia el segundo pico de carga, al final del día de la curva comienza a descender.



**Figura 6.28. Errores por horas.** En x-eje son las horas del día. En y-eje son los errores dados por (6.9). Fuente Propia.

La Figura 6.29 muestra los errores por hora expresada en PDF, y muestra los intervalos entre la media y la media  $\pm$  desviación estándar, y la media y la media  $\pm 2x$  desviación estándar. La mayoría de los errores se concentran en el primer intervalo  $-62\%$ – como se evidencia con los porcentajes mostrados en la Tabla 6.6. Un total del 21% y 17% de los errores se encuentran por encima y por debajo del primer intervalo respectivamente.



**Figura 6.29. Errores de la predicción de los dato por horas.** En x-eje son los errores dados por (6.9). En y-eje son las densidades de probabilidad. Fuente Propia.

Tabla 6.6. Distribución de errores por horas. Fuente Propia.

Variable	Valor	Porcentaje
Media	0.024	2.40%
Desviación estándar (Std.)	0.0030	0.30%
Nº errores por encima x1 Std.	5	20.83%
Nº errores entre x1 Std.	15	62.50%
Nº errores por debajo x1 Std.	4	16.67%
Nº errores por encima x2 Std.	0	0.00%
Nº errores entre x2 Std.	23	95.83%
Nº errores por debajo x2 Std.	1	4.17%

La Figura 6.30 muestra las predicciones de las curvas de carga de tres días con un error medio de día bajo, en donde: a) representa la previsión de 15/02/2010 con un error promedio de 1.20%; b) representa la previsión de 18/05/2010 con un error promedio de 1.10%; y c) representa 21/12/2010 con un error promedio de 1.13%. Como podemos ver, la predicción de la curva de carga coincide casi por completo con la curva de carga real.

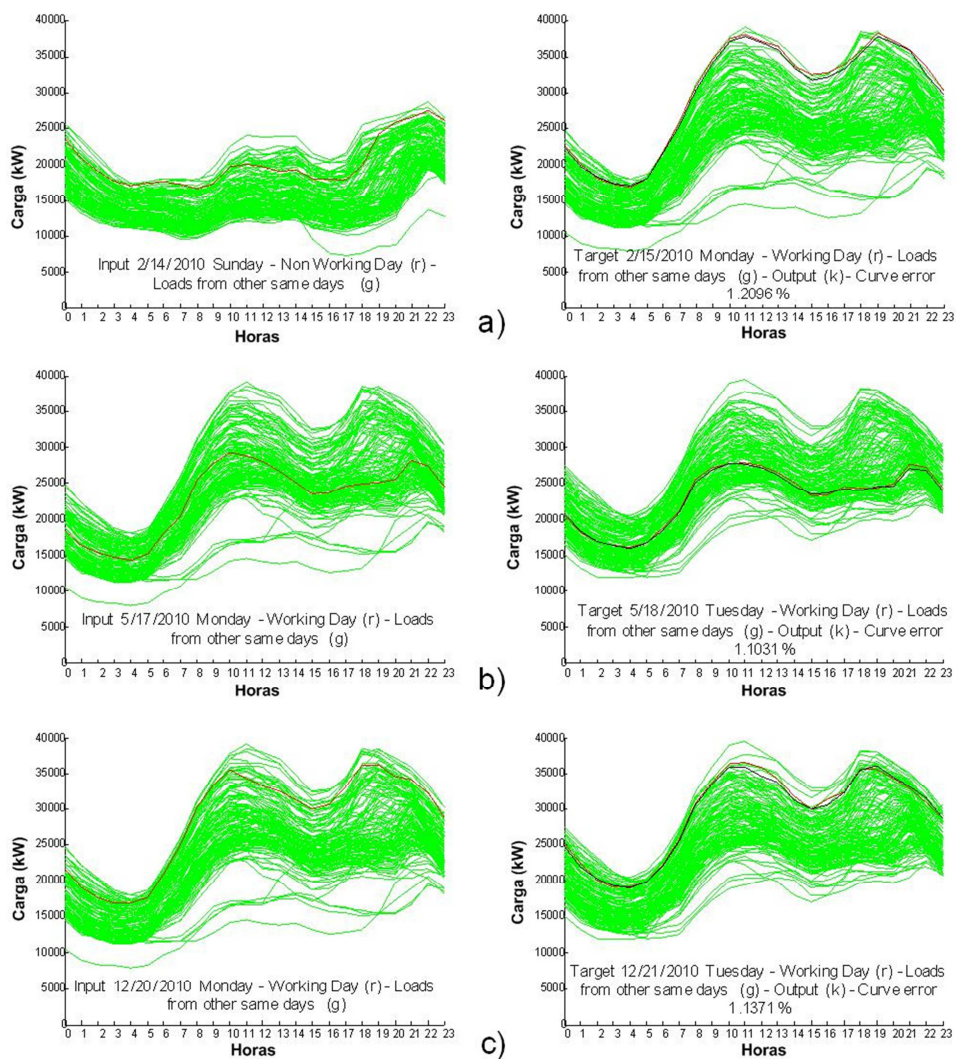


Figura 6.30. Predicciones con errores medios bajos: a) 15/02/2010; b) 18/05/2010; c) 21/12/2010. Fuente Propia.

### 6.3.3.2. *Coste computacional*

Como se indicó anteriormente, se ha empleado MatLab para implementar la ANN, así como el resto de código desarrollado para tareas adicionales (estimación del error, figuras, etc.). Se utilizó una computadora de escritorio con un procesador *Intel Core 2 vPro 3.4 GHz 2 GB de RAM*.

El método de computación es la siguiente: el conjunto de entrenamiento se importa desde la base de datos *Historical Data*, tal como se muestra en la Figura 6.23, a continuación, se inicia la fase de aprendizaje de ANN. Para este estudio, se han empleado el 70% de todos los datos disponibles y el tiempo total de cómputo para dicha fase ha sido de 16 minutos y 43.

Cuando la red está entrenada (según en la Figura 6.25), a continuación, se comienza la fase de operación para predecir la curva de carga de cada uno de los días. El equipo tardó 2 minutos y 49 segundos para procesar el total de los días a predecir, ver las curvas de carga e insertar en la base de datos. Por lo tanto, aproximadamente, el ordenador necesita 0.59 segundos para producir una predicción para un día.

### 6.3.4. *Análisis de resultados*

#### 6.3.4.1. *Distribución del error*

Como se muestra en la Figura 6.26, y apoyado por los resultados mostrados en la Figura 6.27 y la Tabla 6.5, el error medio diario está dentro del rango media  $\pm$  desviación estándar, lo que significa que los errores oscilaron entre 1.45% y 3.35%, que son unos resultados óptimos. La Figura 6.27 también muestra que los días con errores medios diarios altos –por encima de 4%– son días especiales; a continuación, se proporcionarán más detalles sobre este aspecto.

Mediante la observación de la media de error por hora, mostrada en la Figura 6.28, en la Figura 6.29 y la Tabla 6.6, se puede apreciar que los errores más importantes se producen en los puntos de inflexión de la curva de carga del pronóstico. Esta coincidencia puede sugerir que la información adicional sobre la forma de la curva a predecir se podría emplear, y así mejorar las previsiones y mejorar las predicciones con errores elevados.

#### 6.3.4.2. *Errores por día de la semana y mes*

La Figura 6.31 representa la evolución del error medio diario, agrupado por día de la semana. La razón por la que los mayores errores medios se producen los viernes, sábados y domingos, es porque el conjunto de entrenamiento (curva de carga) de dichos días tiene una mayor dispersión (curvas poco parecidas), y como resultado, la incertidumbre es mayor en los ajustes de los pesos después del entrenamiento, y por tanto, aumentan los errores. Además, en los sábados y los domingos, las curvas de carga son significativamente diferentes –tanto valor del consumo como la



forma— de las de los demás días de la semana. Los viernes son también días especiales, ya que marca el comienzo del fin de semana y la demanda de energía eléctrica es menor que en el resto de los días.

La Figura 6.32 muestra la evolución de los errores medios diarios, agrupados por meses. Octubre y noviembre son menos días debido a los valores extremos eliminados (outliers), el error medio diario agrupado por mes oscila entre el 2 y el 3%, lo que evidencia una buena exactitud en los pronósticos.

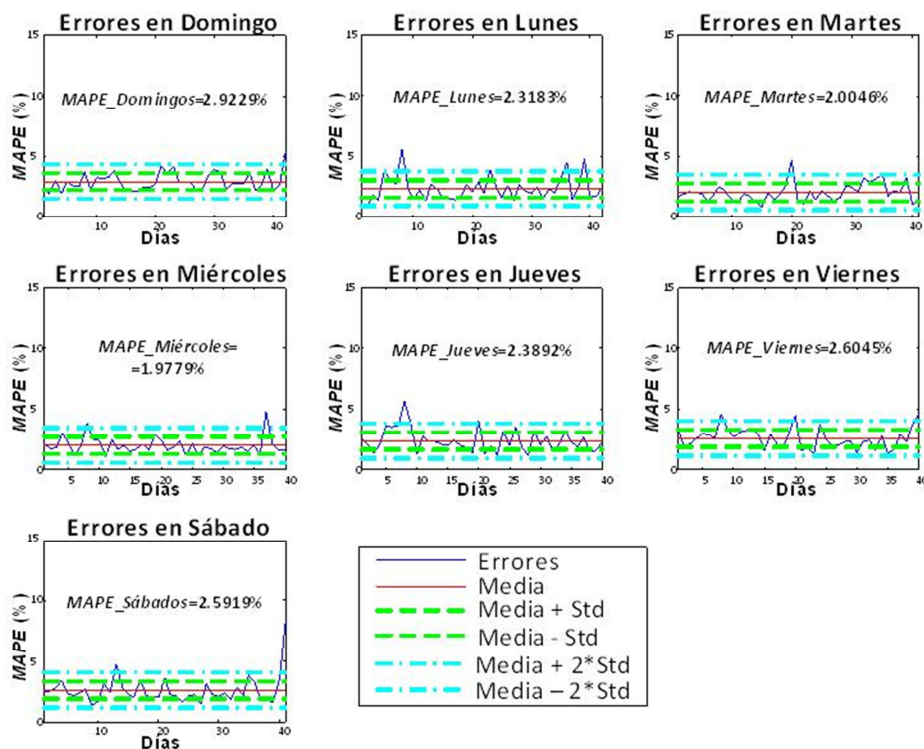


Figura 6.31. Errores por día agrupados por día de la semana. En x-eje son los errores dados por (6.7). En y-eje son cada uno de los días predichos. Fuente Propia.

### 6.3.4.3. Análisis de error

El propósito de esta sección es presentar los errores de predicción más significativos y tratar de analizar la razón de dichos errores. Para finalizar, esta sección resume las conclusiones extraídas de esta experiencia.

La predicción para el 02/04/2010, con un error promedio de 4.34%. Este día es Viernes Santo (Semana Santa), y el día anterior también es una fiesta nacional y, en consecuencia, existen pocos pares de patrones con las mismas características, los cuales habrán sido empleados en la fase de aprendizaje. Por este motivo, la influencia de pares similares en el ajuste de los pesos neuronales en la fase de aprendizaje habrá sido muy reducida.

La predicción para el 01/05/2010, con un error promedio de 4.77%. La curva de carga en este sábado festivo es similar a la de los sábados laborables del mismo mes; en comparación con el día anterior, el viernes anterior al sábado presenta la mitad de la carga; además, la forma de la curva es

irregular, especialmente en el origen de la curva, sin embargo, hacia el final, la carga es similar a los otros viernes del mismo mes.

La predicción para el 24/06/2010, con un error promedio de 3.92%. Ese jueves es un día perteneciente a las fiestas locales (aunque no es festivo el jueves), llamado Jueves la Saca, siendo la carga claramente inferior a la de otro día similar (jueves) y, además, el miércoles (laborable) anterior al jueves es un día de la festividad local, presentando un comportamiento de curva de carga anormal para ese tipo de días.

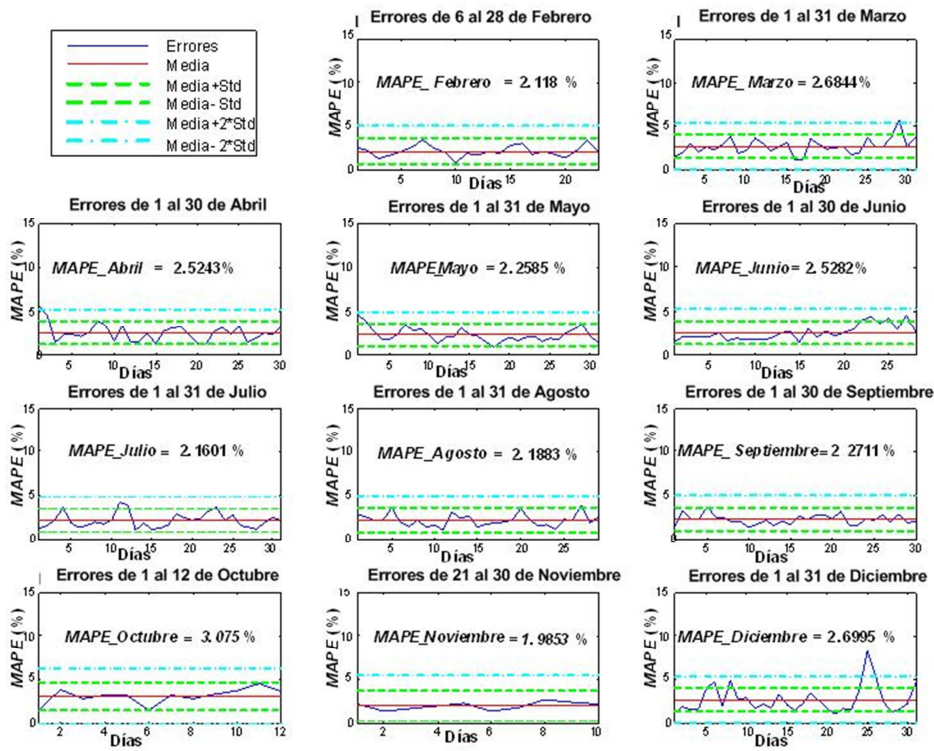


Figura 6.32. Errores por día agrupados por meses. En x-eje son los errores dados por (6.7). En y-eje son cada uno de los días predichos del correspondiente mes. Fuente Propia.

La predicción para el 08/12/2010 con un error medio del 4.73%. Ese día es un miércoles festivo donde la curva de carga es menor que en los miércoles laborables del mismo mes; sólo había un miércoles similar en 2008, sin embargo, el modelo ANN predice un aumento de la demanda y la curva comienza a subir antes de la curva real, causando un error. El martes antes del miércoles festivo es más bajo que el resto de martes del mismo mes, por esta razón, la previsión de la curva de carga comienza a descender antes de tiempo, siendo el final de la curva atípica, y es mucho más baja que la curva de carga real.

La predicción para el 25/12/2010, con un error promedio de 8.04%. Ese día es un sábado de Navidad, donde la carga es menor que en otros sábados del mismo mes, y el primer pico se produce más tarde de lo habitual. El día anterior, es Nochebuena y es laborable, pero su curva de carga es inferior a la curva de carga media para todo el mes e incluso para años anteriores. Aunque la víspera

de Navidad no es un día festivo, la demanda es mucho menor que en días normales laborables, lo que conduce a errores de predicción significativos.

La predicción para el 31/12/2010 con un error promedio de 4.78%. Ese día es la víspera de Año Nuevo, y fue un viernes laborable, año en que sólo había dos viernes laborables con una curva de carga similar; esa curva del viernes es inferior a la de otros viernes laborables y ligeramente superior a la de los domingos del mismo mes. Este día presenta un perfil bajo en comparación con otros jueves del mismo mes, y presenta una forma atípica entre las 11h y 16h, todos estos factores han hecho que el error de la predicción aumentara.

#### 6.3.4.4. Dependencia del error de la fase de operación con el número de patrones de la fase de aprendizaje

La asociación entre el error medio obtenido durante la fase de operación, y el número de patrones presentados a ANN durante la fase de aprendizaje se va a analizar. A tal efecto, diferentes números de patrones se presentaron al mismo modelo durante la fase de aprendizaje: inicialmente, 150 patrones se introdujeron en el modelo, para posteriormente incrementar gradualmente (en pasos de 50 patrones) el número de patrones hasta alcanzar los 700 patrones. Una vez el modelo entrenado, se ejecutó la misma fase de operación para todos ellos. La arquitectura óptima para cada red se consiguió de la manera siguiente: en primer lugar, se empleó un *script* para probar todas las funciones de entrenamiento y el rendimiento de la fase de aprendizaje; a continuación, y desde 3 neuronas, se fue aumentando el número de las mismas, repitiendo todas las posibles combinaciones de los dos anteriores parámetros citados. La razón para el ajuste de cada modelo al presentar más patrones, es que la arquitectura del modelo es totalmente dependiente del número de patrones de entrada, como se muestra en la Tabla 6.7.

**Tabla 6.7. Relación entre el error de la fase de operación y el número de patrones de la fase de aprendizaje.**

Fuente Propia.

Número de red	Nº de patrones	Nº de neuronas	Error fase operación
1	150	3	4.78%
2	200	4	4.18%
3	250	4	3.94%
4	300	4	3.48%
5	350	6	3.26%
6	400	6	3.04%
7	450	12	2.82%
8	500	12	2.65%
9	550	13	2.62%
10	600	13	2.58%
11	650	15	2.48%
12	700	16	2.41%

Los datos anteriores se introdujeron en MatLab, ajustando los valores de los errores de cada modelo mediante un polinomio cúbico, siendo el resultado

$$Y = -1e^{-0.08} \cdot X^3 + 2.2e^{-0.05} \cdot X^2 - 0.017 \cdot X + 6.8 \quad (6.10)$$

donde  $Y$  es el error medio de la fase de operación, y  $X$  es el número de patrones que se utilizan en la fase de aprendizaje. La asociación entre el error medio y el número de patrones se evidencia en la Figura 6.33. La figura evidencia que la arquitectura no puede mejorar aún más al aumentar el número de patrones en la fase de aprendizaje, a partir de determinado número de patrones. La ANN habrá llegado entonces a su fase de madurez, con patrones adicionales no parece mejorar el error medio de la fase de operación. Vale la pena señalar que el error medio de los 730 días en el conjunto de pruebas era 2.40%. Por medio de (6.10), se puede estimar que al utilizar 750 y 800 patrones obtendríamos un error promedio en la fase de operación de 2.38% y 2.33% respectivamente; la evolución del error de la fase de operación puede verse en la Tabla 6.8, donde se ha tenido en cuenta el incremento al pasar de una red a otra. Para evaluar qué error se ha mejorado mediante el aumento del número de neuronas, se utilizó la siguiente

$$\Delta error_{networks\_patterns} = \frac{|error_i - error_j|}{patterns_i - patterns_j} \quad (6.11)$$

donde  $i = 2, \dots, 12$ ;  $j = 1, \dots, 11$ ; con los datos de la Tabla 6.8  $error_i$ ,  $error_j$ ,  $patterns_i$  and  $patterns_j$ . Altos valores obtenidos por (6.11) sugieren una mejora significativa del error con respecto al número de patrones.

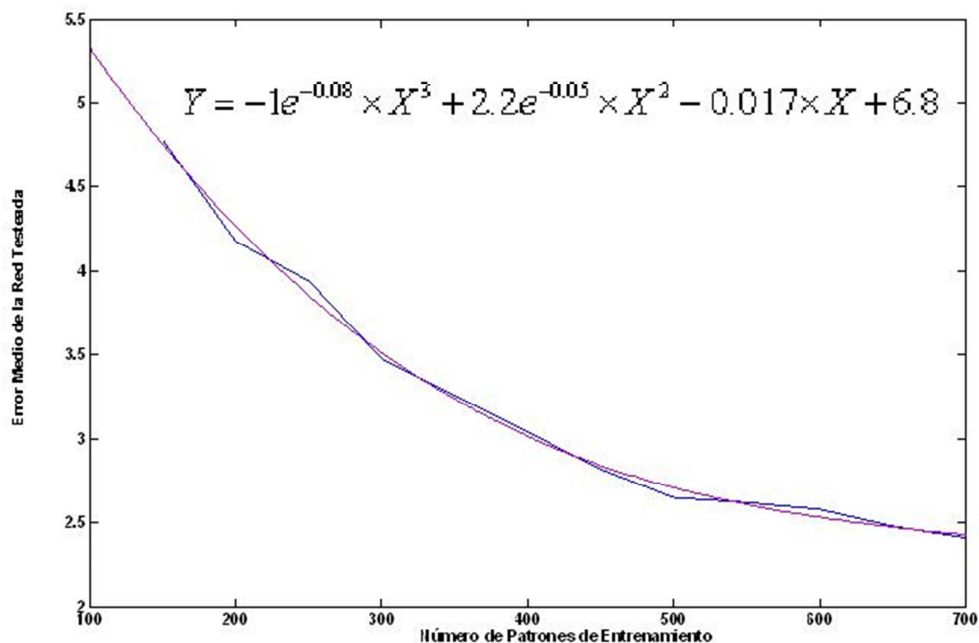


Figura 6.33. Evolución del error medio de la fase de operación con respecto al número de patrones empleados en la fase de aprendizaje. La curva ajusta está en color rojo y la curva real en color azul. Fuente Propia.

Tabla 6.8. Variación del error entre las diferentes redes con respecto al número de patrones. Fuente Propia.

Red $i-j$	Patrones añadidos	Indicador dado por (6.11)
2-1	50	0.01199
3-2	50	0.00478
4-3	50	0.00428
5-4	50	0.00428
6-5	50	0.00455
7-6	50	0.00440
8-7	50	0.00340
9-8	50	0.00060
10-9	50	0.00080
11-10	50	0.00200
12-11	50	0.00140

#### 6.3.4.5. Comparación con otros resultados

En principio, los resultados de este trabajo pueden ser comparados directamente con otros métodos de predicción de la curva de carga validados en entornos de *microgrids*. Al igual que en (Chan *et al.*, 2011), el trabajo aquí presentado se basa en un modelo de predicción *MLP*. El enfoque seguido en este trabajo emplea curvas de carga de días anteriores ( $d-1$ ) como entrada del modelo para predecir las curvas de carga de los días ( $d$ ) en la salida, lo que permite una íntima relación entre la entrada y la salida, y un ajuste interno de los pesos más eficiente que los obtenidos en los modelos de Chan *et al.* ya citados, los cuales utilizan como entradas grupos de curvas de hasta tres días antes del día de pronosticar. Esto explicará los mejores resultados obtenidos con el presente estudio, con un *MAPE* de entre 2% y 5% frente a cifras del orden de 15% de aquél.

Por último, es importante mencionar que la mayoría de las obras de la literatura sólo estudian *MAPE* diarios. En este estudio, sin embargo, se ha presentado el *MAPE* detallado hora por hora, para poder obtener conclusiones de interés.

#### 6.3.5. Conclusiones y estudios futuros

En este estudio se propone un modelo basado en *ANN* para *STLF*, en entornos desagregados (ciudad pequeña, entorno rural o *microgrids*), con un sencillo *MLP* basado en la arquitectura. A tal efecto, las variables relevantes de la entrada han sido seleccionadas con el fin de minimizar los errores de la predicción. Como se ha comentado anteriormente, la previsión al desagregar resulta más compleja, debido al aumento de la variabilidad de las curvas de carga, como consecuencia de la desagregación. Además, una predicción exacta en este tipo de entornos dependerá de las variables empleadas y la forma en que se introducen en la *ANN*.

También se ha mostrado numéricamente que existe una estrecha relación entre los errores de predicción y el número de patrones presentados en la fase de aprendizaje, por lo que es necesario seleccionar cuidadosamente los datos a emplear.

Finalmente, se ha demostrado que el concepto de predicción de carga mediante *ANN* son aplicables en entornos desagregados, del tipo de ciudad pequeña o *microgrid*, obteniéndose muy buenos resultados, y mostrando que podrían alcanzarse un error total de la fase de operación en torno a 2.4%.

#### **6.4. SHORT-TERM LOAD FORECASTING PARA ENTORNOS DESAGREGADOS BASADO EN ARTIFICIAL NEURAL NETWORKS. LA FUERZA DE LA CLASIFICACIÓN Y CLUSTERIZADO**

Una vez presentados los resultados de la sección 6.3, y teniendo en cuenta lo aportado en la sección 6.2, donde por medio de un sistema en cascada de *SOM* y algoritmo *k-means*, se construye una herramienta selectora del clúster a asignar a un día del que se pretende hacer *STLF*, se ha planteado presentar un modelo para *STLF* para entornos desagregados (entorno rural, ciudad pequeña o *microgrid*), con los mismos datos y el mismo emplazamiento que los presentados en la sección 6.3, pero realizando un reconocimiento previo mediante *SOM* y un clusterizado posterior con *k-means*, y realizar tantos modelos *MLPs* para *STLF* como clústers resulten.

Los resultados obtenidos podrán ser directamente comparables con los presentados en la sección 6.3; además, se realizará el mismo estudio de la evolución del error de la fase de operación para cada una de las *MLPs* resultantes frente al número de patrones de la fase de aprendizaje. En este caso, el objetivo del estudio será ver la posible mejora del error medio del modelo si van incorporándose futuros patrones a cada uno de los clústers resultantes.

##### **6.4.1. Introducción sobre el estudio**

La estrategia presentada en este estudio consistirá en aprovechar la clasificación y posterior clusterizado de las curvas de carga, para posteriormente entrenar tantos modelos *MLPs* para *STLF* como clústers se obtengan. La finalidad es que en cada clúster existan patrones de curvas de carga similares entre ellos, con la idea de que cada *MLP* se especialice en dichas curvas, y que el ajuste posterior, consecuencia de *STLF*, sea más preciso.

Se pretende comparar los resultados con la sección 6.3, ya que el emplazamiento seleccionado y el periodo de datos es similar. Como la eficiencia de los modelos *ANN* depende en gran medida de la cantidad de patrones en la fase de aprendizaje, y con el objetivo de comparar resultados similares, se realizará un estudio de la evolución del error de la fase de operación con el número de patrones de la fase de aprendizaje. En el presente estudio, cada clúster que se obtenga

tendrá un número de patrones para su fase de aprendizaje (de la *MLP*) muy inferior a la disponible en la sección 6.3 (el total de los datos de la fase de aprendizaje), es por esto, que se presentará dicha dependencia entre el número de patrones en la fase de aprendizaje y el error medio en la fase de operación/validación del modelo, para de esta forma poder inferir conclusiones acerca de la eficiencia del modelo presentado en esta sección frente al de 6.3, donde no había clustering.

### 6.4.2. Arquitectura del modelo

El modelo desarrollado se compone de tres etapas, tal y como se muestra en la Figura 6.34. La primera etapa utiliza los datos históricos (calendario y el consumo de electricidad) para realizar un reconocimiento de patrones de las curvas de carga diferentes utilizando *SOM*. En la segunda etapa, los resultados de *SOM* son clusterizados utilizando el algoritmo *k-means*, con el fin de obtener una clasificación neural óptima. La partición generada a partir de estas dos etapas se estudia, con el fin de diseñar un selector de clúster, lo que permite determinar a qué clúster pertenece un día específico de acuerdo a sus parámetros de calendario (lunes o viernes, día de fiesta o no, etc.); por ejemplo, un clúster determinado de días con bajo consumo es más probable que contenga domingos y festivos, entonces, el selector de clúster está diseñado para asignar los domingos y festivos a este grupo. En la tercera etapa, una *MLP* dedicada para cada clúster es entrenada, para posteriormente predecir correctamente la curva de carga (*STLF*). En la fase de operación (una vez que las redes han sido entrenadas), el selector de clúster decide el clúster apropiado para el día a predecir, por lo que la *MLP* del clúster escogido se ejecuta, y así obtener la predicción.

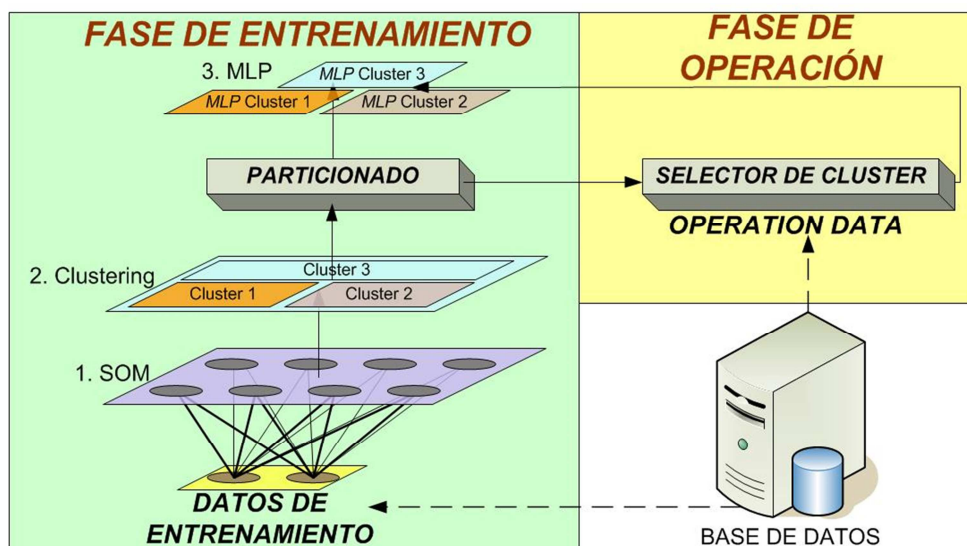


Figura 6.34. Modelo general. Fuente Propia.

Las principales características de *SOM* en este estudio son: dimensión *SOM* es  $5 \times 5$  (25 neuronas), que permite una buena dispersión de los datos; la forma del mapa es hexagonal; la

función de vecindad empleada es *gaussiana*; el vector de inicialización es *linear*; el algoritmo de aprendizaje es del tipo *batch*; y el número de entradas de la red es 27. Las variables de entrada de *SOM* son: mes de cada patrón de entrada (*enero = 1, ..., diciembre = 12*); día de la semana de cada patrón de entrada (*domingo = 0, ..., sábado = 6*); laborabilidad de cada entrada patrón (*festivo=1* y *laborable = 2*); y 24 valores de consumo de electricidad por hora, lo que representa la curva de carga diaria.

El resultado del reconocimiento de patrones realizado por *SOM* se utiliza como entrada para el algoritmo *k-means*, cuyos parámetros de configuración son los siguientes: número máximo de clústers, número máximo de pruebas y algoritmo *k-means* de tipo *batch*. El clustering se realiza de forma desatendida.

La arquitectura *MLP* muestra en la Figura 6.35. Las variables en la capa entrada y en la capa de salida, para cada una de las redes de cada uno de los clústers, serán las mismas que las presentadas en la sección 6.3.2.2.

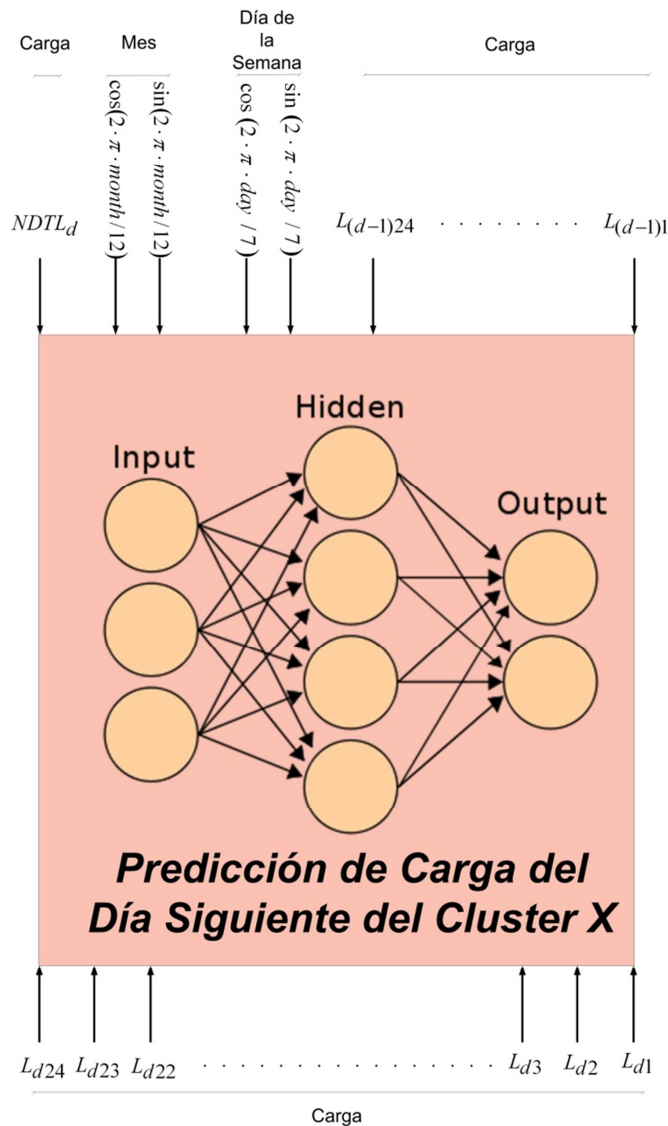


Figura 6.35. Arquitectura *MLP* para cada uno de los clústers. Fuente Propia.



Para optimizar la topología de cada *MLP*, se utilizó un *script* variando todos los parámetros posibles, obteniendo la configuración óptima siguiente: la función de entrenamiento *Bayesian Backpropagation Regulation*; función de rendimiento de la red *Sum Squared Error*; el número de entradas en la red fue de 29; y el número de neuronas de la capa oculta fue de 8, 6 y 10 para el clúster 1, 2 y 3 respectivamente.

El esquema de operación *on-line* del predictor se muestra en la Figura 6.36, y es empleado para predecir el día  $d$ . El predictor selecciona los datos de entrada referidos al día  $d-1$ . En paralelo, el selector de clúster decide a qué clúster pertenece el día  $d$ . Después de esta decisión, la *MLP* del clúster seleccionado se ejecuta junto con los datos seleccionados por el predictor y la estimación de  $NDTL_d$ . Durante la fase de operación, donde la predicción del día  $d$  se obtiene, se vuelve a emplear la fórmula (6.7) para el control del error medio del día. Una vez la fase de operación terminada, y predichas la totalidad de curvas de carga de todos los días, se emplea la fórmula (6.8) para obtener el error medio de la predicción de toda la fase de operación. Para el control del error horario durante la fase de operación, se vuelve a emplear la fórmula (6.9).

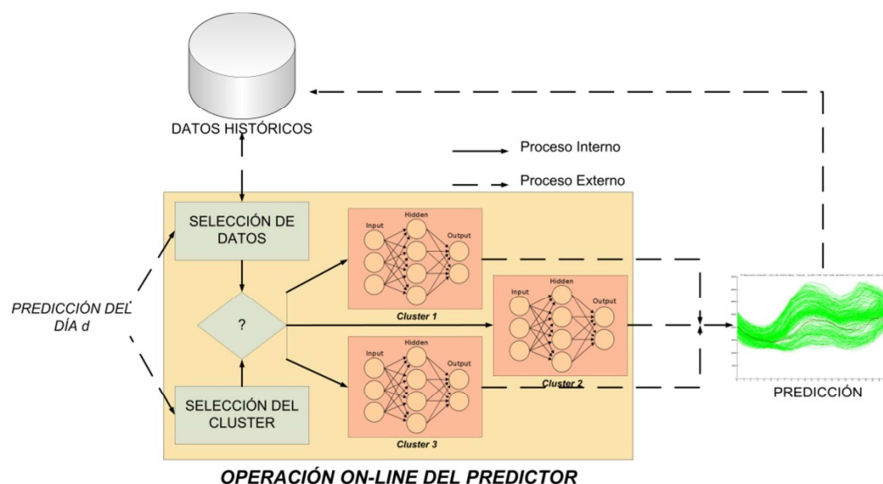


Figura 6.36. Operación *On-Line* del predictor. Fuente Propia.

### 6.4.3. Resultados

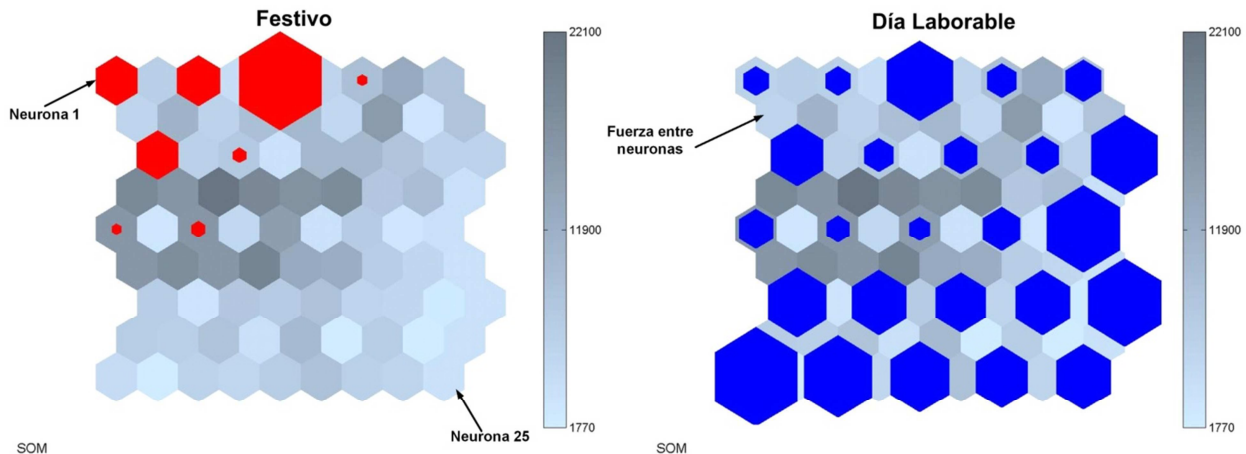
A lo largo de esta sección, se presentarán los resultados de la fase de operación. El 70% de los datos (sin outliers) se utilizan para el reconocimiento de patrones por medio de *SOM*; el resultado se clusterizará por medio del algoritmo *k-means*, y finalmente, los datos que pertenecen a cada clúster se emplearán para la fase de aprendizaje de una *MLP* para cada clúster. El restante 30% de los datos se usará para obtener la predicción de las curvas de carga en la fase de operación.

#### 6.4.3.1. Reconocimiento de patrones con *SOM*

Después de un análisis preliminar, se encontraron los siguientes resultados: en cuanto al tipo de día, las vacaciones se separan de las jornadas de trabajo; refiriéndose a los meses, los patrones de

los meses con estacionalidad similar se agrupan en las neuronas proximales; con respecto a los días de la semana, los patrones vinculados a sábados y domingos se agrupan en las neuronas cercanas, mientras que el resto de los días lo hacen de una manera diferente.

Este análisis visual previo, se realizará de manera desatendida posteriormente, una vez se realice la fase de clustering. La Figura 6.37 muestra la forma en que se realiza la activación de las neuronas (patrones de sábado y domingo), encontrándose todos esos patrones en un espacio específico, mientras que en el resto de días se encontrarán en otro espacio complementario al mostrado.



**Figura 6.37. Mapa de activación para laborabilidad. El mapa representa el significado de cada neurona: número de neurona y fuerza entre neuronas. Fuente Propia.**

#### 6.4.3.2. Clustering con *k-means*

Los resultados obtenidos por las diferentes evaluaciones de los clústers con el algoritmo *k-means*, muestran que el número óptimo de clústers es tres.

Una vez que se hace el clustering, recuerda al análisis visual previo que se hizo en la sección 6.4.3.1 tras *SOM*.

Las particularidades de estos clústers, incluyendo el número de neuronas integradas en cada uno de ellos, se indican a continuación:

- Clúster 1: días laborables de enero, febrero, marzo, abril, noviembre, diciembre y octubre (del 15 al 31). Las neuronas: 3, 4, 5, 8, 9, 10, 13, 14, 15 y 20.
- Clúster 2: sábados, domingos y festivos. Neuronas: 1, 2, 6, 7, 11, 12 y 16.
- Clúster 3: días laborables de mayo, junio, julio, agosto, septiembre y octubre (del 1 al 14). Las neuronas: 17, 18, 19, 21, 22, 23, 24 y 25.

La Figura 6.38 muestra las curvas que pertenecen a cada uno de los tres clústers, es fácil identificar la gran similitud entre las curvas de carga en el mismo clúster, y la diferencia apreciable cuando se comparan con las de los otros clústers.

En el anexo E se entregan las figuras con la representación de los patrones de carga que pertenecen a cada una de las neuronas (tras *SOM*). Analizando figura a figura se puede observar el correcto clusterizado, al poder observar como los patrones de las curvas de carga de cada neurona del mismo clúster se asemejan a la perfección.

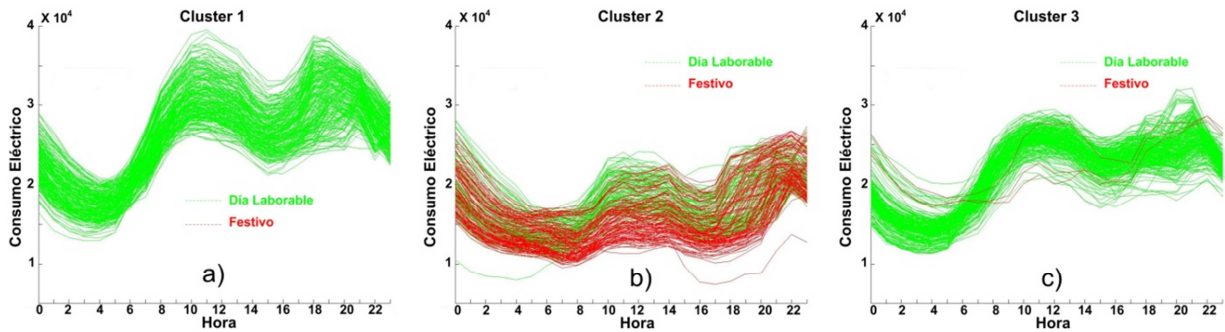


Figura 6.38. Curvas de carga de los tres clústers: a) clúster 1; b) clúster 2; c) clúster 3. Fuente Propia.

### 6.4.3.3. Resultados generales

Una vez que las tres redes *MLP* han sido entrenadas con el conjunto de patrones correspondientes a cada clúster, la fase de operación se lleva a cabo. Los días de predicción (*d*) pasar a través del selector de clúster, y se decide a qué grupo pertenece *d*.

Por cada día a predecir se obtiene la curva de carga, y su error medio del día se calcula con (6.7). Por ejemplo, en la Figura 6.39, se representa la evolución de los errores de los días predichos del clúster 1, con la media, media  $\pm$  desviación estándar y media  $\pm 2x$  desviación estándar. En el eje de ordenadas, se muestra el error por unidad y el eje de abscisas representa el día de la predicción de la fase de operación. Se emplea la misma configuración de las etiquetas que en la sección 6.3. La media de error de toda la fase de operación para cada clúster se obtiene usando (6.8), siendo 2.7615% para el clúster 1, 3.1755% para el clúster 2 y 2.7159% para el clúster 3.

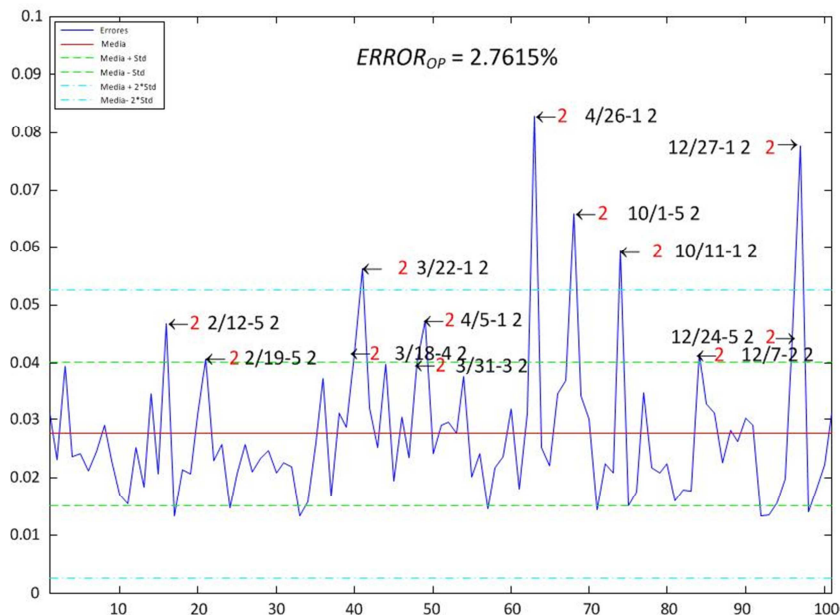


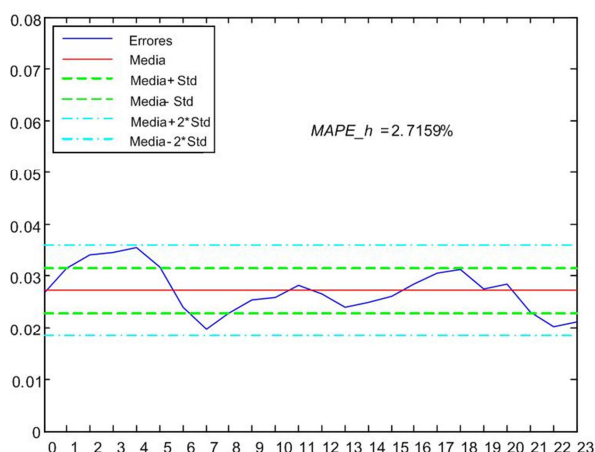
Figura 6.39. Errores por día en clúster 1. Fuente Propia.

La mayoría de los errores se producen en el primer intervalo, tal como se muestra en la Tabla 6.9, en la que se representa el número de días con error en los siguientes tramos: media y media  $\pm$  desviación estándar; media y media  $\pm 2x$  desviación estándar; por encima de la desviación estándar y por encima de la  $2x$  desviación estándar.

La Figura 6.40 muestra los errores por horas calculadas con (6.9), para el clúster 3, el mayor porcentaje de error se muestra en áreas concretas de la curva de carga (similar análisis que en la sección 6.3).

**Tabla 6.9. Distribución de errores por días en cada clúster. Fuente Propia.**

CLÚSTER 1		CLÚSTER 2		CLÚSTER 3	
Variable	%	Variable	%	Variable	%
Media	2.76	Media	3.17	Media	2.71
Des. Std.	1.25	Des. Std.	1.12	Des. Std.	1.18
Sobre x1 Std.	11.88	Sobre x1 Std.	13.97	Sobre x1 Std.	13.97
Entre x1 Std.	81.18	Entre x1 Std.	78.49	Entre x1 Std.	76.34
Bajo x1 Std.	6.69	Bajo x1 Std.	7.52	Bajo x1 Std.	9.67
Sobre x2 Std.	4.95	Sobre x2 Std.	3.22	Sobre x2 Std.	5.37
Entre x2 Std.	95.04	Entre x2 Std.	96.77	Entre x2 Std.	94.62
Bajo x2 Std.	0.00	Bajo x2 Std.	0.00	Bajo x2 Std.	0.00

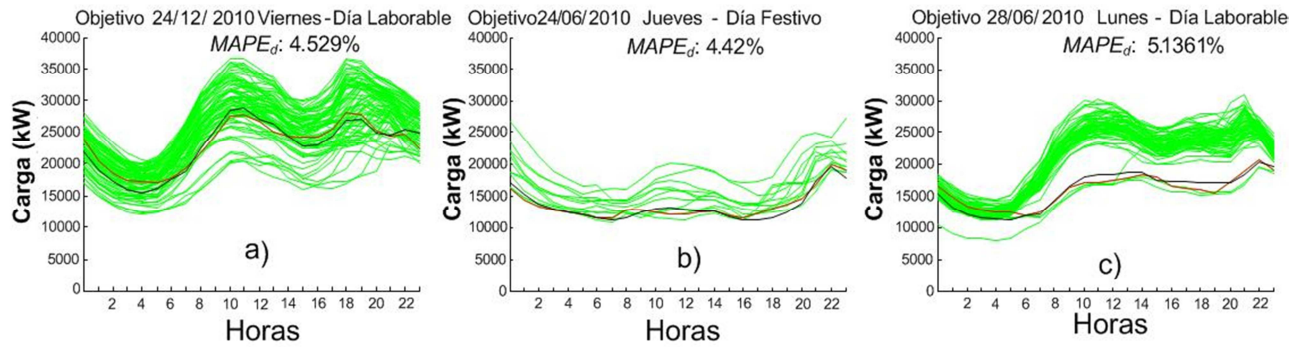


**Figura 6.40. Errores por hora en clúster 3. Fuente Propia.**

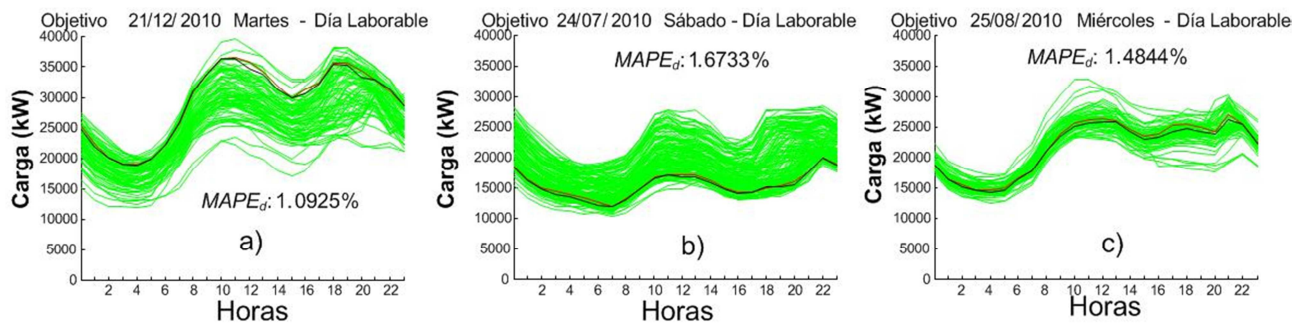
La Figura 6.41 representa las curvas de predicción de tres días con un alto porcentaje de error, cada uno perteneciente a un clúster diferente, donde: a) 24/12/2010 (clúster 1) con un error de 4.529%; b) 24/06/2010 (clúster 2) con un error de 4.42%; y c) 28/06/2010 (clúster 3) con un error de 5.1361%. La curva roja en la figura muestra la curva real (la curva que quiere ser predicha), la curva negra es la predicción y las curvas verdes representan todas las curvas similares a las del día que se pretende predecir (el mismo día de la semana y laborabilidad).

Por otra parte, la Figura 6.42 muestra las curvas de predicción de tres días con un bajo porcentaje de error (cada uno perteneciente a un clúster diferente), donde: a) 21/12/2010 (clúster 1)

muestra un error de 1.0925%; b) 24/07/2010 (clúster 2) con un error de 1.6733%; y c) 25/08/2010 (clúster 3) con un error de 1.4844%.



**Figura 6.41. Predicciones con error elevado: a) clúster 1: 24/12/2010; b) clúster 2: 24/06/2010; c) clúster 3: 28/06/2010. Fuente Propia.**



**Figura 6.42. Predicciones con error bajo: a) clúster 1: 21/12/2010; b) clúster 2: 24/07/2010; c) clúster 3: 25/08/2010. Fuente Propia.**

## 6.4.4. Análisis de resultados

### 6.4.4.1. Distribución de errores

Como se ve en la Figura 6.39 y la Tabla 6.9, la mayoría de los errores medios del día se incluyen en el rango  $\text{media} \pm \text{desviación estándar}$ , que corresponde a los errores entre 4.01% – 1.51% para el clúster 1, el 4.29% – 2.05% para el clúster 2 y 3.89% – 1.53% para el clúster 3. Estos rangos se pueden considerar aceptables.

Como se ve en la Figura 6.40, el error medio por hora muestra que las desviaciones más altas corresponden a áreas particulares de la curva de carga, y es para todos los grupos. Por lo tanto, es posible utilizar parte de la información de la curva de carga a predecir, en relación con las zonas antes mencionadas, que permite una predicción más precisa.

### 6.4.4.2. Errores por día de la semana y mes

En las Figuras 6.43, 6.44 y 6.45 se muestra la evolución del error medio por día en cada clúster (agrupado por día de la semana). Es necesario destacar que en el clúster 2 (sábados, domingos y festivos), los festivos (no fin de semana) representan un número muy reducido de días,

por eso no se ha incluido en la figura. En todos los clústers, todos los viernes (comienzo del fin de semana) tienen un comportamiento con menos consumo que el resto de los días laborables (de lunes a jueves), por esta razón, su error medio resulta ligeramente superior en promedio a la del resto de los días.

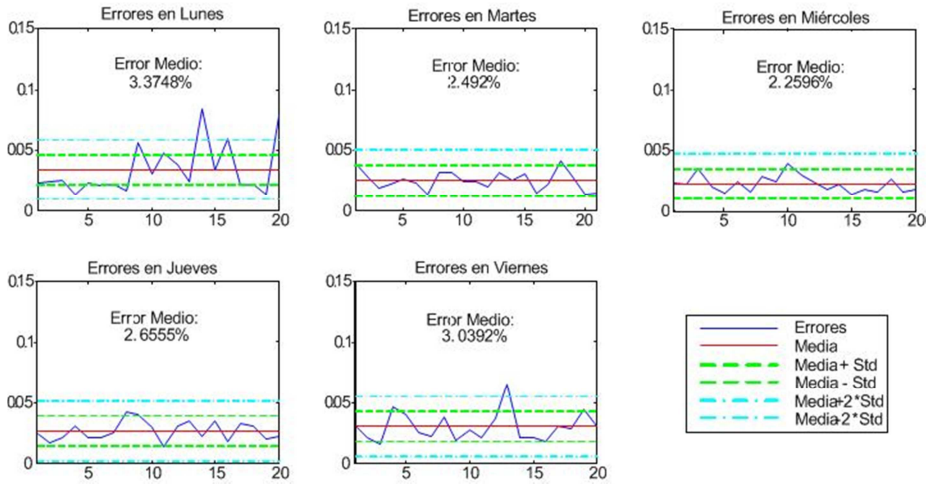


Figura 6.43. Errores por días: clúster 1. El eje de ordenadas muestra el error de predicción (por unidad), y el eje de abscisas representa el número de días predichos para el día de la semana representado. Fuente Propia.

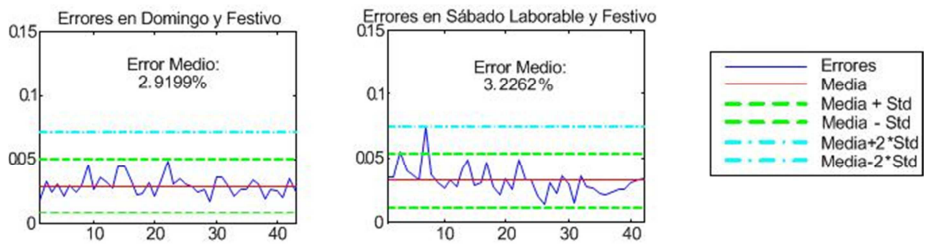


Figura 6.44. Errores por días: clúster 2. El eje de ordenadas muestra el error de predicción (por unidad), y el eje de abscisas representa el número de días predichos para el día de la semana representado. Fuente Propia.

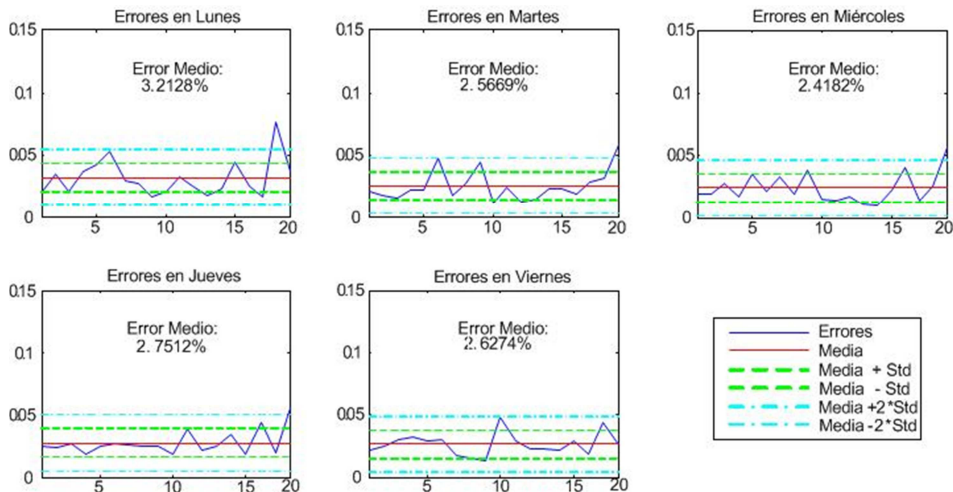


Figura 6.45. Errores por días: clúster 3. El eje de ordenadas muestra el error de predicción (por unidad), y el eje de abscisas representa el número de días predichos para el día de la semana representado. Fuente Propia.

En los clústers 1 y 3, los mayores errores medios corresponden a los viernes, por la misma razón mencionada arriba, y los lunes tiende a tener patrones de consumo más bajos, debido a la

inercia del fin de semana, por eso, se convierten en días más complicados de predecir con la información que dispone el clúster. En el clúster 2, los errores elevados de días festivos de lunes a viernes se pueden explicar por el bajo número de patrones con características similares a ellos en la fase de aprendizaje.

Si el mismo estudio se lleva a cabo pero agrupando por meses, el error medio por meses varía entre 3.5302% a 2.199% en el clúster 1, de 3.8811% a 2.3646% en el clúster 2 y de 3.3739% a 2.2385% en el clúster 3. Desde el punto de vista mensual, la predicción puede considerarse óptima.

**6.4.4.3. Dependencia del error de la fase de operación con el número de patrones de la fase de aprendizaje**

Como se comentó en la sección 6.4, se considera muy importante demostrar la dependencia del error medio (fase de operación) con el número de patrones de la fase de aprendizaje (para cada clúster), teniendo en cuenta que la topología de la red varía con el número de patrones. Los resultados de la evolución del error medio de la fase de operación se muestran en la Tabla 6.10. Se muestra la mejora del porcentaje del error en el inicio de clúster 3 frente a los otros dos grupos, porque las curvas de carga en el clúster 3 tienen mayor parecido entre ellas.

**Tabla 6.10. Dependencia del error medio de la fase de operación con el número de patrones de la fase de aprendizaje. N.N.: número de la red; Ptt.: patrones; N.: número de neuronas; Error (%) error medio de la fase de operación. Fuente Propia.**

CLÚSTER 1				CLÚSTER 2				CLÚSTER 3			
N.N.	Ptt.	N.	Error (%)	N.N.	Ptt.	N.	Error (%)	N.N.	Ptt.	N.	Error (%)
1	50	3	9.6388	1	50	3	7.5828	1	50	3	4.5069
2	75	3	8.1319	2	75	3	6.5873	2	75	4	3.5910
3	100	4	5.5339	3	100	4	5.3916	3	100	4	2.9098
4	125	5	4.1130	4	125	4	3.8672	4	125	6	2.8705
5	150	6	4.10125	5	150	4	3.5104	5	150	8	2.8376
6	175	6	3.7287	6	175	4	3.4759	6	175	9	2.7825
7	200	7	3.2268	7	200	4	3.3451	7	200	10	2.7371

La Figura 6.46 muestra la evolución del error de la fase de operación frente al número de patrones de la fase de aprendizaje, para cada clúster. Posteriormente, los resultados se ajustan con una función para cada clúster, resultando las siguientes funciones

$$Y = 42.0149 \cdot X^{(-0.48474)} \tag{6.12}$$

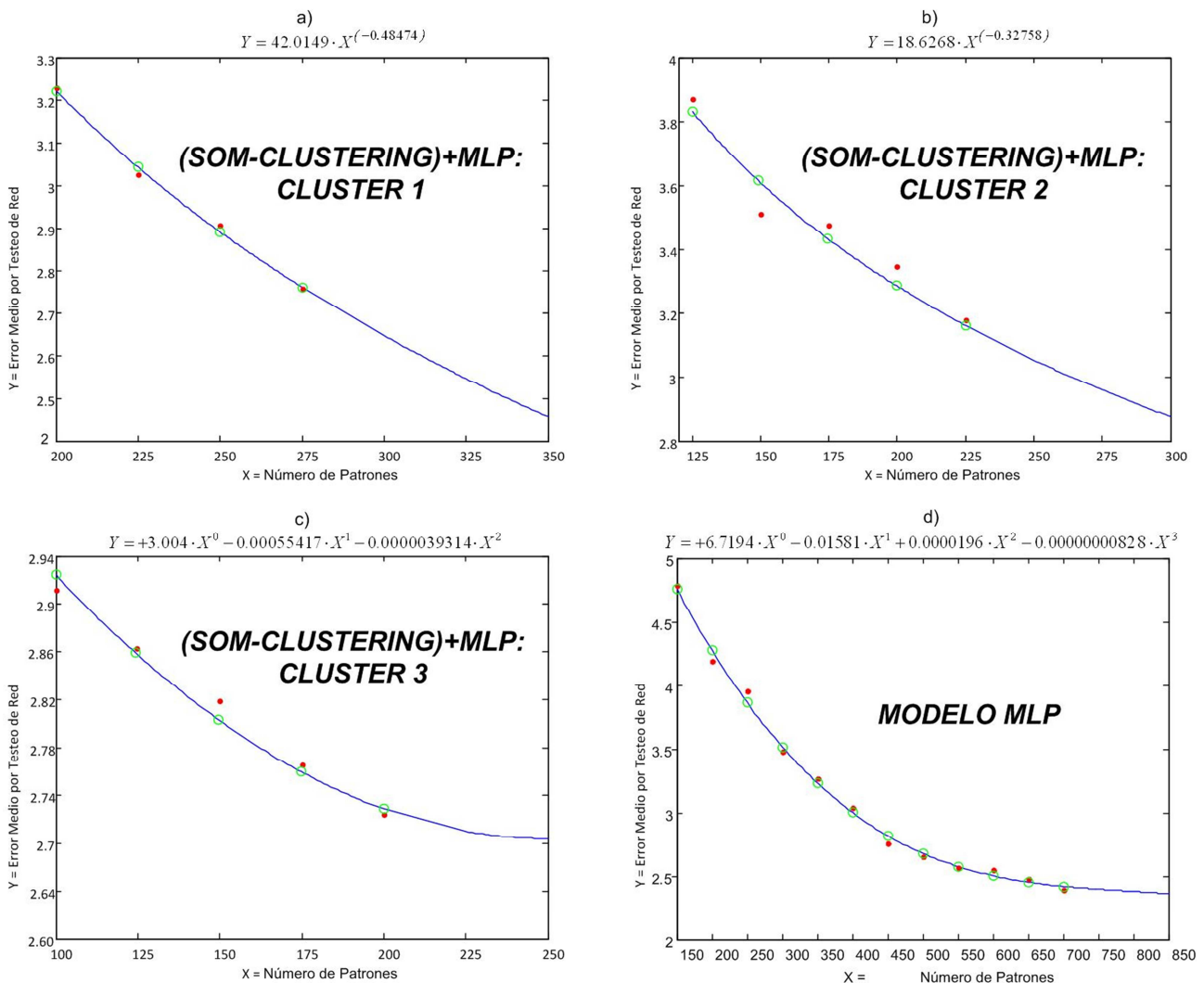
$$Y = 18.6268 \cdot X^{(-0.32758)} \tag{6.13}$$

$$Y = 3.9005 + 3.9005 \cdot X - 0.012207 \cdot X^2 \tag{6.14}$$

donde  $Y$  es el error medio en la fase de operación y  $X$  representa el número de patrones en la fase de aprendizaje; (6.12) corresponde al clúster 1 (Figura 6.46a), (6.13) al clúster 2 (Figura 6.46b) y (6.14) al clúster 3 (Figura 6.46c). Para comparar la mejora, se compara frente al modelo *MLP*

obtenido en la sección 6.3, donde no se realizó ningún reconocimiento de patrones seguido de clusterizado, donde la Figura 6.46d representa la evolución del error con respecto al número de patrones.

Como se aprecia en la Figura 6.46d, el modelo no presenta una tendencia a disminuir el porcentaje de error a partir de los 700 patrones, sin embargo, los tres clúster (secuencia *SOM+k-means+MLP*) tienen una tendencia de mejora, teniendo en cuenta que la prueba tiene un número de 275, 225 y 200 patrones para cada uno de los tres clústers, evidenciando que no presentan un límite, cuestión que si sucedía al observar la Figura 6.46d. Por lo tanto, se ha demostrado la mejora empleando esta sistemática para hacer *STLF* de las curvas de carga, ya que cada clúster está especializado en curvas similares, según la decisión del selector de clúster, obteniendo un error de predicción menor.



**Figura 6.46.** Evolución del error medio de la fase de operación en función del número de patrones de la fase de aprendizaje: a) clúster 1 con 275 patrones; b) clúster 2 con 225 patrones; c) clúster 3 con 200 patrones; d) MLP sección 6.3 con 700 patrones. Fuente Propia.



### **6.4.5. Conclusiones y estudios futuros**

En este estudio, se ha presentado un modelo basado en tres niveles: un reconocimiento de patrones con *SOM*, un posterior clusterizado por medio del algoritmo *k-means*, y finalmente un modelo *MLP* para cada uno de los clústers obtenidos para hacer *STLF*. El modelo produce errores bajos en comparación con otros modelos simples que no están especializados para cada clusterizado de curvas de carga similares.

Por otra parte, se ha presentado una metodología para controlar la evolución error de la fase de operación con el número de patrones de la fase de entrenamiento. Se pretende disponer de una metodología de control de la posible mejora del modelo planteado en base a la adición futura de patrones. La adición de patrones supondrá un ajuste en la topología de la red y el consiguiente reentrenamiento del modelo.

## **6.5. ANÁLISIS EXPERIMENTAL DE LA RELEVANCIA DE LAS VARIABLES DE ENTRADA PARA LA PREDICCIÓN DE LA DEMANDA ELÉCTRICA AGREGADA DEL DÍA SIGUIENTE POR MEDIO DE REDES NEURONALES ARTIFICIALES**

Como se verá en esta sección, existe una relación importante de ciertos parámetros con respecto a la demanda agregada del día siguiente. Además, como se ha visto en secciones anteriores y se verá en la siguiente, la demanda agregada estimada del día siguiente es fundamental como parámetro de entrada de modelos predictivos de *STLF*. Por tanto, en esta sección se presentará la evolución mejorada de diferentes modelos para predecir la demanda agregada del día siguiente por medio de *ANNs*. Las propuestas de las variables de dichos modelos, se basarán en un análisis previo de las variables más relevantes con respecto a la demanda agregada, y la predicción se realizará con *MLP*. En concreto, se plantearán siete modelos basados en *MLP*, añadiendo progresivamente variables de entrada seleccionadas del análisis de la demanda agregada frente a las variables climáticas. Se presentarán los modelos presentando los resultados de la predicción con datos obtenidos en un entorno real.

### **6.5.1. Introducción sobre el estudio**

Desde ya hace mucho tiempo la importancia de estimaciones de la demanda agregada ha quedado patente. Hsu and Chen (2003) presentan un modelo basado en *ANNs* para la predicción de la demanda agregada de tres zonas de Taiwan, pudiéndose emplear los resultados para encontrar oportunidades potenciales de negocio. Pilo *et al.* (2007) muestran un controlador central de *microgrid*, que persigue obtener beneficios económicos agregando consumidores individuales y

comerciales, empleando *ANN* para planificar los generadores, introduciendo la demanda agregada entre otros inputs. Carpaneto and Chicco (2008) entregan una caracterización de consumos residenciales agregados alimentados desde una subestación eléctrica, donde un parámetro crítico son los patrones de demanda agregada estacionales. Fan *et al.* (200) investigan la demanda eléctrica y los datos climáticos de una *utility*, demostrando la dependencia entre ellas en su área de control, y presentando un sistema de predicción de demanda multi-regional empleando la demanda agregada para el factor que analiza la diversidad de carga.

Pudjianto *et al.* (2007) presentan la *VPP* como el principal vehículo para rentabilizar los Recursos de Energía Distribuidos (*Distributed Energy Resources – DER*) dentro de un sistema de energía, donde la estimación de la demanda se convierte en un input principal de la *CVPP*, para posteriormente interactuar con la *TVPP*. Ruiz *et al.* (2009) proporcionan un algoritmo de optimización para gestionar una *VPP* compuesta por un gran número de clientes con dispositivos con control termostático, donde entre los parámetros dinámicos del algoritmo se encuentra la predicción de la demanda agregada en un horizonte de tiempo determinado. Las *VPPs* plantean un reto para la predicción de la demanda y la generación, que se puede abordar mediante un modelo de gestión que tenga en cuenta los diferentes elementos que la componen, y que cooperen para obtener predicción de la demanda por medio de *ANN*, en entornos desagregados como muestran Hernández *et al.* (2013a).

Paoletti *et al.* (2011) abordan los problemas de predicción de demanda eléctrica para redes de distribución con demanda activa, descomponiendo la carga en dos componentes, la llamada base (demanda agregada representando los diferentes patrones estacionales) y un término residual, dependiendo ambos de las fluctuaciones estocásticas y los efectos de la demanda activa. Mousavi and Abyaneh (2011) entregan un modelo de carga dependiente de la tensión, preciso y flexible, siendo la solución más prometedora para modelar la demanda agregada, identificando a ésta como parámetro crítico en la red de distribución.

Ipakchi and Alboye (2009) presentan su visión de la transición de la actual red de energía a *SG*, enfatizando principalmente en *DR*, *EV*, almacenamiento de energía y *DG*, siendo como elemento clave la visión de la demanda agregada de la zona a controlar. Naphade *et al.* (2011) exponen, que para la consecución de ciudades inteligentes, se requiere innovar en planificación, gestión y operación, poniendo como ejemplo de futuro a las *utilities*, que combinarán sofisticados modelos de predicción de la demanda (horaria, agregada, etc.) basado en históricos junto a tráfico real que podría impactar en la demanda futura. Los entornos de futuro, englobados dentro de *Smart World*, deben conocer la relación entre las variables climáticas locales y su demanda agregada, para nuevas aplicaciones de futuro, como muestran Hernández *et al.* (2012a).

## 6.5.2. Datos para la experiencia y metodología

La necesidad de estimar la demanda agregada se ha presentado arriba, además, la dependencia de ciertas variables climáticas con dicha demanda se hace patente en entornos desagregados. Por tanto, se presentará un estudio de relación de las variables climáticas frente a la demanda agregada, así como un análisis de auto-correlación de la demanda agregada con respecto a días anteriores, para plantear diferentes modelos de predicción de demanda agregada, con diferentes entradas de la ANN en base a los resultados de los análisis, comparando los resultados de los diferentes modelos.

Para la implementación y las pruebas de este sistema, se han empleado los mismos datos de la sección 6.1, un entorno desagregado con un tamaño y comportamientos de consumo que podría considerarse similar a la de una *microgrid*. Los datos de demanda y climáticos empleados han sido comentados en la sección 6.1.

### 6.5.2.1. Análisis de las variables relevantes

El objetivo de este estudio es tener una idea inicial de las posibles variables a emplear como entradas de un modelo basado en ANN para la predicción de la demanda agregada del día siguiente.

El estudio se centrará en dos ámbitos diferentes:

- La propia demanda agregada: se pretende detectar la influencia de la demanda agregada de días anteriores con respecto a la del día que se pretende predecir, para lo cual se realizará un análisis de auto-correlación.
- Variables climáticas: se tratará de detectar qué variables climáticas tienen una relación más clara con respecto a la demanda agregada.

Para poder entrenar la red neuronal necesitamos saber con qué intensidad se relaciona la demanda agregada de un día con respecto a la misma variable de los días anteriores, a fin de tener claro qué inputs hay que introducir en la red. La función estadística de auto-correlación, que se define como la correlación cruzada de una señal consigo misma, resulta de gran utilidad para encontrar patrones repetitivos, o periódicos, dentro de una señal.

En estadística, la auto-correlación de una serie temporal discreta de un proceso  $X_t$  es simplemente la correlación de dicho proceso con una versión desplazada en el tiempo de la propia serie temporal. Si  $X_t$  representa un proceso estacionario de segundo orden con un valor principal de  $\mu$  se define entonces:

$$R(k) = \frac{E[(X_t - \mu)(X_{t-k} - \mu)]}{\sigma^2} \quad (6.15)$$

donde  $E$  es el valor esperado y  $k$  el desplazamiento temporal considerado (llamado desfase). Esta función varía dentro del rango  $[-1, +1]$ , donde  $+1$  indica una total correlación,  $-1$  indica una total anticorrelación y  $0$  indica que no existe correlación. En base a esta pequeña explicación teórica de la auto-correlación y utilizando la función “*autocorr*” de MatLab, se realiza un estudio para obtener los coeficientes de auto-correlación de los valores de la demanda agregada.

Con respecto a las variables climáticas, este trabajo presenta un estudio para tratar de determinar la relación existente entre ellas y la demanda agregada, con el fin de intentar determinar aquellas variables susceptibles de ser empleadas en los modelos de predicción, mediante un análisis de correlación entre la demanda agregada y todas las variables climáticas disponibles. En estadística y probabilidad, una correlación mide la fuerza y la dirección de la relación lineal entre dos variables aleatorias. Hay varios coeficientes que miden el grado de correlación, pero el más extendido es el coeficiente de correlación de Pearson, que es el empleado en este estudio. Se obtiene dividiendo la covarianza de dos variables por el producto de sus desviaciones estándar. La correlación de Pearson puede ser definida como un índice que mide hasta qué punto dos variables cuantitativas están relacionados. Los coeficientes de correlación pueden variar de  $-1.00$  a  $+1.00$ :

- $r = +1$  representa una positiva correlación lineal perfecta.
- $0.0 < r < 0.09$  no existe correlación.
- $0.1 < r < 0.25$  hay una pequeña positiva correlación lineal.
- $0.26 < r < 0.55$  hay una media positiva correlación lineal.
- $0.56 < r < 1.0$  hay una fuerte positiva correlación lineal.
- $r = 0$  indica que ambas variables no están correlacionadas.
- $-1 < r < 0$  representa una negativa correlación lineal.
- $r = -1$  representa una negativa correlación lineal perfecta.

#### 6.5.2.2. Cálculo del error

Para cada uno de los modelos presentados, es preciso tener ciertos indicadores de la exactitud de las predicciones realizadas, una vez completada la fase de operación. El proceso de evaluación de los resultados de la predicción de la demanda eléctrica no es una labor sencilla, tal y como reflejan Hippert *et al.* (2001), señalando además, como las *utilities* se muestran reacias a aplicar nuevas técnicas no lineales para la predicción, al no disponer de una sistemática de evaluación de resultados que la industria pueda comprender. Se ha decidido por tanto emplear tres medidas de error complementarias, aplicadas a cada uno de los modelos:

- *MAPE*: error recomendado por la industria y permite comparar con otros trabajos empleados en la literatura:

$$MAPE = 100 \cdot \frac{\sum_{i=1}^n \left| \frac{A_L(i) - \hat{A}_L(i)}{A_L(i)} \right|}{n} \quad (6.16)$$

donde  $A_L(i)$  es la demanda agregada correspondiente al día  $i$ ,  $\hat{A}_L(i)$  es la demanda agregada estimada correspondiente al mismo día y finalmente  $n$  es el tamaño de la muestra.

- Raíz Cuadrada del Error Cuadrático Medio (*Root Mean Square Error – RMSE*): el *MAPE* es poco sensible a los errores que están por encima de la media más 2 veces la desviación estándar, sin embargo los errores grandes y poco frecuentes son de gran importancia para las *utilities*. Es evidente que la función cuadrática da más peso a los errores grandes que la función valor absoluto, es por esto que se ha decidido tener en cuenta. El error *RMSE* se define como:

$$RMSE = \sqrt{\frac{\sum_{i=1}^n \left( \frac{A_L(i) - \hat{A}_L(i)}{A_L(i)} \right)^2}{n}} \quad (6.17)$$

- *Maximum Error (ME)*: este índice complementa a los anteriores y evalúa la máxima desviación entre la predicción y el valor real. Un único efecto muy grande puede tener consecuencias desastrosas en un sistema de producción. El *ME* se define como:

$$ME = \max_i |A_L(i) - \hat{A}_L(i)| \quad (6.18)$$

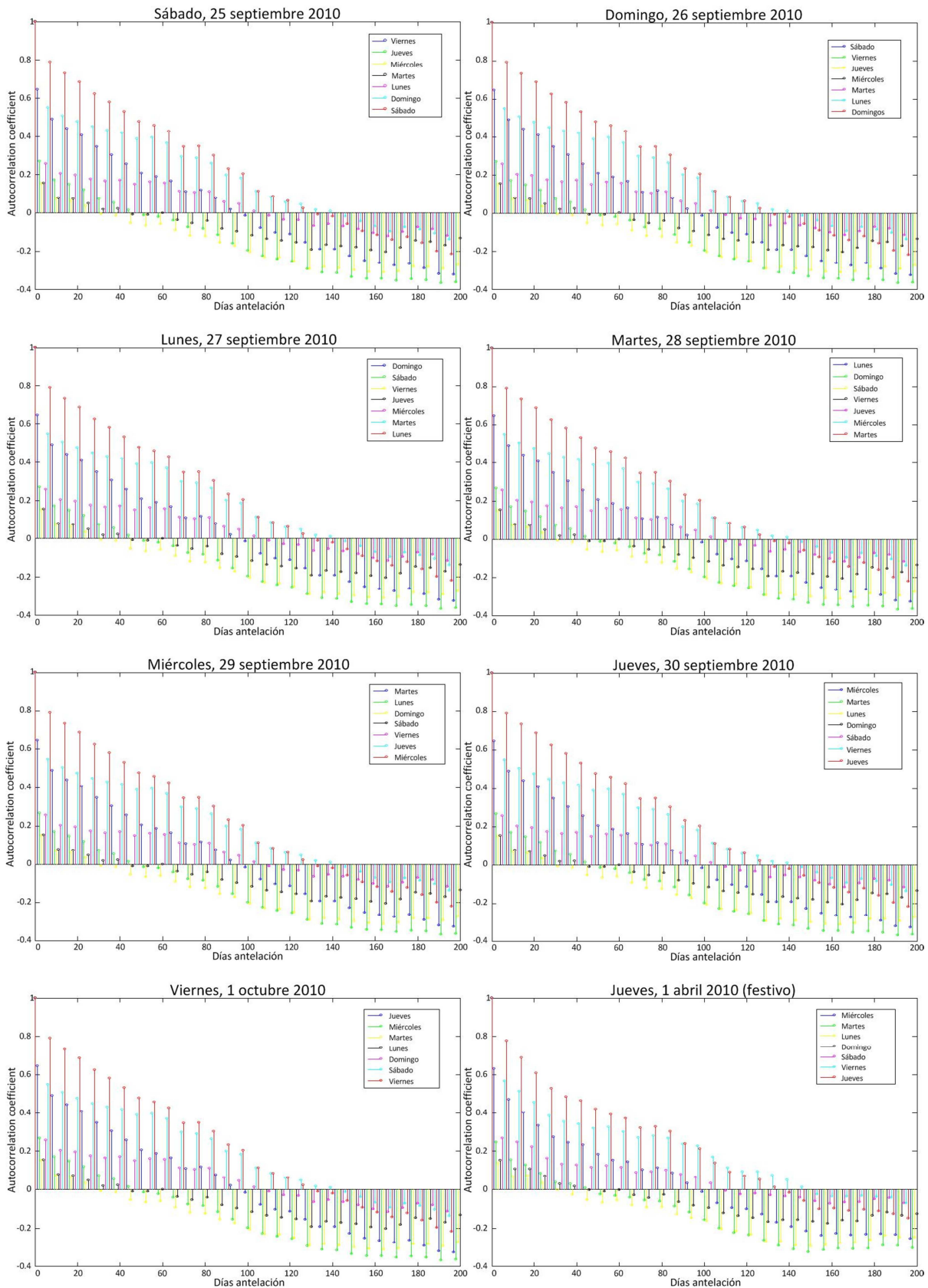
### 6.5.3. Análisis de las variables de relevancia para la producción de la demanda eléctrica agregada

En esta sección se presentarán los resultados de la relación de la demanda agregada consigo misma, y su importancia con respecto a la demanda agregada de días anteriores. También se mostrará la relación de la demanda agregada con las variables climáticas. De esta manera, se tendrá una base científica para proponer diferentes modelos de predicción en la siguiente sección.

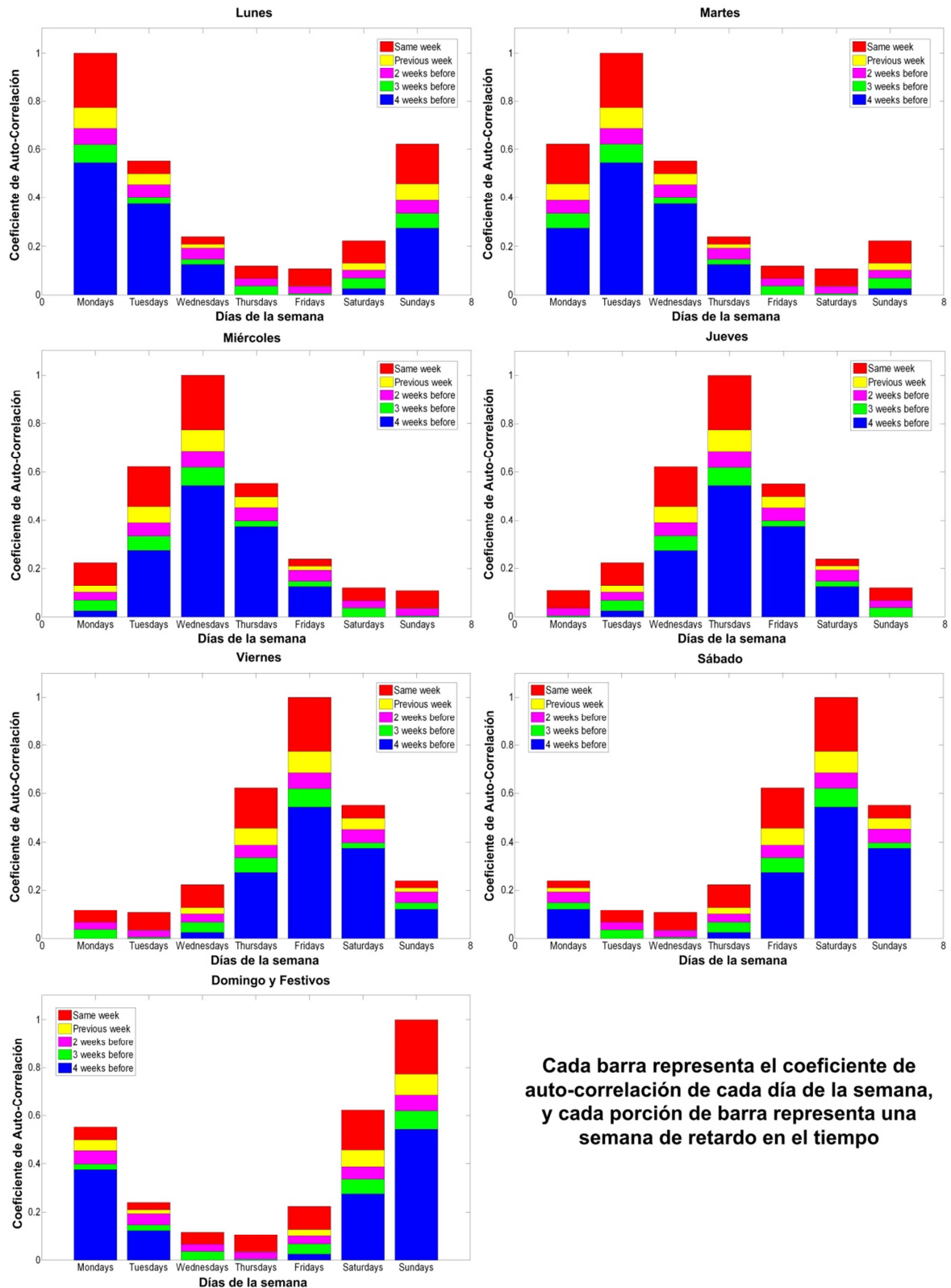
#### 6.5.3.1. Auto-correlación de la demanda agregada

Una vez se introduce en MatLab la totalidad de los patrones con la demanda agregada de todos los días, se obtiene una primera evidencia clara de nuestra suposición inicial: los días que más relación tienen con el día a predecir son los tres o cuatro días similares (día de la semana) de las semanas anteriores, así como su día inmediatamente anterior. Como el interés se centra en tratar de entrenar correctamente el modelo de red que se pretende construir, únicamente se tendrán en cuenta los días hasta 200 días antes del día analizado, a fin de poder estimar el número óptimo de inputs a introducir en la *ANN*, viendo de esta manera todo con más detalle. Es importante destacar el hecho

de que la auto-correlación se hizo con la totalidad de patrones, pero las gráficas se han ampliado al rango  $[0, 200]$  para ver los resultados.



**Figura 6.47. Coeficientes de auto-correlación de cada uno de los días de la semana frente al resto de patrones, y separado por día de la semana. Fuente Propia.**



Cada barra representa el coeficiente de auto-correlación de cada día de la semana, y cada porción de barra representa una semana de retardo en el tiempo

Figura 6.48. Coeficientes medios de auto-correlación entre valores de demanda agregada según día de la semana. Fuente Propia.

En la Figura 6.47 se presenta el análisis de siete días, cada uno se corresponde con cada uno de los días de la semana y, además, se muestra el análisis de un día festivo. Se muestran los días de la semana, los coeficientes de auto-correlación más elevados se corresponden al día

inmediatamente anterior, y a días similares al que se efectúa el análisis, hasta cierto valor del pasado.

Se deduce que tomando como referencia todos aquellos valores de correlación superiores a 0.6, del conjunto total de días se pueden obtener dos subgrupos:

- Los días laborables correlacionan con el día inmediatamente anterior a sí mismos y con los días iguales a ellos de las tres semanas anteriores.
- De la misma manera, los días festivos correlacionan con el día inmediatamente anterior a sí mismos y con los días iguales a ellos de las dos semanas anteriores.

La Figura 6.47 muestra días concretos escogidos de manera aleatoria. Con la finalidad de poder generalizar nuestro análisis, se plantea un análisis global de los valores medios de los coeficientes de auto-correlación.

Tras obtener para cada día los valores de los coeficientes de auto-correlación con respecto a los treinta días previos, se promedian dichos valores con los del análisis realizado de manera análoga, para todos los días similares en cuanto a día de la semana.

De esta manera, se obtiene un valor medio de la auto-correlación entre un día cualquiera de la semana y sus treinta días anteriores, que es lo que se muestra en la Figura 6.48.

### **6.5.3.2. Variables climáticas**

Debido a que el interés inicial es obtener una visión global de la relación de las variables climáticas frente a la demanda agregada, se ha realizado un análisis de correlación empleando la totalidad de los patrones.

En la Tabla 6.11 se muestra la matriz que refleja los coeficientes de correlación entre las columnas de una matriz que contiene los valores de la demanda eléctrica agregada y las variables climáticas, donde se comprueba que las variables climáticas con mayor relación frente a la demanda agregada son: la temperatura media, la humedad relativa media y la radiación solar agregada.

La temperatura media y radiación solar agregada tienen correlación negativa, cuanto más temperatura y radiación menor consumo de energía y viceversa, en cambio, la correlación mostrada por la humedad relativa es positiva.

En la Figura 6.49 se muestra en tres gráficos, la regresión lineal de las variables con mayor correlación con respecto a la demanda agregada, pudiendo comprobar como los puntos se agrupan de manera considerable, y observando igualmente la pendiente positiva de la humedad relativa y la negativa de la temperatura y radiación solar. Los resultados anteriormente mencionados están en línea con lo presentado por Hernández *et al.* (2012a), donde también se expuso un amplio trabajo de la correlación existente entre las variables climáticas y la demanda eléctrica.



Tabla 6.11. Correlación entre demanda agregada y variables climáticas. Demanda Agregada (*Aggregated Load – AL*), Temperatura Media (*Mean Temperature – MT*), Humedad Relativa Media (*Mean Relative Humidity – MRH*), Precipitación Agregada (*Aggregated Precipitation – AP*), Radiación Solar Agregada (*Aggregated Solar Radiation – ASR*), Dirección de Viento Media (*Mean Wind Direction – MWD*), Velocidad de Viento Media (*Mean Wind Speed – MWS*) y Presión Media (*Mean Pressure – MP*). Fuente Propia.

	<i>AL</i>	<i>MT</i>	<i>MRH</i>	<i>AP</i>	<i>ASR</i>	<i>MWD</i>	<i>MWS</i>	<i>MP</i>
<i>AL</i>	1.00000	-0.4508	0.3226	*	-0.4239	0.25437	0.20064	0.02355
<i>MT</i>	-0.4508	1.00000	-0.61656	*	-0.09186	-0.19514	-0.23928	0.05872
<i>MRH</i>	0.32157	-0.61656	1.00000	*	0.27902	0.08449	0.24059	-0.06951
<i>AP</i>	*	*	*	*	*	*	*	*
<i>ASR</i>	-0.4239	-0.09186	0.27902	*	1.00000	0.16316	0.01912	0.00309
<i>MWD</i>	0.25437	-0.19514	0.08449	*	0.16316	1.00000	0.32152	-0.07974
<i>MWS</i>	0.20064	-0.23928	0.24059	*	0.01912	0.32152	1.00000	-0.08816
<i>MP</i>	0.02355	0.05872	-0.06951	*	0.00309	-0.07974	-0.08816	1.00000

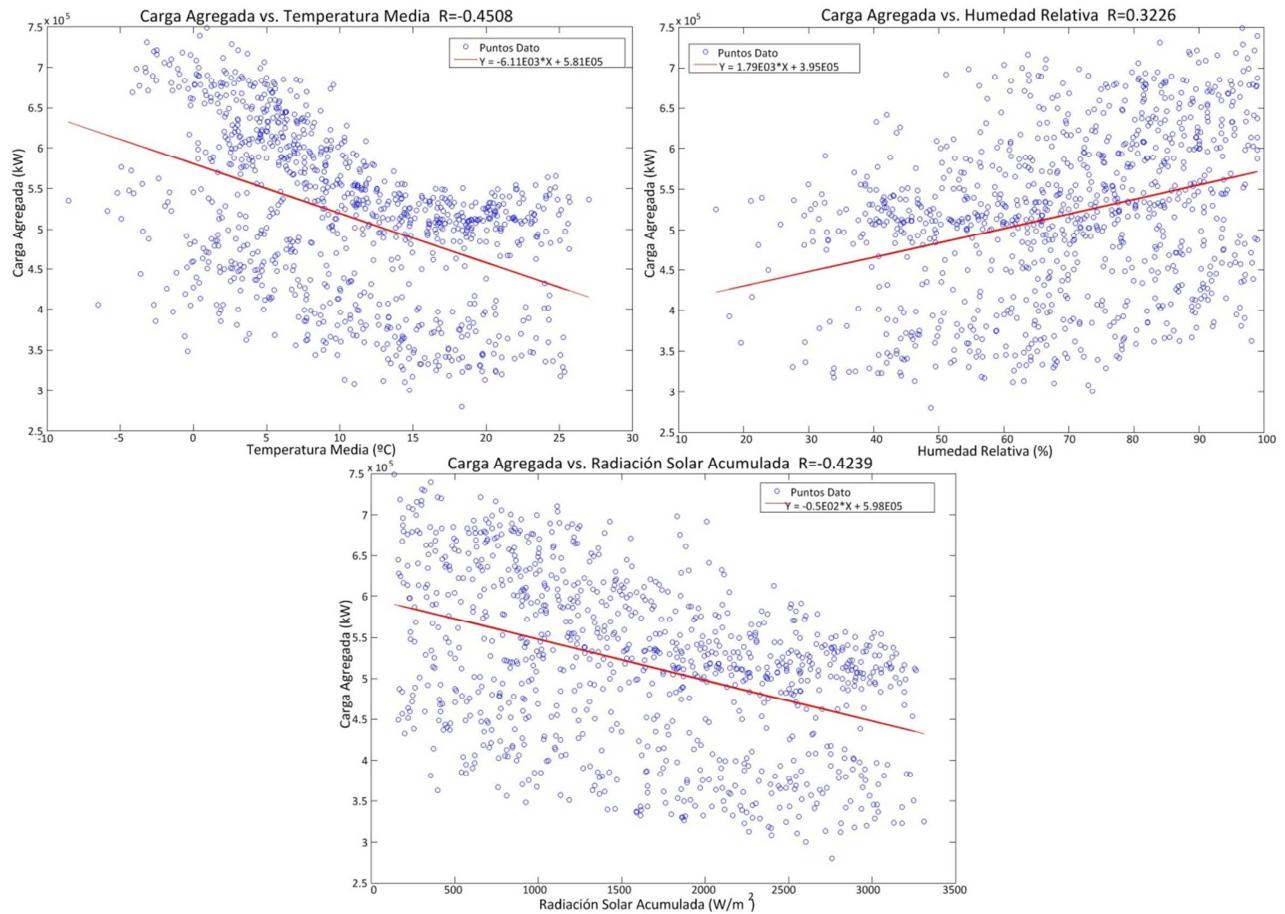


Figura 6.49. Regresión lineal de la demanda agregada frente a la temperatura media, humedad relativa y radiación solar. En el eje de las y demanda agregada en kW y en el eje de las x °C, % humedad relativa y W/m<sup>2</sup>. Se indica el valor del coeficiente de correlación para cada variable climática frente a demanda agregada. Fuente Propia.

#### 6.5.4. Modelos de predicción de la demanda agregada presentados

En esta sección, todos los modelos presentados tendrán como salida la misma variable, la demanda agregada. En base a la experiencia del equipo investigador y de los resultados obtenidos en la sección 6.5.3, se presentarán diferentes modelos de predicción, basados en *MLP*, según la Figura 6.25. A continuación, se detallan los modelos analizados, donde la incorporación de variables inputs a cada uno de ellos se entenderá al presentar los resultados de cada modelo en la siguiente sección:

- Predicción con Demanda/Carga Agregada (*Forecast with Aggregated Load – F\_AL*): según lo demostrado en la sección anterior, la demanda agregada tiene una clara relación con la demanda agregada del día anterior y con las demandas agregadas del mismo día de la semana, pero de las tres semanas anteriores, independientemente del tipo de día (en cuanto a laborabilidad y día de la semana). Por este motivo, en el modelo *F\_AL* se han seleccionado como entradas del modelo las demandas agregadas del día anterior, y las de los tres días similares en cuanto a día de la semana de las tres semanas anteriores. Haciendo un total de 4 entradas y 1 objetivo.
- Predicción con Demanda/Carga Agregada y Laborabilidad (*Forecast with Aggregated Load and Workability – F\_AL\_W*): a las variables de *F\_AL* se añade en este modelo la laborabilidad de los días pasados, así como la del día a predecir. Haciendo un total de 9 entradas y 1 objetivo.
- Predicción con Demanda/Carga Agregada, Laborabilidad y Día de la Semana (*Forecast with Aggregated Load, Workability and Day of the Week – F\_AL\_W\_DW*): a las variables de *F\_AL\_W* se añade en este modelo el día de la semana (*domingo = 0, lunes = 1, ..., viernes = 5, sábado = 6*), en forma de seno y coseno, tanto de los días pasados como del día a predecir. Haciendo un total de 19 entradas y 1 objetivo. Para las variables cíclicas se ha seguido el mismo planteamiento que en la sección 6.3.2.2, ya que se comprobó su éxito al introducir las variables en forma de senos y cosenos.
- Predicción con Demanda/Carga Agregada, Laborabilidad, Día de la Semana y Temperatura Media (*Forecast with Aggregated Load, Workability, Day of the Week and Mean Temperature – F\_AL\_W\_DW\_MT*): a las variables de *F\_AL\_W\_DW* se añade en este modelo la temperatura media de los días pasados, así como la del día a predecir. Haciendo un total de 24 entradas y 1 objetivo.
- Predicción con Demanda/Carga Agregada, Laborabilidad, Día de la Semana y Humedad Relativa (*Forecast with Aggregated Load, Workability, Day of the Week and Relative Humidity – F\_AL\_W\_DW\_RH*): a las variables de *F\_AL\_W\_DW* se añade en este modelo la

humedad relativa de los días pasados, así como la del día a predecir. Haciendo un total de 24 entradas y 1 objetivo.

- Predicción con Demanda/Carga Agregada, Laborabilidad, Día de la Semana y Radiación Solar (*Forecast with Aggregated Load, Workability, Day of the Week and Solar Radiation – F\_AL\_W\_DW\_SR*): a las variables de *F\_AL\_W\_DW* se añade en este modelo la radiación solar de los días pasados, así como la del día a predecir. Haciendo un total de 24 entradas y 1 objetivo.
- Predicción con Demanda/Carga Agregada, Laborabilidad, Día de la Semana y Todas Climáticas (*Forecast with Aggregated Load, Workability, Day of the Week and all Weather – F\_AL\_W\_DW\_allW*): a las variables de *F\_AL\_W\_DW* se añade en este modelo la temperatura media, humedad relativa y radiación solar de los días pasados, así como la del día a predecir. Haciendo un total de 34 entradas y 1 objetivo.

Para todos los modelos, los patrones seleccionados en la fase de aprendizaje de la ANN son sometidos a *cross-validation*: 70% *training*, 15% *validation* y 15% *testing*. El volumen total de los datos se presentó en la sección 6.1 y se repartirán de la siguiente manera: 70% fase de aprendizaje y 30% fase de validación/operación. La fase de operación consistirá en realizar la predicción de cada uno de los días pertenecientes a dicho conjunto, una vez que cada modelo de red haya sido entrenado con su conjunto de datos. Para cada modelo, de cada día predicho se obtendrá su *MAPE* con (6.16), y una vez concluida la fase de operación, se procederá a obtener *RMSE* con (6.17) y *ME* con (6.18), para poder tener una comparativa entre modelos.

### 6.5.5. Resultados

Para todos los modelos, se programó un *script* para completar las etapas de definición/creación y aprendizaje de la red, así como la realización la fase de operación. Para cada *script* de cada modelo, y para todas las funciones de entrenamiento de MatLab, se realizaron las tres etapas anteriores, desde 1 hasta 20 neuronas que forman la única capa oculta, y por cada combinación –número de neuronas y función de entrenamiento– 100 iteraciones para obtener una estabilidad en los resultados.

Las funciones de entrenamiento disponibles son: *traingd* es descenso del gradiente *backpropagation*; *traingdm* es descenso del gradiente con *momentum backpropagation*; *traingda* es descenso del gradiente con *gradient adaptive learning rate backpropagation*; *traingdx* es descenso del gradiente con *momentum and adaptive learning rate backpropagation*; *trainrp* es *resilient backpropagation*; *traingcf* es gradiente conjugado *backpropagation* con *Fletcher-Reeves updates*; *traingcp* es gradiente conjugado *backpropagation* con *Fletcher-Ribière updates*; *traingcb* es gradiente conjugado *backpropagation* con *Powell-Beale restarts*; *traingcg* es *scaled conjugated*

*gradient backpropagation*; *trainbfg* is *Broyden–Fletcher–Goldfard–Shanno (BFGS) quasi–Newton backpropagation*; *trainoss* es *one-step secant backpropagation*; *trainlm* es *Levenberg-Marquardt backpropagation*; y *trainbr* es *Bayesian regulation backpropagation*. Como función de rendimiento de red, que sirve para finalizar el aprendizaje de la red, se empleó *Sum Squared Error*, ya que es la óptima según MatLab.

En la Tabla 6.12, se muestra el *MAPE* de la fase de operación de cada uno de los siete modelos planteados en la sección 6.5.4, para la predicción de la demanda agregada. En la tabla, se presenta para cada modelo, la totalidad de las funciones de entrenamiento y el número de neuronas óptimo tras predecir la demanda agregada de todos los días de la fase de operación (tras 100 iteraciones en cada caso). En negrita se muestran los valores óptimos de cada modelo que hacen un *MAPE* menor de la fase de operación. Se puede apreciar como las funciones de entrenamiento *traingd*, *traingdm* y *traingdx*, presentan errores *MAPE* mínimos de la fase de operación mayores al 25%, obteniéndose para un número de neuronas bajo, por lo que el *MAPE* continuará creciendo para el resto de números de neuronas. Además, con independencia de los siete modelos planteados, la función de entrenamiento que proporciona un *MAPE* menor de la fase de operación es *trainbr*, siendo la desviación estándar la menor de todas igualmente. Los detalles de la evolución de cada modelo se irán presentando a continuación.

**Tabla 6.12. Comparativa de los *MAPE* mínimos entre los distintos modelos de predicción, tras 100 iteraciones de cada uno de ellos y cada función de aprendizaje. Siendo: (1) número de neuronas óptimo para modelo y función de aprendizaje; (2) *MAPE* (%) del número de neuronas óptimo; y (3) desviación estándar (%) del número de neuronas óptimo. Fuente Propia. (Continúa).**

Tabla 6.12. Fuente Propia.

	traingd			traingdm			traingda			trainidx			trainrp			traincgr			traincgp		
	(1)	(2)	(3)	(1)	(2)	(3)	(1)	(2)	(3)	(1)	(2)	(3)	(1)	(2)	(3)	(1)	(2)	(3)	(1)	(2)	(3)
<i>F_AL</i>	1	27.84	9.13	1	27.99	8.70	13	8.39	0.96	1	27.07	8.67	7	6.90	0.46	11	6.81	0.37	13	6.98	0.53
<i>F_AL_W</i>	5	40.87	18.85	5	39.26	18.59	6	7.09	1.64	5	42.20	19.12	7	5.34	0.42	15	5.26	0.42	12	5.37	0.51
<i>F_AL_W_DW</i>	1	30.92	9.04	1	28.47	8.67	8	5.31	1.09	1	29.70	10.38	7	3.90	0.44	13	3.61	0.36	13	3.67	0.51
<i>F_AL_W_DW_MT</i>	1	29.90	8.99	1	28.93	10.08	10	5.89	1.05	1	30.39	8.82	8	4.21	0.47	12	3.83	0.39	11	3.99	0.51
<i>F_AL_W_DW_RH</i>	1	27.67	8.01	1	29.52	9.64	9	5.98	1.15	1	29.35	9.19	8	4.16	0.54	15	3.68	0.42	16	3.82	0.41
<i>F_AL_W_DW_SR</i>	1	28.96	9.89	1	29.01	9.24	12	5.89	1.21	1	29.80	8.44	8	4.13	0.60	17	3.66	0.48	15	3.77	0.40
<i>F_AL_W_DW_allW</i>	1	28.14	9.43	1	29.01	9.31	5	6.54	1.68	1	28.42	9.00	7	4.44	0.56	16	3.93	0.49	18	3.99	0.46

	traincgb			traincsg			trainbfg			trainoss			trainlm			trainbr		
	(1)	(2)	(3)	(1)	(2)	(3)	(1)	(2)	(3)	(1)	(2)	(3)	(1)	(2)	(3)	(1)	(2)	(3)
<i>F_AL</i>	13	6.85	0.36	13	6.96	0.43	10	6.98	0.37	15	7.04	0.45	13	7.06	1.36	11	6.37	0.22
<i>F_AL_W</i>	18	5.31	0.51	15	5.43	0.52	8	5.49	0.35	15	5.54	0.51	19	5.33	0.84	7	4.44	0.15
<i>F_AL_W_DW</i>	16	3.66	0.43	13	3.72	0.57	10	4.06	0.34	12	3.79	0.46	18	3.96	0.85	4	3.13	0.17
<i>F_AL_W_DW_MT</i>	14	3.95	0.48	14	3.96	0.56	8	4.42	0.49	11	4.21	0.68	15	4.53	0.79	3	3.26	0.18
<i>F_AL_W_DW_RH</i>	15	3.85	0.41	15	3.84	0.51	8	4.35	0.45	8	4.01	0.64	14	4.52	1.45	3	3.12	0.17
<i>F_AL_W_DW_SR</i>	16	3.73	0.47	18	3.79	0.58	9	4.43	0.64	8	4.05	0.62	15	4.46	1.64	4	2.98	0.15
<i>F_AL_W_DW_allW</i>	15	4.02	0.40	14	4.05	0.48	10	4.64	0.51	9	4.29	0.64	17	4.94	1.29	3	3.18	0.19

Nuestros modelos empiezan con un número bajo de entradas (4), para ir incorporándoles más variables al modelo. Con *trainbr* se puede observar, como la mayor información proporcionada por las nuevas variables, suponen un error *MAPE* de la fase de operación menor como norma general, y además, una menor necesidad de neuronas en la capa oculta en la mayoría de los casos.

Complementando a lo anterior, en la Figura 6.50 se muestran las evoluciones de los *MAPE* de la fase de operación, según funciones de aprendizaje y número de neuronas, para todos los modelos, tras realizar 100 iteraciones por cada caso. Se aprecia como *trainbr* mejora al resto de funciones de aprendizaje, detectando similitudes de comportamiento de ciertas funciones de aprendizaje en todos los modelos.

En la Figura 6.51 se representa la evolución de los *MAPE* de cada uno de los días predichos en la fase de operación, para cada uno de los modelos presentados. Se han etiquetado los valores de predicción con *MAPE* más elevado, según la siguiente notación: Mes/Día–Día de la Semana Laborabilidad (*Month/Day–Day\_of\_the\_Week Workability*). Además, se muestra el valor de *MAPE*, *RMSE* y *ME* para la fase de operación completa de cada modelo.

El modelo *F\_AL* está construido simplemente con los inputs de la demanda agregada de los días con más relación con respecto al día a predecir, tal y como se analizó en la sección 3.1. Tal y como se muestra en la Figura 6.51, los errores *MAPE* más elevados se producen al hacer las siguientes predicciones:

- Los días festivos entre semana, así como ciertos días posteriores a estos últimos (inercia de la festividad). Al no disponer de más información sobre los patrones, como por ejemplo la laborabilidad, la red no ha tenido en cuenta al ajustar pesos en la fase de aprendizaje, no pudiendo predecir con más exactitud los días que se corresponden con estos dos grupos de curvas de carga.
- Los días comprendidos entre el 23 y el 28 de Junio, que corresponden a la semana de fiestas locales. Exceptuando el jueves 24 que es festivo y el domingo 27 que también lo es, el resto de días hasta el 28 son laborables, pero con comportamiento de festivo, y al igual que antes, la red no tiene dicha información de laborabilidad. Son días que no son festivos en el calendario, pero sí en cuanto a su patrón de comportamiento, de aquí su dificultad en la predicción.
- Elevado número de sábados con errores comprendidos entre el 15% y el 25%. Puede explicarse teniendo en cuenta que estos días son laborables (la red predice un valor de demanda agregada elevado) pero los valores de demanda agregada son menores que los de lunes a viernes, empleando este último tipo de día para predecir los sábados.

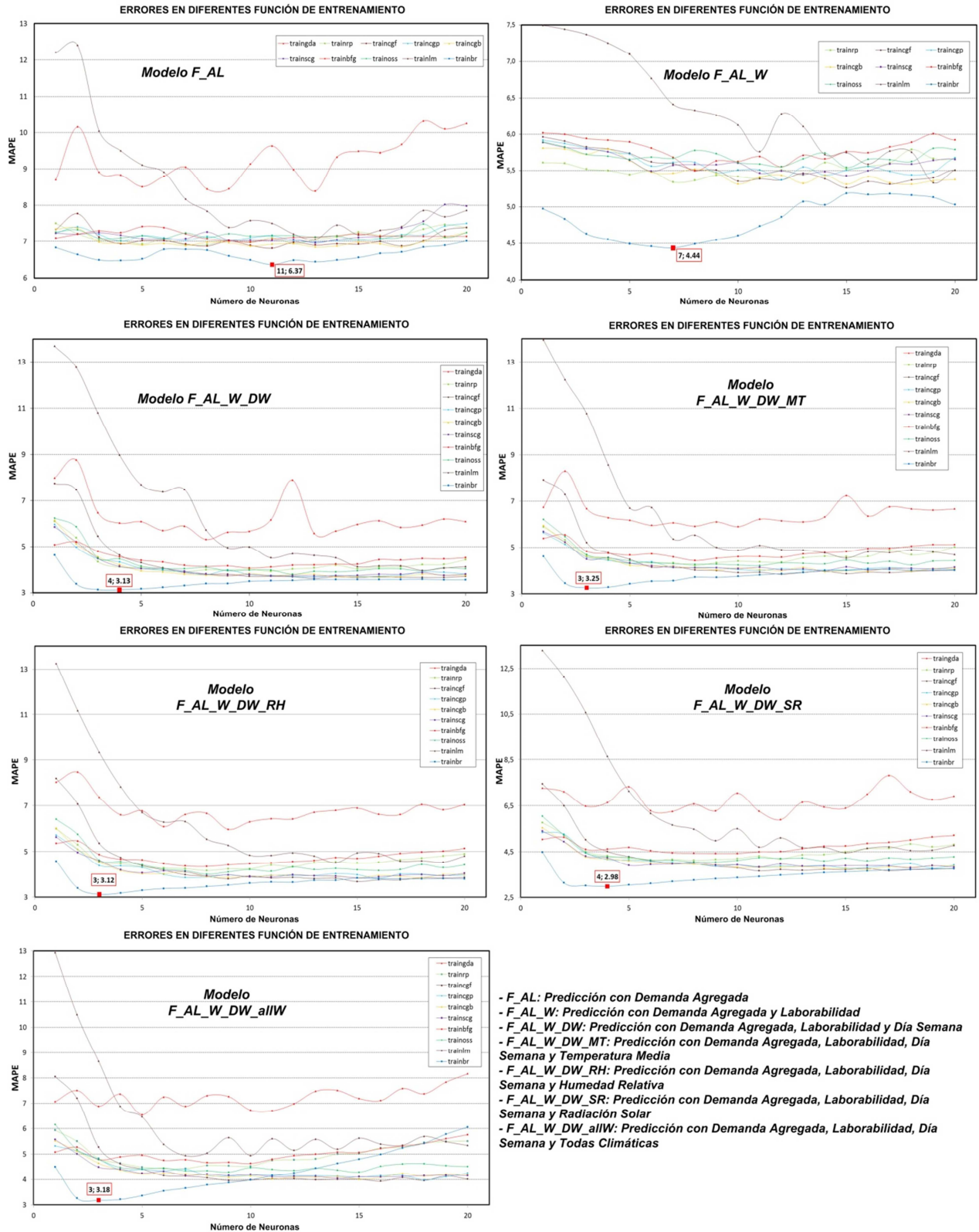


Figura 6.50. Evolución de MAPE (%) de la fase de operación de cada uno de los modelos, para cada una de las funciones de entrenamiento y número de neuronas variables. En la gráfica se indica el valor de MAPE (%) y neuronas óptimas para cada modelo. Fuente Propia.

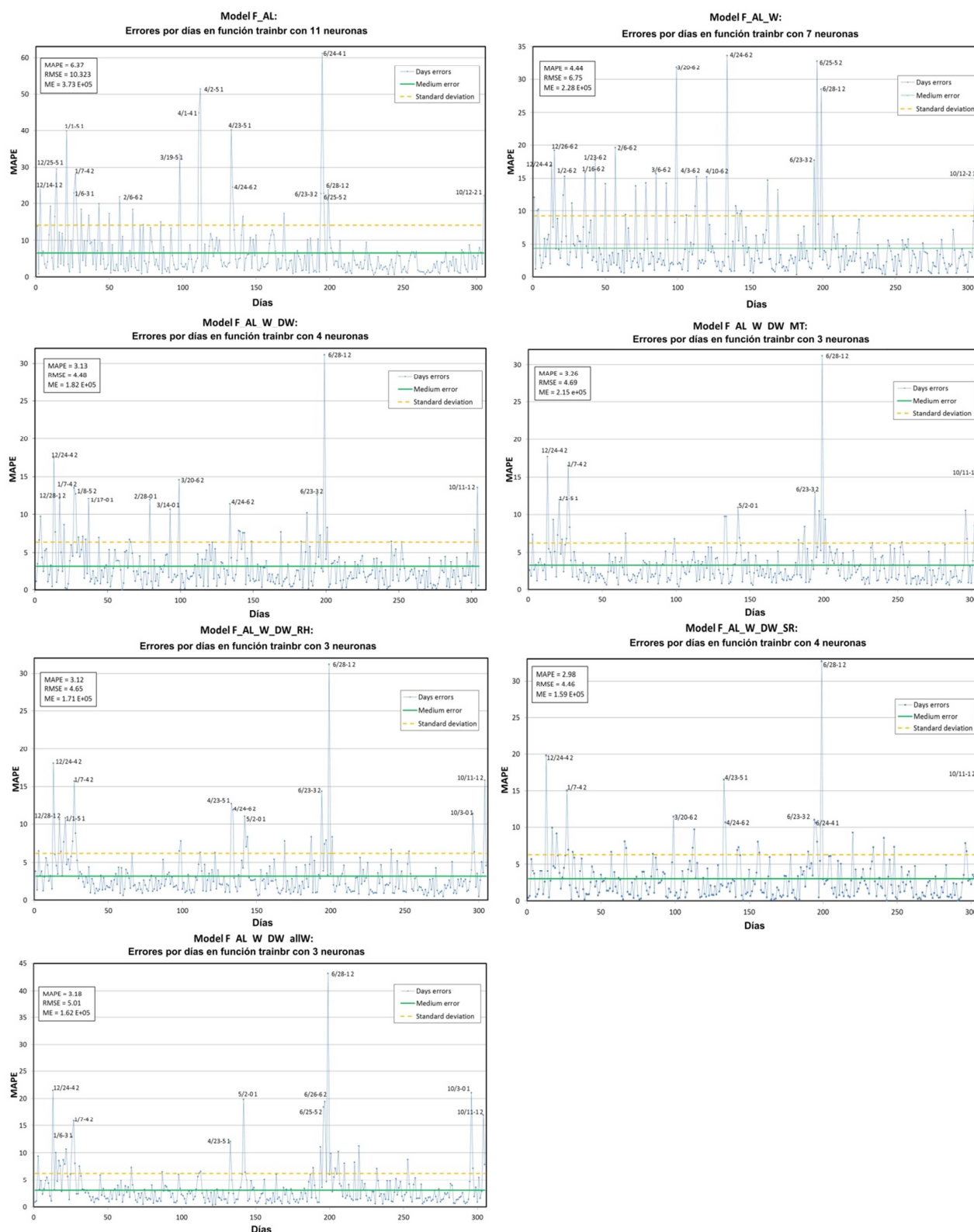


Figura 6.51. Evolución de  $MAPE$  (%) de todos los días de la fase de operación de cada uno de los modelos, para *trainbr* y número de neuronas óptimo. Se indica valor  $MAPE$  (%),  $RMSE$  (adimensional) y  $ME$  (W) de la fase de operación de cada modelo. Fuente Propia.

Del análisis anterior, se evidencia la necesidad de indicar al modelo la laborabilidad de los días, por lo que en el modelo  $F_{AL\_W}$  se introducen las laborabilidades de los días seleccionados como input y del día a predecir la demanda agregada. Como se aprecia en la Figura 6.51, el modelo



sigue fallando en la semana de la festividad local, y en los sábados, sin embargo, se ha conseguido predecir los días festivos con un error menor que la media en todos los casos, es decir, que la idea de introducir la laborabilidad de los patrones como input ha supuesto una mejora.

A fin de mejorar los resultados del modelo anterior, se complementan los inputs de aquella con los días de la semana, en forma de seno y coseno, de los días seleccionados como input y del día a predecir la demanda agregada, obteniendo el modelo  $F\_AL\_W\_DW$ . Como se aprecia en la Figura 6.51, el modelo falla al predecir los días entre dos festivos consecutivos, al ser patrones muy parecidos a los sábados, pero sin ninguna periodicidad haya podido ser detectada por el modelo. El modelo sigue fallando en la semana de la festividad local, pero en menor medida. Se encuentran ciertos domingos para los que la predicción parece ser notablemente errónea; sin embargo, la escala del eje del error ha disminuido y estos errores que parecen elevados, son similares a los cometidos en los modelos anteriores. Además, la predicción de los sábados se ha mejorado notablemente.

En estos tres modelos, al ir introduciendo nuevas variables a la ANN ( $F\_AL$ ,  $F\_AL\_W$ ,  $F\_AL\_W\_DW$ ), se observa como la concentración de los  $MAPE$  elevados de los días disminuye, y se puede ver al etiquetar el error de los días en la Figura 6.51 cada vez a partir de una cota inferior menor, siendo para  $F\_AL$  un 20%, y para los otros dos modelos un 15% y un 10% respectivamente; además, en estos dos últimos modelos, los errores por encima del 20% decrecen notablemente, y en concreto para  $F\_AL\_W\_DW$  aparece tan sólo un valor.

Lo indicado anteriormente, se cumple para la totalidad de los modelos con variables climáticas presentados ( $F\_AL\_W\_DW\_MT$ ,  $F\_AL\_W\_DW\_RH$ ,  $F\_AL\_W\_DW\_SR$  y  $F\_AL\_W\_DW\_allW$ ), comportándose de manera muy similar a  $F\_AL\_W\_DW$ . Los modelos presentan un  $MAPE$  de la fase de operación muy similar a  $F\_AL\_W\_DW$ , aunque cabe destacar la mejora ligera con respecto a éste que proporciona  $F\_AL\_W\_DW\_SR$ . Sin embargo, en todos los modelos con variables climáticas, se comete un error ligeramente superior al predecir determinados días festivos y anteriores/posteriores a éstos, por tanto, sería posible emplear un modelo con variables climáticas para predecir días laborables, y emplear  $F\_AL\_W\_DW$  para el resto de días. En concreto,  $F\_AL\_W\_DW\_SR$  presenta un  $MAPE$  de la fase de operación en días laborables de 2.57% frente a 2.79% de  $F\_AL\_W\_DW$ , en cambio, el  $MAPE$  de la fase de operación para festivos y anteriores/posteriores a éstos de  $F\_AL\_W\_DW$  es 9.87% frente a 13.57% de  $F\_AL\_W\_DW\_SR$ .

Para concluir, se presenta en la Figura 6.52 la fase de operación por meses, por medio del  $MAPE$ ,  $RMSE$  y  $ME$ . Observando  $MAPE$  (para todos los modelos), el error es alto en el mes de junio, comparado con los meses próximos a junio; esto es debido a las fiestas locales comentadas anteriormente. Se aprecia también el error alto (comparado con el resto de meses), y para todos los modelos, en: diciembre, enero y octubre; debido a la presencia de ciertas festividades que complican la predicción.

Se comprueba el parecido de los modelos que contienen climáticas con  $F\_AL\_W\_DW$ , pero destacando la mejora significativa en los meses de diciembre, mayo y septiembre de  $F\_AL\_W\_DW\_SR$ , presentando además una relativa estabilidad de  $MAPE$  en los diferentes meses, con excepción de junio y octubre.

Observando  $RMSE$ , se aprecia como los modelos con climáticas tienen pocos errores elevados, mientras que  $F\_AL$  y  $F\_AL\_W$  presentan en todos sus meses la existencia de errores elevados.  $F\_AL\_W\_DW$  se comporta de manera similar a los modelos climáticos con respecto al control de esos errores elevados, con excepción de diciembre, lo que demuestra que el aporte de variables climáticas mejora la cantidad de errores elevados respecto a la media del mes.

Con respecto a  $ME$ , destacar el evidente parecido entre los modelos climáticos y  $F\_AL\_W\_DW$ , presentando los errores máximos (en  $W$ ) en los meses de diciembre, enero, junio y octubre.

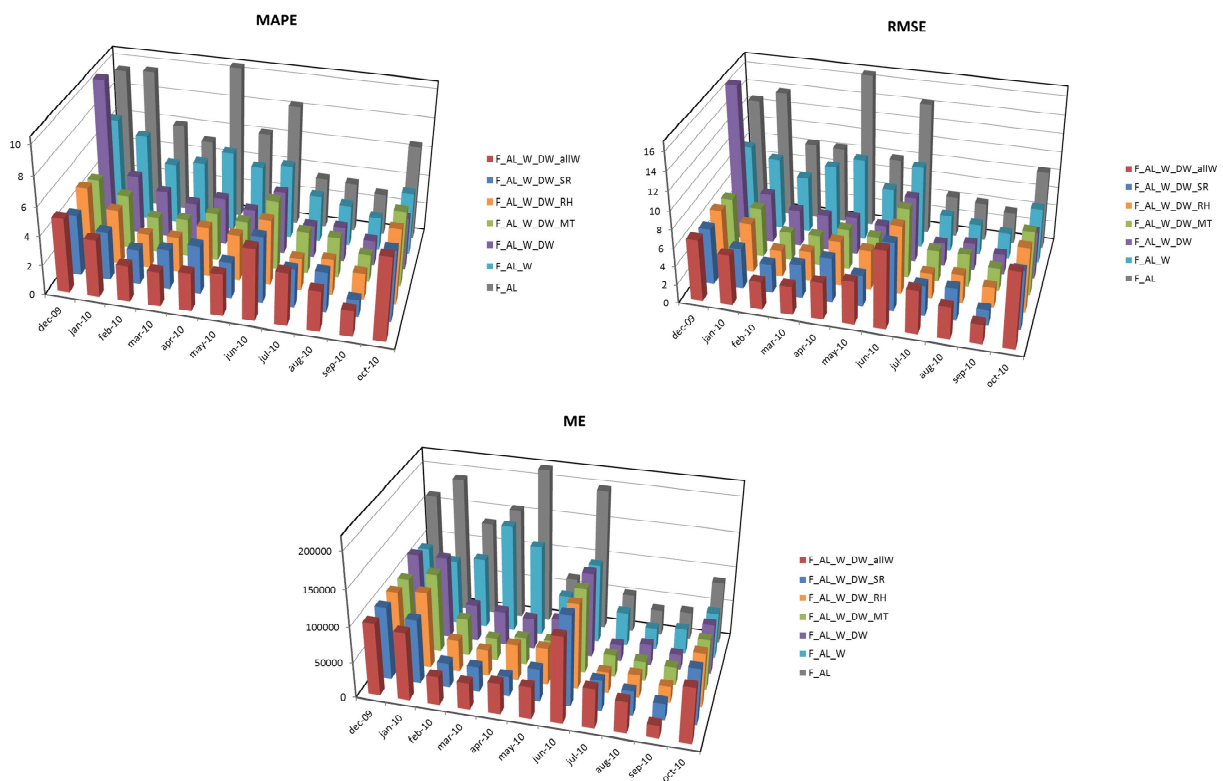


Figura 6.52. Evolución de  $MAPE$  (%),  $RMSE$  (adimensional) y  $ME$  ( $W$ ) de todos los días de la fase de operación de cada uno de los modelos por meses. Fuente Propia.

### 6.5.6. Conclusiones y estudios futuros

En los últimos años, la aparición de nuevos escenarios de operación y funcionamiento del sistema eléctrico ( $SG/VPP/Microgrids$ ) han vuelto a requerir la necesidad de plantear nuevos modelos de predicción, tanto de la demanda como de la  $DG$ . Para el correcto funcionamiento de los anteriores escenarios, se precisarán predicciones exactas, tanto de diferentes parámetros (demanda

agregada, curva de carga, etc.) como de horizontes de tiempo (día siguiente, semana siguiente, etc.). Por tanto, la comunidad científica tiene la oportunidad de continuar investigando en el campo de las predicciones aplicadas a estos nuevos entornos, por medio de técnicas derivadas de la *AI* o por sistemas híbridos.

Los modelos de predicción de demanda, se han centrado hasta el momento en grandes regiones o países enteros. Con la aparición de los entornos citados arriba, se plantea la posibilidad de hacer estudios más detallados de la influencia de las variables que afectan a la demanda eléctrica, ya que al ser entornos más desagregados, permite el empleo de variables locales que afecten al entorno particular, como son las variables climáticas, calendarios específicos, hábitos determinados, etc.; en esta sección se ha presentado un análisis de variables climáticas sobre la demanda agregada, así como la propia influencia de ésta con respecto a días pasados, con el fin de tener una idea de las variables a emplear en los modelos de predicción a proponer. Esta metodología, puede emplearse como referencia para estudios posteriores, y ayudar a identificar estas relaciones o incluso otras con nuevas variables, para así plantear modelos de predicción evolucionados, tanto para la demanda agregada como para otros parámetros de interés a estimar.

Se han presentado siete modelos de predicción de la demanda agregada, tratando de presentar mejoras al aportar nuevas variables de entrada a los modelos. El aporte de nuevas variables se ha basado en el estudio de las variables más relevantes para la estimación de la demanda agregada. Con los resultados de cada modelo, se ha planteado la posibilidad de emplear el mejor modelo con climáticas ( $F\_AL\_W\_DW\_SR$ ) o el mejor modelo sin ellas ( $F\_AL\_W\_DW$ ), dependiendo la época del año en la que se esté. Con independencia del modelo, se plantea como necesario el ajuste a posteriori de las predicciones de ciertos días especiales (fiestas locales y ciertos días festivos), en base a técnicas complementarias, por lo que se seguirán desarrollando modelos híbridos, con *ANN* y lógica fuzzy.

## **6.6. SHORT-TERM LOAD FORECASTING PARA ENTORNOS DESAGREGADOS BASADO EN REDES NEURONALES ARTIFICIALES CON INFORMACIÓN DEL DÍA A PREDECIR**

De las conclusiones extraídas en las secciones 6.3 y 6.4, los modelos de predicción para *STLF* concentran sus errores más elevados de la curva de carga en las zonas de cambio de concavidad y convexidad, o dicho de otra forma, al pasar por sus valores máximos y mínimos de la curva de carga. En concreto, las curvas de carga analizadas en esta obra presentan dos puntos máximos correspondiendo con los dos valores de carga pico, y dos mínimos que corresponden a los dos valores de carga valle.

En este estudio se presenta un modelo basado en *ANN* para hacer *STLF* del día siguiente en entornos desagregados (entorno rural, ciudad pequeña o *microgrid*), empleando como variables de entrada a la red, valores estimados del día que se pretende predecir. Estos puntos de interés del día a predecir son los dos valores máximos (picos), los dos valores mínimos (valles) y la demanda agregada, que junto a otras variables harán que el modelo ajuste mejor la predicción. Los resultados serán comparados con los resultados obtenidos en 6.3; la arquitectura completa será mostrada, con los modelos de predicción de picos, valles y demanda agregada, los cuales serán procesados de manera previa a *STLF* del día siguiente.

### 6.6.1. Introducción sobre el estudio

Tal y como presentan Chan *et al.* (2012), *SG* posibilita el flujo bidireccional de la energía eléctrica entre las *utilities* y los consumidores, y pudiendo adaptar más fácilmente el incremento de fuentes de generación renovable, fomentando la participación de los usuarios finales de la energía en el ahorro de la misma y cooperando con el mecanismo *DR*. El objetivo principal de *DR* es la reducción de la carga pico dentro de su entorno de control, bien sea una red de distribución, una *SC* o una *microgrid*.

O'Toole and Clarke (2012) muestran como los periodos valle obliga a las *utilities* a reducir la producción de los generadores renovables, siendo además difícil de predecir dicha producción. Los periodos pico necesitarán de periodos adicionales, los cuales permanecerán latentes en otros periodos. Una vez presentada la importancia de la carga pico y la carga valle, presentan la *SG* como un sistema multicapa y propone un enfoque de adaptación dinámica entre capas para permitir la flexibilidad necesaria y coordinación precisa. El modelo identificará acciones adaptativas en respuesta a ajustes identificados, agrupado todo en un marco estadístico predictivo en tiempo real para anticiparse a los desajustes y permitir adaptaciones.

Vista la importancia de la predicción de la carga pico, se han realizado trabajos sobre Predicción de la Carga Pico (*Peak Load Forecasting – PLF*). Mohan and Kumar (2002) presentan un modelo basado en *MLP*, con diferentes métodos de gradiente conjugado, para hacer *PLF* del día siguiente y hasta siete días después. Para la predicción de la carga pico del día siguiente se consigue un rango de *MAPE* de 1.06% a 3.39%, y para siete días de horizonte de predicción un rango de *MAPE* de 1.72% a 4.15%.

McSharry *et al.* (2005) presentan una evolución de diez años de carga pico, desde una visión de día de la semana y mes. Posteriormente presentan un modelo para *PLF* con horizonte de predicción de un año por medio de una función compuesta por variables climáticas, estacionalidad y laborabilidad. La validación del modelo se hace con un año completo con un *MAPE* de 2.52%.

Amin–Naseri and Soroush (2008) presentan un modelo híbrido para *PLF* del día siguiente, empleando una clasificación previa de patrones similares, e introduciendo variables climáticas para mejorar la predicción. Para la clasificación emplean el modelo *SOM* junto al índice de Davies–Boundin; posteriormente emplean una red *MLP* para cada clúster y hacer *PLF*. Su resultado es de cinco clústers, la *MLP* de cada uno de los cinco clústers tiene un número de neuronas de la capa oculta de 36, 10, 31, 17 y 2 respectivamente. Para el aprendizaje y validación del modelo híbrido emplean cuatro años de datos, y obteniendo una *MAPE* para cada uno de los cinco clústers de 1.56%, 2.13%, 3.51%, 2.04% y 2.01% respectivamente.

Maksimovich and Shiljkut (2009) plantean tres modelos de *PLF* anual. Los modelos intentan eliminar, evitar o reducir el problema existente en los predictores clásicos cuando se produce una reducción brusca en la evolución con los años de la carga pico. El primer modelo se construye como superposición de dos efectos, el primero se basa en el crecimiento de la carga pico con el tiempo (modelo lineal) y el segundo en la influencia restrictiva del cambio del sistema tarifario (modelo exponencial). El segundo modelo emplea “*modeling slower peak load’s saturation by chain index*”. En el tercer modelo se presenta una metodología para seleccionar el periodo de tendencia relevante y extrapolar los resultados para obtener la estimación. En el peor de los casos se obtiene un *MAPE* de 32% y en el mejor de los casos 3.7%.

Como exponen Hyndman and Fan (2010), la predicción de la demanda a largo plazo juega un papel fundamental para el futuro de las plantas de producción y la posibilidad del aumento de las capacidades de transporte. En el contexto de largo plazo, los planificadores deben adoptar una visión probabilística de las posibles cargas pico, por tanto, los autores presentan una metodología para predecir la densidad de carga pico a largo plazo. El modelo emplea un histórico de diez años y se valida con un año, para dar estimación de la carga pico semanal y anual.

Moazzami *et al.* (2013) presentan un sistema híbrido para *PLF* del día siguiente. Plantean un módulo para descomposición en baja y alta frecuencia de los datos mediante Wavelets, y posteriormente, para cada una de las dos descomposiciones emplear una *ANN* optimizada con algoritmos genéticos y hacer la predicción. Las *ANNs* empleadas son Feed–Forward Generalizada (*Generalized Feed–Forward*), *MLP* y *SVM*; todas entrenadas con el algoritmo *Levenberg–Marquardt*. Todos los modelos de *ANN* tienen 16 entradas para obtener la carga pico, dos capas ocultas con un número de neuronas de: *GFF–Low* (10/10); *GFF–High* (16/16); *MLP–Low* (20/20); *MLP–High* (20/20); *SVM* (no tiene capa oculta). Los modelos fueron entrenados con cinco años y medio y validados con un mes. El *MAPE* varía 1.2% a 1.576%.

En los trabajos anteriores se han presentado diferentes modelos para hacer *PLF* con diferentes horizontes de predicción. No obstante, también se ha empleado información de la carga pico como parámetro de entrada para hacer *STLF*. Jain and Satish (2008) presentan un sistema

híbrido basados en modelos estadísticos y ANN para *STLF*, donde el sistema está formado por cuatro módulos: *Basic SVM*; *Peak and Valley ANN*; *Averager&Forecaster*; y *Adaptative Combiner*. En este modelo, un estimador de la carga pico and carga valle es empleado separadamente, donde las variables de ANN (tanto para estimar carga pico como carga valle) son: carga pico/valle del día anterior, temperatura de la carga pico/valle del día anterior y temperatura media del día anterior.

Amral *et al.* (2008) presentan tres modelos diferentes para hacer *STLF*. El primer modelo entrega los 24 valores a la vez por medio de una *MLP*, teniendo 24 variables de entrada y 12 neuronas en su capa oculta. El segundo modelo predice la carga pico y la carga valle por medio de dos *MLP* con 38 variables de entrada y 6 neuronas en la capa oculta, para posteriormente emplear ambas predicciones para escalar una media normalizada de las curvas de carga del histórico y obtener *STLF* del día siguiente. El tercer modelo presenta 24 *MLPs* en paralelo, una para cada hora del día, con 24 variables de entrada y 6 neuronas en la capa oculta, para todas las *MLPs*. Disponen de un año de datos para la fase de entrenamiento y otro año más para la validación, en esta última han excluido festivos al considerarlos “*días irregulares*”. No se entrega el error cometido para *PLF* and Predicción de la Carga Valle (*Valley Load Forecasting – VLF*), siendo el *MAPE* medio del año para cada uno de los tres modelos: 3.87%, 2.81% y 1.38%.

### 6.6.2. Descripción de los datos y marco metodológico

Los resultados del modelo planteado serán comparados con los resultados obtenidos en 6.3, por tanto, se han empleado los mismos datos.

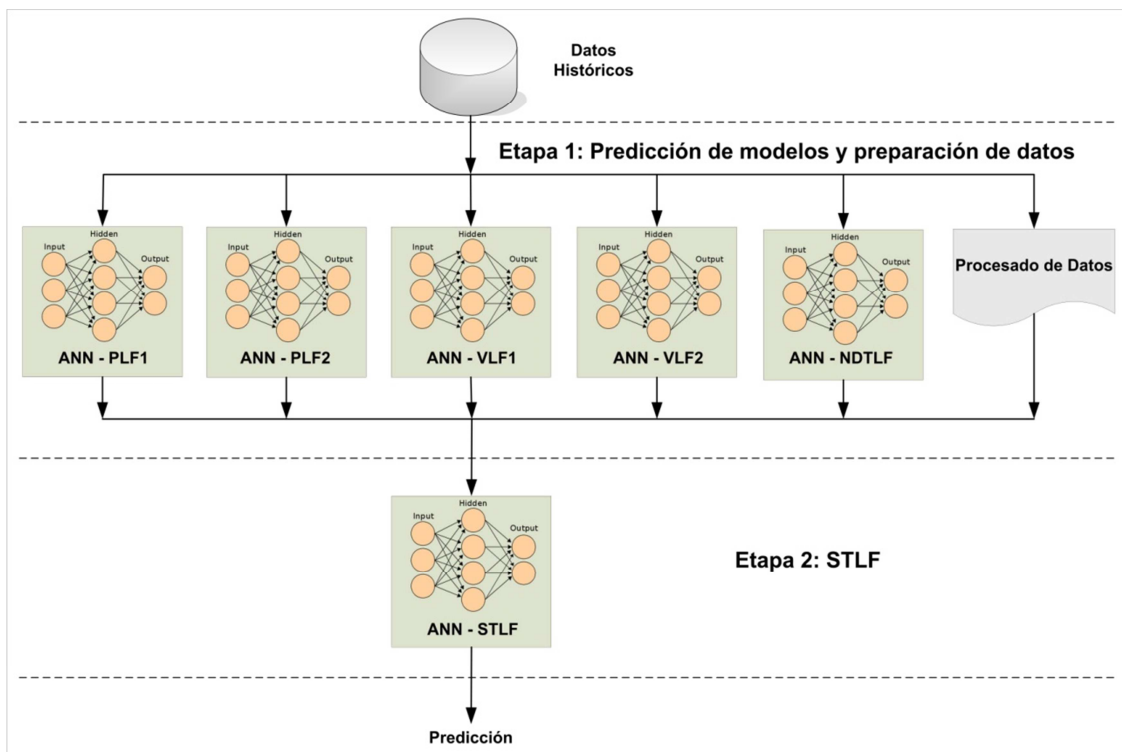


Figura 6.53. Diagrama de bloques de la metodología propuesta. Fuente Propia.

Se plantea una metodología para *STLF* por medio de variables de entrada basada en detalles estimados del día a estimar. El marco metodológico aparece representado en la Figura 6.53, y puede resumirse en dos etapas: estimación de los modelos y preparación de los datos; *STLF* del día siguiente.

La totalidad de los datos se almacenan en una base de datos, la cual estará conectada con la etapa primera del marco metodológico presentado. En esta primera etapa se realizarán seis acciones diferentes:

- *ANN-PLF1*: red *MLP* encargada de obtener la estimación de la primera carga pico del día a predecir. Esta red recibirá las variables de entrada de la base de datos y presentará su estimación a la segunda etapa.
- *ANN-PLF2*: red *MLP* encargada de obtener la estimación de la segunda carga pico del día a predecir. Esta red recibirá las variables de entrada de la base de datos y presentará su estimación a la segunda etapa.
- *ANN-VLF1*: red *MLP* encargada de obtener la estimación de la primera carga valle del día a predecir. Esta red recibirá las variables de entrada de la base de datos y presentará su estimación a la segunda etapa.
- *ANN-VLF2*: red *MLP* encargada de obtener la estimación de la segunda carga valle del día a predecir. Esta red recibirá las variables de entrada de la base de datos y presentará su estimación a la segunda etapa.
- *ANN-NDTLF*: red *MLP* encargada de obtener la estimación *NDTL*. Esta red recibirá las variables de entrada de la base de datos y presentará su estimación a la segunda etapa.
- Procesado de Datos (*Processed data*): se encargará de procesar los datos procedentes de la base de datos, y que la red *STLF* de la segunda etapa espera como variables de entrada, incluidos los resultados de las cinco anteriores acciones.

La segunda etapa consistirá en una red *MLP* para hacer *STLF* del día siguiente (*ANN-STLF*). La red necesitará los datos procedentes de la primera etapa para usarlos como variables de entrada. Una vez presentadas dichas variables, la red entregará la estimación de los 24 valores de demanda eléctrica del día siguiente.

### 6.6.3. Arquitectura del modelo y evaluación del rendimiento del modelo

A continuación se detallarán cada una de las redes empleadas para cada *MLP* indicada en la sección 6.6.2; se presentarán cada una de las capas (entrada, oculta y salida), indicando las variables de la capa de entrada, número de neuronas de la capa oculta y variables de la capa de salida.

La manera de optimizar las *MLPs*, tanto para la determinación de sus neuronas en la capa oculta como del mejor algoritmo de entrenamiento, se suele hacer de manera heurística. Por tanto,

se empleará realizar un *script* automático desarrollado en MatLab, donde se variarán todos los posibles parámetros (número de neuronas de la capa oculta, función de aprendizaje, función de rendimiento de la red durante el aprendizaje, etc.), seleccionando al final la mejor topología para cada una de las redes planteadas. El número de neuronas de la capa oculta se indicará en las siguientes secciones, siendo el resto de parámetros comunes para todas las redes: función de aprendizaje *trainbr* (*Bayesian Regulation Backpropagation*) y función de rendimiento de la red *Sum Squared Error*.

Al igual que en los anteriores estudios, todas las variables de entrada que lleven implícita ciclicidad (día de la semana y mes), serán introducidas a su red correspondiente como dos variables en lugar de una variable, en forma de seno y coseno.

### 6.6.3.1. Estructura ANN-PLFx

La estructura presentada sirve tanto para ANN-PLF1 como para ANN-PLF2, siendo la única diferencia que empleará valores de la primera carga pico cuando sea 1 o de la segunda carga pico cuando sea 2. Como referencia de la red, se han empleado los resultados de Ghomi *et al.* (2010), donde se presenta una red para PLF con un MAPE de 1.57% y dos capas ocultas. En este trabajo, la red dispondrá de tan sólo una capa por motivos de eficiencia del objetivo global (ANN-STLF).

Entrada: 9

- $PL1 \text{ or } 2_{(d-1)}, PL1 \text{ or } 2_{(d-2)}, PL1 \text{ or } 2_{(d-3)}, PL1 \text{ or } 2_{(d-4)}, PL1 \text{ or } 2_{(d-5)}, PL1 \text{ or } 2_{(d-6)}, PL1 \text{ or } 2_{(d-7)}$   
: carga pico de los siete días anteriores al día a predecir.
- $PLT_{(d-1)}$ : temperatura de la hora de la carga pico del día anterior a predecir.
- $AvgT_{(d-1)}$ : temperatura media del día anterior a predecir.

Salida: 1

- $PL1 \text{ or } 2_d$ : carga pico 1 ó 2 del día a predecir.

Ocultas: 3 neuronas.

### 6.6.3.2. Estructura ANN-VLFx

La estructura presentada sirve tanto para ANN-VLF1 como para ANN-VLF2, siendo la única diferencia que empleará valores de la primera carga valle cuando sea 1 o de la segunda carga valle cuando sea 2. Para el diseño de la red se sigue la misma filosofía que ANN-PLF.

Entrada: 9

- $VL1 \text{ or } 2_{(d-1)}, VL1 \text{ or } 2_{(d-2)}, VL1 \text{ or } 2_{(d-3)}, VL1 \text{ or } 2_{(d-4)}, VL1 \text{ or } 2_{(d-5)}, VL1 \text{ or } 2_{(d-6)}, VL1 \text{ or } 2_{(d-7)}$   
: carga valle de los siete días anteriores al día a predecir.
- $VLT_{(d-1)}$ : temperatura de la hora de la carga valle del día anterior a predecir.



- $AvgT_{(d-1)}$ : temperatura media del día anterior a predecir.

Salida: 1

- $VL1$  or  $2_d$ : carga valle 1 ó 2 del día a predecir.

Ocultas: 3 neuronas.

### 6.6.3.3. Estructura ANN–NDTLF

A continuación se indica la estructura de la red ANN–NDTLF, indicando cada una de las variables empleadas, tanto para la capa de entrada como para la capa de salida. Como referencia de la red, se han empleado los resultados de la sección anterior, donde tras un análisis de la influencia de las variables climáticas sobre la demanda agregada, y la influencia de la demanda agregada con respecto a ella misma en días pasados. Se analizan varias redes con diferentes entradas, siendo la red con mejores resultados *Forecast with Aggregated Load–Workability–Day of the Week–Solar Radiation (F\_AL\_W\_DW\_SR)*, con un MAPE de 2.98%.

Entrada: 24

- $TL_{(d-1)}, TL_{(d-7)}, TL_{(d-14)}, TL_{(d-21)}$ : la carga total de un día tiene una clara relación con la carga total del día anterior y con las cargas totales del mismo día de la semana (de las tres semanas anteriores), independientemente del tipo de día (laborabilidad y día de la semana). Por este motivo, se han seleccionado como inputs de la red las cargas totales del día anterior, y las de los tres días similares en cuanto a día de la semana de las tres semanas anteriores.
- $W_{(d-1)}, W_{(d-7)}, W_{(d-14)}, W_{(d-21)}, W_d$ : laborabilidad de los días comentados en el punto anterior, así como la laborabilidad del día a predecir. La codificación es: *festivo=1* y *laborable=2*.
- $DW_{(d-1)}, DW_{(d-7)}, DW_{(d-14)}, DW_{(d-21)}, DW_d$ : día de la semana en forma de seno y coseno (según fórmula sección 6.3.2.2), tanto de los días pasados mencionados en el primer punto, como del día a predecir. La codificación empleada para los días de la semana es: *domingo=0, lunes=1, ..., viernes=5, sábado=6*.
- $SW_{(d-1)}, SW_{(d-7)}, SW_{(d-14)}, SW_{(d-21)}, SW_d$ : radiación solar de los días pasados mencionados en el primer punto, así como la del día a predecir.

Salida: 1

- $NDTL_d$ .

Ocultas: 4 neuronas.

### 6.6.3.4. Estructura ANN–STLF

A continuación se indica la estructura de la red ANN–STLF, indicando cada una de las variables empleadas, tanto para la capa de entrada como para la capa de salida.

Entrada: 33

- $L_{(d-1)1}, L_{(d-1)2}, L_{(d-1)3}, \dots, L_{(d-1)24}$  : correspondiente a los 24 valores de la curva de carga del día anterior al día a predecir.
- Día de la semana de  $d - 1$ : esta variable se introduce como dos, en forma de senos y cosenos, a través de  $\sin[(2 \cdot \pi \cdot day) / 7]_{(d-1)}$  y  $\cos[(2 \cdot \pi \cdot day) / 7]_{(d-1)}$ , con  $day$  de 0 a 6 (  $domingo = 0, lunes = 1, martes = 2, miércoles = 3, jueves = 4, viernes = 5, sábado = 6$  ).
- Mes de  $d - 1$ : esta variable se introduce como dos, en forma de senos y cosenos, a través de  $\sin[(2 \cdot \pi \cdot month) / 12]_{(d-1)}$  y  $\cos[(2 \cdot \pi \cdot month) / 12]_{(d-1)}$ , con  $month$  de 1 a 12 (  $enero = 1, febrero = 2, marzo = 3, \dots, diciembre = 12$  ).
- $PL1_d$  y  $PL2_d$ : los dos valores máximos de la curva de carga del día a predecir, llamados carga pico 1 y 2 respectivamente.
- $VL1_d$  y  $VL2_d$ : los dos valores mínimos de la curva de carga del día a predecir, llamados carga valle 1 y 2 respectivamente.
- $NDTL_d$ .

Salida: 24

- $L_{(d)1}, L_{(d)2}, L_{(d)3}, \dots, L_{(d)24}$  : correspondiente a los 24 valores de la curva de carga del día a predecir.

Ocultas: 22 neuronas.

### 6.6.3.5. Evaluación del rendimiento del modelo

Para toda la fase de operación, en la cual obtendremos la predicción de la curva de carga del día  $d$ , calcularemos su  $MAPE$  por medio de (6.7).

Una vez obtenido  $MAPE_d$  de cada uno de los días correspondientes a la fase de operación, se procede a obtener el error medio de dicha fase por medio de (6.8).

Para controlar el error de la predicción de la curva de carga por horas, se representará dicho error gráficamente teniendo en cuenta el total de los días a predecir durante la fase de operación. El error medio de la predicción para cada una de las 24 horas vendrá dado por (6.9).

El objetivo del estudio es  $ANN-STLF$ , no obstante se entregará el error medio de cada una de las redes pertenecientes a la primera etapa a través de (6.7), agrupando los resultados por meses tras aplicar (6.8) a cada mes y red.

## 6.6.4. Resultados

### 6.6.4.1. Resultados

En la Tabla 6.13 se muestran los *MAPE* agrupados por meses para cada una de las redes de estimación de la primera etapa.

**Tabla 6.13. *MAPE* de los días de la fase de operación, agrupado por meses. Se muestra cada una de las redes de la primera etapa. Fuente Propia.**

	<i>MAPE</i> (%) <i>ANN-PLF1</i>	<i>MAPE</i> (%) <i>ANN-PLF2</i>	<i>MAPE</i> (%) <i>ANN-VLF1</i>	<i>MAPE</i> (%) <i>ANN-VLF2</i>	<i>MAPE</i> (%) <i>ANN-NDTLF</i>
<i>Febrero</i>	2.03	2.04	1.92	1.88	2.39
<i>Marzo</i>	2.35	2.38	2.02	1.96	2.71
<i>Abril</i>	2.30	2.15	2.12	2.05	3.26
<i>Mayo</i>	2.13	2.16	1.96	1.95	2.47
<i>Junio</i>	2.30	2.22	2.09	2.00	4.50
<i>Julio</i>	2.04	2.08	1.87	1.90	2.63
<i>Agosto</i>	2.13	2.09	2.01	1.99	2.74
<i>Septiembre</i>	2.20	2.15	2.03	2.04	1.20
<i>Octubre</i>	2.84	2.67	2.22	2.10	4.86
<i>Noviembre</i>	2.24	2.20	2.05	2.00	4.19
<i>Diciembre</i>	2.99	2.53	2.17	2.02	3.31
<b><i>Media anual</i></b>	<b>2.32</b>	<b>2.24</b>	<b>2.04</b>	<b>1.99</b>	<b>3.11</b>

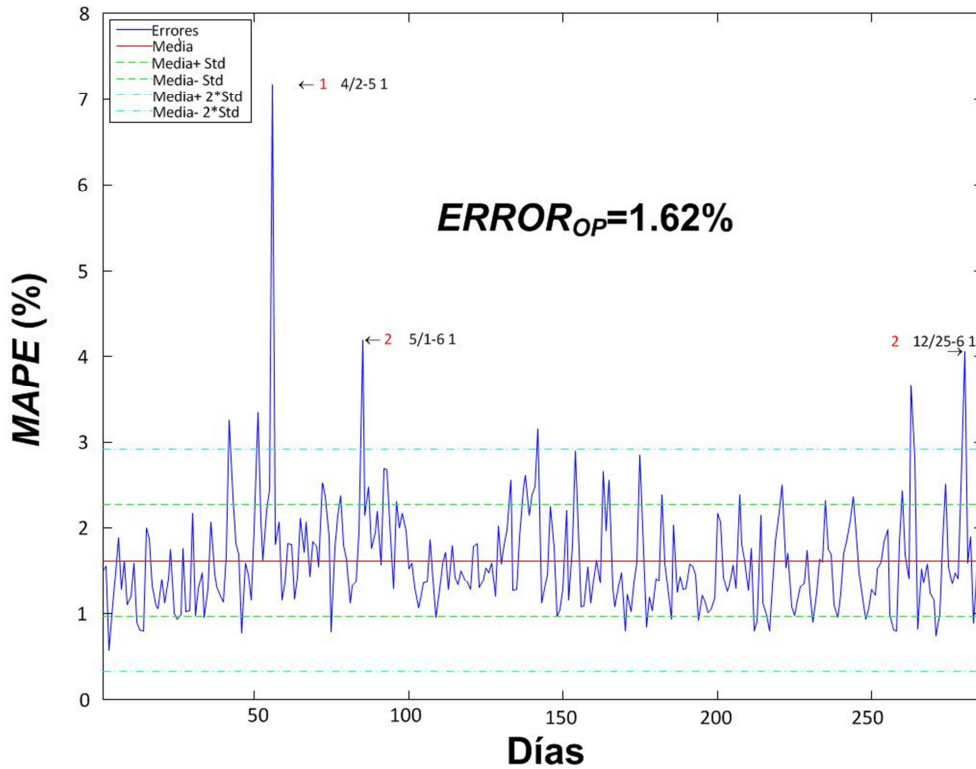
Los *MAPEs* obtenidos para *PLF* and *VLF* son óptimos si los comparamos con los resultados de Mohan and Kumar (2002) cuyo *MAPE* varía entre 1.06% y 3.39%. De manera similar ocurre al compararlos con los resultados de McSharry *et al.* (2005), cuyos *MAPEs* para los cinco clústers son 1.56%, 2.13%, 3.51%, 2.04% y 2.01% respectivamente. Por contra, tan solo *ANN-VLF2* se podría aproximar a los resultados obtenidos por Moazzami *et al.* (2013), con *MAPE* entre 1.2% y 1.57%.

Una vez finalizada la fase de operación para *ANN-STLF*, se representa la evolución de (6.7) en la Figura 6.54 para cada uno de los días de dicha fase. El error medio de toda la fase de operación viene dado por (6.8), siendo su valor 1.62%, el cual es representado junto con la media,  $\text{media} \pm \text{desviación estándar}$  y  $\text{media} \pm 2 \times \text{desviación estándar}$  en la misma figura.

Amral *et al.* (2008) analizaron tres modelos para *STLF* del día siguiente. En concreto el segundo modelo emplea *PLF* y *VLF* como datos de un estimador estadístico que posteriormente obtenía *STLF*, siendo su *MAPE* de 2.81%. El *MAPE* aquí presentado, con los cuatro puntos característicos (*PLF1*, *PLF2*, *VLF1* y *VLF2*) ya descritos, mejora sustancialmente el de aquél.

El *MAPE* obtenido en la sección 6.3 es de 2.47%. Este valor es superior al 1.62% obtenido en esta sección, por lo que podemos asegurar que emplear valores estimados del día a predecir hacen

mejorar ostensiblemente los resultados de la fase de operación, en concreto, la mejora viene propiciada por la información de los dos picos y los dos valles de la curva de carga a predecir.



**Figura 6.54. Errores por día en la fase de operación. En x-eje se representa los días. En y-eje son los errores por (6.7). Fuente Propia.**

En la Figura 6.55 se muestran los errores con *PDF*, donde los intervalos entre la media y  $media \pm$  desviación estándar, y  $media \pm 2 \times$  desviación estándar son representados. Como muestra la figura, la mayoría de los errores –79%– están en el primer intervalo. El porcentaje de distribución de los errores se muestra en la Tabla 6.14. En la sección 6.3 se obtuvo una desviación estándar de 0.95% mientras que en este trabajo se ha obtenido un 0.65%, por lo que los errores se encuentran concentrados alrededor de la media del error, que como se ha visto es inferior a la de aquel trabajo. En este trabajo el 79% de los errores de los días están entre la  $media \pm$  desviación estándar, mientras que en los resultados de la sección 6.3 el porcentaje descendía al 72%. El porcentaje de los errores de día por encima de  $media \pm 2 \times$  desviación estándar en este trabajo es de 2.45%, inferior al 4.21% que se obtenía en la sección 6.3, por lo que los errores muy altos han disminuido, evidencia que puede apreciarse en la Figura 6.54.

En la Figura 6.56 se muestran los errores por hora ocurridos durante la fase de operación. Al igual que se mostraba en la sección 6.3, la red sigue teniendo su errores más altos en las proximidades a los puntos singulares ya comentados (picos y valles), coincidiendo con los cambios de concavidad–convexidad que se producen en las inmediaciones de dichos puntos. Pero por el contrario, en este trabajo los errores totales por hora han disminuido considerablemente.

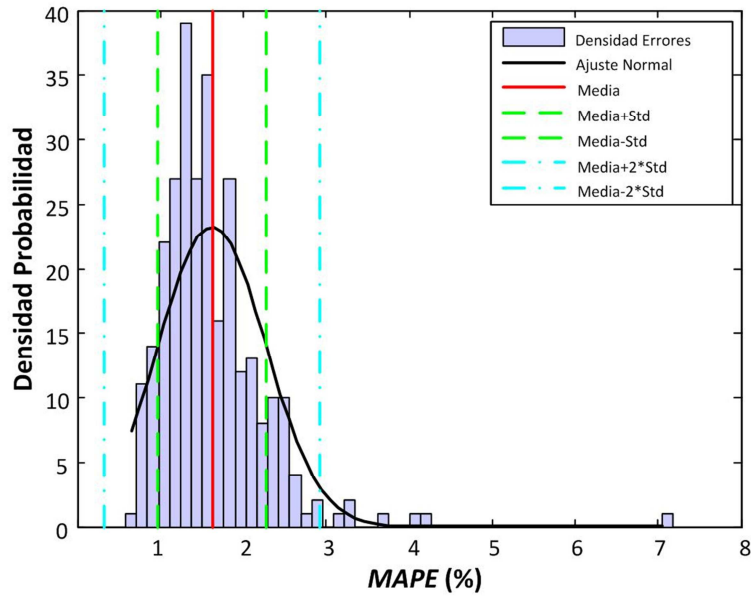


Figura 6.55. Datos de los errores de las curvas de carga en la fase de operación. En x-eje son los errores por (6.8). En y-eje son las densidades de probabilidad. Fuente Propia.

Tabla 6.14. Distribución de errores por día. Fuente Propia.

<i>Variable</i>	<i>Valor</i>	<i>Porcentaje (%)</i>
<i>Median</i>	0.162	1.62
<i>Desviación Estándar (Std.)</i>	0.0065	0.65
<i>No. de errores encima x1 Std.</i>	34	11.89
<i>No. de errores entre x1 Std.</i>	227	79.37
<i>No. de errores debajo x1 Std.</i>	25	8.74
<i>No. de errores encima x2 Std.</i>	7	2.45
<i>No. de errores entre x2 Std.</i>	279	97.55
<i>No. de errores debajo x2 Std.</i>	0	0.00

En la Figura 6.57 se muestran los errores por hora con *PDF*, donde los intervalos entre la media y media  $\pm$  desviación estándar, y media  $\pm 2 \times$  desviación estándar son representados. Como muestra la figura, la mayoría de los errores  $\sim 70\%$  están en el primer intervalo. El porcentaje de distribución de los errores se muestra en la Tabla 6.15. En este trabajo el 79% de los errores por hora se encuentran entre la media y media  $\pm$  desviación mientras que en la sección 6.3 el porcentaje era del 62.5%, por lo que los errores por hora se encuentran más concentrados alrededor de la media.

En la Figura 6.58 se representa la predicción de tres días con error medio más bajo, siendo: a) representa la predicción de 02/08/2010 con error de 0.58%; b) representa la predicción de 03/23/2010 con error de 0.77%; y c) representa la predicción de 12/16/2010 con error de 0.75%.

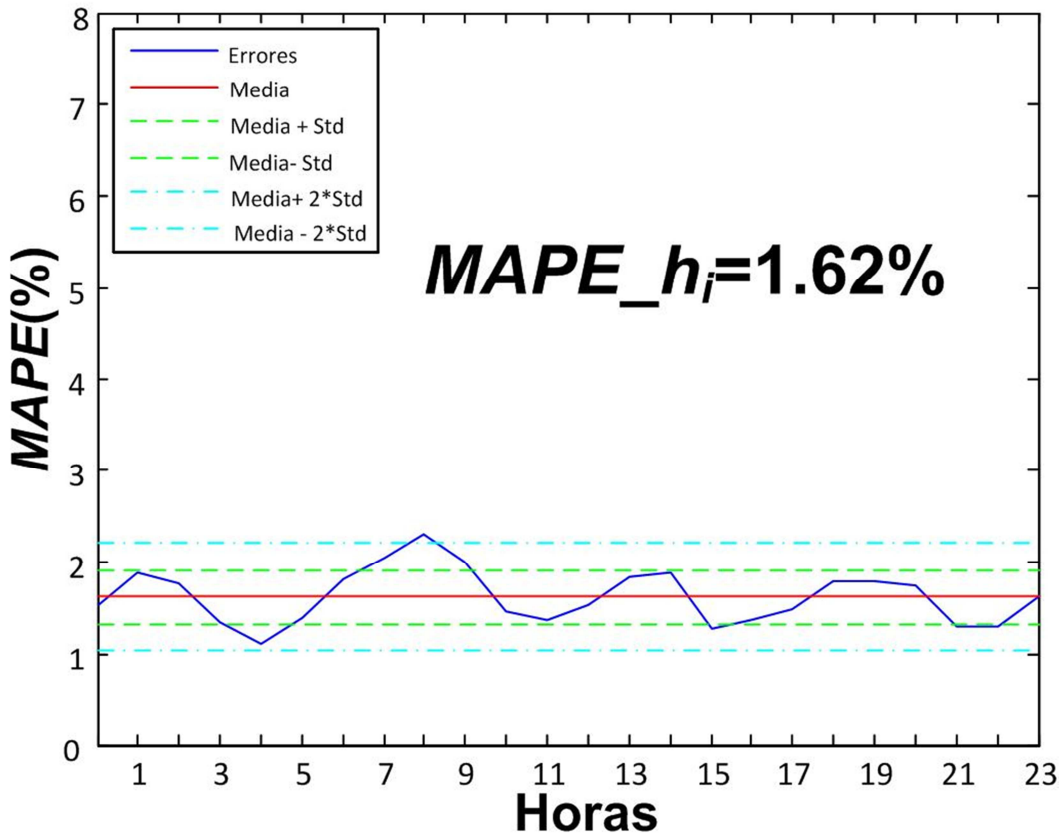


Figura 6.56. Errores por hora en la fase de operación. En x-eje son las 24 horas. En y-eje son los errores por (6.9). Fuente Propia.

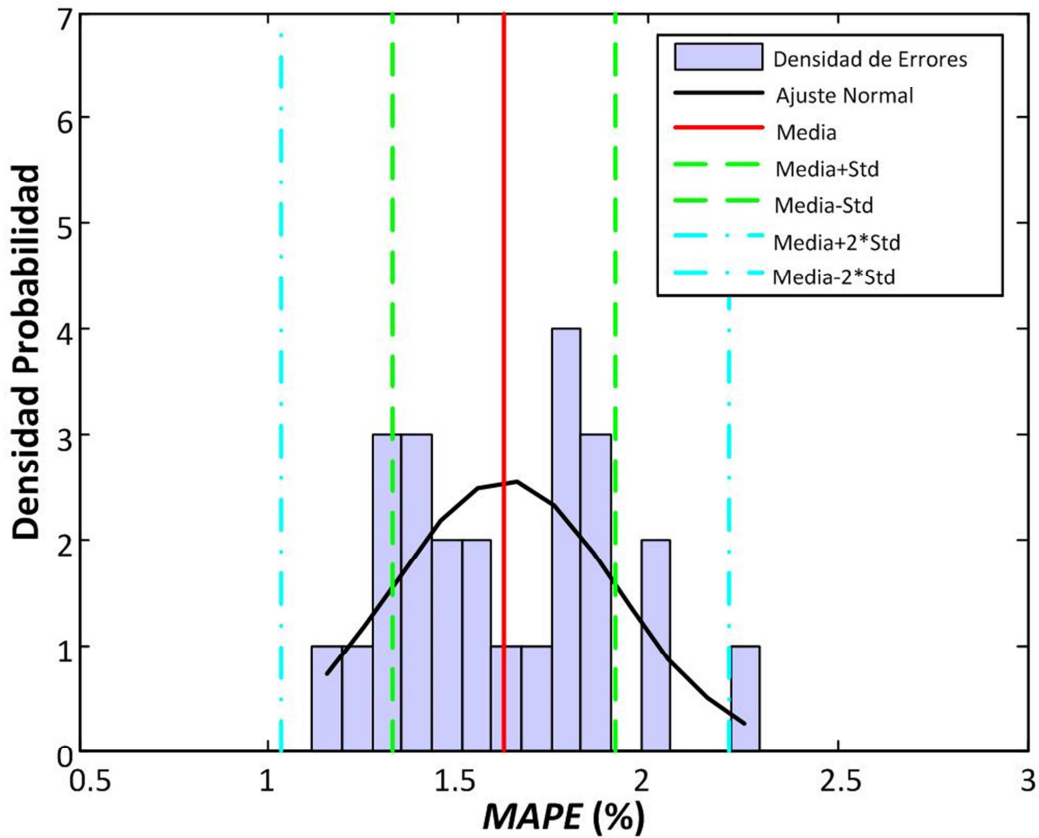


Figura 6.57. Datos de los errores por hora de las curvas de carga en la fase de operación. En x-eje son los errores por (6.9). En y-eje son las densidades de probabilidad. Fuente Propia.

Tabla 6.15. Distribución de errores por hora. Fuente Propia.

Variable	Valor	Porcentaje (%)
Media	0.0162	1.62
Desviación Estándar (Std.)	0.0065	0.65
No. de errors por encima de x1 Std.	3	12.50
No. de errors entre x1 Std.	17	70.83
No. de errors por debajo de x1 Std.	4	16.67
No. de errors por encima de x2 Std.	1	4.17
No. de errors entre x2 Std.	23	95.83
No. de errors por debajo de x2 Std.	0	0.00

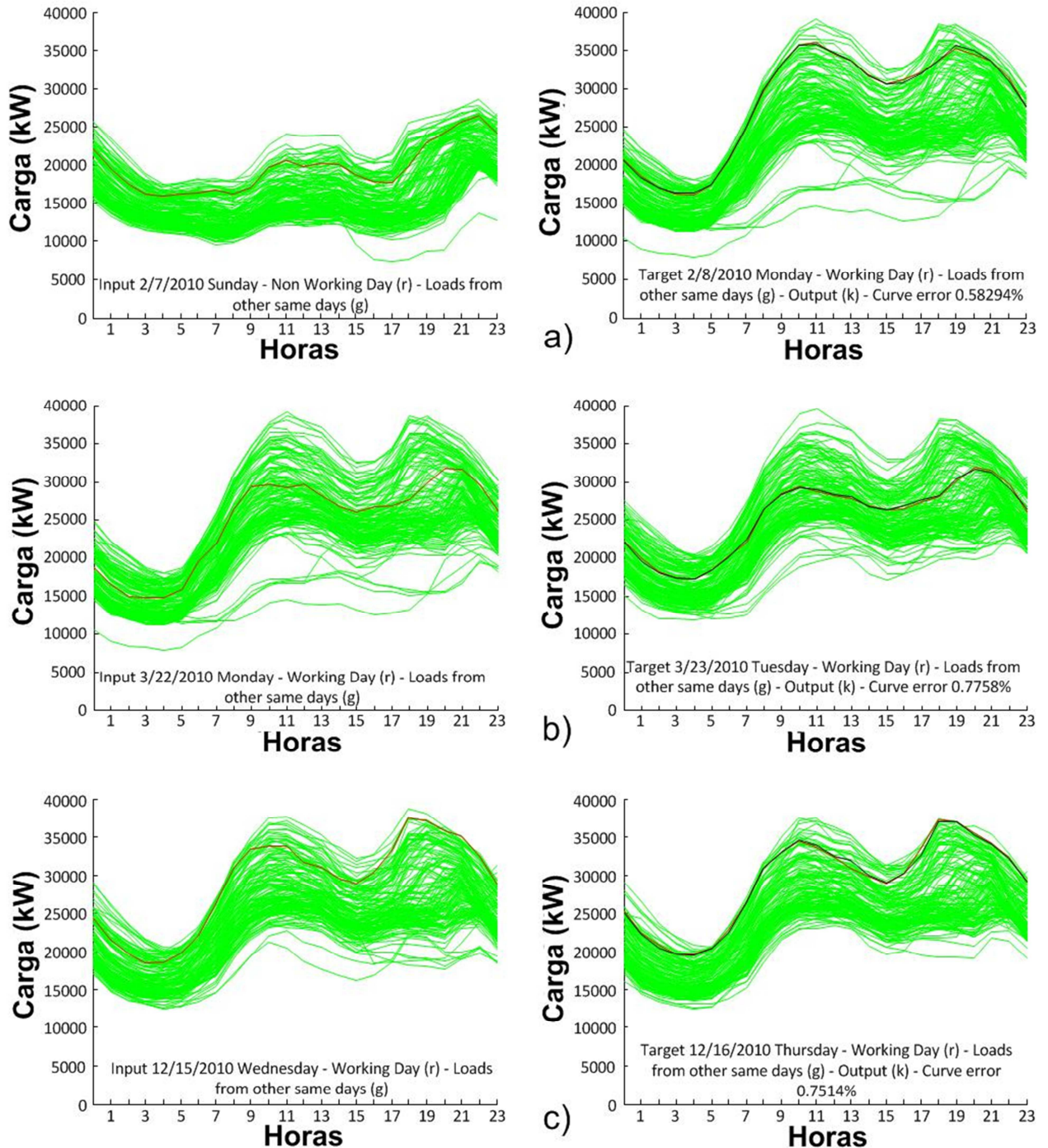


Figura 6.58. Predicciones con error medio bajo: a) 08/02/2010; b) 23/03/2010; c) 16/12/2010. Fuente Propia.

Los errores más bajos encontrados en la sección 6.3 eran de 1.20%, 1.10% y 1.13%, superiores a los resultados del presente trabajo. Con respecto a la Figura 6.58, la gráfica de la izquierda muestra en rojo la curva del día anterior al que vamos a predecir y en verde todas las curvas que tienen las mismas características que ésta, en cuanto a día de la semana y laborabilidad. La gráfica de la derecha muestra en rojo la curva real que se pretende predecir, en negro la predicción realizada y en verde todas las curvas con las mismas características que el día de la predicción, en cuanto a día de la semana y laborabilidad.

#### 6.6.4.2. Coste computacional

Se ha utilizado MatLab para realizar la experiencia, tanto para implementar la ANN, como para el resto de código necesario para tareas adicionales (cálculo de errores, figuras, etc.). Se ha empleado un ordenador de sobremesa con un procesador *Intel Core2 vPro 3.4GHz 2GB RAM*.

Una vez optimizadas las diferentes redes, y para todas ellas, se han realizado tanto la fase de aprendizaje (70% de los datos) como la de operación (30% de los datos) 100 veces. En la Tabla 6.16 se muestra la medida de tiempo para cada una de las dos fases, y por red.

De los resultados obtenidos podemos concluir que en operación real, el tiempo necesario para poder obtener los 24 valores de demanda del día siguiente será de aproximadamente 1 segundo.

Hay que tener en cuenta, que para obtener una predicción, las estimaciones de las cuatro redes de la primera etapa podrían hacerse en paralelo, por lo que el tiempo para disponer de los cinco valores en el peor de los casos sería de 0.38 segundos, y debería sumarse al tiempo necesario para obtener ANN-STLF, que vendría a ser de 0.62 segundos. Se asume que *Processed data* tiene un tiempo de ejecución inferior a 0.38 segundos.

**Tabla 6.16. Tiempo empleado en la fase de aprendizaje y de operación para cada red. Fuente Propia.**

	Tiempo fase de aprendizaje	Tiempo fase de operación
<i>ANN-PLF1</i>	8 minutos y 17 segundos	57 segundos
<i>ANN-PLF2</i>	8 minutos y 25 segundos	58 segundos
<i>ANN-VLF1</i>	8 minutos y 02 segundos	53 segundos
<i>ANN-VLF1</i>	8 minutos y 05 segundos	54 segundos
<i>ANN-NDTLF</i>	15 minutos y 45 segundos	1 minuto y 50 segundos
<i>ANN-STLF</i>	22 minutos y 15 segundos	2 minutos y 57 segundos

Con los resultados de Mohan and Kumar (2002), *PLF* con *MLP* emplea un tiempo para la fase de operación de 104.79 segundos, para un conjunto de datos similar. El tiempo es superior al conseguido en este trabajo para *PLF*. Moazzami *et al.* (2013), para un conjunto de entrenamiento doble que en el trabajo aquí presentado y *PLF*, incluyendo el entrenamiento del algoritmo genético,



la operación y el cómputo de ANN, emplean un tiempo de 705 segundos para un modelo de ANN en el peor de los casos, y de 163 segundos para el modelo con menor tiempo.

### 6.6.5. Análisis de resultados

#### 6.6.5.1. Distribución del error

De lo visto en la Figura 6.54, y apoyado por los resultados reflejados en la Figura 6.55 y Tabla 6.14, se afirma que el error medio de toda la fase de operación está comprendido entre 1.55% y 1.68%, que son mejores resultados que los obtenidos en la sección 6.3, donde el rango se extendía entre 1.45% y 3.35%.

Analizando el error medio por horas de la Figura 6.56, junto con lo representado en la Figura 6.57, Tabla 6.15, vemos que a pesar de obtener mejores resultados que en la sección 6.3, los errores más elevados siguen produciéndose en determinadas zonas de cambio de concavidad–convexidad.

Por lo tanto, el aporte de información del día siguiente, por medio de los cuatro puntos característicos y la demanda total, hace que las predicciones mejoren sustancialmente con respecto a las obtenidas en la sección 3, pero sigue pareciendo interesante dotar a la red de nueva información que mejore los resultados, información que deberá centrarse en esas zonas de la curva.

#### 6.6.5.2. Errores por día de la semana y mes

En la Tabla 6.17 se presenta el error medio de la fase de validación por día de la semana. Se han presentado los resultados comparándolos con los obtenidos en la sección 6.3. Es apreciable la clara reducción del error, destacando la bajada de 1.03% en los domingos.

**Tabla 6.17. MAPE (%) de la fase de operación por día de la semana. Comparativa de ANN-STLF con los resultados de la sección 6.3. Fuente Propia.**

	<i>Domingo</i>	<i>Lunes</i>	<i>Martes</i>	<i>Miércoles</i>	<i>Jueves</i>	<i>Viernes</i>	<i>Sábado</i>
<b>Sección 6.3</b>	2.92	2.31	2.01	1.97	2.38	2.60	2.59
<b>ANN-STLF</b>	1.89%	1.57	1.34	1.41	1.56	1.64	1.91

En la Tabla 6.18 se presenta el error medio de la fase de operación por meses. Al igual que para los meses, los resultados se han comparado con los obtenidos en la sección 6.3. Si se obtiene la diferencia entre los MAPEs para cada mes y se agrupan en: meses fríos (octubre, noviembre, diciembre, febrero y marzo) y meses cálidos (abril, mayo, junio, Julio, agosto y septiembre); al hacer la media de las diferencias de dichos grupos, se obtiene una disminución de 0.96% y 0.65% para cada uno de los grupos respectivamente, a favor del modelo presentado aquí. Los meses fríos presentan mayores variaciones entre los picos y los valles, por lo que en este caso, el aporte de

información de dichos puntos, hace que las predicciones mejoren en mayor medida que para el resto de meses donde las diferencias no son tan evidentes.

**Tabla 6.18. MAPE (%) de la fase de operación por meses. Comparativa de ANN-STLF con los resultados de la sección 6.3. Fuente Propia.**

	<i>Febrero</i>	<i>Marzo</i>	<i>Abril</i>	<i>Mayo</i>	<i>Junio</i>	<i>Julio</i>	<i>Agosto</i>	<i>Sep.</i>	<i>Oct.</i>	<i>Nov.</i>	<i>Dic.</i>
<b>Sección 6.3</b>	2.11	2.68	2.52	2.25	2.52	2.16	2.18	2.27	3.07	1.98	2.69
<b>ANN-STLF</b>	1.32	1.59	1.93	1.81	1.78	1.53	1.45	1.48	1.72	1.41	1.66

### 6.6.5.3. Análisis del error

En la sección 6.3 se expusieron los motivos que explicaban los errores elevados de la predicción de ciertos días al emplear la red de dicho trabajo. Básicamente se podían resumir en: pocas parejas de patrones en la fase de entrenamiento similares al día con error elevado en su predicción; predecir sábados festivos, ya que los patrones de entrenamiento no presentaban esta característica; días entre semana no festivos pero inmersos en fiestas patronales de la localidad (Jueves la Saca); festivos entre semana con día anterior con consumos atípicamente más reducidos; día de Navidad que coincide que su día anterior (Nochebuena) es un día con consume muy bajo.

A continuación, en la Tabla 6.19, se mostrarán los resultados de los días con error elevado que se presentaron en la sección 6.3, para demostrar que con la estructura de variables de entrada de este estudio, esas predicciones mejoran considerablemente, a excepción de 02/04/2010. Destacar la reducción sorprendente de los días 25/12/2010, 26/12/2010 y 31/12/2010.

**Tabla 6.19. MAPE (%) de los días de la sección 6.3 con error más elevado. Comparativa con ANN-STLF; al ser todos los días de 2010, la fecha se muestra con formato MM/DD. Fuente Propia.**

	<b>04/02</b>	<b>05/01</b>	<b>06/24</b>	<b>06/25</b>	<b>06/27</b>	<b>06/29</b>	<b>07/11</b>	<b>10/11</b>	<b>12/06</b>	<b>12/08</b>	<b>12/25</b>	<b>12/26</b>	<b>12/31</b>
<b>Sección 6.3</b>	4.34	4.77	3.92	3.84	4.04	4.36	4.07	4.54	4.77	4.73	8.04	5.06	4.8
<b>ANN-STLF</b>	<b>7.17</b>	4.19	2.44	2.62	2.38	3.15	2.89	2.35	1.68	3.66	4.05	1.59	1.53

En la Figura 6.59 se representa la predicción del día 02/04/2010: a) con la sección 6.3; y b) con el planteamiento aquí presentado. La curva real (roja) parece que hace un levantamiento prematuro de la misma en torno a las 5h, para luego decaer y levantar definitivamente a partir de la 7–8h. En este caso, *VLF1* parece haber sido el causante del error final de ANN-STLF, ya que como se ve en la curva negra (predicción) en b), el pico aparece retrasado en la predicción y hace que

tanto la predicción del primer ascenso y del primer segundo descenso (círculos en la figura) acumulen un error elevado.

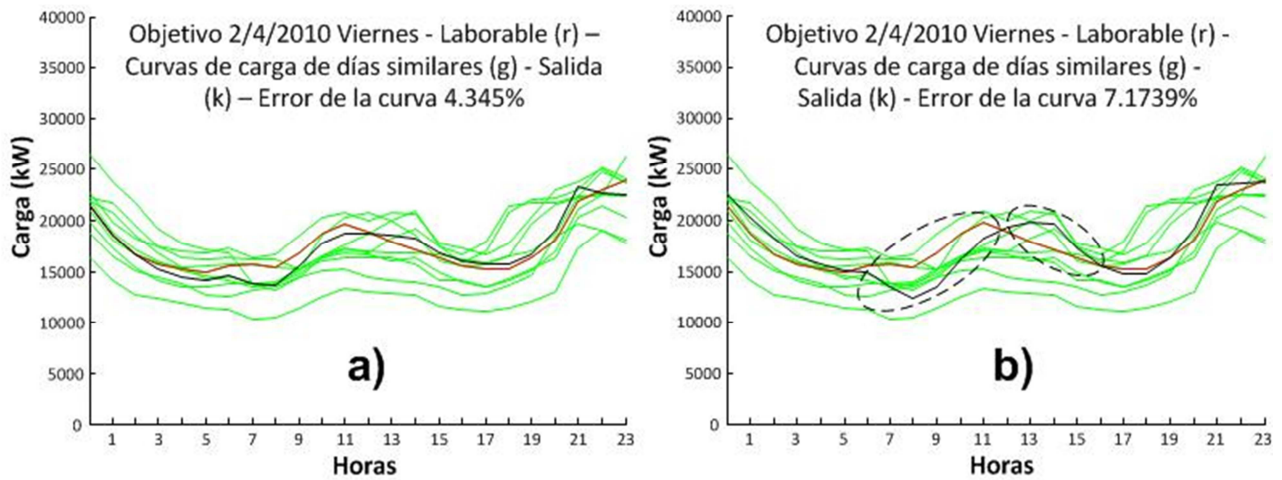


Figura 6.59. Predicción de 02/04/2010: a) con sección 6.3; b) con ANN-STLF. Fuente Propia.

Para finalizar, se listarán las predicciones con errores más bajos, correspondiéndose todas ellas con días laborables de lunes a viernes. Debemos recordar que son los días con el mayor número de patrones. Días con errores bajos y *MAPE* inferior a 0.92%: 02/02/2010 – 0.58%; 17/02/2010 – 0.91%; 18/02/2010 – 0.81; 19/02/2010 – 0.81; 23/03/2010 – 0.77%; 21/04/2010 – 0.79%; 27/07/210 – 0.80%; 03/08/2010 – 0.84%; 09/09/2010 – 0.80%; 10/09/2010 – 0.92%; 14/09/2010 – 0.81%; 28/09/2010 – 0.90%; 02/12/2010 – 0.81%; 03/12/2010 – 0.80; 10/12/2010 – 0.83%; 16/12/2010 – 0.75; 28/12/2010 – 0.89%.

### 6.6.6. Conclusiones

La principal contribución en este estudio de esta sección, es haber demostrado que empleando información del día a predecir es posible mejorar la salida de la red para hacer *STLF*. En concreto, la estimación de los dos valles y los dos picos de la curva de carga del día a predecir, junto a otras variables de entrada presentadas, hará que la estimación global del día siguiente mejore sustancialmente. Esto se ha demostrado realizando una comparación con los resultados del modelo de la sección 6.3, donde las variables de entrada del modelo eran diferentes a las aquí empleadas.

Además, la experiencia se ha realizado con datos pertenecientes a entorno desagregado, donde los perfiles de las curvas de carga muestran pendientes (ascendentes y descendentes) más abruptas que las de los entornos con mayor agregación (grandes extensiones o países). El despliegue de *microgrids* es una cuestión inminente, con lo que será preciso su control y operación; en términos de consumo, lo que se produce es una desagregación de la curva de carga dificultando su predicción. A lo anterior se une la ya existente dificultad que tiene la predicción de la demanda.

Además, las *microgrids* son lugares idóneos para hacer *DR*, ya que entre sus objetivos estará el tratar de ajustar la generación a la demanda, por lo que la predicción de esta última se considera primordial para el balance de las mismas. *DR* no se concibe sin que se tenga un conocimiento lo más exacto posible de la demanda, por lo tanto, el desarrollo de modelos basados en *ANN* para *STLF* serán necesarios, y tendrán mucho margen de desarrollo y mejora.

## **6.7. UNA ARQUITECTURA BASADA EN SISTEMAS MULTI-AGENTES PARA LA GESTIÓN DE UNA SMART GRID Y LA PREDICCIÓN DE LA DEMANDA ELÉCTRICA EN VIRTUAL POWER PLANTS**

Los recientes avances tecnológicos en las áreas de las tecnologías de generación y de la información están ayudando a cambiar el sistema de suministro eléctrico con el fin de conseguir mayor eficiencia energética y sostenibilidad. Las *SGs* introducen inteligencia en la red de energía para optimizar el uso de los recursos. A fin de que esta inteligencia sea eficaz, es necesario recuperar la información sobre el funcionamiento de red junto con otros datos, tales como variables ambientales, para de esta forma poder modificar el comportamiento de los elementos de red. Esta sección presenta un modelo basado en *MAS* para *VPP*, un concepto nuevo en la planta de generación de energía que ya no se produce en grandes instalaciones, pero que es el resultado de la cooperación de generadores más pequeños y de elementos inteligentes. La propuesta no sólo se centra en la gestión de los diferentes elementos, sino que incluye un conjunto de agentes integrados con *ANN* para la previsión de la demanda de energía de los usuarios domésticos finales dentro de la *VPP*.

### **6.7.1. Introducción**

Siguiendo con los principios “*verdes*”, las tendencias y las restricciones impuestas por la creciente demanda y los precios del petróleo y de otros combustibles, los sistemas de generación de la energía y la distribución han estado en constante evolución en los últimos años, tratando de conseguir modelos más sostenibles y compatibles con el medio ambiente. Esta será la clave para lograr una economía sostenible (energía) de acuerdo con las disposiciones del Tratado de Lisboa. Además, los mercados energéticos se enfrentan a un conjunto de desafíos globales. Estos incluyen:

- La desregulación global.
- Elementos innovadores se están integrando gradualmente en la red de energía, tales como *DER* o *DG* (Coster *et al.*, 2011).
- La irrupción masiva de *EVs*.

- Incremento de la participación de los consumidores: los consumidores modificarán su comportamiento de acuerdo con las tendencias “*verdes*” y su ahorro económico (Rahimi and Ipakchi, 2010).

Por otra parte, las *utilities* han sido animadas, por medio de políticas gubernamentales, a incorporar nuevos dispositivos inteligentes. Por ejemplo, los *SMs* serán desplegados en los puntos finales de consumo (hogares de los usuarios, industrias, comercios, etc.), y recolectarán de datos a través de los *DCs*. Además, las *Smart Homes* permitirán el control autónomo de los dispositivos de energía eléctrica (Baladrón *et al.*, 2010; Han and Lim, 2010).

Este aumento de la inteligencia en los nodos de la red eléctrica ha llevado al concepto de las *SGs*: redes inteligentes de energía capaces de realizar adaptación autónoma de sus elementos para optimizar el consumo de recursos (por ejemplo, minimizar el transporte de energía debido a la producción localizada, remodelar la demanda mediante el uso de dispositivos automatizados en *Smart Homes*, etc.).

Dentro de las *SGs*, emerge un modelo de producción de energía denominado *VPP* (Wille-Haussmann *et al.*, 2010), donde la planta de energía ya no es una instalación monolítica, sino una suma de pequeños elementos cooperantes, e inteligentes. La agregación de los generadores y las cargas se unen como una unidad autónoma físico o lógico, donde todos los elementos funcionan coherentemente, como pueda ser un parque industrial, una zona residencial con módulos fotovoltaicos o los recursos distribuidos de una *utility*.

Sin embargo, la gestión de estas *VPP* representa un desafío para las *TIC*. *VPP* comprende un gran número de elementos diferentes, diseñados para resolver los problemas locales, pero que necesitan interactuar juntos con el fin de comportarse como una unidad (*VPP*). Protocolos de comunicación y herramientas son necesarias para permitir que estos elementos se puedan comunicar, tomar decisiones que afectan a varias entidades, y coordinar los comportamientos para llevar a cabo tareas complejas. Al mismo tiempo, los dispositivos de una *VPP* son extremadamente heterogéneos, complicando todavía más su comunicación y coordinación.

En el estado actual de la técnica, modelos *MAS* (McArthur *et al.*, 2007) se han propuesto para controlar los elementos de las *SGs*, especialmente para las grandes redes de área de energía, como por ejemplo en el apoyo a un mercado de la energía inteligente (Duan and Deconinck, 2009; Pinto *et al.*, 2009). *MAS* es un paradigma de computación novedoso en el cual múltiples entidades dentro de un entorno, influyen y son influidos, por otras entidades en el mismo entorno. Además, la previsión de carga es un problema recurrente en las redes de energía, y como tal, se han propuesto algunas soluciones (Wang *et al.*, 2011), aunque la mayoría están diseñados para grandes áreas geográficas.

En esta sección se presenta una solución de arquitectura para la gestión de *VPP* mediante la aplicación de un *MAS*, proponiendo una solución para la predicción de la demanda en entornos desagregados (*microgrids*) y en concreto para el entorno en *Smart Home*. Si bien las obras existentes se centran en la gestión de grandes áreas así como la previsión de demanda en estos entornos agregados, nuestro modelo gestiona *VPP* y la previsión de carga será en entornos más locales y desagregados.

### **6.7.2. Virtual Power Plants**

El concepto de *VPP* fue explicado con detalle en la sección 2.4, no obstante, se cree necesario recordar ciertos detalles e incluso ampliar con algunas informaciones para así entender el planteamiento de arquitectura presentado en esta sección.

A medida que nuevos dispositivos inteligentes y los *DER* se incorporan en la generación de energía, se convierte en necesaria la transformación de antiguas infraestructuras en redes inteligentes hacia un nuevo paradigma para asegurar la coordinación entre todas las entidades implicadas. *VPP* emerger como un sistema descentralizado y auto-organizado como solución inteligente.

Un *VPP* desempeña dos funciones principales. La primera es la *CVPP*, dirigida a la optimización económica. *CVPP* firma contratos de energía con *DER*, a cambio de los ingresos del generador optimizado, tratando de ajustar sus parámetros técnicos y económicos, y creando un sistema estable de compra/venta de energía. Una vez que las ofertas son aceptadas, la *CVPP* controla la ejecución del contrato. La segunda es la *TVPP*, que gestiona el funcionamiento óptimo y seguro del sistema de acuerdo a las limitaciones físicas, garantizando la viabilidad técnica del programa de generación/consumo basado en las fechas presentadas por *CVPP*.

En resumen, *VPP* facilita la integración de nuevos dispositivos inteligentes y *DERs* en el mercado eléctrico mayorista, ofreciendo gestión de sistemas y servicios de apoyo. La Figura 6.60 muestra algunos de los elementos que componen una *VPP*: como generadores (sistemas fotovoltaicos, plantas eólicas, plantas de biomasa, etc.), los consumidores (*Smart Home* y *Smart Building*) y algunos productores/consumidores (*EV* y almacenamiento).

Por otra parte, el tratamiento de la información en la *VPP* se hace de una manera separada de los componentes físicos. Por ejemplo, los generadores en lugares diferentes pueden ser controlados de forma coherente para tratar de optimizar su operación, tanto a nivel local en cada uno de sus entornos físicos, como de manera global y conectados a la red como una sola entidad; de manera similar, un área física puede alojar simultáneamente varias *VPP*.

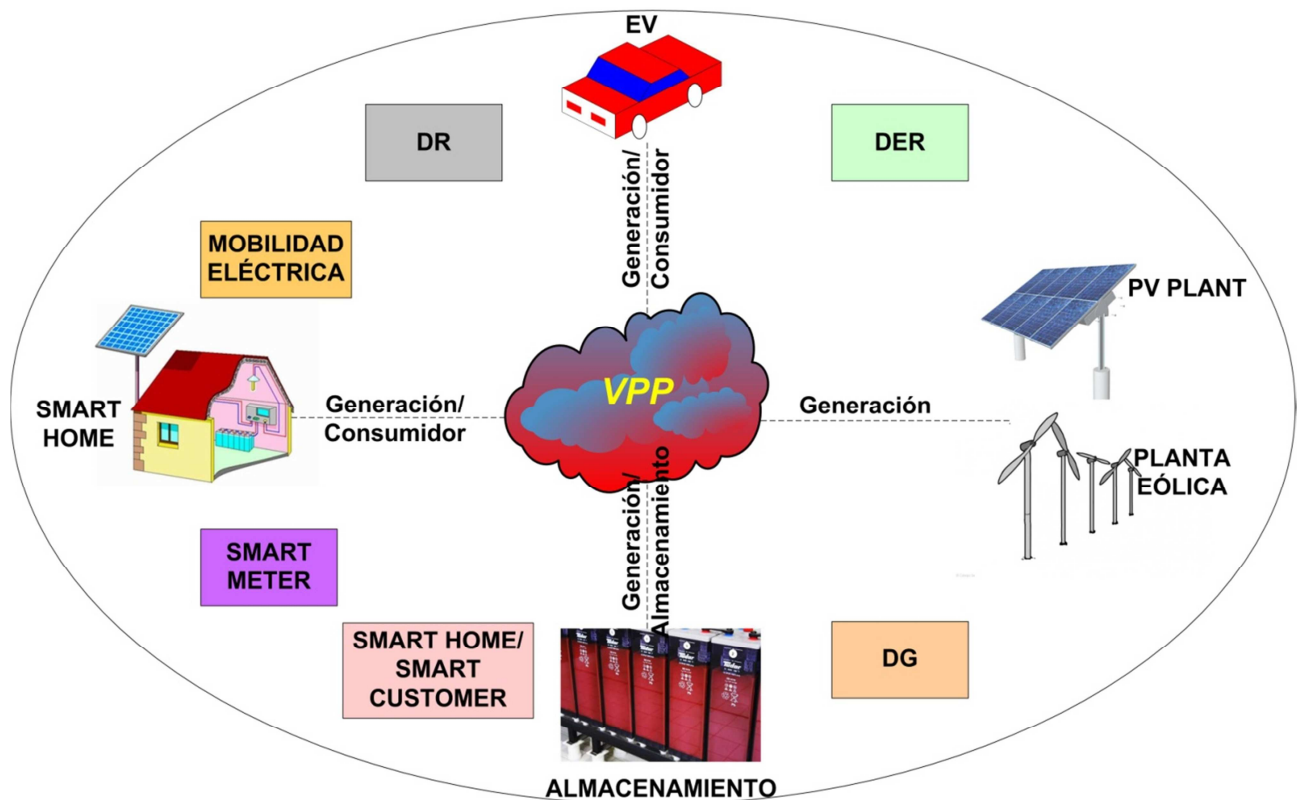


Figura 6.60. Elementos de una VPP. Fuente Propia.

Por último, VPP utiliza medidas de seguridad para garantizar la integridad de los datos y la privacidad. Esto se ha convertido en un campo de investigación crítica (McDaniel and McLaughlin, 2009; Rashvand *et al.*, 2010), porque VPP debe manejar información personal de manera privada. VPP está sujeta a fallos y puede ser objeto de ataques físicos e incluso de virus.

### 6.7.3. Tecnologías de la Información en el escenario Smart Grid

#### 6.7.3.1. Tecnologías de la Información en la estructura física de la Virtual Power Plant

VPP confía en la inteligencia distribuida, la cual supervisa y controla todos sus elementos. La Figura 6.61 muestra la arquitectura de control genérico de un VPP. El flujo de datos VPP comienza desde las fuentes de datos, tales como SM o las estaciones meteorológicas, y se envía los datos a los agentes inteligentes de baja tensión (*Low Voltage – LV*) en las subestaciones, por medio de los DCs. Las funciones más básicas se llevan a cabo en el nivel más bajo de la estructura jerárquica, donde los dispositivos toman decisiones con respecto a sus propias actividades locales, tales como la activación de un generador o apagar un dispositivo doméstico inteligente. El control de VPP (VPPC) coordina estas operaciones de acuerdo con el estado global interno de la VPP (Lloret *et al.*, 2011), y proporciona interfaces con la red de alimentación externa (representado por la segunda VPP en la figura) de acuerdo con la indicación externa del estado de otros elementos inteligentes. Por ejemplo, a través de estas interfaces VPP, las empresas pueden facilitar la creación de un

mercado de la energía inteligente, o *VPP* de varios polígonos industriales y zonas residenciales pueden intercambiar información de demanda y generación, compartiendo energía localmente en lugar de comprarla en un mercado global.

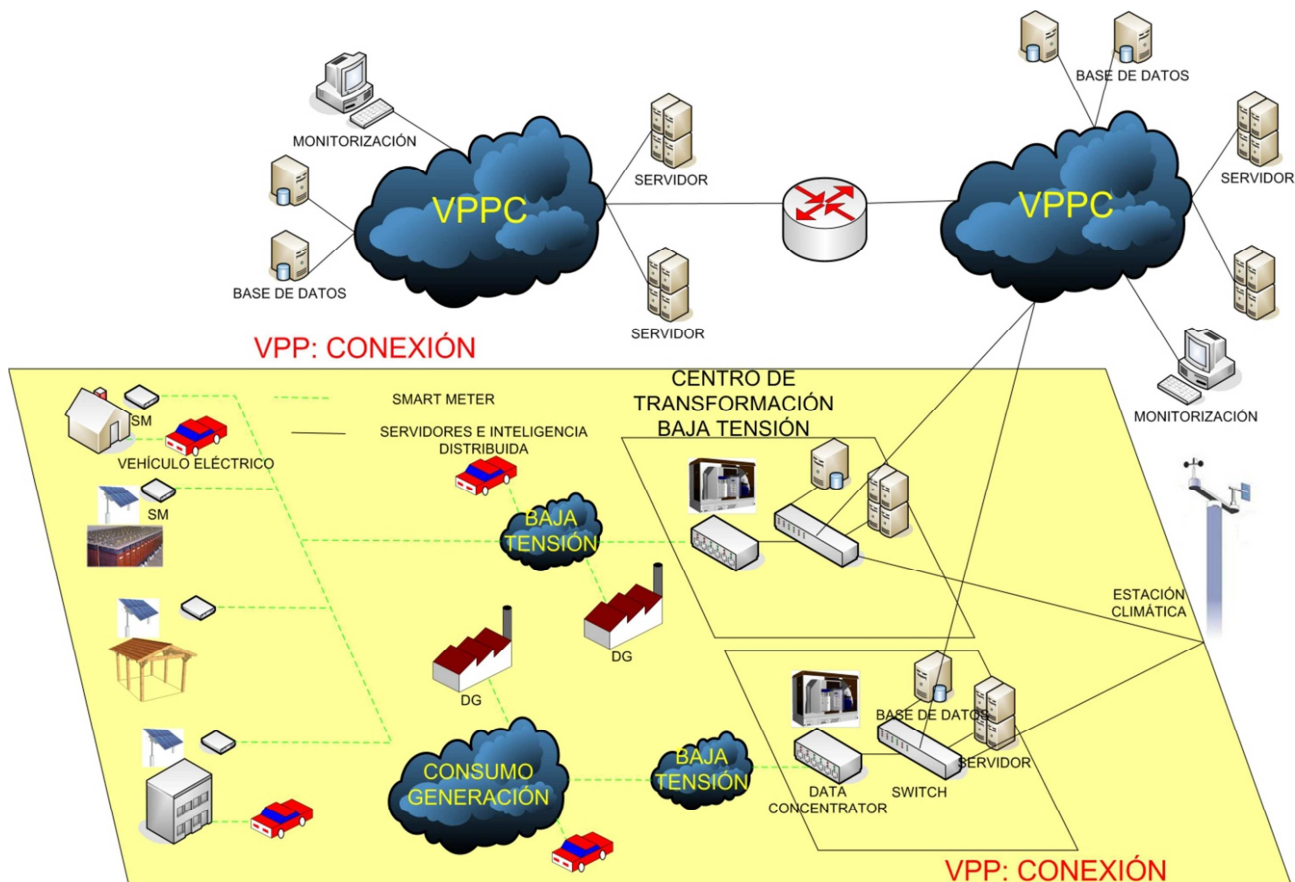


Figura 6.61. Inteligencia Distribuida a través de una VPP. Fuente Propia.

### 6.7.3.2. Procesos de control externos/internos en una Virtual Power Plant

Hay tres tipos de procesos que deben de ser controladas dentro de un *VPP*: cuestiones relacionadas con el mercado (*CVPP*), las relacionadas con el rendimiento y los problemas funcionales (*TVPP*), y las que involucran a ambas a la vez. Los agentes y procesos externos de la red de energía están listados aquí, indicando si dependen de *TVPP* y/o *CVPP*:

- Estado de la red (*TVPP*): se propaga el estado (por ejemplo eventos, fallos, etc.).
- Estado de la generación (*CVPP* y *TVPP*): comparte información acerca de la capacidad de generación actual y la previsión de generación futura.
- El estado de la demanda (*CVPP* y *TVPP*): distribuye la información sobre la demanda actual y las previsiones de demanda.
- Ofertas (*CVPP* y *TVPP*): las solicitudes/ofertas de energía de/desde elementos de la *VPP* hacia otros externos de la *VPP*.
- Tarifa (*CVPP*): difunde precios de la energía que ofrece la *VPP*.



- *Roaming* de generadores y cargas (*CVPP* y *TVPP*): comparte información acerca de los elementos que permiten intercambio de energía entre las *VPPs*.

Los procesos de control internos de *VPPC* se enumeran a continuación, indicando si dependen de *TVPP* y/o *CVPP*:

- Solución de problemas (*TVPP*): gestiona los posibles fallos de la red eléctrica y de la comunicación.
- Control de la generación (*CVPP* y *TVPP*): asegura que *DG* no cause desequilibrios red.
- Predicción de la generación (*CVPP*): predice la capacidad de generación por elemento.
- Control de la demanda (*CVPP* y *TVPP*): controla la demanda de energía en su área a través de los ratios de consumo de los clientes y en base a sus contratos.
- Predicción de demanda (*CVPP*): predice la demanda de la energía de los usuarios finales.
- Gestión del balance de la energía (*CVPP*): ofrece diferentes tarifas para optimizar el balance final de la energía, de acuerdo con la actual estrategia de control de la *VPP*.
- Actualizador de la tarifa (*CVPP*): informa a los clientes, en tiempo real, sobre las tasas fijadas por el gestor del balance de la energía.
- Control de cliente (*CVPP* y *TVPP*): gestiona los servicios del cliente.
- Control de estabilidad de red (*TVPP*): monitoriza y controla el rendimiento de red de la *VPP*, así como la calidad, la fiabilidad, y los parámetros de seguridad.
- Comunicación con otras *VPP* (*CVPP* y *TVPP*): comunicaciones bidireccionales con otros elementos, a nivel de *VPP*.
- Control de estación meteorológica (*TVPP*): gestión de la estación/es meteorológica/s, recopila y procesa sus datos; sirve en bruto y procesa la información para cualquier otro elemento de la arquitectura (por ejemplo, para la predicción de la demanda y la generación).
- Control del almacenamiento (*CVPP* y *TVPP*): monitoriza y controla el almacenamiento de energía de la *VPP*.

### 6.7.3.3. Enfoque con *Muti-Agent System*

El entorno de una *VPP* es una estructura compleja de procesos de toma de decisiones, funcionando por separado pero interdependientes. Las arquitecturas distribuidas son difíciles de controlar con los enfoques centralizados tradicionales. Por lo tanto, es necesario un nuevo sistema de control, y el paradigma de programación *MAS* se adapta perfectamente porque en estos sistemas, muchos actores interactúan con el objetivo competir o cooperar. Los agentes *Software* locales se centran en los subsistemas locales, e influyen en el sistema global mediante la negociación con los otros agentes *Software*; por otro lado, *MAS* implementa procesos de toma de decisión de forma abierta, flexible y extensible.

En este trabajo se presenta un modelo *MAS* para *VPP* con dos aspectos: la previsión de la demanda y la coordinación de los productores/consumidores con el fin de equilibrar la producción de energía. La primera es una solución pura de agentes, ya que no se centra en crear un marco, sino en la codificación de la solución bajo los principios de *MAS*. Por lo tanto, el algoritmo de predicción se controla utilizando agentes.

El segundo aspecto es abordado de manera diferente. En lugar de implementar el algoritmo distribuido, se crea un marco *MAS* implementando un modelo de roles diferentes y tareas en torno a la *VPP*, y especificando el comportamiento de cada agente. El diseño de este marco se lleva a cabo utilizando una solución de *Software* orientado a agentes llamado *INGENIAS* (Rodríguez–Fernández and Gómez–Sanz, 2010). *INGENIAS* sigue el paradigma basado en modelos; por lo tanto, los diagramas mostrados son convertidos en código del programa o documentos.

#### **6.7.4. Modelo Arquitectural para la Gestión y Predicción**

##### **6.7.4.1. Un marco orientado a Muti–Agent System para el control de Virtual Power Plant**

Se ha diseñado *MAS* teniendo en cuenta el comportamiento que esperamos de *VPP*. Es un diseño inspirado en las discusiones del grupo de trabajo durante el *workshop SOAR* (Rodríguez–Fernández and Gómez–Sanz, 2010). Los comportamientos se expresan en tres casos de uso en la que participan tres actores. Hay un productor que genera la electricidad, los consumidores exigen una cantidad fija de potencia; también hay consumidores flexibles que hacen ofertas especiales para reducir el consumo de energía cuando escasez. El mercado es asumido para regular el consumo de energía donde los consumidores acuerdan mediante contratos consumir menos energía a cambio de incentivos del mercado. Los tres casos de uso se definen como sigue:

- *Un productor identifica una escasez de energía.* En este caso, el productor detecta el problema y contacta con todos los consumidores flexibles para pedirles reducir su consumo de energía.
- *Un consumidor identifica una escasez de energía y contacta con el productor.* El consumidor junto con el productor necesita investigar lo que ha ocurrido.
- *Un consumidor requiere más energía.* Algunas instalaciones, como una gran fábrica, en ocasiones pueden demandar energía adicional. En este caso, el productor debe contactar con consumidores flexibles para pedirles que reduzcan la potencia consumida.

Se presenta un *MAS* que se ocupa de estos casos de uso. Los agentes del sistema están representados a través de roles, que son partes reutilizables de funcionalidad. Los agentes del sistema se representan en la Figura 6.62b, que coordinan los agentes ya identificados: productor, consumidor y consumidor flexible. Cada rol persigue un objetivo, expresado con la relación

*GTPursues*. Por ejemplo, además de la capacidad de consumo, el consumidor flexible trata de reducir su consumo cuando sea posible o cuando sea solicitado. Se espera que los roles cumplan con sus objetivos y se puede extender otros roles utilizando la relación *ARoleInheritance*, lo que implica que el rol extendido adquiere todos los objetivos y las tareas asociadas del rol principal.

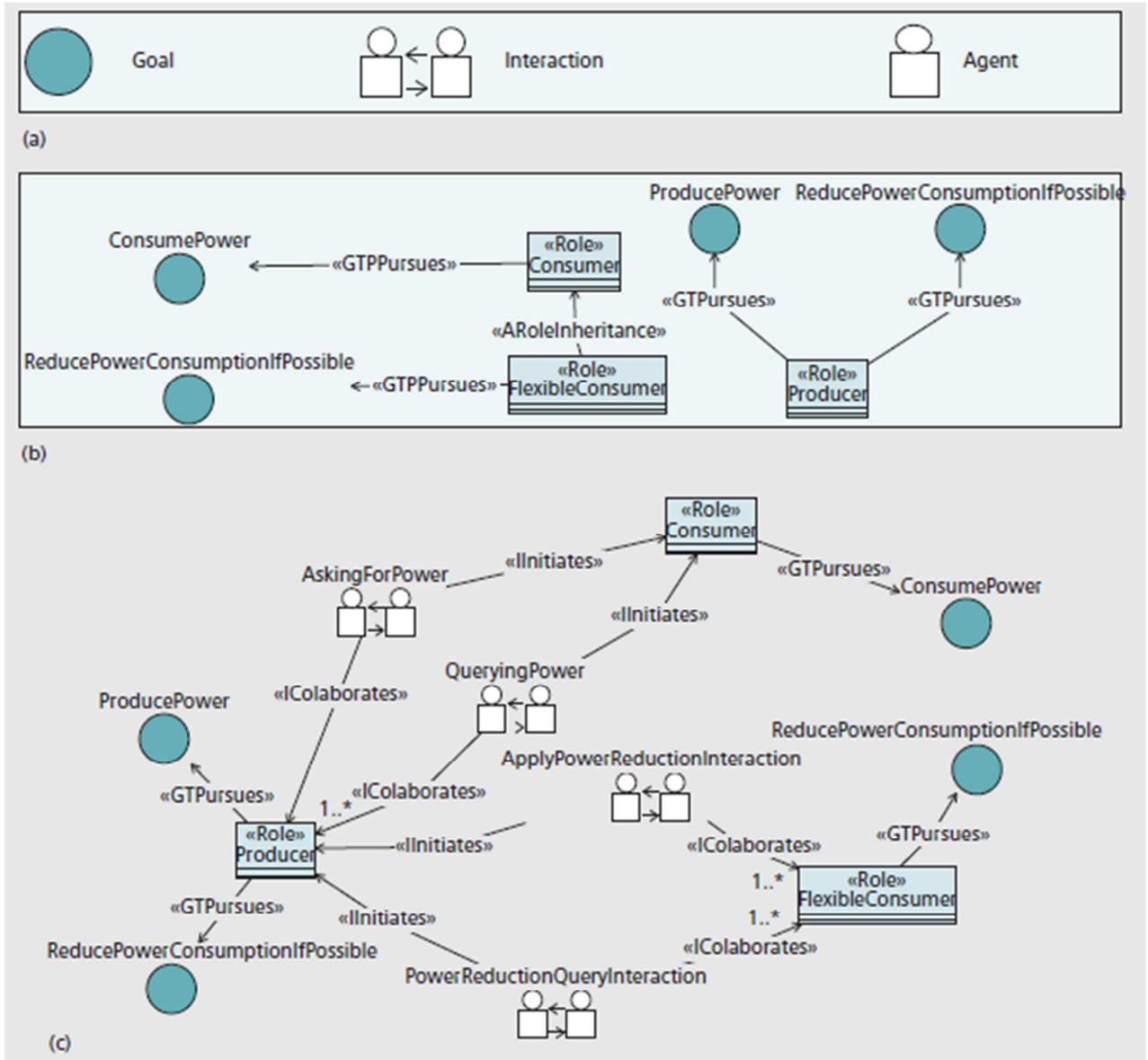


Figura 6.62. Modelo MAS del sistema de gestión y predicción: a) Notación *INGENIAS*; b) Actores; c) Interacción entre agentes. Fuente Propia.

Los agentes se espera que interactúe con el fin de realizar los casos de uso descritos anteriormente. Las interacciones se muestran en la Figura 6.62c involucran intercambios de la información entre agentes. En una interacción siempre hay sólo un agente para iniciar la interacción, llamado el iniciador (relación *IInitiates*), y al menos otros agentes están esperando para interactuar, que se llaman colaboradores (relación *IColaborates*). La figura 6.62c muestra que un consumidor pide a los productores la energía que precisa y a qué precios. Se trata de una interacción *one-to-many*, lo que significa que todos los productores de la red eléctrica son consultados. Una

vez que un único productor se selecciona, la disposición se lleva a cabo mediante una interacción diferente, *AskingForPower*. Estas interacciones se inician cuando hay necesidad de potencia adicional (tercer caso de uso) o se detecta escasez de energía por parte del cliente (segundo caso de uso). En este último caso, el cliente se adheriría a nuevas fuentes de energía. Cuando hay insuficiente energía para satisfacer la demanda y el productor lo sabe (primer caso de uso), interactuará con los clientes flexibles para reducir su consumo de energía. Cuando la reducción de energía se consigue, se inicia la interacción con *ApplyPowerReduction*.

Se ha implementado un modelo completo en base a *MAS*, por medio de *INGENIAS*. La Figura 6.63 muestra una de las pantallas utilizadas por la infraestructura *INGENIAS* para depurar las actividades realizadas por los agentes. *INGENIAS* emplea los modelos gráficos vistos en la Figura 6.62 para comunicarse con el programador. De esta manera, el programador recibe información de depuración usando los mismos conceptos aplicados durante el diseño. En este caso, el foco de la depuración es una actividad de consumo que toma como entrada un evento *Power Shortage*. El comportamiento diseñado para la tarea de partida implica una interacción de tipo *QueryingPower* (Figura 6.62c) con los productores de energía eléctrica alternativos. La cantidad solicitada de kilovatios de cada uno se almacena en una ranura de la entidad *Power Query*. Sin el modo de depuración, el agente debería proceder de manera automática y obtener la cantidad requerida de electricidad.

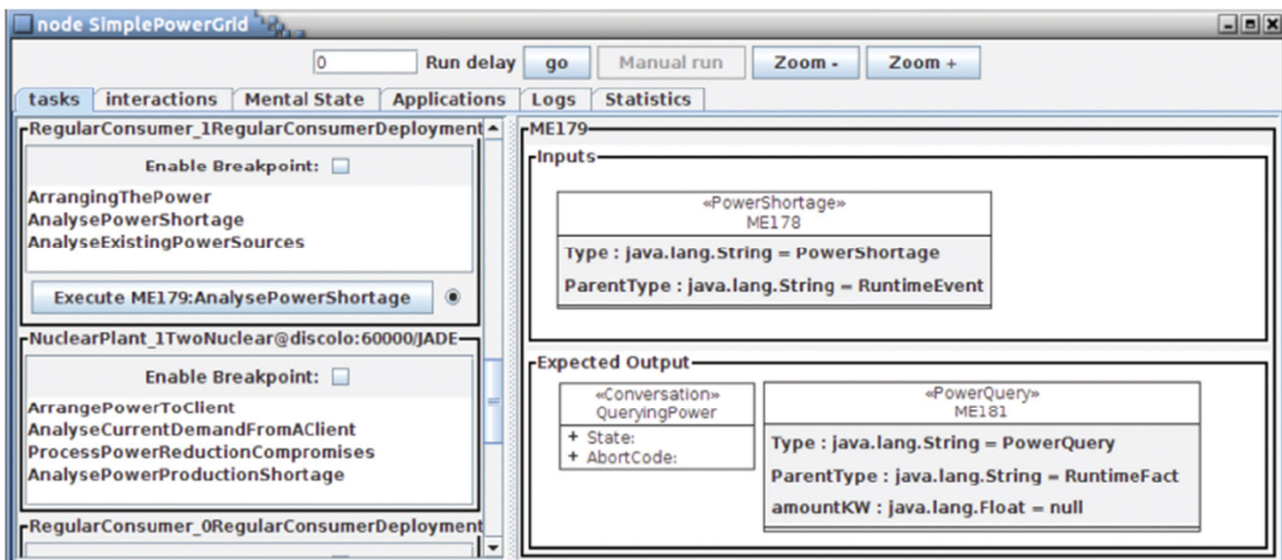


Figura 6.63. Ejecución de la especificación *MAS* con *INGENIAS*. Fuente Propia.

#### 6.7.4.2. Modelo Muti-Agent System para la predicción de la demanda eléctrica en Smart Home/Smart House dentro de Virtual Power Plant

De todos los procesos de control que se ha explicado anteriormente, uno de los más críticos para el funcionamiento de la *VPP* es la predicción de la demanda, lo que permite a la *VPP* saber de antemano la cantidad de energía que se requiere para que la planificación de su generación. Aquí

describimos la implementación que debería tener *MAS* para la Previsión de la Demanda (*Demand Forecasting – DeF*) en una *VPP*.

El objetivo de *DeF MAS* es predecir la demanda de energía de los usuarios de la *VPP* por sectores desagregados. Para realizar esta tarea, *VPP* necesita comunicarse con otros agentes y colaborar con el resto de procesos para garantizar la estabilidad la *VPP*. Las tareas de agente de *DeF* son:

- *DeF – Agente de Control de Históricos (Historical control Agent – DEF-HcA)*: este agente decide qué datos históricos utilizarse para el reentrenamiento de la red.
- *DeF – Agente de Control de Smart Meter (Smart Meter control Agent – DeF-McA)*: este agente es responsable de asegurar que los *SMs* están enviando datos de consumo correctamente.
- *DeF – Agente de Control de Smart House/Home (Smart House/Home Data control Agent – DeF-DcA)*: estos agentes se coordinan con *DeF-McA* para verificar la relevancia de los datos obtenidos, y comunicar esa información al agente *DeF-DpA*.
- *DeF – Agente de Preprocesado de Datos (Data preprocessing Agent – DeF-DpA)*: estos agentes son responsables de la normalización de los datos introducidos en base de datos, y además, detectar y resolver potenciales irregularidades.
- *DeF – Agente de Predicción (Forecasting Agent – DeF-FoA)*: estos agentes se encargan de activar la predicción de la demanda de acuerdo con un reloj común controlado por el sistema *DeF*.
- *DeF – Agente de Control de Re-entrenamiento (Retraining control Agent – DeF-RcA)*: este agente es responsable de controlar cuando la arquitectura del modelo de predicción necesita ser reentrenado.
- *DeF – Agente de Control de Nuevo Smart House/Home (New Smart House/Home control Agent – DeF-NcA)*: estos agentes están encargados de crear todos los agentes necesarios e informar de la aparición de este nuevo cliente al resto de agentes. También elimina los agentes cuando un cliente ya no utiliza los servicios de *VPP*.
- *DeF – Agente de Control Externo (External control Agent – DeF-EcA)*: la tarea de estos agentes es comunicar con el sistema *DeF* para aprender acerca de los intercambios de clientes entre las diferentes *VPPs* e informar al resto de agentes implicados en la *VPP*.

#### 6.7.4.3. Predicción de la demanda eléctrica

El sistema presentado en la sección anterior utiliza un subsistema de *DeF*, representado en la Figura 6.64. Este subsistema está basado en *ANNs*; mientras que la literatura presenta una gran cantidad de ejemplos de predicción de la demanda en entornos agregados (como países o grandes

extensiones), ésta tiene por objeto proporcionar una predicción de la demanda de energía en entornos pequeños (por ejemplo pequeñas ciudades).

ANNs son un paradigma de computación basada en la agregación de elementos pequeños de computación, llamados neuronas, que están interconectadas de manera similar al funcionamiento del cerebro humano. ANN puede presentar varias arquitecturas, y normalmente se configura para aprender de la experiencia a partir de los datos de la fase de aprendizaje. El sistema de predicción completo se muestra en la Figura 6.64a, donde las curvas de carga y variables climáticas se introducen en la base de datos.

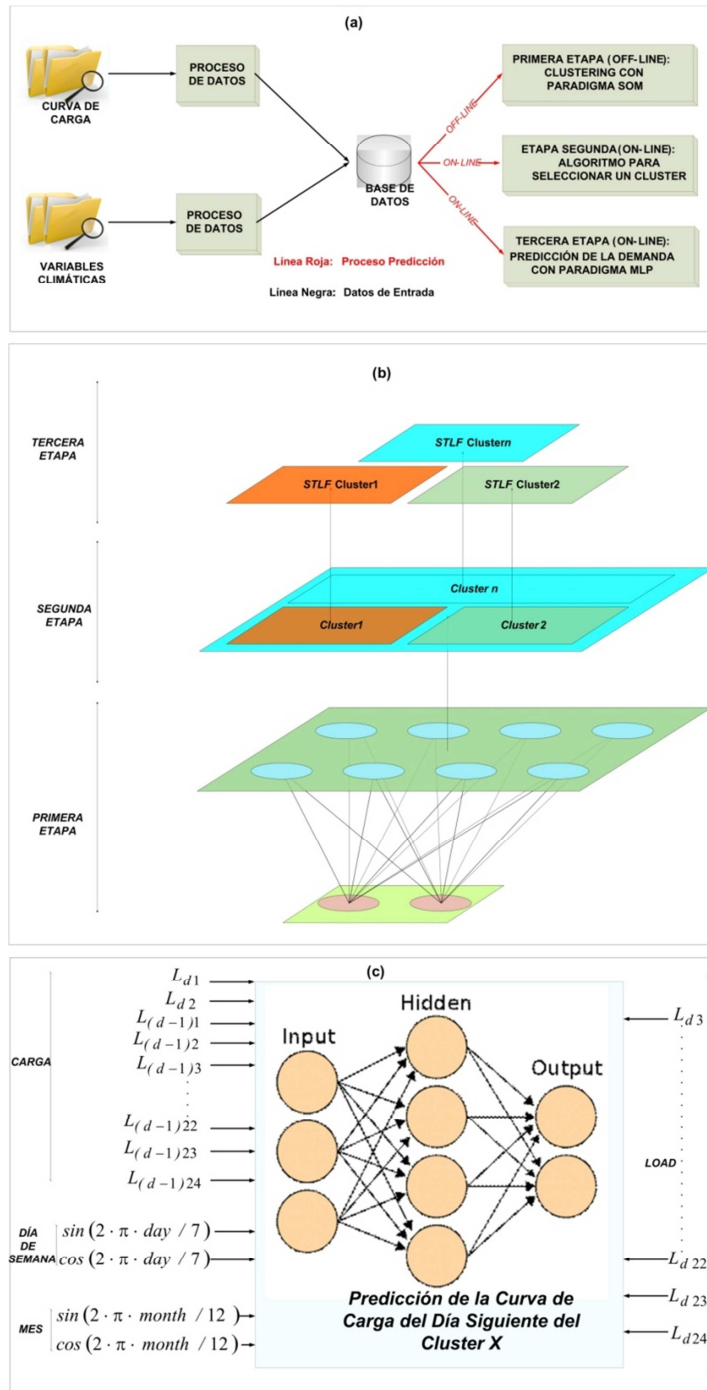


Figura 6.64. a) Sistema completo; b) Representación de las tres etapas; c) Entradas y salidas de los modelos MLP. Fuente Propia.

La primera etapa es un mapa de auto-organización *SOM* para el reconocimiento de patrones (y posterior clasificado) sin supervisión de los diferentes patrones de demanda a través de la curva de carga. Los patrones de demanda con características similares se agrupan juntos. Un grupo podría representar los días con menor consumo, como los fines de semana de verano y días festivos, mientras que un grupo diferente podría contener los días de mayor consumo, como los días de entre semana de invierno. Esta etapa se representa en la Figura 6.64b, y utiliza los siguientes parámetros como entrada: tres variables meteorológicas (*temperatura media*, *radiación solar global*, y *humedad relativa*), el tipo de sector (*industrial=0*, *comercial=1*, *doméstico=2*, *administración=3*, y *otros usos=4*), el día de la semana (*domingo=0*, *lunes=1*, ..., *viernes=5*, *sábado=6*), el mes (*enero=1*, *febrero=2*, ..., *noviembre=11*, *diciembre=12*) y la curva de carga.

La segunda etapa es un algoritmo para seleccionar a qué grupo pertenece un día, empleando para la decisión las variables meteorológicas, datos del calendario (día de la semana, mes y día festivos o día de trabajo) y la curva de carga.

Conjunto de *ANNs*, basado en *MLP*, uno para cada uno de los grupos identificados; realizan una predicción de las siguientes 22 horas de demanda (salida *ANN*). Esto se representa en la Figura 6.64c con las siguientes variables de entrada para cada patrón presentado en la fase de aprendizaje:

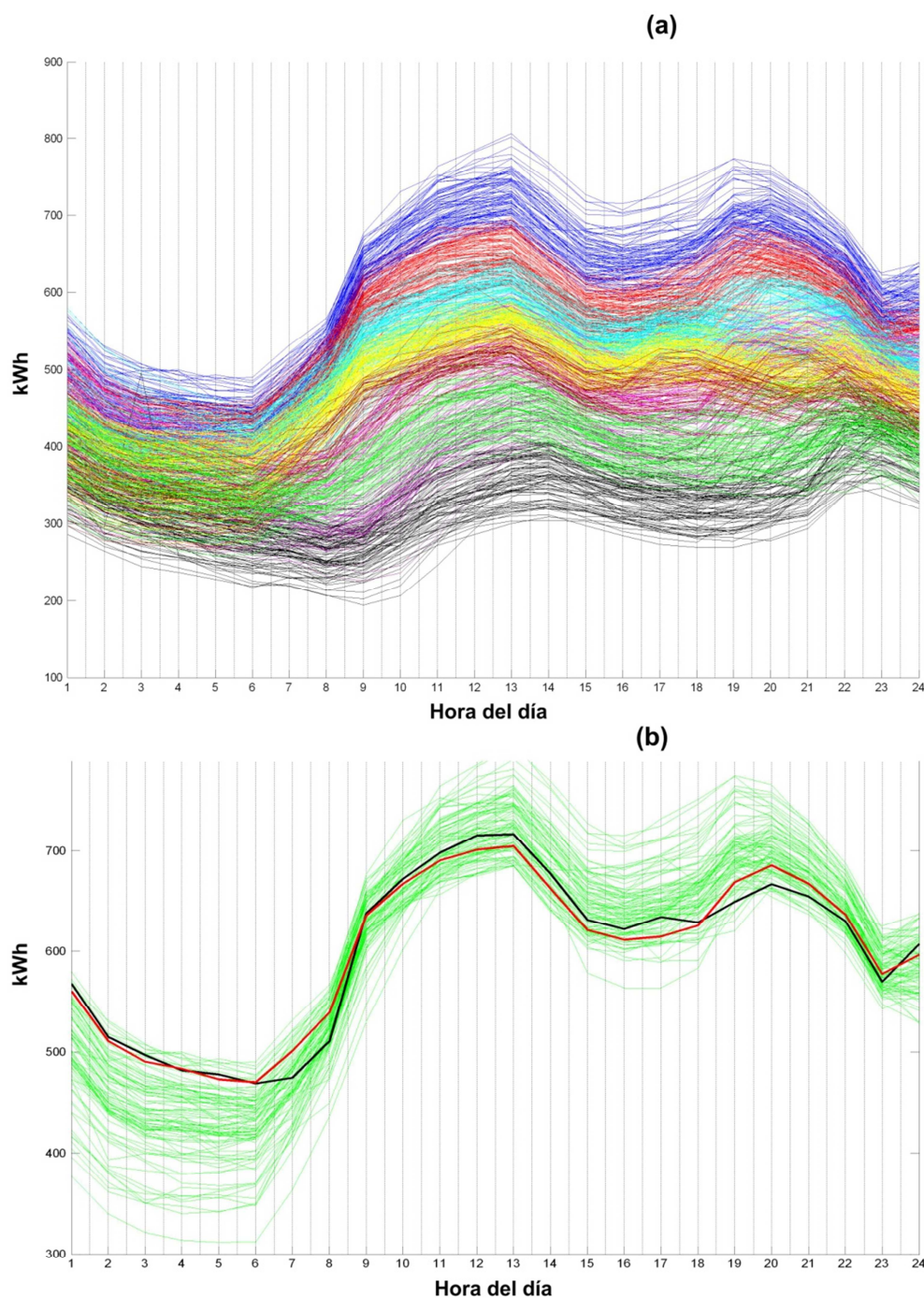
- $L_{d1}$  y  $L_{d2}$ : representan los dos valores de la curva de carga del día siguiente, de modo que la planificación de la generación/almacenamiento puede hacerse por adelantado por los agentes apropiados del sistema.
- $L_{(d-1)1}$ ,  $L_{(d-1)2}$ ,  $L_{(d-1)3}$ , ...,  $L_{(d-1)24}$ : representan los 24 valores de la curva de carga del día anterior.
- Día de la semana  $d-1$ : presentado como dos términos en forma de seno y coseno (para mejorar el rendimiento de *ANN*),  $\sin[(2 \cdot \pi \cdot \text{day}) / 7]_{(d-1)}$  y  $\cos[(2 \cdot \pi \cdot \text{day}) / 7]_{(d-1)}$ , con los días de 0 a 6 (*domingo=0*, *Lunes=1*, *martes=2*, *miércoles=3*, ..., *sábado=6*).
- Mes  $d-1$ : presentado como dos términos en forma seno y coseno,  $\sin[(2 \cdot \pi \cdot \text{day}) / 12]_{(d-1)}$  y  $\cos[(2 \cdot \pi \cdot \text{day}) / 12]_{(d-1)}$ , con los meses de 0 a 12 (*enero=0*, ..., *diciembre=12*).

Como se muestra en la Figura 6.64, las tres etapas están conectadas, aunque la primera etapa se lleva a cabo fuera de línea.

La Figura 6.65 ilustra un ejemplo del modelo *ANN*. La figura 6.65a presenta las curvas de carga agrupados después de la primera etapa, con cada grupo identificado por un color diferente. Es fácil ver que los patrones diarios de carga con características similares se agrupan. Hay varios hechos dignos de mención acerca de la carga: una región de bajo consumo de energía se observa durante las horas nocturnas, entre las 22h y 8h, mientras que los dos picos se encuentran

normalmente a las 11h–12h y 18h–19h; durante el período de bajo consumo, los patrones son bastante similares, y sus diferencias se incrementan durante el periodo de alto consumo.

La Figura 6.65b presenta un ejemplo de predicción de curva de carga, con la curva de carga objetivo en negro, la predicción en rojo, y las otras curvas de carga que pertenecen a su agrupación en verde.



**Figura 6.65. Resultados de predicción de curva de carga: a) Clusterizado de las curvas de carga (cada color es un clústers); b) Ejemplo de predicción de curva de carga: curva objetivo en negro, predicción en rojo y otras curvas del mismo grupo/clúster en verde. Fuente Propia.**



### 6.7.5. Conclusión

En este estudio se ha presentado un diseño *MAS* para su implementación en *VPP*. Este enfoque de la computación distribuida es perfectamente capaz de modelar las diferentes entidades involucradas en *VPP*, facilitando su interacción y gestión colaborativa. Aplicaciones con *MAS* han sido desplegadas en la literatura, pero por lo general, dedicadas a la gestión de las estructuras a gran escala (*SG*). Por el contrario, este trabajo presenta un modelo detallado de gestión de *VPP*.

A fin de que los agentes de *VPP* puedan tomar decisiones, necesitan información detallada acerca de lo que está sucediendo en la red, así como predicciones acerca de lo que potencialmente va a suceder en el futuro. Es por eso que este diseño *MAS* se ha complementado con un algoritmo de predicción de la demanda basado en *ANN*. Los algoritmos existentes de predicción se han utilizado para estimar demandas en zonas extensas (naciones, grandes regiones, etc.), pero se ha propuesto una predicción de demanda en un entorno desagregado y de menor tamaño, como pueda ser una *VPP* o una *microgrid*.



## CONCLUSIONES GENERALES Y LÍNEAS FUTURAS

En este último Capítulo se expondrán las conclusiones finales de la *Tesis Doctoral*, tratando de identificar los aportes realizados en la presente obra. Además, se listarán las revistas científicas donde se han validado los resultados obtenidos y, finalmente, se detallarán las líneas de futuro asociadas a la *Tesis Doctoral*.

### CONCLUSIONES GENERALES

Desde los orígenes del sistema eléctrico se ha tenido interés en tener una visión del comportamiento de la demanda, unas veces para su simple entendimiento y comprensión y, la mayoría de las veces, para poder realizar una planificación de las unidades de generación para poder abastecer a los consumidores. Se ha visto, que hace casi un siglo se empleaban sistemas tan sencillos como tarjetas perforadas con información de medias de los consumos anteriores, y así poder tener ciertas nociones del comportamiento futuro de la demanda. Poco a poco, a partir de la década de los cuarenta del siglo pasado, se comenzaron a aplicar modelos lineales para la predicción de la demanda, los cuales fueron evolucionando, hasta los modelos *ARMA* y sus variantes, que fueron empleados indiscutiblemente hasta la década de los ochenta del pasado siglo. La principal dificultad de estos modelos radica en tratar de interpretar el carácter aleatorio de la demanda e introducirlo en forma de ecuación en el modelo, dicho de otra manera, los parámetros caprichosos que afectan a la curva de carga, como las variables climáticas, efecto estacional, laborabilidad, entre otros, debe de ser perfectamente entendido y materializado en forma de ecuación matemática.

Con respecto al interés de la predicción de la demanda, se han presentado desde los orígenes, diferentes horizontes de predicción, siendo el de mayor interés para las empresas de energía la predicción de las 24 horas del día siguiente, ya que les permitirá obtener precios de la energía en los mercados inter/intra-diario.

A pesar de que hacia 1943 McCulloch and Pitts (1943) presentaron la posibilidad de aplicar el comportamiento de las neuronas al entorno computacional, no es hasta finales de los ochenta, comienzos de los noventa del pasado siglo, cuando comienzan a aplicarse modelos basados en *ANN* para la predicción de la demanda. La mejora en la operación de entrenamiento y, sobre todo, la capacidad de generalización y habilidad para detectar las no linealidades implícitas en la demanda eléctrica, hizo que los modelos basados en *ANN* se comenzaran a aplicar en mayor medida que los modelos *ARMA* y similares, en la etapa final del pasado siglo y en los comienzos del siguiente. A fecha de hoy, estos modelos son empleados en mucha mayor medida por la comunidad científica y la industria, debido a su extremada eficacia, rapidez de conformado de los ensayos y obtención de

resultados. El investigador se debe centrar en plantear las entradas de interés al modelo, y dejando el reto de encontrar asociaciones complejas intrínsecas con el problema a la propia red, obteniendo resultados asombrosos.

Además, el presente siglo ha traído consigo la conceptualización de nuevos entornos de generación, transporte, distribución y consumo de la energía eléctrica, que orbitan alrededor de *SG*. De la misma forma, la cada vez mayor penetración de la *DG* en las redes eléctricas, y la necesidad de una operación más eficiente del sistema, evitando las pérdidas no necesarias en transporte y distribución, ha propiciado la aparición de entornos desagregados (*microgrid*, *Smart Building*, entre otros), donde deberán convivir la generación, consumo, almacenamiento, y como no, información y conocimiento de lo que acontece y sucederá alrededor de estos elementos.

Esta desagregación y, por lo tanto, la aparición de estos nuevos “*espacios inteligentes*”, conllevarán la necesidad de seguir trabajando en presentar modelos de predicción, tanto de la demanda como de la generación, que con independencia del tipo de modelo, posibilitarán la introducción de nuevas variables más ajustadas al problema (climatología a escala altamente local, comportamiento y estancia de usuarios, empleo de las instalaciones, etc.), para poder emplearlos en los sistemas de gestión de operación y planificación de estos entornos de futuro. En estos entornos continuarán siendo de total interés las predicciones de: carga de la próxima hora, carga pico del día siguiente, carga total (agregada) del día siguiente y la curva de carga; por ejemplo, piense en la necesidad de planificar el almacenamiento eléctrico, para lo cual será necesario tener una previsión del estado de dicho almacenamiento para mañana, y saber la demanda/carga agregada/total del día siguiente, para poder hacer un ajuste; o suponga que en una *microgrid* se dispone de una planta de generación base (micro-turbina hidráulica) para hacer Afeitado de Picos (*peak shaving*), será indispensable conocer la carga pico, y así saber la cantidad de agua a turbinar para el próximo día.

Por último, la inteligencia llevará consigo el despliegue de medida, que por tanto, posibilitará el disponer de datos más recientes, creando históricos de comportamientos más actualizados, posibilitando que los modelos puedan ser ajustados (entrenados en el caso de las *ANN*) de manera más periódica y, por tanto, disponer de información de los cambios inherentes a la demanda (por ejemplo la influencia de los cambios sociales y económicos de la zona bajo control) que cada vez se producirán de manera más rápida en este mundo tan cambiante.

A continuación se enumerarán las conclusiones extraídas de la *Tesis Doctoral* presentada:

- Se ha planteado un concepto, que será el que aglutine a todos los existentes, llamado *Smart World*, y que servirá a comunidades diferentes para obtener información variada, la cual procederá de diferentes tecnologías e infraestructuras. Los objetivos globales de *Smart World*, los cuales son compartidos por los distintos “*Smart*” en mayor o menor medida, han sido presentados. El diseño de aplicaciones, servicios y protocolos, que cumplan uno,

algunos o todos los objetivos arriba indicados, independientemente que sean para *SG*, *microgrid*, *SC*, *SE*, etc.; formarán parte de un concepto de nivel superior, que será *Smart World*, formando así una jerarquía de entornos agrupados, y teniendo como origen a *Smart World*.

En concreto, se ha demostrado la relación existente entre ciertas variables climáticas y el comportamiento de la demanda eléctrica en un entorno semejante a una *microgrid*. La cambiante situación socioeconómica, junto a los cambios en la climatología en los últimos tiempos, hace que sea necesario una visión detallada de la forma en que las variables climáticas evolucionan en su efecto con la demanda eléctrica, necesitando por tanto, tratar de solapar medida proveniente de sensórica desplegada en diferentes infraestructuras para poder emplearla en aplicaciones novedosas que se empezarán a demandar en estos nuevos espacios de futuro (*SG*, *SC*, *microgrid*, *Smart Building*, etc.).

Los resultados obtenidos entre variables climáticas y demanda eléctrica, permitirán el diseño para validar nuevos modelos basados en *ANN*, para la predicción de la demanda eléctrica de entornos desagregados (subestación, *microgrid*, etc.). Para ello, se han empleado datos de un entorno real y poder obtener la clara relación entre algunas variables climáticas y la demanda eléctrica.

A modo de ejemplo, se ha presentado un posible desarrollo basado en *MAS* donde se recoge y trata la información de la sensórica desplegada en estaciones meteorológicas, para tratar de ayudar a los pronósticos de la demanda eléctrica.

- Se han aplicado técnicas no lineales junto a un algoritmo de clusterizado para tratar de agrupar las curvas de carga de un polígono industrial. En concreto, se ha empleado el paradigma *SOM* para un primer reconocimiento de patrones, para mediante una segunda etapa por medio del algoritmo de clusterizado *k-means* formar clúster de curvas de carga similares. El comportamiento del consumo diario de un polígono industrial real ha sido analizado mediante el clustering de días diferentes, de acuerdo con sus curvas de carga, y los patrones de comportamiento han sido correctamente identificados por el sistema y, además, de una manera completamente no supervisado.

Tras este agrupamiento con curvas de carga similares, se pretende poder acudir mediante un algoritmo decisor al clúster más idóneo, para posteriormente poder hacer una predicción de la demanda eléctrica del día siguiente.

Este tipo de algoritmos serán de gran utilidad para los *agregadores* de estos nuevos entornos (*microgrid*, *Smart Building*, etc.), ya que podrán planificar de una forma óptima sus fuentes de generación, su almacenamiento, pudiendo además hacer indicaciones de ajustes de la demanda si fuera preciso.

- Se ha propuesto un modelo basado en *ANN* para *STLF*, en entornos desagregados (ciudad pequeña, entorno rural o *microgrids*). La arquitectura del sistema está basada en una red *MLP*. Las variables relevantes de la entrada han sido seleccionadas con el fin de minimizar los errores de la predicción. Como se ha comentado anteriormente, la previsión al desagregar resulta más compleja, debido al aumento de la variabilidad de las curvas de carga, como consecuencia de la desagregación. Además, una predicción exacta en este tipo de entornos dependerá de las variables empleadas y la forma en que se introducen en la *ANN*.

También se ha presentado una metodología para poder entender la evolución del error de las predicciones con respecto al número de patrones de la fase de aprendizaje en los modelos basados en *ANN*. Se ha mostrado numéricamente que existe una estrecha relación entre los errores de predicción y el número de patrones presentados en la fase de aprendizaje, por lo que es necesario seleccionar cuidadosamente los datos a emplear.

En estos entornos de futuro (ciudades pequeñas, pueblos, *microgrids*, etc.), sujetos además a los cambios socioeconómicos, requerirán la actualización adecuada de las herramientas de pronóstico. Además, al disponer de datos provenientes de los *SMs*, se pueden plantear los momentos oportunos en los que volver a entrenar los modelos con datos recientes, a modo de “*ventana deslizante*” de datos para la fase de aprendizaje.

Finalmente, se ha demostrado que el concepto de predicción de carga mediante *ANN* son aplicables en entornos desagregados, del tipo de ciudad pequeña o *microgrid*, obteniéndose muy buenos resultados, y mostrando que podrían alcanzarse un error total de la fase de operación en torno a 2.47%.

- Tomando como punto de partida los buenos resultados obtenidos anteriormente en la predicción de la curva de carga del día siguiente, unido a las conclusiones del estudio del polígono industrial, se ha creído interesante el probar a agrupar las curvas de carga y tratar de obtener una predicción mejorada.

Se ha planteado un modelo basado en tres niveles: un reconocimiento de patrones mediante *SOM*, un posterior clusterizado por medio del algoritmo *k-means*, y finalmente un modelo *MLP* para cada uno de los clústers obtenidos para hacer *STLF*. El objetivo de este planteamiento es tratar de formar grupos con curvas de carga especialmente parecidas, de modo que el posterior ajuste de la predicción (mediante *MLP*) obtenga resultados mejorados, ya que durante su fase de aprendizaje se habrán empleado curvas de carga similares, obteniendo unos ajustes sinápticos mejorados.

De la misma forma, y basándose en la metodología para controlar la evolución error de la fase de operación con respecto al número de patrones de la fase de aprendizaje, se ha confirmado que la evolución del error frente al número de patrones tiene mejora (para cada

una de las *MLPs* de cada grupo), habiéndose demostrado analíticamente, que al disponer de más patrones para cada clúster, el modelo *MLP* (tras un nuevo entrenamiento) proporcionará mejores resultados de predicción.

- Para el correcto funcionamiento de los escenarios de futuro (*SG/VPP/microgrids*), se precisarán herramientas que permitan su monitorización y control. Estas últimas, se basarán en modelos de predicciones exactas de diferentes parámetros (demanda agregada, curva de carga, etc.). Por tanto, se ha presentado un análisis de variables climáticas sobre la demanda agregada, así como la propia influencia de ésta con respecto a días pasados, con el fin de tener una idea de las variables a emplear en los modelos de predicción a proponer. Esta metodología, puede emplearse como referencia para estudios posteriores, y ayudar a identificar estas relaciones o incluso otras con nuevas variables, para así plantear modelos de predicción evolucionados, tanto para la demanda agregada como para otros parámetros de interés a estimar.

Se han presentado siete modelos de predicción de la demanda agregada, tratando de presentar mejoras al aportar nuevas variables de entrada a los modelos. El aporte de nuevas variables se ha basado en el estudio anteriormente comentado. Con los resultados de cada modelo, se ha planteado la posibilidad de emplear el mejor modelo con climáticas (*F\_AL\_W\_DW\_SR*) o el mejor modelo sin ellas (*F\_AL\_W\_DW*), dependiendo la época del año en la que se esté.

Ha vuelto a ponerse de manifiesto la necesidad de conocer las variables más relevantes con respecto a la demanda agregada. En estos nuevos entornos acotados físicamente, disponer de información meteorológico o de otra índole, supondrá la posibilidad de emplear en los modelos predictivos nuevas variables que conseguirán predicciones más exactas. Además, el pronóstico de la demanda agregada del día siguiente es de especial importancia para poder hacer *STLF* del día siguiente, ya que se mejora la predicción ostensiblemente tal y como se ha visto en las primeras experiencias iniciales.

- Se ha demostrado que empleando información del día a predecir es posible mejorar la predicción al hacer *STLF*. La información empleada del día siguiente, la cual se ha estimado igualmente, han sido: los dos valles del día a predecir, los dos picos del día a predecir y la demanda agregada del día a predecir; que junto a otras variables de entrada presentadas, hará que la estimación global del día siguiente mejore sustancialmente. Para validarlo se ha realizado la misma experiencia con el modelo presentado en la sección 6.3. El error de toda la fase de operación ha sido de 1.62%, frente al 2.47% que se obtuvo sin la información estimada de los dos picos y los dos valles de la curva de carga del día a predecir (si se empleó en aquel modelo la estimación de la demanda agregada).

En estos nuevos entornos desagregados, se presenta como fundamental intentar anticipar valores de la curva a predecir. En este caso estimaciones de las zonas críticas (cambio de concavidad–convexidad) ha supuesto una mejora considerable del pronóstico. Otro tipo de información, como por ejemplo la planificación de procesos productivos (en el caso de polígonos industriales) o estancias de empleados o usuarios de las instalaciones, podrían mejorar las estimaciones de la demanda eléctrica si se introdujeran en el modelo de aprendizaje.

El despliegue de *microgrids* es una cuestión inminente, con lo que será preciso su control y operación; además, las *microgrids* son lugares idóneos para hacer *DR*, ya que entre sus objetivos estará el tratar de ajustar la generación a la demanda, por lo que la predicción de esta última se considera primordial para el balance de las mismas. Por lo tanto, el desarrollo de modelos basados en *ANN* para *STLF* serán necesarios, y la exactitud de las predicciones determinará la mejor operación de estos entornos de futuro.

- Parece inminente la incorporación de fuentes de generación en lugares cercanos a los consumos. En este escenario, las ciudades y/o entornos rurales parecen los destinados para este nuevo despliegue de *DG*, y si se pretende conseguir los objetivos de *Horizon 2020*, la mayoría de estos nuevos recursos de generación deberían ser renovables. Para poder gestionar y monitorizar tanto los generadores dispersos como las cargas de las mismas zonas, surge el concepto de *VPP*, el cual tratará de controlarlos (generación y demanda), viéndolos como un agregado, necesitando para tal fin el despliegue de elementos de medida e inteligencia.

Se ha presentado un diseño conceptual basado en *MAS* para su implementación en *VPP*, siendo capaz de modelar los diferentes actores involucrados en *VPP*, facilitando su interacción y gestión colaborativa. Se han detallado los procesos internos y externos a la *VPP*, desde el punto de vista de la gestión de las predicciones de la demanda.

A fin de que los agentes de *VPP* puedan tomar decisiones, necesitan información detallada acerca de lo que está sucediendo en la red, así como predicciones acerca de lo que potencialmente va a suceder en el futuro. El diseño conceptual *MAS* se ha complementado con un algoritmo de predicción de la demanda basado en *ANN*. La arquitectura propuesta del modelo de predicción se ha presentado en su conceptualización.

## **VALIDACIÓN DE RESULTADOS**

Seguidamente se listarán las revistas científicas donde se han validado los resultados propuestos en la *Tesis Doctoral*. Además, se identificarán los diferentes proyectos de I+D+i donde



CIEMAT participa, y algunas de las actuaciones propuestas tienen que ver con los desarrollado en esta *Tesis Doctoral*.

### ***Publicaciones en revistas científicas***

Como ya se comentó en la metodología de la *Tesis Doctoral*, los resultados derivados de los estudios planteados en la *Tesis Doctoral* han sido validados en diferentes revistas científicas. A continuación se listan las revistas y el título del artículo:

- “*Journal*” ***Sensors*** (Impact Factor: 1.739 (2011); 5–Year Impact Factor: 2.060 (2011)) bajo el título “*Performance Study of the Application of Artificial Neural Networks to the Completion and Prediction of Data Retrieved by Underwater Sensors*”. Enviado el 27 de diciembre de 2011 y publicado el 2 de febrero de 2012.
- “*Journal*” ***Sensors*** bajo el título “*A Study of the Relationship between Weather Variables and Electric Power Demand inside a Smart Grid/Smart World Framework*”. Enviado el 6 de Julio de 2012 y publicado el 27 de agosto de 2012.
- “*Journal*” ***IEEE Communications Magazine*** (5–Year Impact Factor: 3.785) en su *Special Issue on Ultimate Technologies and Advances For Future Smart Grid – UTASG*, bajo el título “*A Multi–Agent System Architecture for Smart Grid Management and Forecasting of Energy Demand in Virtual Power Plants*”. Enviado el 4 de octubre de 2012 y publicado el 4 de enero de 2013.
- “*Journal*” ***Energies*** (Impact Factor: 1.865 (2011)) en su *Special Issue on Smart Grid and the Future Electrical Network*, bajo el título “*Classification and Clustering of Electric Demand Patterns in Industrial Parks*”. Enviado el 9 de octubre de 2012 y publicado el 12 de diciembre de 2012.
- “*Journal*” ***Energies*** en su *Special Issue on Hybrid Advanced Techniques for Forecasting in Energy Sector*, bajo el título “*Short–Term Load Forecasting for Microgrids Based on Artificial Neural Networks*”. Enviado el 28 de noviembre de 2012 y publicado el 5 de marzo de 2013.
- “*Journal*” ***Energies*** bajo el título “*Experimental Analysis of the Input Variables’ Relevance to Forecast Next Day’s Aggregated Electric Demand Using Neural Networks*”. Enviado el 29 de marzo de 2013 y publicado el 6 de junio de 2013.
- “*Journal*” ***IEEE Communications Surveys and Tutorials*** (Impact Factor: 3.692) en su *Special Issue on Energy and Smart Grid*, bajo el título de “*A survey on electric power demand forecasting: future trends in Smart Grids, Microgrids and Smart Buildings*”. Enviado el 2 de abril de 2013, en estos momentos en fase de aceptación.

- “Journal” *International Journal of Green Energy* (Impact Factor: 1.188 (2011); 5–Year Impact Factor: 1.038 (2011)) bajo el título “*Artificial Neural Networks for Short–Term Load Forecasting in Microgrids Environment*”. Enviado el 1 de junio de 2013, en estos momentos en fase de aceptación.
- “Journal” *Energies* bajo el título “*Improved Short–Term Load Forecasting Based on Two–Stage Predictions with Artificial Neural Networks in Microgrids Environment*”. Enviado el 18 de julio de 2013 y publicado el 29 de agosto de 2013.

## ***Proyectos en desarrollo***

En estos momentos *CIEMAT* participa activamente en proyectos de I+D+i que, en mayor o menor medida, tienen que ver con la aplicación de los resultados obtenidos en esta *Tesis Doctoral*. A continuación se expondrán las principales características de tres de los proyectos, cuyos acrónimos son: *S.O.R.I.A (+ x -)*, *MIRED–CON* y *OVI–RED*.

### *S.O.R.I.A (+ x -)*

El resultado esperado es un programa informático que permitirá gestionar y optimizar el consumo energético en ayuntamientos de tamaño pequeño y medio, incluyendo la gestión de sus flotas de *EVs* y de posibles fuentes de generación renovable. El proceso en su conjunto se llevará a cabo mediante la captación de información procedente de las instalaciones, equipamientos, infraestructuras y vehículos, a partir de la cual se podrá conocer cómo y dónde se consume la energía, tomar decisiones y plantear actuaciones de optimización de consumos energéticos y de mejora de rendimientos, integrándose todo ello en un programa de fácil utilización y manejo. Determinados elementos de decisión del *Software*, orientado a la toma de decisiones de mejora, estarán desarrollados bajo el paradigma *MAS*, integrándose en su conjunto con el resto de la herramienta de aplicación.

*S.O.R.I.A. (+ x -)* aborda el planteamiento de soluciones mediante algoritmos predictivos (demanda eléctrica y generación con renovables) capaces de valorar consumos, calcular la mejora del rendimiento energético en instalaciones, equipamientos y en una flota, en este caso simulada, de vehículos eléctricos, y mostrar la cantidad de energía no consumida con su correspondiente ahorro de gases de efecto invernadero. El objetivo que se persigue es facilitar a las administraciones públicas municipales indicaciones que les permita planificar e implementar medidas destinadas a influir en el modo de consumir energía en el ámbito de los servicios públicos, de manera que se produzcan los cambios deseados en la curva de carga.

### *MIRED–CON*

Desde hace ya unos años, se está tratando de realizar un cambio conceptual sobre el actual sistema energético para, principalmente, obtener mayor sostenibilidad y eficiencia energética, de

una manera más compatible con el medio ambiente, siendo este cambio de modelo de sistema una de las claves para el desarrollo de una mejor economía sostenible de acuerdo con el tratado de Lisboa. La necesidad de los consumidores de información respecto a su consumo, así como el poder hacer realidad la *DR*.

Reglamentado por los diferentes Gobiernos, se les ha planteado a las *utilities* el reto de incorporar nuevos dispositivos inteligentes en lugares que hasta el momento no existían, por ejemplo en los puntos finales de consumo donde aparecerán para la toma de medida los *SMs* y en los intermedios, principalmente centros de transformación, los concentradores de datos provenientes de los *SMs* a través de *DCs*.

Según datos de la Comisión Europea, se estima que para 2020 la penetración de generación distribuida en red en Europa será en torno al 20–25% de la generación total y en 2030 esta cifra se establecerá en torno al 30–35%. En esta revolución energética juegan un papel principal los cambios en la infraestructura eléctrica. Es necesaria una profunda transformación del sistema eléctrico y de las líneas de transmisión y distribución convencionales hacia una red de suministro eléctrico interactiva y estándar. Importantes avances tecnológicos y cambios regulatorios y normativos que deberán llevarse a cabo a corto–medio plazo, serán la base para el nuevo modelo de suministro eléctrico.

El proyecto *MIRED–CON* consistirá en la instalación de una infraestructura de medida y control avanzados sobre una red que pretende ser autogestionada energéticamente, convirtiendo a esta nueva red en una referencia de lo que pueden ser las redes de distribución del futuro. Esta red de generación distribuida contendrá generadores de tecnologías diferentes (eólicos, fotovoltaicos, minihidráulica, entre otros), almacenamiento (baterías como sistema de base, volante de inercia como sistema rápido) y cargas pasivas y activas.

Para ello se partirá de una red disponible en *CEDER–CIEMAT* que presenta unas características interesantes de distintos niveles de tensión, topología diversa, varios centros de transformación de diferente tipo, generación distribuida, consumos de distinto tipo, cargas controlables y almacenamiento de energía.

El objetivo del proyecto es poder gestionar los elementos activos de la red para conseguir un determinado funcionamiento o situación que interese a la propia red o a la *utility* que suministra su potencia. Para ello se podrán establecer distintos criterios de operación:

- Mínimo consumo de la red (para aumentar el nivel de autonomía).
- Mínimo nivel de carga en las líneas de transmisión de la red.
- Mínimo nivel de carga en un determinado punto de la red.
- Estado de mínimas pérdidas.
- Maximización del beneficio económico.

- Crear una nueva tecnología Comunicaciones en Línea de Energía (*Power Line Communications – PLC*) en media tensión que satisfaga los siguientes requisitos de aplicación:
  - Alcance > 5 km en líneas de media tensión (sin derivación).
  - Throughput de 500 kbps (nivel de aplicación).
  - Adaptación dinámica a condiciones del canal. Desarrollo opción tecnológica basada en Multiplexación por División de Frecuencias Ortogonales (*Orthogonal Frequency Division Multiplexing – OFDM*), extensión de Evolución de la Medida Inteligente PowerLine (*PowerLine Intelligent Metering Evolution – PRIME*) a media tensión.

Validación y demostrar dicha tecnología y pruebas de rendimiento en red de *CEDER–CIEMAT*. En función del criterio que se elija, el sistema se programará con unas determinadas actuaciones (que para una determinada red serán estudiadas con anterioridad para establecer un algoritmo que en todo momento determine las consignas de operación para cada uno de los elementos) que fijarán la operación de cada dispositivo conectado a la red.

La comunicación de la medida bajo *PLC* parece la solución más sensata y menos costosa, ya que la infraestructura eléctrica estará desplegada y no parece necesario el sobrecargar el espectro radioeléctrico. Además, y adicionalmente a la medida, surge la oportunidad de emplear *PRIME* como tecnología de control de los diferentes elementos distribuidos en un espacio controlado (generación y almacenamiento).

Las ventajas de disponer de una infraestructura avanzada de medida y control de generación, almacenamiento y posible demanda, de cara a la propia red serán:

- Control y estado de los tramos de línea existentes, teniéndolos en cuenta en la operación del sistema y con su consiguiente beneficio de seguridad.
- Optimización de la generación y consumo en el tramo local de la red de distribución, de forma que se pueda presentar un balance de energía casi neutro hacia la red de transporte, y de esta forma optimizar la capacidad del sistema y mejorar la eficacia en el uso de las fuentes renovables y de generación distribuida.
- Minimización de las pérdidas locales.
- Posibilitar la penetración de las renovables y de manera distribuida.
- *PLC+PRIME* como solución Última Milla (*last mile*) para medida y supervisión y control en baja tensión.

El atractivo para a las *utilities*:

- *PLC+PRIME* como solución al control a través de media tensión, desde su centralización hacia sus elementos distribuidos de generación y almacenamiento.

- La inteligencia y control de la generación, almacenamiento y demanda, repercutirán en un suministro de la energía de manera más eficiente y controlada por parte de la *utility*.
- Eliminación de las pérdidas por transporte de energía innecesarias.

Para poder realizar lo anterior, y como elementos importantes del proyecto, caben destacar los siguientes:

- El control central del entorno de pruebas inteligente, deberá disponer de predicciones de la demanda y de la generación. Además, y debido a que los centros de transformación clásicos se convertirán en “*nodos inteligentes de transformación*”, al dotarlos de elementos de comunicaciones y con cierta inteligencia, estas predicciones serán elaboradas para uno de los nodos pertenecientes a la red. Una vez realizadas las estimaciones (demanda y generación), se mandarán a cada uno de los anteriores nodos.
- Estos “*nodos inteligentes de transformación*”, además de albergar capacidades de medida y protección, permitirán integrar inteligencia adicional. Se desplegará una arquitectura basada en *MAS*, a nivel de centro de transformación, donde se recibirán planes energéticos (basados en las predicciones) desde el centro de control, y cada nodo podrá dialogar con sus iguales para establecer las reglas de operación del almacenamiento y generación gestionable a su cargo.

### OVI-RED

El proyecto *OVI-RED* propone diseñar, desarrollar e implementar un sistema para la gestión de un conjunto de *microgrids*, las cuales a su vez, gestionan individualmente los recursos contenidos dentro de su *microgrid* local con presencia de almacenamiento energético distribuido de variada tecnología, capacidad energética y gestionabilidad, utilizando como base principal el concepto de *VPP*.

El sistema propuesto estará compuesto por diversas capas de control y gestión jerarquizadas que permitan operar los recursos energéticos distribuidos de la *microgrid* en base a criterios de estabilidad y fiabilidad de la red, optimización del despacho de energía y maximización del beneficio económico proporcionado por el intercambio energético con la red principal. El proyecto abordará el desarrollo de los sistemas de control a distintos niveles para cumplir debidamente los requerimientos relacionados con la gestión de aspectos técnicos y con los aspectos económicos relacionados con la optimización del intercambio energético con la red principal de distribución. La gestión de una *microgrid* se puede considerar desde dos puntos vista:

- Técnico. Integra el control operativo de los diferentes recursos distribuidos, manteniendo la estabilidad y la fiabilidad de una *microgrid*. Además existe la posibilidad de proporcionar distintos tipos de servicios (reserva, regulación de frecuencia y tensión, etc.) a la red principal.

- Económico. Integra y gestiona de forma óptima el despacho de energía según criterios económicos del mercado de la energía facilitando las transacciones comerciales a través de los mercados de compra-venta de energía. Desde un punto de vista económico la gestión óptima de la energía se consigue por medio del ajuste de la generación y demanda de la *microgrid* en base a los precios y casación del mercado. Para este cometido, resultará fundamental el disponer de modelos de predicción de la demanda y la generación de cada una de las tres *microgrids* participantes en el proyecto.

## LÍNEAS FUTURAS

Se presenta la oportunidad para el desarrollo de nuevas herramientas y aplicaciones que empleen datos procedentes de sensórica implantada en diferentes infraestructuras (eléctrica, agua, estaciones climáticas, etc.). Estas nuevas herramientas/aplicaciones serán demandadas por la sociedad presente/futura, ya que una sociedad informada y formada será una sociedad más inteligente. En esta *Tesis Doctoral*, se ha planteado un concepto aglutinador de los entornos “*Smart*” llamado *Smart World*, precisamente para llamar la atención de la necesidad de herramientas/aplicaciones que aglutinen información procedente de diferentes infraestructuras, y con unos objetivos comunes, los cuales servirán para satisfacer las necesidades de sociedades futuras.

La aparición de medida desplegada en las redes de energía, junto a la inteligencia asociada a la misma, posibilitará la aplicación de técnicas ya empleadas en el pasado, pero para nuevos entornos desagregados. Esto es el caso de las predicciones de la demanda eléctrica, que junto a las estimaciones de la *DG*, serán piezas fundamentales en la operación y control de los entornos desagregados, como puedan ser las *microgrids*, *Smart Building*, etc.; además, con la posibilidad de integrar elementos *Software* (“*inteligentes*”) dentro del *Hardware*, se abre un nuevo horizonte para nuevas aplicaciones de futuro.

La posibilidad de datos en estos nuevos entornos desagregados, permitirá la operación de los mismos de una manera controlada, posibilitando además la irrupción de fuentes de generación renovable en ellos. Con respecto a los modelos de predicción, los entornos desagregados como *microgrids* o *Smart Buildings*, permitirán el uso de nuevas variables que desde una perspectiva agregada (regiones extensas o países) no se podrán concebir. Por ejemplo, un país como España tiene una variación de temperatura considerable, por lo que un modelo que aglutine la demanda eléctrica de todo el país, difícilmente podrán emplear un valor de temperatura, radiación solar o humedad relativa que represente todo el país; en cambio, en estos nuevos espacios delimitados territorialmente (edificio, pueblo, polígono industrial, ciudad), estas variables climáticas (o de otro tipo) sí que representarán claramente al emplazamiento y su relación con la demanda eléctrica será

clara, pudiéndose emplear directamente en los modelos. Otro tipo de información a emplear como variables de los modelos, podrán ser las que definan hábitos directos de comportamiento, como puedan ser: estimaciones de habitabilidad de los inquilinos de una vivienda (*Smart Home* o *Smart Building*); o planificaciones de operación en entornos industriales, los cuales podrán servir a los modelos predictivos de la demanda eléctrica para ajustar la predicción.

Con independencia del modelo, y la complejidad del mismo, para la predicción de la demanda eléctrica, se precisará una integración sencilla y rápida de dichos modelos con las herramientas de gestión de la demanda. De esta forma, los investigadores podrán proponer nuevos modelos, que una vez testeados y validados, se podrán poner en producción de manera rápida, sin mucha intervención de recursos humanos.

Con respecto a las *ANN* para la predicción de la demanda, parece que con modelos “sencillos” como *MLP* se obtienen unos resultados óptimos. Lo que parece apropiado es el empleo de sistemas en varias etapas, ya que la exactitud del modelo parece que se mejorará, teniendo en cuenta un número de patrones similares. Otra alternativa, aunque con mayor complejidad, son los sistemas híbridos. Por tanto, para estos nuevos entornos desagregados, se plantea el reto de optar por modelos sencillos (*MLP* o similares) y centrarse en la detección de nuevas variables a incorporar, o continuar trabajando en modelos de múltiples etapas o híbridos.

Como se ha mostrado en esta *Tesis Doctoral*, la principal ventaja de los modelos de predicción basados en *ANN* con respecto a los modelos *ARMA* (o similares) es que el investigador deja la responsabilidad de interpretar las no linealidades al modelo (en el caso de *ANN*), por lo que el investigador puede centrarse en otras cuestiones, como la detección de nuevas variables que aporten información que ayude a obtener predicciones más exactas. Por el contrario, la complicación de los modelos *ANN* radica en que necesitan la optimización del modelo, esta optimización depende principalmente del número de patrones de la fase de entrenamiento y de la complejidad arquitectural de la *ANN* (número de variables de entradas, número de capas ocultas, número de neuronas de cada capa, etc.). Por tanto, el empleo de *algoritmos genéticos* o técnicas de *neurocomputación* para la optimización de los modelos de manera desatendida serán de especial interés. Aunque los modelos de predicción pueden considerarse cajas negras que esperan unas entradas para obtener unas salidas, deben entenderse como inmersos en un *TODO*, donde la aparición de nuevos datos (potenciales patrones) e incluso nuevas variables a emplear en los modelos, harán que dichos modelos puedan ser adaptados en el tiempo. Por lo que, en segundo plano es posible proponer nuevos modelos de predicción alternativos a los existentes en producción, y si de manera desatendida, la propuesta de optimización de la arquitectura puede ser automatizada, sería de gran utilidad para los ingenieros, los cuales en última instancia deberán tomar la decisión de si sustituyen o no el modelo de predicción actual.

Continuando con lo anterior, se plantea la necesidad de sistemas inteligentes que sean capaces de controlar y entender la evolución de los errores de la predicción de los modelos predictivos que están en producción. El cometido de estos sistemas inteligentes será el detectar anomalías de las predicciones, alertando a modelos basados en lógica fuzzy o sistemas expertos y tratar de corregir las desviaciones. Otra funcionalidad podría ser la de decidir cuándo es el momento oportuno para volver a re-entrenar el modelo que esté operativo. Por último, estos sistemas podrían valorar si interesa o no la incorporación de las nuevas variables propuestas para ser empleadas en el modelo, pudiendo ser tan sencillo como un análisis de correlación o la optimización y ejecución del nuevo modelo resultante al incorporar dichas variables (*algoritmo genético*) y la obtención del error de predicción para compararlo con el modelo operativo. Estos sistemas con inteligencia podrán ser planteados en base a *MAS*, ya que por un lado podrán existir agentes encargados de contactar con la sensoria interna y externa al sistema eléctrico y, además, tener agentes especializados en cada una de las funcionalidades anteriormente comentadas.

Como se mencionó en esta *Tesis Doctoral*, tanto la arquitectura del modelo *ANN* como la exactitud de la predicción teniendo en cuenta la generalización, dependerá del número de patrones en la fase de entrenamiento. Además, habrá que tener en cuenta que con el despliegue masivo de los *SMs* en los entornos citados en la *Tesis Doctoral*, posibilitará el disponer de patrones de consumo más recientes y, que además, reflejarán los cambios de hábitos de la demanda eléctrica. Por tanto, es posible disponer de un sistema que, de manera automática, plantee el conjunto de datos óptimo en las sucesivas fases de re-entrenamiento. Conceptualmente, este planteamiento puede ser conocido como “*ventana deslizante de datos*”, ya que en cada momento variará en cuanto a número de datos, y será controlado por algún agente con inteligencia suficiente y que operará en segundo plano.

Debemos recordar que las *microgrids* o *Smart Building* serán emplazamientos donde coexistirán la demanda eléctrica y la generación (con fuentes renovables o convencionales). Por tanto, los modelos predictivos de la demanda deberán tener una visión global del sistema, ya que normalmente, las fuentes de generación renovable (fotovoltaica, mini-eólica, etc.) estarán dentro del circuito de fuerza del edificio de un *prosumer*, por tanto, aguas abajo del *SM* del edificio o instalación. Dicho de otra forma, si la generación del propio edificio es consumido internamente, el *SM* de la instalación registrará una medida “*no real*” de la necesidad de la misma, ya que en este caso la necesidad real sería la media del *SM* del edificio y lo aportado por la generación interna. Si hay *SMs* tanto en el edificio como en la generación no habrá problema, ya que con la energía importada y exportada en ambos dispositivos, se puede tener un perfil del hábito del comportamiento real. Pero debe reflexionarse, en que en este tipo de *prosumer*, el comportamiento de la demanda debe tratarse desde una perspectiva global, debiendo tenerse en cuenta el conjunto en su totalidad, y deberá tenerse presente para los modelos predictivos, ya que una predicción de la



demanda con datos del *SM* del edificio, dará lugar a errores altos si la generación es fotovoltaica y no se han tenido en cuenta variables como radiación solar en el modelo, ya que la fotovoltaica producirá mucha energía en un día despejado de mayo, mientras que otro día similar de ese mes con nubosidad alta producirá una disminución en la energía generada, y por tanto el *SM* del edificio registrará valores de consumo más elevados de lo habitual para dichos días.

Antes de finalizar, a lo largo de la *Tesis Doctoral* ha quedado patente la necesidad de disponer modelos predictivos de la generación. En concreto, en las *microgrids* y las *VPPs*, la *DG* aparecerá como fundamental para obtener los objetivos de estos entornos, entre los cuales, el más importante será la eficiencia energética. Estos modelos dependerán de las variables climáticas locales del emplazamiento, para el caso de la fotovoltaica y eólica. La metodología para estos modelos puede ser similar a la de los modelos aquí presentados para la demanda eléctrica, teniendo en cuenta que la predicción del recurso (radiación solar y viento) será fundamental.



---

## BIBLIOGRAFÍA

- Abe, R., Taok, H., and McQuilkin, D. (2011). Digital Grid: Communicative Electrical Grids of the Future. *IEEE Transactions on Smart Grid*, 2(2):399–410.
- Abou–Hussien, M.S., Kandlil, M.S., Tantawy, M.A., and Farghal, S.A. (1981). An Accurate Model for Short–Term Load Forecasting. *IEEE Transactions on Power Apparatus and Systems*, PAS–100(9):4158–4165.
- Abreu, J.M., Camara, F., and Ferrao, P. (2012). Using pattern recognition to identify habitual behavior in residential electricity consumption. *Energy and Buildings*, 49:479–487.
- Abu–El–Magd, M.A. and Sinha, N.K. (1981). Two New Algorithms for On–Line Modelling and Forecasting of the Load Demand of a Multinode Power System. *IEEE Transactions on Power Apparatus and Systems*, PAS–100(7):3246–3252.
- Al–Hader, M., Rodzi, A., Sharif, A., and Ahmad, N. (2009). SmartCity Components Architecture. In *International Conference on Computational Intelligence, Modelling and Simulation, (CSSim '09)*, September 2009, Brno (Czech Republic), pages 93–97.
- Alfuhaid, A.S., El–Sayed, M.A., and Mahmoud, M.S. (1997). Cascaded artificial neural networks for short–term load forecasting. *IEEE Transactions on Power Systems*, 12(4):1524–1529.
- Alahakoon, D., Halgamuge, S.K., and Srinivasan, B. (2000). Dynamic self–organizing maps with controlled growth for knowledge discovery. *IEEE Transactions on Neural Networks*, 11(3):601–614.
- Alkon, D.L. (1989). Almacenamiento de memoria y sistemas neuronales. *Investigación y Ciencia*, 194:14–23.
- Amjady, N. (2007). Short–Term Bus Load Forecasting of Power Systems by a New Hybrid Method. *IEEE Transactions on Power Systems*, 22(1):333–341.
- Amjady, N., Keynia, F., and Zareipour, H. (2010). Short–Term Load Forecast of Microgrids by a New Bilevel Prediction Strategy. *IEEE Transactions on Smart Grid*, 1(3):286–294.
- Amin, S.M. and Giacomoni, A.M. (2012). Smart Grid – Safe, Secure, Self-Healing. *IEEE Power and Energy Magazine*, 10(1):33–40.
- Amin–Naseri, M.R. and Soroush, A.R. (2008). Combined use of unsupervised and supervised learning for daily peak load forecasting. *Energy Conversion and Management*, 49(6):1302–1308.
- Amral, N., King, D., and Ozveren, C.S. (2008). Application of Artificial Neural Network for Short Term Load Forecasting. In *Universities Power Engineering Conference, 2008. UPEC 2008. 43<sup>rd</sup> International*, Sep. 2008, Padova (Italy), pages 1–5.

- Anderson, J., Silverstein, J., Ritz, S., and Jones, R. (1977). Distinctive features, categorical perception and probability learning: some applications on a neural model. *Psychological Review*, 84:413–451.
- Anduaga, J., Boyra, M., Cobelo, I., García, E., Gil, A., Jimeno, J., Laresgoiti, I., Oyarzabal, J.M., Rodríguez, R., Sánchez, E., Santiago, F.J., Turienzo, E., Zabala, E., and Perea, E. (2008). La microrred, una alternativa de futuro para un suministro energético integral. *TECNALIA, Corporación Tecnológica*.
- Anelli, R., Caprio, U.D., Marchese, V., and Pozzi, S. (1978). Short Term Prediction of Stationary Load Processes with a Correlation Function Finite Sum of Exponentials. In *7<sup>th</sup> PSCC*, Aug. 1978, pages 401–408.
- Aracil, J. and Gordillo, F. (1997). Dinámica de sistemas. *Alianza Editorial*.
- Arahal, M.R., Cepeda, A., and Camacho, E.F. (2002). Input variable selection for forecasting models. In *15<sup>th</sup> triennial world congress of the IFAC*, July 2002, Barcelona (Spain), pages 1–6.
- Arahal, M.R., Berenguel, M., and Rodríguez, F. (2006). Técnicas de predicción con aplicaciones en la ingeniería. *Secretariado de Publicaciones de la Universidad de Sevilla*.
- Arbib, M.A. (1998). The Handbook of Brain Theory and Neural Networks. *Ed. MIT Press*.
- Asber, D., Lefebvre, S., Saad, M., and Desbiens, C. (2007). Modeling of Distribution Loads for Short and Medium–Term Load Forecasting. In *IEEE Power Engineering Society General Meeting, 2007*, June 2007, Tampa (US), pages 1–5.
- Ashok, S. and Banerjee, R. (2001). An optimization mode for industrial load management. *IEEE Transactions on Power Systems*, 16(4):879–884.
- Auman, R.J. and Hart, S. (1994). Handbook of Game Theory with Economic Applications. *Ed. North Holland*, vol. 2.
- Austin, J.A. (1962). How to Do Things with Words. *Ed. Oxford University press*.
- Austin, J.A. (1970). Philosophical Papers. *Ed. Oxford University press*.
- Axelrod, R. (1994). The Evolution of Co–operation. *Basic Books*.
- Baclawski, K., Kokar, M.K., Kogut, P.A., Hart, L., Smith, J., Letkowski, J., and Emery, P. (2002). Extending the Unified Modeling Language for ontology development. *Software and Systems Modeling*, 1(2):142–156.
- Badri, M.A. (1996). Neural networks of combination of forecasts for data with long memory pattern. In *IEEE International Conference on Neural Networks*, June 1996, Washington DC (US), pages 359–364.
- Baharudi, Z. and Kamel, N. (2008). Autoregressive Method in Short Term Load Forecast. In *2<sup>nd</sup> IEEE International Conference on Power and Energy (PECon 08)*, Dec. 2008, Johor Bahru (Malasya), pages 1603–1608.

- 
- Bakirtzis, A.G., Petridis, V., Kiartzis, S.J., Alexiadis, M.C., and Maissis, A.H. (1995). A neural network short term load forecasting model for the Greek power system. *IEEE Transactions on Power Systems*, 11(2):858–863.
- Baladrón, C., Aguiar, J.M., Gobernado, J., Carro, B., and Sánchez–Esguevillas, A. (2010). User-driven context aware creation and execution of home care applications. *Annals of Telecommunications*, 65(9–10):545–556.
- Bartlett, D., Harthoorn, W., Kehoe, M., and Schloss, R.J. (2011). Enabling integrated city operations. *IBM Journal of Research and Development*, 55(1&2):168–177.
- Bauer, B., Müller, J.P., and Odell, J. (2001). Agent UML: A Formalism for Specifying Multiagent Interaction. *International Journal of Software Engineering and Knowledge Engineering*, 11(3):1–24.
- Baum, E.B. and Haussler, D. (1989). What size net gives valid generalization? *Neural Computation*, 1(1):151–160.
- Bäumer, C. and Magedanz, T. (1999). Grasshopper – A Mobile Agent Platform for Active Telecommunication Networks. *Lecture Notes in Computer Science*, 1699:19–32.
- Beale, M.H., Hagan, M.T., and Demuth, H.B. (2010). Neural Network Toolbox™ 7 User’s Guide. *The MathWorks, Inc.* <http://www.mathworks.com>, (Accessed 09/07/2011).
- Bel, I., Valenti, A., Maire, J., Corera, J.M., and Lang, P. (2007). Innovative Operation with Aggregated Distributed Generation. In *19<sup>th</sup> International Conference on Electricity Distribution*, May 2007, Vienna (Austria), pages 1–4.
- Belik, B., Nelson, D.J., and Olive, D.W. (1978). Use of the Karhunen–Loeve expansion to analyze hourly load requirements for a power utility. *IEEE Power Engineering Society Winter Meeting*, A78:225–230.
- Bellifemine, F., Poggi, A., and Rimassa, G. (2001). Developing Multi–Agent Systems with JADE. *LNAI*, 1571:89–103.
- Bergenti, F., Gleizes, M–P., and Zambonelli, F. (2004). Methodologies and Software Engineering for Agent Systems – The Agent–Oriented Software Engineering Handbook. *Kluwer Academic Publishers*.
- Berlanga, A., Sanchís, A., Isasi, P., and Molina, J.M. (2002). Neural Network Controller Against Environment: A Coevolutionary Approach to Generalize Robot Navigation Behavior. *Journal of Intelligent and Robotic Systems*, 33(2):139–166.
- Bigus, J.P., Schlosnagle, D.A., Pilgrim, J.R., Mills III, W.N., and Diao, Y. (2002). ABLE: A toolkit for building multiagent autonomic systems. *IBM Systems Journal*, 41(3):350–371.
- Bishop, C.M. (1994). Neural Networks and their applications. *Review of Scientific Instruments*, 65(6):1803–1832.
-

- Bianchini, M., Frascioni, P., and Gori, M. (1995). Learning without local minima in radial basis function networks. *IEEE Transactions on Neural Networks*, 6(3):749–756.
- Björkskog, C., Jacucci, G., Mikkola, T., Bertoncini, M., Gamberini, I., Torstensson, C., Nieminen, T., Briguglio, I., Andriani, P., and Firoentino, G. (2010). BeAware: A framework for residential services on energy awareness. In *Proceedings of the International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies*, Oct. 2010, Florence (Italy), pages 1–8.
- Blasco, B.J. (1999). Control predictivo basado en modelos mediante técnicas de optimización heurística. Aplicación a procesos no lineales y multivariantes. *PhD thesis, Universidad Politécnica de Valencia (Spain)*.
- Booch, G., Rumbaugh, J., and Jacobson, I. (1998). *The Unified Modeling Language, User Guide*. Ed. Addison–Wesley.
- Booklets European Commission (2009). Your Guide to the Lisbon Treaty. <http://ec.europa.eu/publications/booklets/others/84/en.pdf>, (Accessed 09/01/2012).
- Box, G.E.P. and Jenkins, G. (1970). Time series analysis: forecasting and control. Ed. Holden–Day.
- Bragatto, T.A.C., Ruas, G.S.I., and Lamar, M.V. (2008). Real–time hand postures recognition using low computational complexity Artificial Neural Networks and Support Vector Machines. In *3<sup>rd</sup> International Symposium on Communications, Control and Signal Processing, 2008 (ISCCSP 2008)*, March 2008, St. Julians (Malta), pages 1530–1535.
- Bratman, M. (1990). What is Intention? In (*Cohen et al., 1990*), pages 15–31.
- Brazier, F.M., Cornelissen, F., Gustavsson, R., Jonker, C.M., Lindeberg, O., Polak, B., and Treur, J. (2004). Compositional verification of a multiagent system for one–to–many negotiation. *Applied Intelligence*, 20(2):95–117.
- Brodie, S.E., Knight, B.W., and Ratliff, F. (1978). The response of the Limulus retina to moving stimuli: A prediction by Fourier synthesis. *The Journal of General Physiology*, 72:129–166.
- Brooks, R.A. (1986). A Robust Layered Control System for a Mobile Robot. *IEEE Journal of Robotics and Automation*, RA–2:14–23.
- Brooks, R.A. (1991). Intelligence without Representation. *Artificial Intelligence*, 47:139–159.
- Bruck, J. (1990). On the convergence properties of the Hopfield model. *Proceedings of the IEEE*, 78(10):1579–1585.
- Bunn, D.W. and Farmer, E.D. (1985). Comparative models for electrical load forecasting. *New York: Wiley*, pages 13–30.
- Busi, N., Ciancarini, P., Gorrieri, R., and Zavattaro, G. (2001). Coordination Models: A Guided Tour. In (*Omicini et al., 2001*), pages 6–24.

- Caire, G., Leal, F., Chainho, P., Evans, R., Garijo, F., Gómez-Sanz, J.J., Pavón, J., Kerney, P., Stark, J., and Massonet, P. (2001). Agent Oriented Analysis using MESSAGE/UML. *Ed. Springer-Verlag*.
- Cammarata, S., McArthur, D., and Steeb, R. (1988). Strategies of Cooperation in Distributed Problem Solving. In *Readings in Distributed Artificial Intelligence*, Ed. Alan H. Bond and Les Gasser, Morgan Kaufmann.
- Campo, R. and Ruíz, P. (1987). Adaptive Weather-Sensitive Short Term Load Forecast. *IEEE Transactions on Power Systems*, 2(3):592–598.
- Carbó, J., Molina, J.M., and Dávila, J. (2003). Trust Management through Fuzzy Reputation. In *International Journal of Cooperative Information Systems*, March 2003, 12(1), pages 135–155.
- Carillo, S. (2010). Smart City: Respondiendo a los retos energéticos del siglo XXI. [www.smartcitymalaga.com](http://www.smartcitymalaga.com), (Accessed 11/20/2011).
- Carpaneto, E. and Chicco, G. (2008). Probabilistic characterisation of the aggregated residential load patterns. *IET Generation, Transmission and Distribution*, 2(3):373–382.
- Carpinteiro, O.A.S. and Reis, A.J.R. (2005). A SOM-based hierarchical model to short-term load forecasting. In *2005 IEEE Russia Power Tech*, Jun. 2005, St. Petersburg (Russia), pages 1–6.
- Cavendon, L. and Rao, A.S. (1996). Bringing About Rationality: Incorporating Plans Into a BDI Agent Architecture. *Lecture Notes in Computer Science*, 1114:601–612.
- Cerrada, M., Cardillo, J., Aguilar, J., and Faneite, R. (2007). Agents-based design for fault management systems in industrial processes. *Computers in Industry*, 58(4):313–328.
- Chan, P.P.K., Chen, W-C., Ng, W.W.Y., and Yeung, D.S. (2011). Multiple Classifier System for Short Term Load Forecast of Microgrid. In *Proceedings of the 2011 International Conference on Machine Learning and Cybernetics*, July 2011, Guilin (China), vol. 3, pages 1268–1273.
- Chan, S.C., Tsui, K.M., Wu, H.C., Hou, Y., Wu, Y-C., and Wu, F.F. (2012). Load/Price Forecasting and Managing Demand Response for Smart Grids. *IEEE Signal Processing Magazine*, 29(5):68–85.
- Chaouachi, A., Kamel, R.M., Andoulsi, R., and Nagasaka, K. (2013). Multiobjective Intelligent Energy Management for a Microgrid. *IEEE Transactions on Industrial Electronics*, 60(4):1688–1699.
- Charytoniuk, W. and Chen, M-S. (2000). Very Short-Term Load Forecasting Using Artificial Neural Networks. *IEEE Transactions on Power Systems*, 15(1):263–268.
- Che, J., Wang, J., and Wang, G. (2012a). An adaptive fuzzy combination model based on self-organizing map and support vector regression for electric load forecasting. *Energy*, 37(1):657–664.

- Che, J., Wang, J., and Tang, Y. (2012b). Optimal training subset in a support vector regression electric load forecasting model. *Applied Soft Computing*, 12(5):1523–1531.
- Chellas, B.F. (1980). Modal Logic: An Introduction. *Ed. Cambridge University Press*.
- Chen, H., Wu, J., and Gao, S. (2006). A Study of AutoRegressive Conditional Heteroscedasticity Model in Load Forecasting. In *2006 International Conference on Power System Technology*, Oct. 2006, Chongqing (China), pages 1–8.
- Chen, C., Duan, S., Cai, T., Liu, B., and Hu, G. (2011). Smart energy management system for optimal Microgrid economic operation. *IET Renewable Power Generation*, 5(3):258–267.
- Chicco, G., Napoli, R., Piglione, F., Postolache, P., Scutariu, M., and Toader, C. (2004). Load pattern-based classification of electricity customers. *IEEE Transactions on Power System*, 19:1232–1239.
- Choi, T.S., Ko, K.R, Park, S.C., Jang, Y.S., Yoon, Y.T., and Im, Y.K. (2009). Analysis of energy savings using smart metering system and IHD (in-home display). In *Transmission & Distribution Conference & Exposition: Asia and Pacific*, Oct. 2009, Seoul (South Korea), pages 1–4.
- Chow, T.W.S. and Tan, H-Z. (1998). Semiblind identification of nonminimum-phase ARMA models via order recursion with higher order cumulants. *IEEE Transactions on Industrial Electronics*, 45(4):663–671.
- Chu, W-C., Chen, Y-P., Xu, Z-W., and Lee, W-J. (2011). Multiregion Short-Term Load Forecasting in Consideration of HI and Load/Weather Diversity. *IEEE Transactions on Industry Applications*, 47(1):232–237.
- Chua, L.O. and Kang, S.M. (1977). Section-wise piecewise-linear functions: Canonical representation, properties, and applications. *Proceedings of the IEEE*, 65(6):915–929.
- Cohen, M.A. and Grossberg, S. (1983). Absolute stability of global pattern formation and parallel memory storage by competitive neural networks. *IEEE Transactions on Systems, Man, and Cybernetics*, 13(5):70–81.
- Cohen, P.R., Morgan, J.L., and Pollack, M.E. (1990). Intentions in Communication. *Ed. MIT Press*.
- Considine, J.T. (2000). The impacts of weather variations on energy demand and carbon emissions. *Resource and Energy Economics*, 22(4):295–314.
- Corera, J.M. (2006). Virtual power plant concept in electrical networks. In *2<sup>nd</sup> International Conference on Integration of Renewable and Distributed Energy Resources*, Dec. 2006, Napa (US), pages 1–32.
- Corrales, T.A. and Auron, J.J. (2000). Nonlinear system identification and overparameterization effects in multisensory evoked potential studies. *IEEE Transactions on Biomedical Engineering*, 47(4):472–486.



- Costanzo, G.T., Zhu, G., Anjos, M.F., and Savard, G. (2012). A System Architecture for Autonomous Demand Side Load Management in Smart Buildings. *IEEE Transactions on Smart Grid*, 3(4):2157–2165.
- Coster, E.J., Myrzik, J.M.A., Kruimer, B., and Kling, W.L. (2011). Integration Issues of Distributed Generation in Distribution Grids. *Proceedings of the IEEE*, 99(1):28–39.
- Couillet, R., Perlaza, S.M., Tembine, H., and Debbah, M. (2012). Electrical Vehicles in the Smart Grid: A Mean Field Game. *IEEE Journal on selected Areas in Communications*, 30(6):1086–1096.
- Czernichow, T. and Muñoz, A. (1995). Variable selection through statistical sensitivity analysis: Application to feedforward and recurrent networks. *Technical Report, INT–SIM*.
- D’Inverno, M. and Luck, M. (2004). Understanding Agent Systems. *Ed. Springer–Verlag*, segunda edición.
- Daneshi, H., Shahidehpour, M., and Choobbari, A.L. (2008). Long–Term Load Forecasting in Electricity Market. In *2008 IEEE International Conference on Electro/Information Technology*, May 2008, Ames (US), pages 395–400.
- Davies, D.L. and Bouldin, D.W. (1979). A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI–1:224–227.
- De Giacomo, G., Lespérance, Y., and Levesque, H.J. (2000). ConLog, a concurrent programming language base on the situation calculus. *Artificial Intelligence*, 121(1–2):109–169.
- Decker, S., Sycara, K., and Williamson, M. (1997). Middle–agents for the Internet. In *International Joint Conference on Artificial Intelligence*, Aug. 1997, Nagoya (Japan), pages 578–583.
- Decker, S., Erdmann, M., Fensel, D., and Studer, R. (1999). OntoBroker: Ontology based Access to Distributed and Semi–Structured Information. In *DS-8 Proceedings of the IFIP TC2/WG2.6 Eighth Working Conference on Database Semantics- Semantic Issues in Multimedia Systems*, Jan. 1999, Rotorua (New Zealand), pages 351–369.
- Deconinck, G. and Decroix, B. (2009). Smart metering tariff schemes combined with distributed energy resources. In *Critical Infrastructures, 2009. CRIS 2009. Fourth International Conference on*, March–April 2009, Linkoping (Sweden), pages 1–8.
- Deihimi, A. and Showkati, H. (2012). Application of echo state networks in short–term electric load forecasting. *Energy*, 39(1):327–340.
- DeLoach, S. (2001). Analysis and design using MaSE and agentTool. In *12<sup>th</sup> Midwest Artificial Intelligence and Cognitive Science Conference (MAICS 2001)*, March–April 2001, Ohio (US), pages 1–7.
- Demanzeau, Y. (2001). La methode VOYELLES, dans *Systèmes Multi–Agents: Des Théories Organisationnelles aux Applications Industrielles*. Mandiau et al., eds. Hermès, hiver.

- Denker, J.S., Schwartz, D., Wittner, B., Solla, S., Howard, R., Jackel, L., and Hopfield, J.J. (1987). Large automatic learning, rule extraction and generalization. *Complex Systems*, 1(5):877–922.
- Dennet, D.C. (1987). *The Intentional Stance*. Ed. MIT Press.
- DeSieno, D. (1988). Adding a conscience to competitive learning. Adding a conscience to competitive learning. In *IEEE International Conference on Neural Networks*, July 1988, San Diego (US), vol. 1, pages 117–124.
- Diederich, S. and Opper, M. (1987). Learning patterns in spin-glass networks by local learning rules. *Phys. Rev. Lett.*, 58(9):949–952.
- Dimeas, A.L. and Hatziaargyriou, N.D. (2005). Operation of a Multiagent System for Microgrid Control. *IEEE Transactions on Power Systems*, 20(3):1447–1445.
- Dong, M., Meira, P.C.M., Xu, W., and Freitas, W. (2012). An Event Window Based Load Monitoring Technique for Smart Meters. *IEEE Transactions on Smart Grid*, 3(2):787–796.
- Doveh, E., Feigin, P., Greig, D., and Hyams, L. (1999). Experience with FNN Models for Medium Term Power Demand Predictions. *IEEE Transactions on Power Systems*, 14(2):538–546.
- Drezga, I. and Rahman, S. (1988). Input variable selection for ANN-based short-term load forecasting. *IEEE Transactions on Power Systems*, 13(4):1238–1244.
- Drezga, I. and Rahman, S. (1999a). Short-term load forecasting with local ANN predictors. *IEEE Transactions on Power Systems*, 14(3):844–850.
- Drezga, I. and Rahman, S. (1999b). Phase-space Short-Term Load Forecasting for Deregulated Electric Power Industry. In *International Joint Conference on Neural Networks*, July 1999, Washington DC (US), vol. 5, pages 3405–3409.
- Duan, R. and Deconinck, G. (2009). Multi-agent coordination in market environment for future electricity infrastructure based on microgrids. In *IEEE International Conference on Systems, Man and Cybernetics, 2009. SMC 2009*, Oct. 2009, San Antonio (US), pages 3959–3964.
- Duan, P., Xie, K., Guo, T., and Huang, X. (2011). Short-Term Load Forecasting for Electric Power Systems Using the PSO-SVR and FCM Clustering Techniques. *Energies*, 4(1):173–184.
- Durfee, E.H. and Lesser, V. (1991). Partial Global Planning: A Coordination Framework for Distributed Hypothesis Formation. *IEEE Transactions on Systems, Man and Cybernetics*, 21(5):1167–1183.
- Durfee, E.H. and Rosenschein, J. (1994). Distributed Problem Solving and Multi-Agent Systems – Comparisons and Examples. In *Proc. Int. Distributed Artificial Intelligence Workshop*, July 1994, Seattle (US), pages 94–104.
- Durfee, E.H. (2001). Distributed Problem Solving and Planning. In (*Marik et al., 2001*), pages 118–149.

- Ephrati, E. and Rosenschein, J. (1994). Divide and Conquer in Multi-Agent Planning. In *Proceedings of the Twelfth National Conference on Artificial Intelligence (AAAI-94)*, Aug. 1994, Seattle (US), pages 375–380.
- Erman, L.D. and Lesser, V.R. (1975). A multi-level organization for problem solving using many diverse cooperating sources of knowledge. In *Proceedings of the Fourth International Joint Conference on Artificial Intelligence*, Sep. 1975, Tbilisi (Russia), pages 483–490.
- Erman, L.D., Hayes-Roth, F., Lesser, V.R., and Reddy, R. (1980). The Hearsay-II Speech-Understanding System: Integrating Knowledge to Resolve Uncertainty. *ACM Comput. Surv.*, 12(2):213–253.
- Ernault, M. and Mattatia, R. (1984). Short term load forecasting: New developments at E.D.F. In *Proc. 8<sup>th</sup> Power Systems Computation Conference*, Aug. 1984, Helsinki (Finland), pages 369–375.
- Erwin, E., Obermayer, K., and Schulten, K. (1992). Self-organizing maps: Stationary states, metastability and convergence rate. *Biological Cybernetics*, 67(1):35–45.
- Fadhilah, A.R., Suriawati, S., Amir, H.H., Izham, Z.A., and Mahendran, S. (2009). Malaysian day-type load forecasting. In *3<sup>rd</sup> International Conference on Energy and Environment, 2009 (ICEE 2009)*, Dec. 2009, Malacca (Malaysia), pages 409–411.
- Fahlman, S.E. and Lebiere, C. (1990). The cascade-correlation learning architecture. *Advances in Neural Information Processing Systems*, 2:524–532.
- Fan, J.Y. and McDonald, J.D. (1994). A real-time implementation of short-term load forecasting for distribution power systems. *IEEE Transactions on Power Systems*, 9(2):988–994.
- Fan, S., Methaprayoon, K., and Lee, W-L. (2009). Multiregion Load Forecasting for System With Large Geographical Area. *IEEE Transactions on Industry Applications*, 45(4):1452–1459.
- Fang, X., Misra, S., Xue, G., and Yang, D. (2012). Smart Grid – The New and Improved Power Grid: A Survey. *IEEE Communications Surveys & Tutorials*, 14(4):944–980.
- Fankhauser, H.R. (1984). A novel approach to on-line short and intermediate-term load forecasting. In *Proc. 8<sup>th</sup> Power Systems Computation Conference*, Aug. 1984, pages 376–380.
- Farhangi, H. (2010). The path of the smart grid. *IEEE Power Energ. Mag*, 8:18–28.
- Farmer, E.D. and Potton, M.J. (1968). Development of online load-prediction techniques with results from trials in the south-west region of the CEGB. *Proceedings of the Institution of Electrical Engineers*, 115(10):1549–1558.
- Feed-in Tariffs. <http://www.fitariffs.co.uk/> (Accessed 23/04/2012).
- Ferber, J. and Gutknecht, O. (1998). A Meta-Model for the Analysis and Design of Organizations in Multi-Agent Systems. In *Proceedings of the Third International Conference on Multi-Agent Systems (ICMAS98)*, IEEE Computer Society Washington, pages 128–135.

- Ferber, J. (1999). *Multiagent Systems – An Introduction to Distributed Artificial Intelligence*. Ed. Addison–Wesley.
- Ferguson, I.A. (1992). *TouringMachines: An Architecture for Dynamic, Rational, Mobile Agents*. PhD thesis, University of Cambridge (UK).
- Fernández, G., Trujillo, M., Sanz, J.F., and Sallán, J. (2010). Spanish microgrids: current problems and future solutions. In *International Conference on Renewable Energies and Power Quality (ICREPQ'11)*, April 2010, Las Palmas de Gran Canaria (Spain), pages 1–6.
- Ferreira, H.C., Lampe, L., Newbury, J., and Swart, T.G. editors (2010). *Power Line Communications. Theory and Applications for Narrowband and Broadband Communications over Power Lines*. Ed. John Wiley and Sons, primera edición.
- Finin, T., Weber, J., Wiederhold, G., Genesereth, M., Fritzson, R., McGuire, J., Shapiro, S., and Beck, C. (1993). *Specification of the KQML Agent–Communication Language*. DARPA Knowledge Sharing Initiative External Interfaces Working Group.
- FIPA (PC00025C). FIPA Interaction Protocol Library Specification. *Foundation for Intelligent Physical Agents*. <http://www.fipa.org/specs/fipa00025/PC00025C.html>, (Accessed 01/22/2012).
- FIPA (SC00029H). FIPA Contract Net Interaction Protocol Specification. *Foundation for Intelligent Physical Agents*. <http://www.fipa.org/specs/fipa00029/SC00029H.html>, Accessed 15/11/2012).
- FIPA (SC00037J). FIPA Communicative Act Library Specification. *Foundation for Intelligent Physical Agents*. <http://www.fipa.org/specs/fipa00037/SC00037J.html>, (Accessed 01/10/2012).
- Flórez, R. and Fernández, J.M. (2008). *Las Redes Neuronales Artificiales Fundamentos Teóricos y Aplicaciones Prácticas*. Ed. Netbiblo.
- Forrest, J.S. (1946). The effects of weather on power–system operation. The effects of weather on power–system operation. *Journal of the Institution of Electrical Engineers – Part I: General*, 93(64):161–163.
- Fox, M.S. (1979). An organizational view of distributed systems. In *Proceedings of the International Conference on Cybernetics and Society*, Oct. (1979), Denver (US), pages 354–359.
- Freeman, F.A. and Skapura, D.M. (1992). *Neural Networks: Algorithms, Applications and Programming Techniques*. Ed. Addison–Wesley.
- Friedman, J. (1991). *Teoría de Juegos con Aplicaciones a la Economía*. Alianza Editorial.
- Fuentes, R., Gómez–Sanz, J.J., and Pavón, J. (2003). Activity Theory for the Analysis and Design of Multi–Agent Systems. In *AOSE 2003, Springer–Verlag, LNCS 2935*, pages 110–122.
- Fuentes, R., Gómez–Sanz, J.J., and Pavón, J. (2009). Activity Theory for the Analysis and Design of Multi–Agent Systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans*, 39(2):282–298.

- Fukushima, K. (1980). Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36(4):192–202.
- Funahashi, K.I. (1989). On the approximate realization of continuous mappings by neural networks. *Neural Networks*, 2(5):183–192.
- Gann, D.M., Dogson, M., and Bhardwaj, D. (2011). Physical-digital integration in city infrastructure. *IBM Journal of Research and Development*, 55(1&2):90–99.
- Galbraith, J. (1973). *Designing Complex Organizations*. Ed. Addison-Wesley.
- Gallant, S.I. (1993). *Neural Network Learning and Expert Systems*. Ed. MIT Press.
- García, J.L., Blasco, J.A., del Brío, M.B., Domínguez, J.A., Barquillas, J., Ramírez, I.J., and Medrano, N.J. (1995). Short-term electric power load-forecasting using artificial neural networks. Part I: Self-organizing networks for classification of day types. In *Proceedings of the Fourteenth IASTED International Conference Modelling, Identification and Control*, Feb. 1995, Igls (Austria), pages 218–222.
- García-Ascanio, C. and Mate, C. (2009). Electric power demand forecasting using interval time series: A comparison between VAR and iMLP. *Energy Policy*, 38:715–725.
- Gardner, E. (1988). The space of interactions in neural networks models. *Journal of Physics*, 21:257–270.
- Garijo, F.J., Gómez-Sanz, J.J., Pavón, J., and Massonet, P. (2001). Multi-Agent System Organization. An Engineering Perspective. In *Pre-Proceeding of the 10<sup>th</sup> European Workshop on Modeling Autonomous Agents in a Multi-Agent World (MAAMAW'2001)*, May 2001, Annecy (France).
- Garijo, F.J. (2002). Tecnología de Agentes: Experiencias y perspectivas para el desarrollo de nuevos servicios y aplicaciones. *Boletic*, 24:1–9.
- Garijo, F.J., Bravo, S., González, J., and Bobadilla, E. (2004). BOGAR\_LN: An Agent Based Component Framework for Developing Multi-modal Services Using Natural Language. *Lecture Notes in Computer Science*, 3040:207–220.
- Gedeon, T.D., Wong, P.M., and Harris, D. (1995). Balancing the bias and variance: Network topology and pattern set reduction techniques. In *Proc. Int. Work. On Artificial Neural Networks (IWANN95)*, June 1995, Torremolinos (Spain), pages 550–558.
- George, T.J., Hatziargyriou, N.D., and Dialynas, E.N. (2007). Two-stage pattern recognition of load curves for classification of electricity customers. *IEEE Transactions on Power System*, 22:1120–1128.
- Georgeff, M. (1983). Communication and interaction in multi-agent planning. In *Proceedings of the Third national Conference on Artificial Intelligence (AAAI-83)*, Aug. 1983, Washington DC (US), pages 125–129.

- Georgeff, M. and Rao, A. (1998). Rational Software Agents: From Theory to Practice. In (*Jennings and Wooldridge, 1998*), pages 139–160.
- Giampapa, J.A., Paolucci, M., and Sycara, K. (2000). Agent interoperation across multiagent system boundaries. In *Proceedings of the fourth international conference on Autonomous agents*, June 2000, Barcelona (Spain), pages 179–186.
- Gillies, D.K.A., Bernholtz, B., and Sandiford, P.J. (1956). A New Approach to Forecasting Daily Peak Loads. *Transactions of the American Institute of Electrical Engineers, Power Apparatus and Systems, Part III*, 75(3):382–387.
- Godfrey, T. and Rodine, C. (2010). Unified Metrics for Management of Smart Grid Home Area Networks. In *2010 IEEE International Conference on Communications Workshops (ICC)*, May 2010, Palo Alto (USA), pages 1–5.
- Goh, T.N., Ong, H.L., and Lee, Y.O. (1986). A new approach to statistical forecasting of daily peak power demand. *Electric Power Systems Research*, 10(2):145–148.
- Gómez-Pérez, A. and Benjamins, V.R. (1999). Overview of Knowledge Sharing and Reuse Components: Ontologies and Problem Solving Methods. In *Proceedings of the IJCAI-99 Workshop on Ontologies and Problem Solving Methods (KRR5)*, Aug. 1999, Stockholm (Sweden), vol. 1, pages 1–15.
- Goodridge, S.G. and Luo, R.C. (1994). Fuzzy Behavior Fusion for Reactive Control of an Autonomous Mobile Robot: MARGE. In *Proceedings 1994 IEEE International Conference on Robotics and Automation*, May 1994, San Diego (US), vol. 2, pages 1622–1627.
- Gómez-Sanz, J.J., Pavón, J., and Garijo, F. (2002). Meta-Modelling of Multi-Agent Systems. *Symposium on Applied Computing*, March 2002, Madrid (Spain).
- Gopalratnam, K. and Cook, D.J. (2007). Online Sequential Prediction via Incremental Parsing: The Active LeZi Algorithm. *IEEE Intelligent Systems*, 22(1):52–58.
- Gorinevsky, D. (1995). On the persistency of Excitation in Radial basis Function Network Identification of Nonlinear Systems. *IEEE Transactions on Neural Networks*, 6(5):1237–1244.
- Grasia (2004). INGENIAS Development Kit. <http://ingenias.sourceforge.net/>, (Accessed 02/15/2011).
- Green Buildings Elements (2012). Japan and the United States Team to Make Maui a Leading Example in Smart Grid Technology. *Green Buildings Elements*. <http://greenbuildingelements.com/2011/06/02/smart-grid-technology/>, (Accessed 05/23/2012).
- Gross, G. and Galiana, F.D. (1987). Short-Term Load Forecasting. *Proceedings of the IEEE*, 75(12):1558–1573.
- Grossberg, S. (1982). Studies of Mind and Brain: Neural Principles of Learning, Perception, Development, Cognition and Motor Control. *Ed. Reidel Press*.

- Grosz, B.J. and Sidner, C.L. (1990). Plans for Discourse. In (*Cohen et al., 1990*), pages 417–444.
- Gruber, T.R. (1993). A Translation Approach to portable Ontology Specification. *Knowledge Acquisition*, 5(2):199–220.
- Gruetter, J.G. (1955). The Application of Business Machines to Electrical Utility Load Forecasting. *Transactions of the American Institute of Electrical Engineers, Power Apparatus and Systems, Part III*, 74(3):854–858.
- Guan, X., Xu, Z., and Jia, Q-S. (2010). Energy-Efficient Buildings Facilitated by Microgrid. *IEEE Transactions on Smart Grid*, 1(3):243–252.
- Guan, C., Luh, P.B., Michel, L.D., Wang, Y., and Friedland, P.B. (2013). Very Short-Term Load Forecasting: Wavelet Neural Networks With Data Pre-Filtering. *IEEE Transactions on Power Systems*, 28(1):30–41.
- Haddady, A. and Sundermeyer, K. (1996). Belief-Desire-Intention Agent Architectures. In (*O'Hare and Jennings, 1996*), pages 169–185.
- Hagan, M.T. and Behr, S.M. (1987). The Time Series Approach to Short Term Load Forecasting. *IEEE Transactions on Power Systems*, 2(3):785–791.
- Halpin, S.M., Smith, J.W., and Litton, C.A. (2001). Designing industrial systems with a weak utility supply. *IEEE Industry Applications Magazine*, 7(2):63–70.
- Hamby, D.M. (1994). A review of techniques for parameter sensitivity analysis of environmental models. *Environmental Monitoring and Assessment*, 32(2):135–154.
- Hamedani, M.E. and Arefifar, S.A. (2006). Distributed generation, reactive sources and network-configuration planning for power and energy-loss reduction. *IEE Proceedings-Generation, Transmission & Distribution*, 153(2):127–136.
- Hamilton, R.F. (1944). The Summation of Load Curves. *Transactions of the American Institute of Electrical Engineers*, 63(10):729–735.
- Han, D-M. and Lim, J-M. (2010). Design and implementation of smart home energy management systems based on zigbee. *IEEE Transactions on Consumer Electronics*, 56(3):1417–1425.
- Han, H-G. and Qiao, J-F. (2012). Adaptive Computation Algorithm for RBF Neural Network. *IEEE Transactions on Neural Networks and Learning Systems*, 23(2):342–347.
- Harris, A. (2012). Smart Buildings. *Engineering & Technology*, 7(6):52–54.
- Harrison, C., Eckman, B., Hamilton, R., Hartswick, P., Kalagnaman, J., Paraszczak, J., and Williams, P. (2010). Foundations for Smarter Cities. *IBM Journal of Research and Development*, 54(4):1–16.
- Hart, S. and Mas-Colell, A. (1997). Cooperation: Game-Theoretic Approaches. *Ed. Springer-Verlag*, vol. F-155.

- Hayes–Roth, F., Waterman, D.A., and Lenat, D.B. (1983). Building Expert Systems. *Ed. Addison–Wesley*.
- Haykyn, S. (1999). Neural Networks. A Comprehensive Foundation. *Ed. Prentice–Hall*, segunda edición.
- Hebb, D.O. (1949). The organization of behaviour. *Ed. John Wiley & Sons*.
- Hecht–Nielsen, R. (1987). Kolmogorov’s mapping neural networks existence theorem. In *Proceedings of the IEEE International Conference on Neural Networks III*, June 1987, San Diego (US), pages 11–13.
- Hecht–Nielsen, R. (1990). Neurocomputing. *Ed. Addison–Wesley*.
- Henley, A. and Peirson, J. (1997). Non–linearities in electricity demand and temperature: parametric versus non parametric methods. *Oxford Bulletin of Economics and Statistics*, 59(1):149–162.
- Hernández–Aranburo, C.A., Green, T.C., and Mugniot, N. (2005). Fuel consumption minimization of a microgrid. *IEEE Transactions on Industry Applications*, 41(3):673:681.
- Hernández, L., Baladrón, C., Aguiar, J.M., Calavia, L., Carro, B., Sánchez–Esguevillas, A., Cook, D.J., Chinarro, D., and Gómez, J. (2012a). A Study of the Relationship between Weather Variables and Electric Power Demand inside a Smart Grid/Smart World Framework. *Sensors*, 12(9):11571–11591.
- Hernández, L., Baladrón, C., Aguiar, J.M., Carro, B., and Sánchez–Esguevillas, A. (2012b). Classification and Clustering of Electricity Demand Patterns in Industrial Parks. *Energies*, 5(12):5215–5228.
- Hernández, L., Baladrón, C., Aguiar, J.M., Carro, B., Sánchez–Esguevillas, A., Lloret, J., Chinarro, D., Gómez, J.J., and Cook, D. (2013a). A multi-agent system architecture for smart grid management and forecasting of energy demand in virtual power plants. *IEEE Communications Magazine*, 51(1):106–113.
- Hernández, L., Baladrón, C., Aguiar, J.M., Carro, B., Sánchez–Esguevillas, A.J., and Lloret, J. (2013b). Short–Term Load Forecasting for Microgrids Based on Artificial Neural Networks. *Energies*, 6(3):1385–1408.
- Hertz, J., Krogh, A., and Palmer, R. (1991). Introduction to the theory of Neural Computation. *Ed. Addison–Wesley*.
- Hida, Y., Ito, Y., Yokoyama, R., and Iba, K. (2010). A study of optimal capacity of PV and battery energy storage system distributed in demand side. In *2010 45<sup>th</sup> International Universities Power Engineering Conference (UPEC)*, Aug. 2010, Cardiff (Wales), pages 1–5.
- Hilera, J.R. and Martínez, V.J. (1995). Redes Neuronales Artificiales. Fundamentos, modelos y aplicaciones. *Ed. RA–MA*.



- Hingorani, N.G. (1995). Introducing custom power. *IEEE Spectrum*, 32(6): 41–48.
- Hinton, G.E. and Sejnowski, T.J. (1986). Learning and relearning in Boltzmann Machines. In *(Rumelhart and McClelland, 1986)*, pages 282–317.
- Hippert, H.S., Pedreira, C.E., and Souza, R.C. Neural networks for short-term load forecasting: A review and evaluation. *IEEE Transactions on Neural Networks*, 16(1):44–55.
- Hjorth, J.S.U. (1994). Computer Intensive Statistical Methods Validation, Model Selection and Bootstrap. *Ed. Chapman and Hall*.
- Ho, K-L., Hsu, Y-Y., Chen, C-F., Lee, T-E., Liang, C-C., Lai, T-S., and Chen, K-K. (1990). Short term load forecasting of Taiwan power system using a knowledge-based expert system. *IEEE Transactions on Power Systems*, 5(4):1214–1221.
- Ho, K-L., Hsu, Y-Y., and Yang, C-C. (1992). Short term load forecasting using a multilayer neural network with and adaptive learning algorithm. *IEEE Transactions on Power Systems*, 7(1):141–149.
- Holzmann, G.J. (1999). Design and Validation of Computer Protocols. *Ed. Prentice Hall*.
- Hong, W-C. (2010). Application of chaotic and swarm optimization in electric load forecasting. *Energy Policy*, 38(10):5830–5839.
- Hong, W-C., Dong, Y., Lai, C-Y., Chen, L-Y., and Wei, S-Y. (2011a). SVR with Hybrid Chaotic Immune Algorithm for Seasonal Load Demand Forecasting. *Energies*, 4(6):960–977.
- Hong, W-C. (2011b). Electric load forecasting by seasonal recurrent SVR (support vector regression) with chaotic artificial bee colony algorithm. *Energy*, 36(9):5568–5578.
- Hong, W-C., Dong, Y., Zhang, W-Y., Chen, L-Y., and Panigrahi, B.K. (2013). Cyclic electric load forecasting by seasonal SVR with chaotic genetic algorithm. *Electrical Power and Energy Systems*, 44(1):604–614.
- Hooke, R.G. and Newark, N.J. (1955). Forecasting the Demand for Electricity. *Transactions of the American Institute of Electrical Engineers, Power Apparatus and Systems, Part III*, 74(3):993–1008.
- Hopfield, J.J. (1982). Neural Networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences of the United States of America*, 79(8):2554–2558.
- Hopfield, J.J. (1984). Neurons with graded response have collective computational properties like those of two-state neurons. *Proceedings of the National Academy of Sciences of the United States of America*, 81(10):3088–3092.
- Hornik, K., Stinchcombe, M., and White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366.
- Hrycej, T. (1992). Modular Learning in Neural Networks. *Ed. John Wiley and Sons*.

- Hsu, Y–Y. and Yang, C–C. (1991). Design of artificial neural networks for short-term load forecasting. Part I: Self-organising feature maps for day type identification. *IEE Proceedings C Generation Transmission Distribution*, 138(5):407–413.
- Hsu, C–C. and Chen, C–Y. (2003). Regional load forecasting in Taiwan–applications of artificial neural networks. *Energy Conversion and Management*, 44(12):1941–1949.
- Huang, S–J. and Shin, K–R. (2003). Short–term load forecasting using threshold autoregressive models. *IEEE Transactions on Power Systems*, 18(2):673–679.
- Huang, A.Q., Crow, M.L., Heydt, G.T., Zheng, J.P., and Dale, S.J. (2011). The Future renewable Electric Energy delivery and Management (FREEDM) System: The Energy Internet. *Proceedings of the IEEE*, 99(1):133–148.
- Hughes, G.E. and Cresswell, M.J. (1996). A New Introduction to modal Logic. *Ed. Routledge*.
- Hush, D.R. and Horne, B.G. (1993). Progress in supervised neural networks. What’s new since Lippmann? *IEEE Signal Processing Magazine*, January:8–39.
- Hyndman, R.J. and Fan, S. (2010). Density Forecasting for Long-Term Peak Electricity Demand. *IEEE Transactions on Power Systems*, 25(2):1142–1153.
- Iglesias, C.A. (1998). Definición de una metodología para el desarrollo de sistemas multiagente. *PhD thesis, Universidad Politécnica de Madrid (Spain)*.
- Ilic, M., Hsieh, E., and Ramanan, P. (2003). Transmission pricing of distributed multilateral energy transactions to ensure system security and guide economic dispatch. *IEEE Transactions on Power Systems*, 18(2):428–434.
- Ipakchu, A. and Albuyeh, F. (2009). Grid of the Future–Are We Ready to Transition to a Smart Grid? *IEEE Power and Energy Magazine*, 7(2):52–62.
- Ircbeginner <http://www.ircbeginner.com/opvinfo/webglossary.html>, (Accessed 11/20/2011).
- Irisarri, G.D., Widergren, S.E., and Yehsakul, P.D. (1982). On–line Load Forecasting for Energy Contron Center Application. *IEEE Transactions on Power Apparatus and Systems*, PAS–101(1):71–78.
- Islam, M.M., Sattar, M.A., Amin, M.F., Xin, Y., and Murase, K. (2009). A New Constructive Algorithm for Architectural and Functional Adaptation of Artificial Neural Networks. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 39(6):1590–1605.
- Jahn, M., Jentsch, M., Prause, C.R., Pramudianto, F., Al–Akkad, A., and Reiners, R. (2010). The Energy Aware Smart Home. In *2010 5<sup>th</sup> International Conference on Future Information Technology (FutureTech)*, May 2010, Busan (Korea), pages 1–8.
- Jain, A. and Satish, B. (2008). Integrated Approach for Short-Term Load Forecasting using SVM and ANN. In *TENCON 2008 – 2008 IEEE Region Conference*, Nov. 2008, Hyderabad (India), pages 1–6.

- Jang, J.S.R., Sun, C.T., and Mizutani, E. (1997). *Neuro-Fuzzy and Soft Computing*. Ed. Prentice-Hall.
- Javed, F., Arshad, N., Wallin, F., Vassileva, I., and Dahlquist, E. (2012). Forecasting for demand response in smart grids: An analysis on use of anthropologic and structural data and short term multiple loads forecasting. *Applied Energy*, 96:150–160.
- Jennings, N.R. (1993). Specification and Implementation of a Belief-Desire-Joint-Intention Architecture for Collaborative Problem Solving. *Journal of Intelligent and Cooperative Information Systems*, 2:289–318.
- Jennings, N.R. and Wooldridge, M.J. (1998). *Agent Technology: Foundations, Applications and Markets*. Ed. Springer.
- Juan, Y-C., Ou-Yang, C., and Lin, J-S. (2009). A process-oriented multi-agent system development approach to support the cooperation-activities of concurrent new product development. *Computers & Industrial Engineering*, 57(4):1363–1376.
- Kamaev, V.A., Shcherbakov, M.V., Panchenko, D.P., Shcherbakova, N.L., and Brebels, A. (2012). Using Connectionist Systems for Electric Energy Consumption Forecasting in Shopping Centers. *Automation and Remote Control*, 73(6):1075–1084.
- Kanchev, H., Lu, D., Colas, F., Lazarov, V., and Francois, B. (2011). Energy Management and Operational Planning of a Microgrid With a PV-Based Active Generator for Smart Grid Applications. *IEEE Transactions on Industrial Electronics*, 58(10):4583–4592.
- Kandel, E.R. and Hawkings, R.D. (1992). Bases biológicas del aprendizaje y de la individualidad. *Investigación y Ciencia*, 194:48–57.
- Kandel, E.R., Schwartz, T.H., and Jessel, T.M. (1999). *Principles of Neural Science*. Ed. McGraw-Hill, cuarta edición.
- Kandil, M.S., El-Debeiky, S.M., and Hasanien, N.E. (2002). Long-Term Load Forecasting for Fast developing Utility Using a Knowledge-Based Expert System. *IEEE Transactions on Power Systems*, 17(2):491–496.
- Kangas, J. (1994). On the analysis of pattern sequences by self-organizing maps. *PhD thesis, Univ. Tec. Helsinki (Finland)*.
- Kalafatis, A., Arifin, N., Wang, L., and Cluett, W.R. (1995). A new approach to the identification of ph processes based on the wiener model. *Chemical Engineering Science*, 50(23):3693–3701.
- Karnouskos, S. and de Holanda, T.N. (2009). Simulation of a SmartGrid City with Software Agents. In *EMS '09 Proceedings of the 2009 Third UK European Symposium on Computer Modeling and Simulation*, Nov. 2009, Athens (Greece), pages 424–429.
- Kaushik, B., Kaur, N., and Kohli, A.K. (2010). A computational study on the design of overall reliability measures of optimized ANN for computer networks with fixed and varying link

- reliabilities. In *2010 IEEE International Conference on Computational Intelligence and Computing Research (ICIC)*, Dec. 2010, Coimbatore (India), pages 1–6.
- Keith, D., Pannu, A., Sycara, K., and Williamson, M. (1997). Designing Behaviors for Information Agents. In *AGENTS'97 Proceedings of the First International Conference on Autonomous Agents*, Feb. 1997, Marina del Rey (US), pages 404–413.
- Kennel, M.B., Brown, R., and Abarbanel, H.D.I. (1995). Determining embedding dimension for phase-space reconstruction using geometrical construction. *Physical Review A*, 45(6):3403–3411.
- Keyhani, A. and Miri, S.M. (1983). On-line weather sensitive and industrial group bus load forecasting for microprocessor-based applications. *IEEE Transactions on Power Apparatus and Systems*, PAS-102(12):3868–3876.
- Khodayar, M.E., Barati, M., and Shahidehpour, M. (2012). Integration of High Reliability Distribution System in Microgrid Operation. *IEEE Transactions on Smart Grid*, 3(4):1997–2006.
- Khotanzad, R., Afkhami-Rohani, R., Lu, T-L., Abaye, A., Davis, M., and Maratukulam, D.J. (1997). ANNSTLF—a neural-network-based electric load forecasting system. *IEEE Transactions on Neural Networks*, 8(4):835–846.
- Kifer, M., Lausen, G., and Wu, J. (1995). Logical Foundations of Object-Oriented and Frame-Based Languages. *Journal of the ACM*, 42(4):741–843.
- Kim, Y.J., Thottan, M., Kolesnikov, V., and Lee W. (2010). A Secure Decentralized Data-Centric Information Infrastructure for Smart Grid. *IEEE Communications Magazine*, 48(11):58–65.
- Kinny, D. and Georgeff, M. (1991). Commitment and Effectiveness of Situated Agents. In *IJCAI'91 Proceedings of the 12<sup>th</sup> international joint conference on Artificial intelligence*, Aug. 1991, Sydney (Australia), vol. 1, pages 82–88.
- Kinny, D., Georgeff, M., and Rao, A. (1997). A Methodology and Modelling Technique for Systems of BDI Agents. *LNAI*, vol. 1038, pages 56–71.
- Klusck, M. and Sycara, K. (2001). Brokering and matchmaking for coordination of agent societies: a survey. In (*Omicini et al. 2001*), pages 197–224.
- Kohonen, T. (1977). Associative Memory. A system theoretical approach. *Ed. Springer-Verlag*.
- Kohonen, T. (1982a). Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43(1):59–69.
- Kohonen, T. (1982b). Analysis of a simple self-organizing process. *Biological Cybernetics*, 44(2):135–140.
- Kohonen, T. (1986). Learning vector quantization for pattern recognition. *Technical Report TKK-F-A601*, Helsinki University of Technology (Finland).
- Kohonen, T. (1988a). An Introduction to Neural Computing. *Neural Networks*, 1(1):3–16.

- Kohonen, T. (1988b). The neural phonetic typewriter. *Computer*, 21(3):1–22.
- Kohonen, T. (1989). Self–Organization and Associative Memory. *Ed. Springer–Verlag*, tercera edición.
- Kohonen, T. (1990). The Self–Organizing Map. *Proceedings of the IEEE*, 78(9):1464–1480.
- Kohonen, T., Mälkisara, K., Simula, O., and Kangas, J. (1991a). Artificial Neural Networks. In *Proceedings of the 1991 International Conference on Artificial Neural Networks (Icann–91)*, June 1991, Espoo (Finland).
- Kohonen, T. (1991b). Self–organizing amps: optimization approaches. In (*Kohonen et al., 1991a*), vol. II, pages 981–990.
- Kohonen, T. (1993). Things you haven’t heard about the Self–Organizing Map. In *IEEE International Conference on Neural Networks*, March–April 1993, San Francisco (US), vol. 3, pages 1147–1156.
- Kohonen, T. (1996). Emergence of invariant-feature detectors in the adaptive-subspace self–organizing map. *Biological Cybernetics*, 75(4):281–291.
- Kok, K., Karnouskos, S., Nestle, D., Dimeas, A., Weidlich, A., Warmer C., Strauss P., Buchholz, B, and Drenkard, S. (2009). Smart Houses for a Smart Grid. In *20<sup>th</sup> International Conference on Electricity Distribution*, June 2009, Prague (Czech Republic), pages 1–4.
- Kornfeld, W. and Hewitt, C. (1981). The Scientific Community Metaphor. *IEEE Transactions on Systems, Man, and Cybernetics*, SMS–11(1):24–33.
- Kosko, B. (1988). Feedback stability and unsupervised learning. In *IEEE International Conference on Neural Networks*, July 1988, San Diego (US), pages 141–152.
- Kosko, B. (1992a). *Neural Networks and Fuzzy Systems*. *Ed. Prentice–Hall*.
- Kosko, B. (1992b). *Neural Networks for Signal Processing*. *Ed. Prentice–Hall*.
- Labrou, Y. and Finin, T. (1997). A Proposal for a New KQML Specification. *TR CS–97–03, Computer Science and Electrical Engineering Department, University of Maryland*.
- Lamedica, R., Prudenzi, A., Sforma, M., Caciotta, M., and Cencelli, V.O. (1996). A neural network based technique for short–term forecasting of anomalous load periods. *IEEE Transactions on Power Systems*, 11(4):1749–1756.
- Lasseter, R., Akhil, A., Marnay, C., Stevens, J., Dagle, J., Guttromson, R., Meliopoulous, A.S., Yinger, R., and Eto, J. (2002). White paper on integration of Distributed Energy resources. The CERTS Microgrid concept. *Consortium for Electric Reliability Technology Solutions, prepared for the US department of Energy*.
- Laverty, D., Morrow, J., Best, R., and Crossley, P. (2010). Telecommunications for Smart Grid: Backhaul Solutions for the Distribution Network. In *2010 IEEE Power and Energy Society General Meeting*, July 2010, Oakland (USA), pages 1–6.

- LeCum, Y., Bottou, L., and Orr, G.B. (1998). Efficient Backpropagation. In (*Orr and Müller, 1998*), pages 189–227.
- Lee, S. (1992). Supervised learning with Gaussian potentials. In (*Kosko, 1992*), pages 9–50.
- Lee, K.Y., Cha, Y.T., and Park, J.H. (1992). Short-term load forecasting using an artificial neural network. *IEEE Transactions on Power Systems*, 7(1):124–132.
- Lee, J., Jung, D., Kim, Y., Lee, Y., and Kim, Y.M. (2010). Smart Grid Solutions, Services, and Business Models Focused on Telco. In *2010 IEEE/IFIP Network Operations and Management Symposium Workshops (NOMS Wksp)*, April 2010, Osaka (Japan), pages 323–326.
- Lee, J., Baik, S., and Lee, C. (2011). Building and Integrated Service Management Platform for Ubiquitous Cities. *Computer*, 44(6):55–63.
- Leong, C., Ramli, A.R., and Perumal, T. (2009). A Rule-Based Framework for Heterogeneous Subsystems Management in Smart Home Environment. *IEEE Transactions on Consumer Electronics*, 55(3):1208–1213.
- Leontiev, A.N. (1978). Activity, Consciousness, and Personallity. *Ed. Prentice-Hall*.
- Lesser, V.R. and Corkill, D.D. (1981). Functionally Accurate, Cooperative Distributed Systems. *IEEE Transactions on Systems, Man, and Cybernetics*, SMCII(1):81–96.
- Lesser, V.R. and Corkill, D.D. (1983). The Distributed Vehicle Monitoring Testbed: A Tool for Investigating Distributed Problem Solving Networks. *AI Magazine*, 4(3):15–33.
- Li, Y., Vilathgamuwa, D.M., and Loh, P.C. (2004). Design, analysis, and real-time testing of a controller for multibus microgrid system. *IEEE Transactions on Power Electronics*, 19(5):1195:1204.
- Li, H., Guo, S., Zhao, H., Su., C., and Wang, B. (2012). Annual Electric Load Forecasting by a Least Squares Support Vector Machine with a Fruit Fly Optimization Algorithm. *Energies*, 5(11):4430–4445.
- Li, H-Z., Guo, S., Li, C-J., and Sun, J-Q. (2013). A hybrid annual power load forecasting model based on generalized regression neural network with fruit fly optimization algorithm. *Knowledge-Based Systems*, 37:378–387.
- Lind, J. (2000). MASSIVE: Software Engineering for Multiagent Systems. *PhD thesis, University of Saarland (Germany)*.
- Linsker, R. (1988). Self-organization in a perceptual network. *Computer*, 21(3):105–117.
- Lippmann, R.P. (1987). An introduction to computing with neural nets. *IEEE ASSP Magazine*, 4(2):4–22.
- Lisserre, M., Sauter, T., and Hung, J.Y. (2010). Future energy systems: Integrating renewable energy resources into smart power grid through industrial electronics. *IEEE Industrial Electronics Magazine*, 4(1):18–37.

- 
- Liu, W.H. (2010). Analytics and Information Integration for Smart Grid Applications. In *IEEE Power and Energy Society General Meeting*, July 2010, Oakland (USA), pages 1–3.
- Ljung, L. and Soderstrom, T. (1983). Theory and Practice of Recursive Identification. *Ed. MIT press*.
- Ljung, L. (1987). System identification – Theory for the user. *Ed. Prentice Hall, Englewood Cliffs, NJ*.
- Lloret, J., Gilg, M., García, M., and Lorenz, P.P. (2011). A Group-Based Protocol for Improving Energy Distribution in Smart Grids. In *2011 IEEE International Conference on Communications (ICC)*, June 2011, Kyoto (Japan), pages 1–6.
- Lo, Z.P. and Bavarian, B. (1991). On the rate of convergence in topology preserving neural networks. *Biological Cybernetics*, 65(1):55–63.
- Lopes, J.A.P. (2009). Advanced MicroGrids as a component for active management of distribution networks. In *International Conference on Power Engineering, Energy and Electrical Drives, 2009 (POWERENG '09)*, March 2009, Lisbon (Portugal), pages 7–8.
- López, M., Valero, S., Senabre, C., and Aparicio, J.A. (2011). SOM neural network approach to load forecasting. Meteorological and time frame influence. In *Proceedings of the 2011 International Conference on Power Engineering and Electrical Drivers*, May 2011, Málaga (Spain), pages 1–5.
- Lorenz, E. (1963). Deterministic nonperiodic flow. *Journal of Atmospheric Sciences*, 20(2):130–148.
- Lou, C–W. and Dong, M–C. (2012). Modeling data uncertainty on electric load forecasting based on Type–2 fuzzy logic set theory. *Engineering Applications of Artificial Intelligence*, 25(8):1567–1576.
- Lu, C–N., Wu, H–T., and Vemuri, S. (1993). Neural network based short term load forecasting. *IEEE Transactions on Power Systems*, 8(1):336–342.
- Lyytinen, K.S. and Rossi, M. (1999). METAEDIT+ A Fully Configurable Multi–User and Multi–Tool CASE and CAME Environment. *Ed. Springer–Verlag*.
- MacGregor, R. (1991). Inside the LOOM Classifier. *ACM SIGART Bulletin*, 2(3):88–92.
- MacQueen, J.B. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the 5<sup>th</sup> Berkeley Symposium on Mathematical Statistics and Probability*, July 1967, Berkeley (US), University of California Press: Berkeley, CA, US, pages 281–297.
- Maes, P. (1990). Situated Agents Can Have Goals. *Journal Robotics and Autonomous Systems*, 6(1&2):49–70.
- Maes, P. (1995). Artificial Intelligence Meets Entertainment: Lifelike Autonomous Agents. *Communications of the ACM*, 38(11):108–114.
-

- Maes, P., Guttman, R.H., and Moukas, A.G. (1999). Agents that Buy and Sell: Transforming Commerce as we Know It. *Communications of the ACM*, 42(3):81–91.
- Maksimovich, S.M. and Shiljkut, V.M. (2009). The Peak Load Forecasting Afterwards Its Intensive Reduction. *IEEE Power Delivery*, 24(3):1552–1559.
- Malone, T. and Crowston, K. (1994). The Interdisciplinary Study of Coordination. *ACM Computing Surveys*, 26(1):87–119.
- Mantooth, H.A. (2010). Smart Grid: Definition, Road map and Breadth of this Grand Challenge. *Connected Planet*, publicaciones digitales on–line. (Accessed 12/20/2011).
- Maren, A.J., Harston, C.T., and Pap, R.M. (1990). Handbook of Neural Computing Applications. *Ed. Academic Press*.
- Marik, V., Stepankova, O., Krautwurmova, H., and Luck, M. (2001). Multi-Agent-Systems and Applications II. *Ed. Springer–Verlag*.
- Marin, F.J., Garcia–Lagos, F., Joya, G., and Sandoval, F. (2002). Global model for short–term load forecasting using artificial neural networks. *IEE Proceedings–Generation, Transmission and Distribution*, 149(2):121–125.
- Marks II, R.J. (1993). Intelligence: Computational Versus Artificial. *IEEE Transactions on Neural Networks*, 4(5):737–739.
- Martín del Brío, B. and Serrano Cinca, C. (1993). Self–organizing neural networks for analysis and representation of data: Some financial cases. *Neural Computing and Applications*, 1(2):193–206.
- Martín del Brío, B. and Sanz, A. (2006). Redes Neuronales y Sistemas Borrosos. *Ed. RA–MA*, tercera edición.
- Martín–Merino, M. (2005). Técnicas Neuronales y Estadísticas para la Predicción de Demanda Eléctrica. *Amarú Ediciones*.
- Mas, A. (2005). Agentes Software y Sistemas Multi–Agentes: Conceptos, Arquitecturas y Aplicaciones. *Ed. Pearson Prentice Hall*.
- Massoud, S. and Wollenberg, B.F. (2005). Toward a smart grid: power delivery for the 21<sup>st</sup> century. *IEEE Power and Energy Magazine*, 3(5):34–41.
- Matous, K., Leps, M., Zeman, J., and Sejnoha, J. (2000). Applying genetic algorithms to selected topics commonly encountered in engineering practice. *Computers Methods in applied mechanics and engineering*, 190(13–14):1629–1650.
- Matsumoto, S. (2010). Echonet: A Home Network Standard. *IEEE Pervasive Computing*, 9(3):88–92.
- Matthewman, P.D. and Nicholson, H. (1968). Techniques for load prediction in the electricity–supply industry. *Proceedings of the Institution of Electrical Engineers*, 115(10):1451–1457.



- May, P.M. and Murray, K. (1991). JESS, A joint expert specciation system—I. Raison d'être. *Talanta*, 38(12):1409–1417.
- McArthur, S.D.J., Davidson, E.M., Catterson, V.M., Dimeas, A.L., Hatziargyriou, N.D., Ponci, F., and Funabashi, T. (2007). Multi-Agent Systems for Power Engineering Applications—Part I: Concepts, Approaches, and Technical Challenges. *IEEE Transactions on Power Systems*, 22(4):1743–1752.
- McClelland, J.L. and Rumelhart, D.E. (1986). Parallel Distributed Processing. vol 2: Psychological and biological models. *Ed. MIT Press*.
- McCulloch, W.S. and Pits, W.A. (1943). A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematics and Biophysics*, 5(4):115–133.
- McDaniel, P.D. and McLaughlin, S.E. (2009). Security and Privacy Challenges in the Smart Grid. *IEEE Security & Privacy*, 7(3):75–77.
- McMenamin, J.S. and Monforte, F.A. (1998). Short-term energy forecasting with neural networks. *The Energy Journal*, 19(4):43–61.
- McSharry, P.E., Bowman, S., and Bloemhof, G. (2005). Probabilistic forecasts of the magnitude and timing of peak electricity demand. *IEEE Transactions on Power Systems*, 20(2):1166–1172.
- Meng, M. and Shang, W. (2008). Research on Annual Electric Power Consumption Forecasting Based on Partial Least-Squares Regression. In *2008 International Seminar on Business and Information Management*, Dec. 2008, Hubei (China), vol. 2, pp. 125–127.
- Meng, M., Niu, D., and Sun, W. (2011). Forecasting Monthly Electric Energy Consumption using Feature extraction. *Energies*, 4(10):1495–1507.
- Messaoud, Y., Chen, H.Y.H., and Fuller, D.Q. (2011). The Influence of Recent Climate Change on Tree Height Growth Differs with Species and Spatial Environment. *PLoS One*, 6(2):1–8.
- Minsky, M. and Papert, S. (1969). Perceptrons. *Ed. MIT Press*.
- Mirasgedis, S., Sarafidis, Y., Georgopoulou, E., Kotroni, V., Lagouvardos, K., and Lalas, D.P. (2007). Modeling framework for estimating impacts of climate change on electricity demand at regional level: Case of Greece. *Energy Conversion and Management*, 48(5):1737–1750.
- Moazzami, M., Khodabakhshian, A., and Hooshmand, R. (2013). A new hybrid day-ahead peak load forecasting method for Iran's National Grid. *Applied Energy*, 101:489–501.
- Mohammed, O., Park, D., Merchant, R., Dinh, T., Tong, C., Azeem, A., Farah, J., and Drake, C. (1995). Practical experiences with an adaptive neural network short-term load forecasting system. *IEEE Transactions on Power Systems*, 10(1):254–265.
- Mohan Saini, L. and Kumar Soni, M. (2002). Artificial neural network-based peak load forecasting using conjugate gradient methods. *IEEE Transactions on Power Systems*, 17(3):907–912.

- Molina, J.M., Sanchís, A., Berlanga, A., and Isasi, P. (2000). An Enhanced Classifier System for Autonomous Robot Navigation in Dynamic Environments. *Intelligent, Automation and Soft Computing Journal (AutoSoft)*, 6(2):113–124.
- Molina, J.M., García, J., Jiménez, F.J., and Casar, J.R. (2003). Cooperative Management of a Net of Intelligent Surveillance Agent Sensors. *International Journal of Intelligent Systems*, 18(3):279–307.
- Moody, J.E. and Darken, C.J. (1989). Fast learning networks of locally-tuned processing units. *Neural Computation*, 1(2):281–294.
- Moody, J. (1992). The effective number of parameters: An analysis of generalization and regularization in nonlinear learning systems. In *D.S. Touretzky (editor) Advances in Neural Information Processing Systems 4*, 1992, San Mateo (US), pages 29–39.
- Mori, H. and Yuihara, A. (2001). Deterministic Annealing Clustering for ANN-Based Short-Term Load Forecasting. *IEEE Transactions on Power Systems*, 16(3): 545–551.
- Mori, H. and Iwashita, D. (2005). ANN-based risk assessment for short-term load forecasting. In *Proceedings of the 13<sup>th</sup> International Conference on Intelligent Systems Application to Power Systems*, Nov. 2005, pages 446–451.
- Mori, H. and Kanaoka, D. (2006). Application of Preconditioned RBFN to Temperature Forecasting for Short-term Load Forecasting. In *2006 IEEE region 10 Conference (TENCON 2006)*, Nov. 2006, Hong Kong (China), pages 1–4.
- Moto-oka, T. (1983). Overview to the Fifth Generation Computer System project. In *Proceeding of the 10<sup>th</sup> annual international symposium on computer architecture*, June (1983), New York (US), vol. 11(3), pages 417–422.
- Mousavi, S.M. and Abyaneh, H.A. (2011). Effect of Load Models on Probabilistic Characterization of Aggregated Load Patterns. *IEEE Transactions on Power Systems*, 26(2):811–819.
- Müller, B. and Reinhardt, J. (1990). Neural Networks. An Introduction. *Ed. Springer-Verlag*.
- Müller, H. (1996). Negotiation Principles. In (*O’Hare and Jennings, 1996*), pages 211–229.
- Müller, J.P. (1997). The Design of Intelligence Agents (LNAI 1037). *Ed. Springer-Verlag*.
- Nadimi, V., Azadeh, A., Pazhoheshfar, P., and Saberi, M. (2010). An Adaptive-Network-Based Fuzzy Inference System for Long-Term Electric Consumption Forecasting (2008–2015): A Case Study of the Group of Seven (G7) Industrialized Nations: U.S.A, Canada, Germany, United Kingdom, Japan, France and Italy. In *2010 Fourth UKSim European Symposium on Computer Modeling and Simulation (EMS)*, Nov. 2010, Pisa (Italy), pages 301–305.
- Nair, A. and Joshi, S.K. (2010). Short Term Load Forecasting Using Probabilistic Neural Network Based Algorithm. In *2010 International Conference on Computational Intelligence and Communication Networks*, Nov. 2010, Bhopal (India), pages 128–132.

- Naphade, M., Banavar, G., Harrison, C., Paraszczak, J., and Morris, R. (2011). Smarter Cities and Their Innovation Challenges. *Computer*, 44(6):32–39.
- Neal, R.M. (1992). Bayesian learning via stochastic dynamics. In *C.L. Giles, S.J. Hanson, and J.D. Cowan (editors) Advances in Neural Information Processing Systems 5*, San Mateo (US), pp. 475–482.
- Neches, R., Fikes, R.E., Finin, T., Gruber, T.R., Senator, T., and Swartout, W.R. (1991). Enabling Technology for Knowledge Sharing. *AI Magazine*, 12(3):36–56.
- Nemes, T. (1962). Kibernetikai Gépek. *Ed. Akademiai Kiado*.
- Nelles, O. (2001). Nonlinear System Identification. *Ed. Springer*.
- Nelson, M.C. and Illingworth, W.T. (1991). A Practical Guide to neural nets. *Ed. Addison–Wesley*.
- Nestle, D. (2009). Integration of DER into Distribution Grid Operation and Decentralized Energy Management. In *Smart Grid Europe*, March 2009, Barcelona (Spain), pages 1–26.
- Newell, A. (1982). The knowledge level. *Artificial Intelligence*, 18:87–127.
- Nieto, M. (1989). Plasticidad sináptica. *Investigación y Ciencia*, 138.
- Nilsson, N. (1998). Artificial Intelligence – A New Synthesis. *Morgan Kaufmann Publishers*.
- Nissen, M.B. (2009). High performance development as distributed generation. *IEEE Potentials*, 28(6):25–31.
- Nose-Filho, K., Plasencia, A.D., and Minossi, C.R. (2011). Short-Term Multinodal Load Forecasting Using a Modified General Regression Neural Network. *IEEE Transactions on Power Delivery*, 26(4):2862–2869.
- Nwana, H.S., Ndumu, D.T., Lee, L.C., and Collis, J.C. (1999). ZEUS: A Toolkit for Building Distributed Multi-Agent Systems. *Applied Artificial Intelligence*, 13(1–2):129–185.
- O’Hare, G.M.P. and Jennings, N.R. (1996). Foundations of Distributed Artificial Intelligence. *Ed. Wiley–Interscience Publication*.
- O’Toole, E. and Clarke, S. (2012). Dynamic Forecasting and Adaptation for Demand Optimization in the Smart Grid. In *2012 International Workshop on Software Engineering for the Smart Grid (SE4SG)*, June 2012, Zurich (Switzerland), pages 30–33.
- Oja, E. (1982). A simplified neuron model as principal component analyzer. *Journal of Mathematical Biology*, 15(3):267–273.
- Omicini, A., Zambonelli, F., and Klusch, M. (2001). Coordination of Internet Agents: Models, Technologies, and Applications. *Ed. Springer–Verlag*.
- Omicini, A. and Ossowski, S. (2003). Objective versus Subjective Coordination in the Engineering of Agent Systems. *Intelligent Information Agents – Lecture Notes in Computer Science*, 2586:179–202.

- Omicini, A., Ossowski, S., and Ricci, A. (2004). Coordination Infrastructures in the Engineering of Multiagent Systems. In (*Bergenti et al., 2004*), pages 273–296.
- Onodera, H., Takeshita, K., and Tamaru, K. (1990). Hardware architecture for Kohonen network. In *IEEE International Symposium on Circuits and Systems*, May 1993, New Orleans (US), vol. 2, pages 1073–1077.
- Orponen, P. (1994). Computational Complexity of Neural Networks: A Survey. *NeuroCOLT Technical Report Series*.
- Orr, G.B. and Müller, K.R. (1998). Neural Networks: Tricks of the Trade. *Ed. Springer–Verlag*.
- Ossowski, S. and García–Serrano, A. (1998). La Coordinación en Sociedades Artificiales de Agentes. *Inteligencia Artificial – Revista Iberoamericana de Inteligencia*, 6(2):46–56.
- Ossowski, S. (1999). Co–ordination in Artificial Agent Societies: Social Structures and Its Implications for Autonomous Problem-Solving Agents. *Ed. Springer–Verlag*.
- Owen, G. (1995). Game Theory. *Academic Press*, tercera edición.
- Oyarzabal, J., Jimeno, J., Ruela, J., Engler, A., and Hardt, C. (2005). Agent Based Micro Grid Management System. In *Proceedings of the International Conference on Future Power Systems*, November 2005, pages 6–18.
- Paoletti, S., Casini, M., Giannitrapani, A., Facchini, A., Garulli, A., and Vicino, A. (2011). Load forecasting for active distribution networks. In *2011 2<sup>nd</sup> IEEE PES International Conference and Exhibition on Innovative Smart Grid Technologies (ISGT Europe)*, Dec. 2011, Manchester (UK), pages 1–6.
- Papalexopoulos, A.D., Hao, S., and Peng, T–M. (1994). An implementation of a neural network based load forecasting models for the EMS. *IEEE Transactions on Power Systems*, 9(4):1956–1962.
- Park, D.C., El–Sharkawi, M.A., Marks II, R.J., Atlas, L–E., and Damborg, M.J. (1991). Electric load forecasting using an artificial neural network. *IEEE Transactions on Power Systems*, 6(2):442–449.
- Park, J–I. (2010). A smart factory operation method for a smart grid. In *Proceedings of the 2010 40<sup>th</sup> International Conference on Computers and Industrial Engineering (CIE)*, July 2010, Awaji City (Japan).
- Parkpoom, S.J. and Harrison, G.P. (2008). Analyzing the Impact of Climate Change on Future Electricity Demand in Thailand. *IEEE Transactions on Power Systems*, 23(3):1441–1448.
- Parunak, H.V.D. (1990). Distributed AI and Manufacturing Control: Some Issues Insights. *Decentralized AI Elsevier Science Publishers*.
- Parunak, H.V.D. and Odell, J. (2002). Representing social structures in UML. In *5<sup>th</sup> International Conference on Autonomous Agents ‘01*, May–June 2001, Montreal (Canada), poster.

- Patil, R.S., Fikes, R.E., Patel–Schneider, McKay, D., Finin, T., Gruber, T., and Neches, R. (1992). The DARPA Knowledge Sharing Effort: Progress Report. *Proceedings of the Third International Conference on Principles Of Knowledge Representation And Reasoning, Morgan Kaufman*.
- Paul, R.J. and Chaney, T.S. (1998). Simulation optimisation using genetic algorithm. *Simulation Practice and Theory*, 6(6):601–611.
- Pavón, J. and Gómez–Sanz, J.J. (2003). Agent Oriented Software Engineering with INGENIAS. In *Multi–Agent Systems and Applications III, 3<sup>rd</sup> International central and Eastern European Conference on Multi–Agent Systems, CEEMAS 2003*, pages 394–403.
- Paysti, K. (1984). Box–Jenkins method in short–term forecasting of grid load in Finland. In *Proc. of 8<sup>th</sup> Power Systems Computation Conference*, Aug. 1984, Helsinki (Finland), pages 357–369.
- Personnaz, L., Guyon, I., and Dreyfus, G. Collective computational properties of neural networks: New learning mechanisms. *Physical Review A*, 34(5):4217–4228.
- Pilo, F., Pisano, G., and Soma, G.G. (2007). Neural Implementation of Microgrid Central Controllers. In *2007 5<sup>th</sup> IEEE International Conference on Industrial Informatics*, July 2007, Vienna (Austria), pages 1177–1182.
- Pilli–Sihvola, K., Aatola, P., Ollikainen, M., and Tuomenvirta, H. (2010). Climate change and electricity consumption–Witnessing increasing or decreasing use and costs? *Energy Policy*, 38(5): 2409–2419.
- Pinto, T., Vale, Z.A., Morais, H., Praça, I.C., and Ramos, C.A. (2009). Multi–agent based electricity market simulator with VPP: Conceptual and implementation issues. In *IEEE Power & Energy Society General Meeting, 2009. PES '09*, July 2009, Calgary (Canada), pages 1–9.
- Pipattanasomporn, M., Feroze, H., and Rahman, S. (2009). Multi–agent Systems in a Distributed Smart Grid: Design and Implementation. In *Proceedings of the Power Systems Conference and Exposition (PSCE '09)*, March 2009, Seattle (US), pages 15–18.
- Piras, A., Germond, A., Buchenel, B., Imhof, K., and Jaccard, Y. (1996). Heterogeneous artificial neural network for short term electrical load forecasting. *IEEE Transactions on Power Systems*, 11(1):397–402.
- Poggio, T. and Girosi, F. (1990). Networks for approximation and learning. *Proceeding of the IEEE*, 78(9):1481–1497.
- Ponce, P. (2011). *Inteligencia Artificial con Aplicaciones a la Ingeniería*. Ed. Marcombo.
- Prechelt, L. (1998). Early Stopping – But When? In *(Orr and Müller, 1998)*, pages 55–70.
- Principe, J.C., Euliano, N.R., and Lefebvre, W.C. (2000). *Neural and Adaptive Systems: Fundamentals Through Simulations*. Ed. John Wiley.

- Psiloglou, B.E., Giannakopoulos, C., Majithia, S., and Petrakis, M. (2009). Factors affecting electricity demand in Athens, Greece and London, UK: A comparative assessment. *Energy*, 34(11):1855–1863.
- Pudjianto, D., Ramsay, C., and Strbac, G. (2008). The Virtual Power Plant: Enabling Integration of distributed generation and demand. *Bulleting Editor Michael Durstewitz ISET*, 2008, pages 1–2.
- Pudjianto, D., Ramsay, C., and Strbac, G. (2009). Virtual power plant and system integration of distributed energy resources. *IET Renewable Power Generation*, 1(1):10–16.
- Puig, S. (2009). La neurona.....y mucho más. *El Pais.COM*. <http://lacomunidad.elpais.com/lentejas/2009/6/21/la-neurona-y-mucho-mas>, (Accessed 15/02/2012).
- Quintana, A., Gómez, J., and Reppen, N.D. (1978). Integration of an Adaptive Short Term Load Forecast Procedure into a New Energy Control Center. In *7<sup>th</sup> PSCC*, Aug. 1978, pages 426–431.
- Rahimi, F. and Ipakchi, A. (2010). Demand Response as a Market Resource Under the Smart Grid Paradigm. *IEEE Transactions on Smart Grid*, 1(1):82–88.
- Rahman, S. and Bhatnagar, R. (1987). An expert system based algorithm for short term load forecast. *IEEE Transactions on Power Systems*, 3(2):392–399.
- Rajurkar, K.P. and Nissen, J.L. (1985). Data-dependent systems approach to short-term load forecasting. *IEEE Transactions on Systems, Man and Cybernetics*, SMC–15(4):532–536.
- Ram, B. (1995). Tariffs and load management: a post privatisation study of the UK electricity supply industry. *IEEE Transactions on Power Systems*, 10(2):1111–1117.
- Ramezani, M., Falaghi, H. and Haghifam, M–R. (2005). Short-Term Electric Load Forecasting Using Neural Networks. In *EUROCON 2005. The International Conference on Compute as a Tool*, Nov. 2005, Belgrade (Serbia & Montenegro), vol. 2, pages 1525–1528.
- Ramón y Cajal, S. (1899–1904). *Textura del Sistema Nervioso del Hombre y de los Vertebrados*. Ed. N. Noya.
- Rao, A.S. and Georgeff, M.P. (1991). Modeling Rational Agents within a BDI–Architecture. In *Proceedings of the 2<sup>nd</sup> International Conference on Principles of Knowledge Representation and Reasoning*, Feb. 1991, San Mateo (US), pages 473–484.
- Rao, A.S. and Georgeff, M.P. (1992). An abstract architecture for rational agents. In *Proceedings of Third International Conference on Principles of Knowledge Representation and Reasoning*, Oct. 1992, San Mateo (US), pages 439–449.
- Rao, A.S. and Georgeff, M.P. (1995). BDI Agents from Theory to Practice. In *Proceedings of the First International Conference on Multi–Agent Systems (ICMAS–95)*, June 1995, San Francisco (US), pages 312–319.

- Rashidi, P. and Cook, D.J. (2009). Keeping the Resident in the Loop: Adapting the Smart Home to the User. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 39(5):949–959.
- Rashidi, P., Cook, D.J., Holder, L.B., and Schmitter–Edgecombe, M. (2011). Discovering Activities to Recognize and Track in a Smart Environment. *IEEE Transactions on Knowledge and Data Engineering*, 23(4):527–539.
- Rashvand, H.F., Salah, K., Calero, J.M.A., and Harn, L. (2010). Distributed security for multi-agent systems – review and applications. *IET Information Security*, 4(4):188–201.
- Razavi, S. and Tolson, B.A. (2011). A New Formulation for Feedforward Neural Networks. *IEEE Transactions on Neural Networks*, 22(10):1588–1598.
- Rejc, M. and Partos, M. (2011). Short–Term Transmission–Loss Forecast for the Slovenian Transmission Power System Based on a Fuzzy–Logic Decision Approach. *IEEE Transactions on Power Systems*, 26(3):1511–1521.
- Ricordel, P.M. (2001). Programmation Orientée Multi–Agents, Développement et Déploiement de Systèmes Multi–Agents Voyelles. *Ed. Institut national Polytechnique De Grenoble*.
- Riedmiller, M. and Braun, H. (1993). A direct adaptative method for faster backpropagation learning: The RPROP algorithm. In *IEEE International Conference on Neural Networks*, March–April 1993, San Francisco (US), pages 586–591.
- Ríos, D. and Ríos, X. (2008). Simulación. Métodos y Aplicaciones. *Ed. RA–MA*, segunda edición.
- Ritter, H., Martinetz, T., and Schulten, K. *Neural Computation and Self–Organizing Maps*. *Ed. Addison–Wesley*.
- Rodrigo, L., García–Serrano, A., and Martínez, P. (2001). Planteamiento pragmático–semántico para la gestión flexible de diálogos en asistentes virtuales. *Revista para el Procesamiento del Lenguaje Natural*, Revista nº 28, sep. 2001.
- Rodrigues, J.S. and Almeida, L.B. (1990). Improving the learning speed in topological maps of patterns. In *Proceedings of International Conference on Neural Networks*, June 1990, Dordrecht (Netherlands), pages 813–816.
- Rodríguez, C., Martínez, F., Davila, M.C., Santiago, L., Tomé, M., Cuevas, C., Sevilla, M., Soto, F., Artaiz, C., López, J., Pérez, H., González, G., Vegas, L.M., and López, S. (1998). Atlas de la demanda eléctrica Española. Proyecto INDEL. *Red Eléctrica de España, S.A.* [http://www.ree.es/sistema\\_electrico/pdf/indel/Atlas\\_INDEL\\_REE.pdf](http://www.ree.es/sistema_electrico/pdf/indel/Atlas_INDEL_REE.pdf), (Accessed 15/01/2011).
- Rodríguez–Fernández, C. and Gómez–Sanz, J.J. (2010). Self-management capability requirements with SelfMML & INGENIAS to attain self-organising behaviours. In *SOAR '10 Proceedings of the second international workshop on Self-organizing architectures*, June 2010, Washington DC (US), pages 11–20.

- Rojas, I. and Palomares, H. (2004). Soft-computing techniques for time series forecasting. In *Proc. Of the European Symposium on Artificial Neural Networks (ESANN)*, April 2004, Bruges (Belgium), pages 93–102.
- Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408.
- Rosenschein, J. and Zlotkin, G. (1994). Rules of Encounter – Designing Conventions for Automated Negotiation among Computers. *Ed. MIT Press*.
- Rowson, R.B. (1952). Electricity supply—a statistical approach to some particular problems. *Proceedings of the IEE – Part II: Power Engineering*, 99(68):151–167.
- Ruiz, N., Cobelo, I., and Oyarzabal, J. (2007). A Direct Load Control Model for Virtual Power Plant Management. *IEEE Transactions on Power Systems*, 24(2):959–966.
- Rumelhart, D.E. and McClelland, J.L. (1986). Parallel Distributed Processing. vol. 1: Foundations. *Ed. MIT Press*.
- Rumelhart, D.E., Hinton, G.E., and Williams, R.J. (1986). Learning representations by backpropagation errors. *Nature*, 323:533–536.
- Russel, B. (1948). Human Knowledge: Its Scope and Limits. *Ed. Routledge Classics*.
- Russel, S. and Norvig, P. (2003). Artificial Intelligence – A Modern Approach. *Ed. Prentice-Hall*, segunda edición.
- Rying, E.A., Bilbro, G.L., and Lu, J-C. (2002). Focused local learning with wavelet neural networks. *IEEE Transactions on Neural Networks*, 13(2):304–319.
- Saele, H. and Grande, O.S. (2011). Demand Response From Household Customers: Experiences From a Pilot Study in Norway. *IEEE Transactions on Smart Grid*, 2(1):102–109.
- Sánchez-Jiménez, M. (2006). European SmartGrids Technology Platform. Vision and Strategy for Europe’s Electricity Networks of the Future. *European Commission: Directorate-General for Research*. (Accessed 01/10/2012).
- Sandholm, T.W. (1999). Distributed Rational Decision Making. In (*Weiss, 1999*), pages 201–258.
- Sandholm, T.W., Larson, K., Anderson, M., Shehory, O., and Thome, F. (1999). Coalition Structures with Worst case Guarantees. *Artificial Intelligence*, 111(1–2):209–238.
- Santacana, E., Rackliffe, G., Tang, L., and Feng, X. (2010). Getting smart. *IEEE Power & Energy Magazine*, 8(2):41–48.
- Sarang, S.R., Dutaa, P., and Jalan, K. (2012). IT Infrastructure for Providing Energy-as-Service to Electric Vehicles. *IEEE Transactions on Smart Grid*, 3(2):591–604.
- Sargunraj, S., Sen Gupta, D.P., and Devi, S. (1997). Short-term load forecasting for demand side management. *IEE Proceedings-Generation, Transmission and Distribution*, 144(1):68–74.



- Sharda, R. and Patil, R.B. (1990). Neural networks as forecasting experts: an empirical test. In *Proceedings of the International Joint Conference on Neural Networks*, June 1990, pages 491–494.
- Sarle, W.S. (1995). Stopped training and other remedies for overfitting. In *Proc. Of the 27<sup>th</sup> Symposium on the Interface of Computing Science and Statistics*, June 1995, Pittsburgh (US), pages 352–360.
- Schneider, A., Takenawa, T., and Schiffman, D.A. (1985). 24-hour electric utility load forecasting. In *Comparative Models for Electrical Load Forecasting*, D. Bunn and E. Farmer, Eds. Wiley.
- Schreiber, G., Wielinga, B., de Hoog, R., Akkermans, H., and Van de Velde, W. (1994). CommonKADS: A comprehensive methodology for KBS development. *IEEE Expert Magazine*, 9(6):28–37.
- Schumacher, M. (2001). Objective Coordination in Multi-Agent System Engineering – Design and Implementation. *Ed. Springer-Verlag*.
- Searle, J.R. (1979). Speech Acts: An Essay in the Philosophy of Language. *Ed. Cambridge University Press*.
- Sebastian, M. and Maire, J. (2008). Towards Virtual Power Plant integration into the Transmission and Distribution grid tools. *Bulleting Editor Michael Durstewitz ISET*, 2008, pages 1–2.
- Sechilariu, M., Baochao, W., and Locment, F. (2013). Building Integrated Photovoltaic System With Energy Storage and Smart Grid Communication. *IEEE Transactions on Industrial Electronics*, 60(4):1607–1618.
- Seel, N. (1989). Agents Theories and Architectures. *PhD thesis, Surrey University Guildford (UK)*.
- Seelye, A.M., Schitter-Edgecombe, M., Das, B., and Cook, D.J. (2012). Application of Cognitive Rehabilitation Theory to the Development of Smart Prompting Technologies. *IEEE Reviews in Biomedical Engineering*, 5:29–44.
- Sheblé, G.B. (2008). Smart Grid Millionaire. *IEEE Power and Energy Magazine*, 6(1):22–28.
- Shepherd, G.M. (1997). The Synaptic Organization of the Brain. *Oxford University Press*, cuarta edición.
- Sichman, J. (1995). Du Raisonnement Social Chez des Agents. *PhD thesis, Institut National Polytechnique de Grenoble (France)*.
- Sichman, J. and Demazeau, Y. (1995). Exploiting Social Reasoning to Deal with Agency Level Inconsistencies. In *Proceedings of the 1<sup>st</sup> International Conference on Multi-Agent Systems*, June 1995, San Francisco (US), pages 352–359.
- Simmons, G.J. (1983). The Prisoners' Problem and the Subliminal Channel. In *Advances in Cryptology: Proceedings of CRYPTO '83*, Aug. 1983, Santa Barbara (US), pages 51–67.
- Simpson, P.K. (1989). Artificial Neural Systems. *Pergamon Press*.

- Singh, M.P. (1994). Multiagent Systems: A Theoretical Framework for Intentions, Know How and Communications. *Lecture Notes in Artificial Intelligence*, 799, Ed. Springer-Verlag.
- Sjöberg, J. (1995). Non-linear system identification with neural networks. *PhD thesis, Linköping University (Sweden)*.
- Skrzypczak, C. (2010). The intelligent home of 2010. *IEEE Communications Magazine*, 25(12):81–84.
- Smith, R.G. (1980). The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver. *IEEE Transactions on Computers*, C-29(12):1104–1113.
- Stevens, C.F. (1979). The Neuron. *Scientific American*, 241(3):54–65.
- Studer, R., Benjamins, V.R., and Fensel, D. (1998). Knowledge Engineering: Principles and Methods. *IEEE Transactions on data and Knowledge Engineering*, 25(1–2):161–197.
- Su, H. and Zhang, Y. (2007). Short-term load forecasting using  $H_\infty$  filter and Elman neural network. In *IEEE International Conference on Control and Automation (ICCA)*, May 2007, Guangzhou (China), pages 1868–1872.
- Sugaya, Y., Omachi, S., Takeuchi, A., and Nozaki, Y. (2012). A Statistical Analysis on Operation Scheduling for an Energy Network Project. *IEEE Transactions on Parallel and Distributed Systems*, 23(9):1583–1592.
- Sun, W., Lu, J., and He, Y. (2005). Information Entropy Based Neural Network Model for Short-Term Load Forecasting. In *2005 IEEE/PES Transmission and Distribution Conference and Exhibition: Asia and Pacific*, Aug. 2005, Dalian (China), pages 1–5.
- Sycara, K. and Zeng, D. (1996). Coordination of Multiple Intelligent Software Agents. *Int. J. Cooperative Information Systems*, 5(2&3):181–212.
- Tang, Z., de Almeida, C., and Fishwick, P. (1991). Time series forecasting using neural networks vs. Box Jenkins methodology. *Transactions of The Society for Modeling and Simulation International – SIMULATION*, 57(5):303–310.
- Taylor, J.W. and Buizza, R. (2002). Neural Network Load Forecasting With Weather Ensemble Predictions. *IEEE Transactions on Power Systems*, 17(3):626–632.
- Teräsvirta, T. and Anderson, H.M. (1992). Characterising nonlinearities in business cycles using smooth transition autoregressive models. *Journal of Applied Econometrics*, 7(1):119–136.
- Tesla, N. (1888). A New System of Alternating Current Motors and Transformers. *American Institute of Electrical Engineers*.
- Thimm, G. and Fiesler, E. (1995). Neural network initialization. In *Proceedings International Workshop on Artificial Neural Networks (IWANN95)*, June 1995, Malaga-Torremolinos (Spain), pages 535–542.

- Thomson, W. (1997). Cooperative Theory of Bargaining I: Classical. In (*Hart and Mas-Colell, 1997*), pages 9–24.
- Tolksdorf, R. (2000). Models of Coordination. In *Proceedings of the First International Workshop on Engineering Societies in the Agent World: Revised Papers*, Ed. Springer-Verlag, pages 78–92.
- Tollenaere, T. (1995). SuperSAB: Fast adaptive back propagation with good scaling properties. *Neural Networks*, 3(5):561–573.
- Tou, J.T. and González, R.C. (1974). Pattern Recognition Principles. Ed. Addison-Wesley.
- Toyoda, J., Chen, M-S., and Inoue, Y. (1970). An Application of State Estimation to Short-Term Load Forecasting, Part I: Forecasting Modeling. *IEEE Transactions on Power Apparatus and Systems*, PAS-89(7):1678–1682.
- Tsekouras, G.J., Hatziaargyriou, N.D., and Dialynas, E.N. (2007). Two-Stage Pattern Recognition of Load Curves for Classification of Electricity Customers. *IEEE Transactions on Power Systems*, 22(3):1120–1128.
- Tuomola, R. (2000). Cooperation: A Philosophical Study. *Kluwer Academic Publishers*.
- Vaccaro, A., Popou, M., Villacci, D., and Terzija, V. (2011). An integrated framework for smart microgrids modelling, monitoring, control, communication, and verification. *Proceedings of the IEEE*, 99(1):119–132.
- Van Wyk, A.B. and Engelbrecht, A.P. (2010). Overfitting by PSO trained feedforward neural networks. In *2010 IEEE Congress on Evolutionary Computation (CEC)*, July 2010, Barcelona (Spain), pages 1–8.
- Varian, H.R. (1992). Microeconomic analysis. Ed. W.W. Norton.
- Vemuri, S., Hovelda, B., and Mohebbi, S. (1986). Short-term load forecasting based on weather load models. In *Proc. IFAC Conf. on Power Systems and Power Plant Control*, Aug. 1986, pages 565–570.
- Verschueren, T., Haerick, W., Mets, K., Develder, C., De Turck, F., and Pollet, T. (2010). Architectures for smart end-user services in the power grid. In *2010 IEEE/IFIP Network Operations and Management Symposium Workshops (NOMS Wksp)*, April 2010, Osaka (Japan), pages 316–322.
- Vesanto, J. and Alhoniemi, E. (2000). Clustering of the self-organizing map. *IEEE Transactions on Neural Network*, 11(3):586–600.
- Von Martial, F (1992). Coordinating Plans of Autonomous Agents. Ed. Springer-Verlag.
- Vukmirovic, S., Erdeljan, A., Kulić, F., and Luković, S. (2010). Software architecture for Smart Metering systems with Virtual Power Plant. In *MELECON 2010 – 2010 15<sup>th</sup> IEEE Mediterranean Electrotechnical Conference*, April 2010, Valletta (Malta), pages 448–451.

- Wang, Q.J. (1997). Using genetic algorithms to optimise model parameters. *Environmental Modelling and Software*, 12(1):27–34.
- Wang, Z.Y. (2006). Development Case-based Reasoning System for Short-term Load Forecasting. In *2006 IEEE Russia Power Engineering Society General Meeting*, Oct. 2006, pages 1–6.
- Wang, J-J., Niu, D-X., and Li, L. (2009). An ARMA cooperate with Artificial Neural Network approach in Short-Term. In *2009 Fifth International Conference on Natural Computation Load Forecasting*, Aug. 2009, Tianjian (China), pages 60–64.
- Wang, Y., Xia, Q., and Kang, C. (2011). Secondary Forecasting Based on Deviation Analysis for Short-Term Load Forecasting. *IEEE Transactions on Power Systems*, 26(2):500–507.
- Wassermann, P.D. (1993). *Advanced Methods in Neural Computing*. Van Nostrand Reinhold Computer Library.
- Wegner, P. (1997). Why Interaction is more Powerful than Computing. *Communications of the ACM*, 40(5):80–91.
- Weiss, G. (1999). *Multiagent System – A Modern Approach to Distributed Artificial Intelligence*. Ed. MIT Press.
- Werbos, P.J. (1974). *Beyond Regression: New Tools for Prediction and Analysis in Behavioral Sciences*. PhD thesis, Harvard University (US).
- Werbos, P.J. (2011). Computational Intelligence for the Smart Grid–History, Challenges, and Opportunities. *IEEE Computational Intelligence Magazine*, 6(3):14–21.
- Wesson, R., Hayes–Roth, F., Burge, J., Stasz, C., and Sunshine, C.A. (1981). Network Structures for Distributed Situation Assesment. *IEEE Transactions on Systems, Man and Cybernetics*, 11(1):5–23.
- Westwick, D. and Verhaegen, M. (1996). Identifying MIMO wiener systems using subspace model identification methods. *Signal Processing*, 52(2):235–258.
- White, H. (1989). Learning in Artificial Neural Networks: A Statistical Perspective. *Neural Computation*, 1(4):425–464.
- Widrow, B. and Hoff, M. (1960). Adaptive Switching Circuits. In *1960 IREWESCON Convention Record Part 4*, Aug. 1960, Los Angeles (US), pages 96–104.
- Widrow, B. and Winter, R. (1988). Neural nets for adaptive filtering and adaptive pattern recognition. *IEEE Computer*, 21(3):25–39.
- Widrow, B. and Lehr, M.A. (1990). 30 years of adaptive neural networks: Perceptron, Madaline and Backpropagation. *Proceedings of the IEEE*, 78(9):1415–1442.
- Wiederhold, G. (1992). Mediators in the architecture of future information systems. *Computer*, 25(3):38–49.

- Wigren, T. (1993). Recursive prediction error identification using the nonlinear wiener model. *Automatica*, 29(4):1011–1026.
- Wille–Hausmann, B., Erge, T., and Wittner, C. (2010). Decentralized optimisation of cogeneration in virtual power plant. *Solar Energy*, 84(4):604–611.
- Wilson, D.R. and Martinez, T.R. (2003). The general inefficiency of batch training for gradient descent learning. *Neural Networks*, 16(10):1429–1451.
- Wong, H.C. and Sycara, K. (2000). Adding security and trust to multi–agent systems. *Applied Artificial Intelligence: An International Journal*, 14(9):927–941.
- Wooldridge, M. and Jennings, N.R. (1995). Agent Theories, Architectures and Languages: A Survey. *Intelligent Agents Lecture Notes in Computer Science*, 890:1–39.
- Wooldridge, M. (1997). Agent–based software engineering. *IEE Proceedings–Software Engineering*, 144(1):26–37.
- Wooldridge, M., Jennings, N.R., and Kinny, D. (2000). The Gaia Methodology for Agent–Oriented Analysis and Design. *Journal of Autonomous Agents and Multi-Agent Systems*, 3(3):285–312.
- Wooldridge, M. (2009). An Introduction to MultiAgent Systems. *Ed. John Wiley and Sons*, segunda edición.
- Wu, X–X. and Liu, J–G. (2009). A New Early Stopping Algorithm for Improving Neural Network Generalization. In *Second International Conference on Intelligent Computation Technology and Automation, 2009. ICICTA '09*, Oct. 2009, pages 15–18.
- Xinghuo, Y., Cecati, C., Dillon, T., and Simoes, M.G. (2011). The new frontier of smart grids. *IEEE Industrial Electronincs Magazine*, 5(3):49–63.
- Yan, Y., Qian, Y., Sharif, H., and Tipper, D. (2012). A Survey on Cyber Security for Smart Grid Communications. *IEEE Communications Surveys & Tutorials*, 14(4):998–1010.
- Yang, H.T., Chen, S.C., and Peng, P.C. (2005). Genetic k–means–algorithm–based classification of direct load–control curves. *IEEE Proceedings–Generation, Transmission and Distribution*, 152(4):489–495.
- Yongchum, L. (2010). Application of Elman Neural Network in Short–Term Load Forecasting. In *2010 International Conference on Artificial Intelligence and Computational Intelligence (AICI)*, Oct. 2010, Sanya (Chinapages), pages 141–144.
- Yu, N–P., Liu, C–C., and Price, J. (2010). Evaluation of Market Rules Using a Multi–Agent System Method. *IEEE Transactions on Power Systems*, 25(1):470–479.
- Yuan, J–L. and Fine, T.L. (1998). Neural network design for small training sets of high dimension. *IEEE Transactions on Neural Network*, 9(2):266–280.
- Yuan, Y.M., Quin, X., Wu, C.L., and Tang, T.Z. (2011). Architecture and Data Vitalization of Smart City. *Advanced Materials Research*, 403–408:2564–2568.

- Yule, G. (1927). On a method of investigating periodicity in disturbed series with special referente to wolfer's sunsport numbers. *Phil. Trans. Roy. Soc. London A*, 226:227–298.
- Zaballos, A., Vallejo, A., and Selga, J.M. (2011). Heterogeneous Communication Architecture for the Smart Grid. *IEEE Network*, 25(5):30–37.
- Zadeh, L.A. (1994). Fuzzy Logic, Neural Networks and Soft Computing. *Communications of the ACM*, 37(3):77–84.
- Zhang, Z. and Ye, S. (2011). Long Term Load Forecasting and Recommendations for China Based on Support Vector Regression. In *2011 International Conference on Information Management, Innovation Managemenet and Industrial Engineering*, Nov. 2011, Shenzhen(China), vol. III, pages 597–602.
- Zhang, T., Zhang, G., Lu, J., Feng, X., and Yang, W. (2012a). A New Index and Classification Approach for Load Pattern Analysis of Large Electricity Customers. *IEEE Transactions on Power Systems*, 27(1):153–160.
- Zhang, W–Y., Hong, W–C., Dong, Y., Tsai, G., Sung, J–T., and Fan, G–F. (2012b). Application of SVR with chaotic GASA algorithm in cyclic electric load forecasting. *Energy*, 45(1):850–858.
- Zapranis, A. and Refenes, A.P. (1999). Principles of Neural Model Identification, Selection and Adequacy. *Ed. Springer*.
- Zareipoor, H., Cañizares, C.A., and Bhattacharga, K. (2010). Economic impact of electricity market price forecasting errors: A demand-side analysis. *IEEE Transactions on Power System*, 25(1):254–262.
- Zhou, L., Xu, F.Y., and Ma, Y.N. (2010). Impact of smart metering on energy efficiency. In *2010 International Conference on Machine Learning and Cybernetics (ICMLC)*, July 2010, London (UK), pages 3213–3218.
- Zurada, J.M. (1992). Analog implementation of neural networks. *IEEE Circuits and Devices Magazine*, 8(5):36–41.

## ABREVIATURAS

<i>a.m.</i>	<i>ante meridiem</i> – Antes del Mediodía
<i>kW</i>	kilovatio
<i>p.m.</i>	<i>post meridiem</i> – Después del Mediodía
<i>AAII</i>	Australian Artificial Intelligence – Institute Instituto de Inteligencia Artificial Australiano
<i>ABLE</i>	Agent Building and Learning Environment – Construcción de Agente y Entorno de Aprendizaje
<i>ACL</i>	Agent Communication Language – Lenguaje de Comunicación de Agente
<i>ADALINE</i>	ADaptative LINear Elements
<i>AEMET</i>	Agencia Estatal de Meteorología – Elementos Lineales Adaptativos
<i>AFCM</i>	Adaptive Fuzzy Combination Model – Modelo de Combinación Fuzzy Adaptativo
<i>AI</i>	Artificial Intelligence – Inteligencia Artificial
<i>AIC</i>	Akaike Information Criterion – Criterio de Información Akaike
<i>ANFIS</i>	Adaptative NeuroFuzzy Inference System – Sistema de Inferencia Neurofuzzy Adaptativa
<i>ANN</i>	Artificial Neural Network – Red Neuronal Artificial
<i>AR</i>	Auto–Regressive – Auto–Regresivo
<i>ARCH</i>	AutoRegressive Conditional Heteroscedasticity – Heteroscedasticidad Condicional AutoRegresiva
<i>ARIMA</i>	Auto–Regressive Integrated Moving Average – Auto–Regresivo Integrada de Media Móvil
<i>ARMA</i>	Auto–Regressive and Moving Average – Auto–Regresivo y Media Móvil
<i>ARMAX</i>	Auto–Regressive and Moving Average with eXogenous inputs – Auto–Regresivo y Media Móvil con Entradas Exógenas
<i>AUML</i>	Agent Unified Modeling Language – Lenguaje de Modelado Unificado de Agente
<i>BDI</i>	Belief–Desire–Intention – Creencia–Deseo–Intención
<i>CA</i>	Chaotic Algorithm – Algoritmo Caótico
<i>CABC</i>	Chaotic Artificial Bee Colony – Colonia de Abejas Artificial Caótica
<i>CAS</i>	Chaotic Ant Swarm – Enjambre de Hormigas Caóticas
<i>CE</i>	Conjunto de Datos de Entrenamiento.
<i>CEDER</i>	Centro de Desarrollo de Energías Renovables
<i>CERTS</i>	Consortium for Electric Reliability Technology Solutions
<i>CF</i>	Conjunto de Datos Futuros

<i>CGA</i>	Chaotic Genetic Algorithm – Algoritmo Genético Caótico
<i>CGASA</i>	Chaotic Genetic Algorithm Simulated Annealing – Algoritmo Genético Caótico Recocido Simulado
<i>CH</i>	Conjunto de Datos Históricos
<i>CI</i>	Computational Intelligence – Inteligencia Computacional
<i>CIEMAT</i>	Centro de Investigaciones Energéticas, Medioambientales y Tecnológicas
<i>CNE</i>	Comisión Nacional de la Energía
<i>CP</i>	Conjunto de Datos de Prueba
<i>CSS</i>	Cascading Style Sheets – Hojas de Estilo en Cascada
<i>CT</i>	Conjunto de Datos de Test
<i>CVPP</i>	Commercial Virtual Power Plant– Planta de Energía Virtual Comercial
<i>DA</i>	Deterministic Annealing – Recocido Determinista
<i>DARPA</i>	Defense Advanced Research Projects Agency – Agencia de Proyectos de Investigación Avanzados de Defensa
<i>DC</i>	Data Concentrator – Concentrador de Datos
<i>DER</i>	Distributed Energy Resources – Recursos de Energía Distribuidos
<i>DG</i>	Distributed Generation – Generación Distribuida
<i>DOM</i>	Document Object Model – Modelo de Objeto de Documento
<i>DR</i>	Demand Response – Respuesta a la Demanda
<i>DSM</i>	Demand Side Management – Gestión del Lado de la Demanda
<i>DSO</i>	Distribution System Operator – Operador del Sistema de Distribución
<i>EMS</i>	Energy Management System – Sistema de Gestión de Energía
<i>EPRI</i>	Electric Power Research Institute – Instituto de Investigación de Potencia Eléctrica
<i>ESN</i>	Echo State Network – Red de Estado Eco
<i>ETP</i>	European Technology Platform– Plataforma Tecnológica Europea
<i>EV</i>	Electrical Vehicle – Vehículo Eléctrico
<i>FASE</i>	Forecast–Aided State Estimator – Pronóstico Asistido por Estimador de Estado
<i>FCM</i>	Fuzzy c–means
<i>FIPA</i>	Foundation for Intelligent and Physical Agents – Fundación para Agentes Inteligentes Físicos
<i>FIR</i>	Finite Impulse Response – Respuesta de Impulso Finita
<i>FOA</i>	Fly Optimization Algorithm – Algoritmo de Optimización de Mosca
<i>FPE</i>	Final Prediction Error – Error de Predicción Final
<i>GA</i>	Genetic Algorithm – Algoritmo Genético
<i>GE</i>	General Electric



---

<i>GEI</i>	Gas Efecto Invernadero
<i>GDE</i>	Gestión de la Demanda Eléctrica
<i>GRNN</i>	Generalized Regression Neural Network – Red Neuronal de Regresión Generalizada
<i>GWh</i>	Gigavatio–hora
<i>I&amp;K</i>	Information and Knowledge – Información y Conocimiento
<i>IAD</i>	Inteligencia Artificial Distribuida
<i>IDK</i>	INGENIAS Development Kit
<i>IDL</i>	Interface Description Language – Lenguaje de Descripción de Interface
<i>JADE</i>	Java Agent DEvelopment framework
<i>JAS</i>	Java Agent Services
<i>JatLite</i>	Java Agent Template Lite
<i>JESS</i>	Joint Expert Speciation System
<i>KIF</i>	Knowledge Interchange Format – Formato de Intercambio de Conocimiento
<i>KQML</i>	Knowledge Query and Manipulation Language – Consulta de Conocimiento y Lenguaje de Manipulación
<i>LIDAR</i>	Light Detection and Ranging
<i>LMS</i>	Least Mean Squares – Mínimos Cuadrados Medios
<i>LR</i>	Linear Regression – Regresión Lineal
<i>LSSVM</i>	Least Squares Support Vector Machine – Máquina de Vector de Apoyo de Mínimos Cuadrados
<i>LTLF</i>	Long–Term Load Forecasting – Predicción de Carga a Largo Plazo
<i>LVQ</i>	Learning Vector Quantization – Cuantificación Vectorial de Aprendizaje
<i>MA</i>	Moving Average – Media Móvil
<i>MADALINE</i>	Many Adalines – Muchos Adalines
<i>MAM</i>	Moving Average Model – Modelo de Media Móvil
<i>MAPE</i>	Mean Absolute Percent Error – Error Absoluto Porcentual de la Media
<i>MAS</i>	Multi–Agent System – Sistema Multi–Agente
<i>MCOV</i>	Modified COVariance – Covarianza Modificada
<i>MCS</i>	Multiple Classifier Systems – Sistema Clasificador Múltiple
<i>MDL</i>	Minimun Description Length – Longitud de Descripción Mínima
<i>ME</i>	Maximun Error– Error Máximo
<i>MGCC</i>	MicroGrid Central Controller – Controlador Central de la Microrred
<i>MINER</i>	Ministerio de Industria y Energía
<i>MIT</i>	Massachusetts Institute Tchnology – Instituto Tecnológico de Massachusetts
<i>MLP</i>	Multi–Layer Perceptron – Perceptron Multicapa

---

<i>MSE</i>	Mean Squared Error – Error Cuadrático Medio
<i>MTLF</i>	Medium–Term Load Forecasting – Predicción de Carga a Medio Plazo
<i>MVA</i>	Megavoltamperio
<i>MW</i>	Megavatio
<i>NARX</i>	Nonlinear AutoRegressive with eXogenous inputs – AutoRegresiva no Lineal con Entradas Exógenas
<i>NDTL</i>	Next Day’s Total Load – Carga Total del Día Siguiente
<i>OFDM</i>	Orthogonal Frequency Division Multiplexing – Multiplexación por División de Frecuencias Ortogonales
<i>OLS</i>	Ordinary Least Squares – Mínimos Cuadrados Ordinarios
<i>ONU</i>	Organización de Naciones Unidas
<i>OPEP</i>	Organización de Países Exportadores de Petróleo
<i>OWL–S</i>	Ontology Web Language base on Semantic Web – Lenguaje Ontológico Basado en Web Semántica
<i>PCA</i>	Principal Component Analysis – Análisis de Componentes Principales
<i>PDF</i>	Probability Density Function – Función de Densidad de Probabilidad
<i>PDP</i>	Parallel Distributed Processing – Procesamiento Distribuido en Paralelo
<i>PLC</i>	Power Line Communications – Comunicaciones en Línea de Energía
<i>PLF</i>	Peak Load Forecasting – Predicción de la Carga Pico
<i>PLS</i>	Partial Least–Square – Mínimos Cuadrados Parciales
<i>PNN</i>	Probabilistic Neural Network – Red Neuronal Probabilística
<i>PPG</i>	Planificación Parcial Global – Planificación Parcial Global
<i>PRIME</i>	PoweRline Intelligent Metering Evolution – Evolución de la Medida Inteligente PoweRline
<i>RAQ</i>	Real Academia Española
<i>RBF</i>	Radial Basis Function – Función de Base Radial
<i>RBNN</i>	Radial Basis Function Neural Network – Red Neuronal de Función Base Radial
<i>RDF</i>	Resource Description Framework – Marco de Descripción de Recurso
<i>RDP</i>	Resolución Distribuida de Problemas
<i>RE</i>	Régimen Especial
<i>REE</i>	Red Eléctrica de España
<i>RMSE</i>	Root Mean Square Error – Raíz Cuadrada del Error Cuadrático Medio
<i>RPC</i>	Remote Procedure Call – Llamada de Procedimiento Remoto
<i>SAX</i>	Simple API for XML – Simple API para XML
<i>SBC</i>	Schwarz Bayesian Criterion – Criterio Bayesiano Schwarz

---

<i>SC</i>	Smart City – Ciudad Inteligente
<i>SEDE</i>	SEñal de la Demanda Eléctrica de la actividad económica
<i>SG</i>	Smart Grid – Red Inteligente
<i>SIFE</i>	Sistema Integrado de Facturación de Energía Eléctrica
<i>SM</i>	Smart Meter – Medidor Inteligente
<i>SOM</i>	Self–Organizing Map – Mapa Auto–Organizador
<i>SSE</i>	Sum of Squared Error – Suma de Error Cuadrático
<i>SSL</i>	Secure Socket Layer – Capa de Enchue Segura
<i>STLF</i>	Short–Term Load Forecasting – Predicción de Carga a Corto Plazo
<i>SVM</i>	Support Vector Machine – Máquina de Vector de Apoyo
<i>SVR</i>	Support Vector Regression – Regresión de Vector de Apoyo
<i>TIC</i>	Tecnología de la Información y Comunicación
<i>TSO</i>	Transmission System Operator – Operador del Sistema de Transporte
<i>TVPP</i>	Technical Virtual Power Plant – Planta de Energía Virtual Técnica
<i>UE</i>	Unión Europea
<i>UML</i>	Unified Modeling Language – Lenguaje de Modelado Unificado
<i>VAR</i>	Vector AutoRegressive – Vector AutoRegresivo
<i>VC</i>	Validación Cruzada
<i>VLF</i>	Valley Load Forecasting – Predicción de la Carga Valle
<i>VPP</i>	Virtual Power Plant – Planta de Energía Virtual
<i>VPPC</i>	Virtual Power Plantr Control – Control de la Planta de Energía Virtual
<i>VSTLF</i>	Very Short–Term Load Forecasting – Predicción de Carga a Muy Corto Plazo
<i>WSDL</i>	Web Services Description Language – Lenguaje de Descripción de Servicios Web
<i>XML</i>	eXtensible Markup Language – Lenguaje de Marca eXtensible
<i>XSL</i>	eXtensible Stylesheet Language – Lenguaje eXtensible de Hojas de Estilo



# APÉNDICE

## ANEXO A. CLASIFICACIÓN DE ANN

A continuación se dará una serie de clasificaciones posibles de las ANNs, basadas en diferentes criterios, y según lo mostrado por Hilera and Martínez (1995), Flórez and Fernández (2008) y Ponce (2011).

**Tabla A.1. Clasificación de las ANN monocapa. Fuente Propia con Datos de Hilera and Martínez (1995).**

TIPOS DE CONEXIONES		MODELO DE RED
Conexiones laterales explícitas	Conexiones autorecurrentes	<i>Brain-State-in-a-Box</i>
		<i>Additive Grossberg</i>
		<i>Shuntinh Grossberg</i>
		<i>Optimal Linear Associative Memory</i>
	No autorecurrentes	<i>Hopfield</i>
		<i>Boltzmann Machine</i>
		<i>Cauchy Machine</i>
Crossbar		<i>Learning Matrix</i>

**Tabla A.2. Clasificación de las ANN multicapa. Fuente Propia con Datos de Hilera and Martínez (1995). (Continúa).**

Nº DE CAPAS	TIPO DE CONEXIONES		MODELO DE RED
2 Capas	Conexiones hacia adelante Feedforward		<i>Adaline/Madaline</i>
			<i>Perceptron</i>
			<i>Linear/Assoc. Rewar. Penalty</i>
			<i>Linear Assoc. Memory</i>
			<i>Optimal Linear Assoc. Mem.</i>
			<i>Drive Reinforcement</i>
	Conexiones adelante/atrás Feedforward/Feedback	Conexiones laterales implícitas y autorecurrentes	<i>Learning Vector Quantizer</i>
			<i>Topology Preserving Map</i>
		Sin conexiones laterales	<i>Bidirectional Assoc. Mem. (BAM)</i>
			<i>Adaptive BAM</i>
			<i>Temporal Assoc. Memory</i>
			<i>Fuzzy Assoc. Memory</i>
		Con conexiones laterales y autorecurrentes	<i>Competitive Adaptive BAM</i>
			<i>Adaptive Resonance Theory</i>

**Tabla A.2. Fuente Propia con Datos de Hilera and Martínez (1995).**

<i>Nº DE CAPAS</i>	<i>TIPO DE CONEXIONES</i>		<i>MODELO DE RED</i>
3 Capas	Conexiones hacia adelante Feedforward	Sin conexiones laterales	<i>Adaptive Heuristic Critic</i>
			<i>Botzmann/Cauchy Machine</i>
	Conexiones hacia adelante Feedforward	Con conexiones laterales	<i>Counterpropagation</i>
			<i>Botzmann/Cauchy Machine</i>
	Conexiones adelante/atrás Feedforward/Feedback		<i>Botzmann/Cauchy Machine</i>
N Capas	Conexiones hacia adelante Feedforward		<i>Backpropagation</i>
	Conexiones hacia adelante Feedforward (Jerarquía de capas)		<i>Cognitron/Neocognitron</i>
			<i>Cauchy Machine</i>

**Tabla A.3. Clasificación de las ANN con aprendizaje no supervisado. Fuente Propia con Datos de Flórez and Fernández (2008).**

<i>TIPO DE APRENDIZAJE NO SUPERVISADO</i>		<i>MODELO DE RED</i>
Aprendizaje Hebbiano	Off Line	<i>Hopfield</i>
		<i>Learning Matrix</i>
		<i>Temporal Assoc. memory</i>
		<i>Linear Assoc. Memory (LAM)</i>
		<i>Optimal LAM</i>
	On Line	<i>Drive Reinforcement</i>
		<i>Fuzzy Assoc. Memory</i>
		<i>Additive Grossberg</i>
		<i>Shunting Grossberg</i>
		<i>Bidirectional Assoc. Memory (BAM)</i>
Aprendizaje Competitivo/Cooperativo	Off Line	<i>Adaptive BAM</i>
		<i>Learning Vector Quantizer</i>
	On Line	<i>Cognitron/Neocognitron</i>
		<i>Topology Preserving Map</i>
		<i>Adaptive Resonance Theory</i>

**Tabla A.4. Clasificación de las ANN en función de la asociación. Fuente Propia con Datos de Flórez and Fernández (2008). (Continúa).**

<i>REDES HETEROASOCIATIVAS</i>	<i>REDES AUTOASOCIATIVAS</i>
<i>Perceptron</i>	<i>Brain-State-in-a-Box</i>

**Tabla A.4. Fuente Propia con Datos de Flórez and Fernández (2008).**

<i>REDES HETEROASOCIATIVAS</i>	<i>REDES AUTOASOCIATIVAS</i>
<i>Adaline/Madaline</i>	<i>Hopfield</i>
<i>Backpropagation</i>	<i>Additive Grossberg</i>
<i>Linear Rewar. Penalty</i>	<i>Shunting Grossberg</i>
<i>Assoc. Rewar. Penalty</i>	
<i>Adaptive Heuristic Critic</i>	
<i>Boltzmann Machine</i>	
<i>Cauchy Machine</i>	
<i>Learning Matrix</i>	
<i>Temporal Assoc. memory</i>	
<i>Linear Assoc. Memory</i>	
<i>Optimal Linear Assoc. Memory</i>	
<i>Drive Reinforcement</i>	
<i>Fuzzy Assoc. Memory</i>	
<i>Counterpropagation</i>	
<i>Bidirectional Assoc. Memory</i>	
<i>Adaptive Bidirectional Assoc. Memory</i>	
<i>Cognitron/Neocognitron</i>	
<i>Topology Preserving Map</i>	
<i>Learning Vector Quantizer</i>	

**Tabla A.5. Clasificación de las ANN con aprendizaje supervisado. Fuente Propia con Datos de Flórez and Fernández (2008).**

<i>TIPO DE APRENDIZAJE SUPERVISADO</i>		<i>MODELO DE RED</i>
Aprendizaje por corrección del error	Off Line	<i>Perceptron</i>
		<i>Adaline/Madaline</i>
		<i>Backpropagation</i>
		<i>Brain–State–in–a–Box</i>
		<i>Counterpropagation</i>
Aprendizaje por refuerzo	On Line	<i>Linear Rewar. Penalty</i>
		<i>Assoc. Rewar. Penalty</i>
		<i>Adaptive Heuristic Critic</i>
Aprendizaje estocástico	Off Line	<i>Botzmann Machine</i>

**Tabla A.6. Clasificación de las ANN en función de la información de entrada y salida. Fuente Propia con Datos de Ponce (2011).**

<b>REDES CONTINUAS</b> <i>E: ANALÓGICA</i> <i>S: ANALÓGICA</i>	<b>REDES HÍBRIDAS</b> <i>E: ANALÓGICA</i> <i>S: BINARIA</i>	<b>REDES DISCRETAS</b> <i>E: BINARIA</i> <i>S: BINARIA</i>
<i>Backpropagation</i>	<i>Perceptron</i>	<i>Discrete Hopfield</i>
<i>Brain–State–in–a–Box</i>	<i>Adaline/Madaline</i>	<i>Learning Matrix</i>
<i>Continous Hopfield</i>	<i>Linear/Associative Penalty</i>	<i>Temporal Assoc. Memory</i>
<i>Linear Assoc. Memory</i>	<i>Adaptive Heuristic Critic</i>	<i>Bidirectional Assoc. Memory</i>
<i>Optimal Linear Assoc. Memory</i>		<i>Cognitron/Neocognitron</i>
<i>Drive Reinforcement</i>		<i>Adaptive Resonance Theory 1</i>
<i>Counterpropagation</i>		<i>Boltzmann Machine</i>
<i>Additive Grossberg</i>		<i>Cauchy Machine</i>
<i>Shunting Grossberg</i>		
<i>Adaptive BAM</i>		
<i>Learning Vector Quantizer</i>		
<i>Topology Preserving Map</i>		
<i>Adaptive Resonance Theory 2</i>		



---

## **ANEXO B. ANN E INTERNET**

A continuación se proponen una serie de direcciones de acceso por *Internet*, con la finalidad de completar el conocimiento alrededor de las ANNs:

### FAQs en torno a ANN

<ftp://ftp.sas.com/pub/neural/FAQ.html>

### Software de ANN

<http://www.neurosolutions.com/>

### Software de ANN

<http://www.statsoft.com/textbook/neural-networks/>

### MatWorks: Neural Network Toolbox

<http://www.mathworks.es/products/neural-network/>

### Laboratorio de computación y ciencias de la informática

<http://www.cis.hut.fi/>

### Journals de Inteligencia

<http://www.journals.elsevier.com/intelligence/>

<http://www.journals.elsevier.com/engineering-applications-of-artificial-intelligence/>

<http://www.journals.elsevier.com/neural-networks/>

<http://www.springer.com/computer/ai/journal/10489>

<http://ieeexplore.ieee.org/xpl/RecentIssue.jsp?punumber=72>

<http://www.inns.org/njournal.asp>

<http://www.sciencedirect.com/science/journal/08936080>

<http://ieeexplore.ieee.org/xpl/RecentIssue.jsp?punumber=5962385>

<http://www.bioinfo.in/contents.php?id=106&page=aim>



## ANEXO C. OUTLIERS DETECTADOS DEL CONJUNTO DE DATOS. DETALLES ESTADÍSTICO

Tabla C.1. Outliers detectados con PCA. Estas curvas de carga no se emplearán en los diferentes conjuntos de datos. Fuente Propia.

N° Patrón	Día	Mes	Año	Día semana
23	23	1	2008	3
156	4	6	2008	3
241	28	8	2008	4
248	4	9	2008	4
252	8	9	2008	1
259	15	9	2008	1
260	16	9	2008	2
280	6	10	2008	1
282	8	10	2008	3
294	20	10	2008	1
603	25	8	2009	2
665	26	10	2009	1
666	27	10	2009	2
897	15	6	2010	2
955	12	8	2010	4
1017	13	10	2010	3
1018	14	10	2010	4
1019	15	10	2010	5
1020	16	10	2010	6
1021	17	10	2010	0
1022	18	10	2010	1
1023	19	10	2010	2
1024	20	10	2010	3
1025	21	10	2010	4
1026	22	10	2010	5
1027	23	10	2010	6
1028	24	10	2010	0
1029	25	10	2010	1
1030	26	10	2010	2
1031	27	10	2010	3
1032	28	10	2010	4
1033	29	10	2010	5
1034	30	10	2010	6
1035	31	10	2010	0
1036	1	11	2010	1
1037	2	11	2010	2
1038	3	11	2010	3
1039	4	11	2010	4
1040	5	11	2010	5
1041	6	11	2010	6
1042	7	11	2010	0
1043	8	11	2010	1
1044	9	11	2010	2
1045	10	11	2010	3
1046	11	11	2010	4
1047	12	11	2010	5
1048	13	11	2010	6
1049	14	11	2010	0
1050	15	11	2010	1
1051	16	11	2010	2
1052	17	11	2010	3
1053	18	11	2010	4
1054	19	11	2010	5

Detalles estadísticos

La sintaxis de la función *corrcoef()* es

$$r = \text{corrcoef}(x)$$

$$r = \text{corrcoef}(x, y)$$

$$[r, p] = \text{corrcoef}(\dots)$$

$r = \text{corrcoef}(x)$  devuelve la matriz  $r$  de coeficientes de correlación calculados a partir de una matriz  $x$  cuyas filas son observaciones (valores) y cuyas columnas son variables. La matriz  $r = \text{corrcoef}(x)$  está relacionada con la matriz de covarianza  $c = \text{cov}(x)$  por

$$R(i, j) = \frac{c(i, j)}{\sqrt{c(i, i)c(j, j)}}$$

$r = \text{corrcoef}(x)$  es la covarianza normalizada sin retardo, esto es, el valor para retardo cero de  $x\text{cov}(x, 'coeff')$ ,  $r = \text{corrcoef}(x, y)$  es equivalente a  $r = \text{corrcoef}([x \ y])$  y  $[r, p] = \text{corrcoef}(\dots)$  devuelve además de la correlación una matriz  $p$  para comprobar la hipótesis de que no exista correlación. Cada valor de  $p$  representa la probabilidad de obtener una correlación igual de grande que la obtenida con los valores observados, empleando valores aleatorios.

La sintaxis de la matriz de covarianza es

$$r = \text{cov}(x)$$

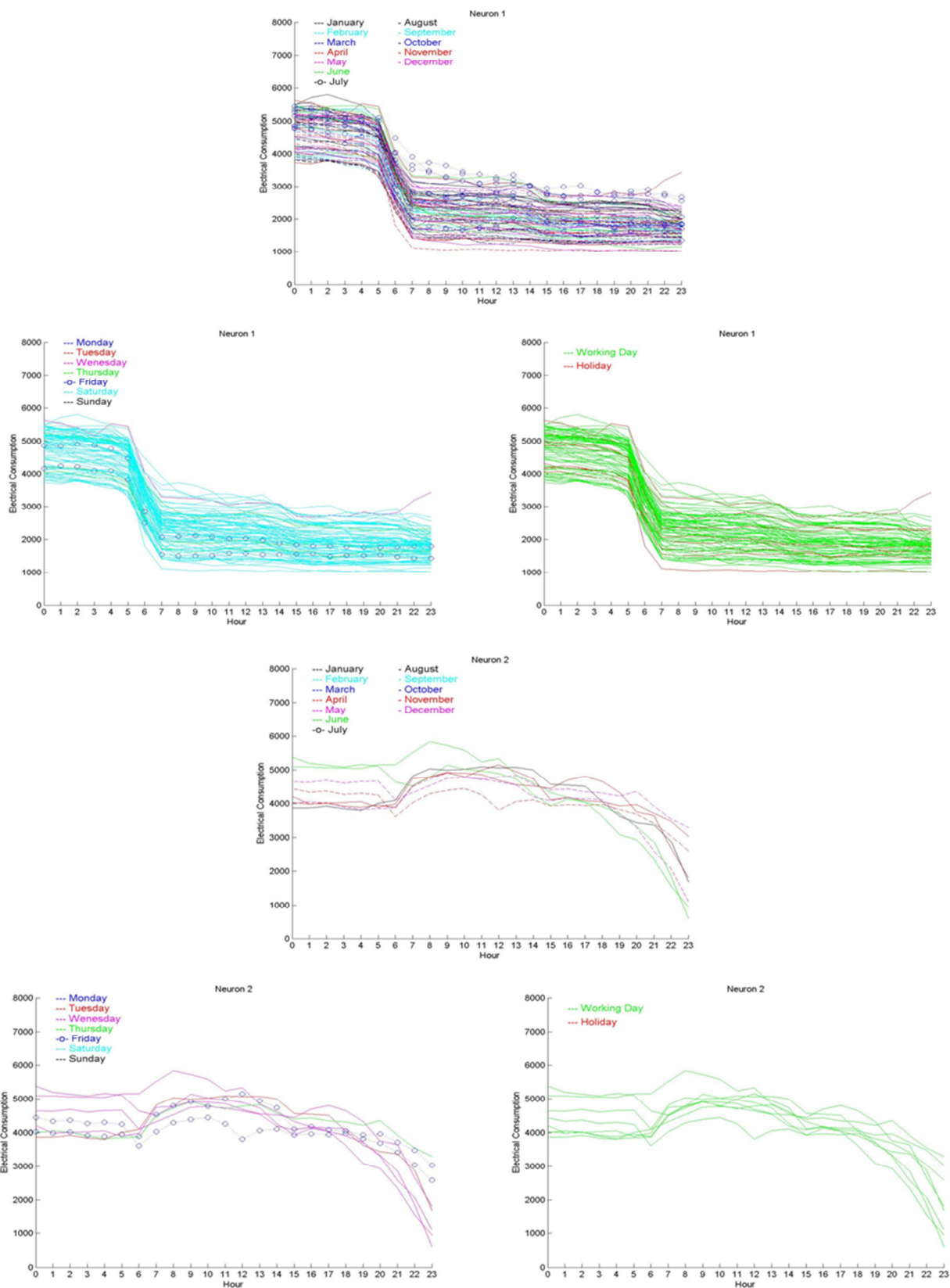
$\text{cov}(x)$  si  $x$  es un vector devuelve la varianza, si  $x$  es una matriz cuyas filas son observaciones (valores) y cuyas columnas son variables. La matriz  $r = \text{cov}(x)$  es la matriz de covarianza.  $(\text{diag}(\text{cov}(x)))$  es un vector de varianzas para cada columna de  $x$  y  $\text{sqr}(\text{diag}(\text{cov}(x)))$  es un vector de desviaciones estándar. Si empleamos  $\text{cov}(x, y)$  donde  $x$  e  $y$  tienen el mismo número de elementos, es equivalente a  $\text{cov}([x \ y])$ .  $\text{cov}(x)$  normaliza el resultado por  $N-1$  donde  $N$  representa el número de muestras. La función de covarianza es

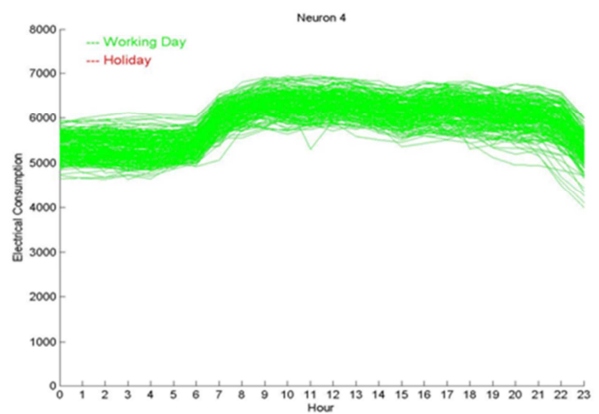
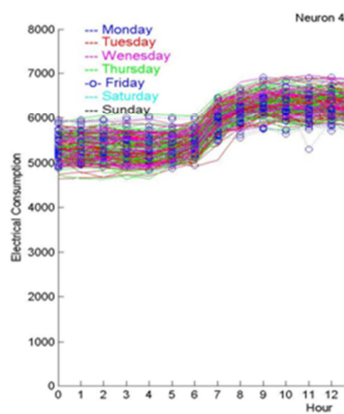
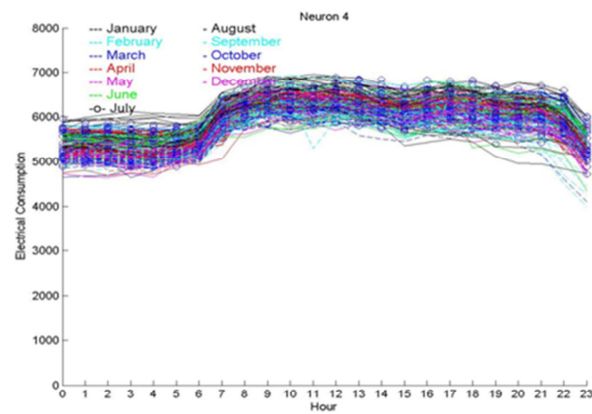
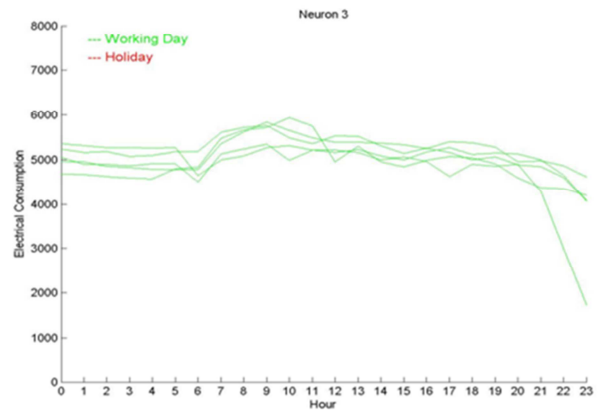
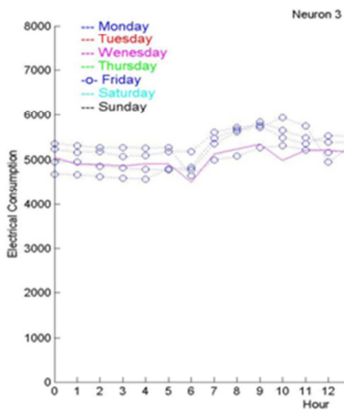
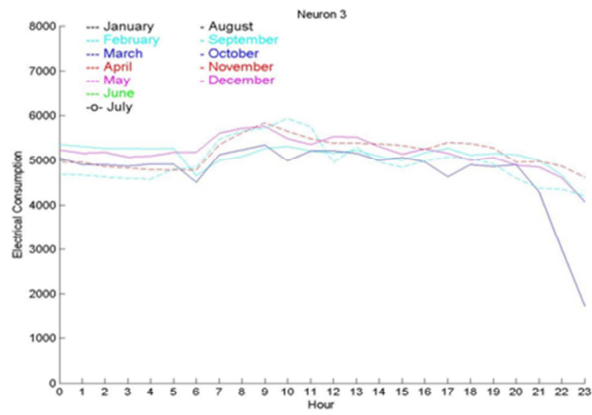
$$\text{cov}(x_1, x_2) = E[(x_1 - \mu_1) - (x_2 - \mu_2)]$$

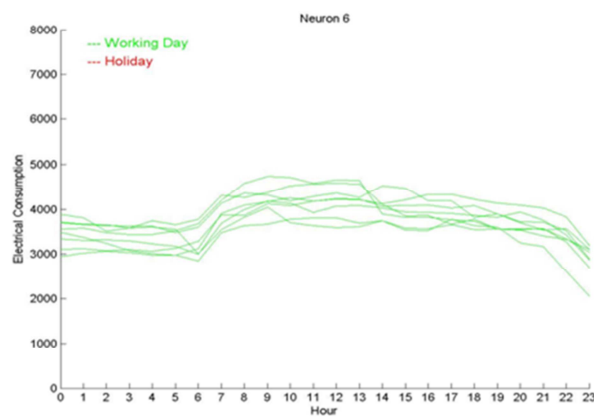
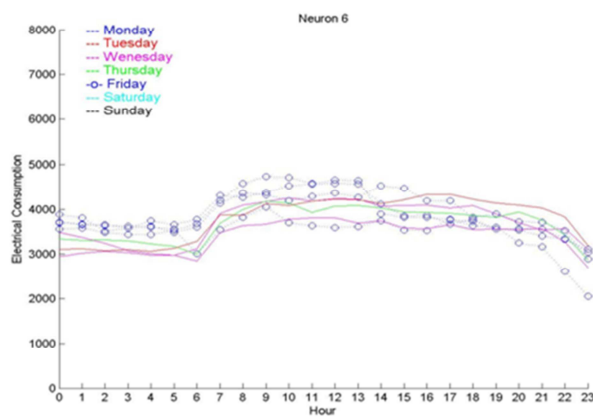
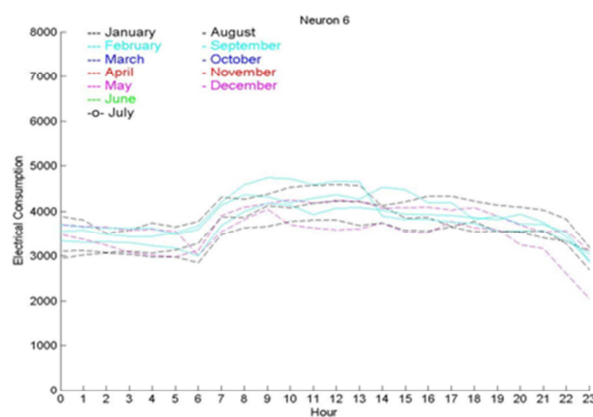
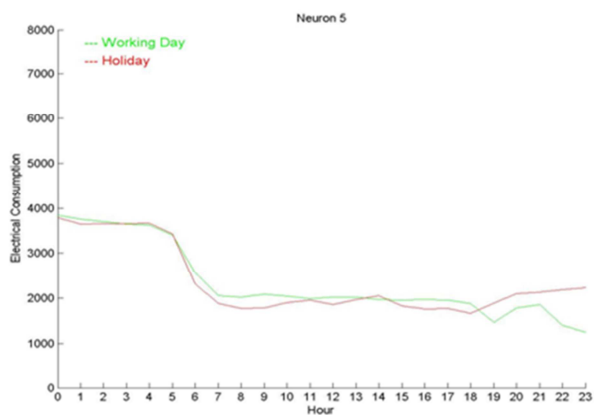
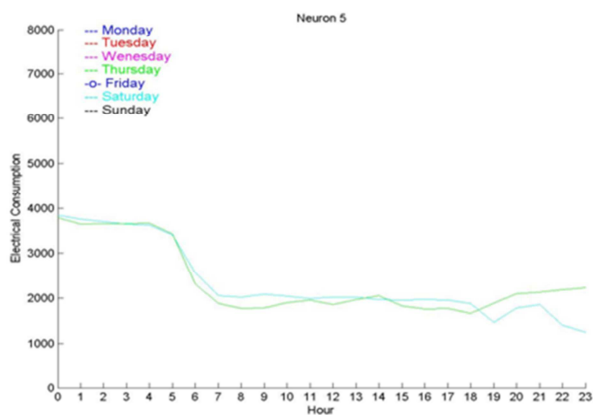
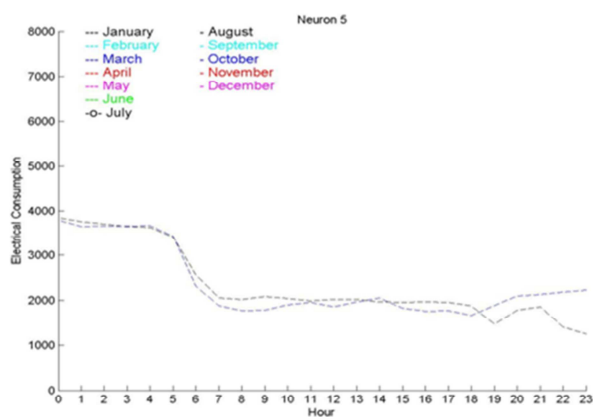
donde  $E$  es la esperanza matemática y  $\mu_1 = Ex_1$ .

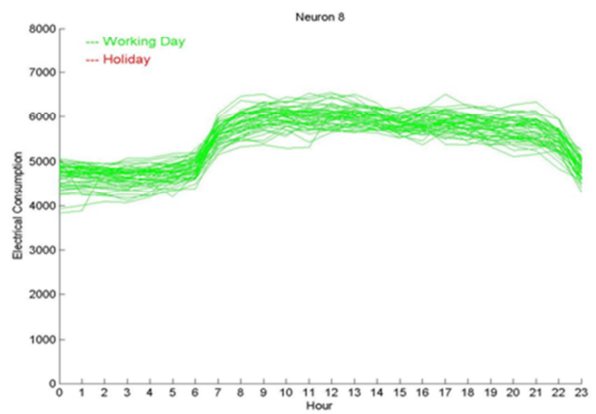
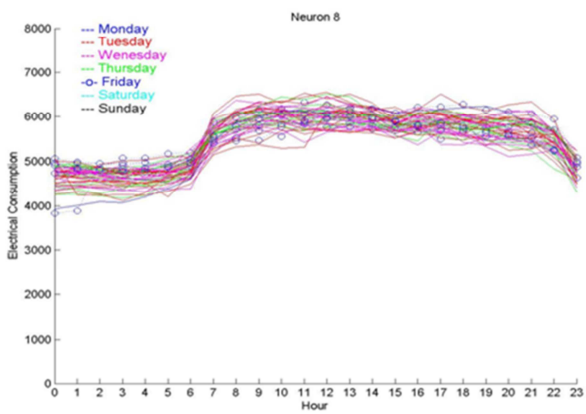
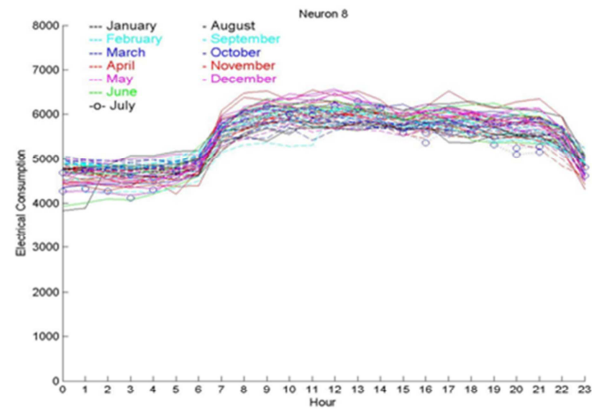
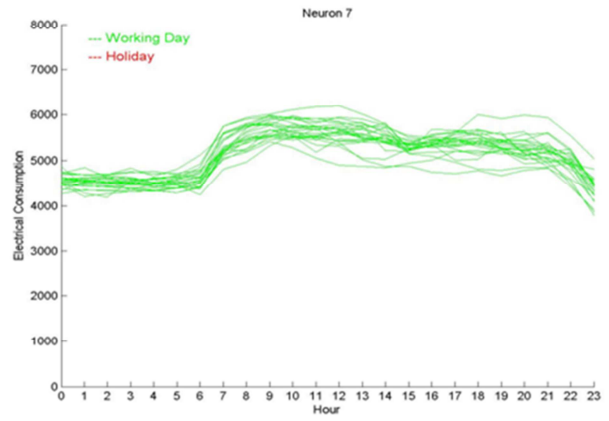
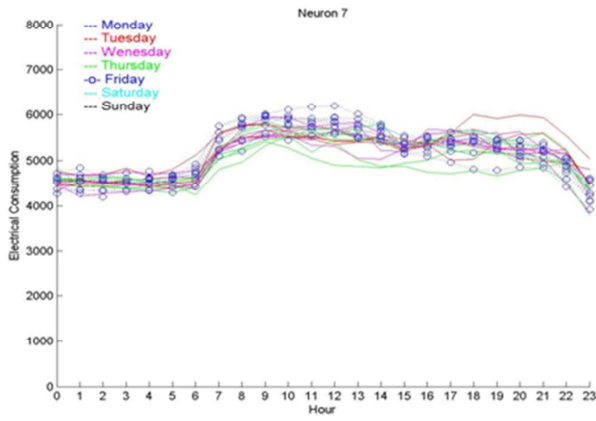
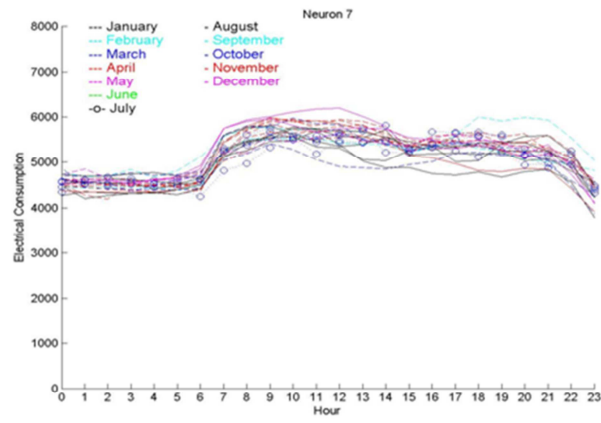
## ANEXO D. CLASIFICACIÓN Y CLUSTERIZADO DE PATRONES DE DEMANDA ELÉCTRICA EN PARQUES INDUSTRIALES. SECCIÓN 6.2.

A continuación se mostrarán los patrones de las curvas de carga que han sido asociadas a cada neurona una vez terminado el reconocimiento de patrones por medio de SOM.

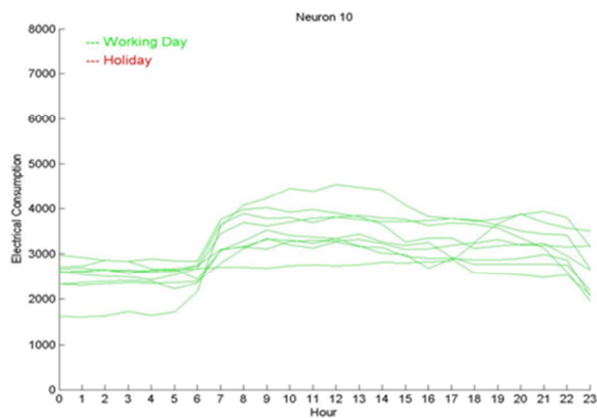
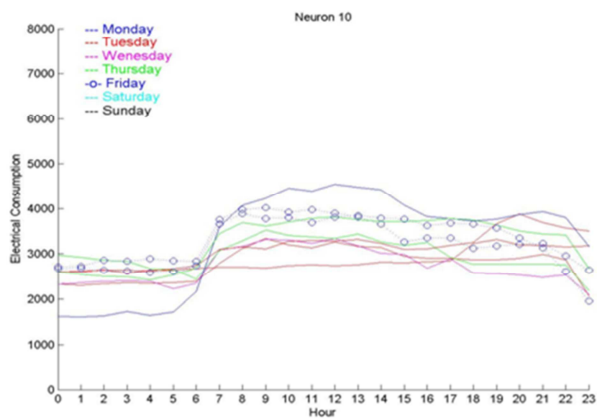
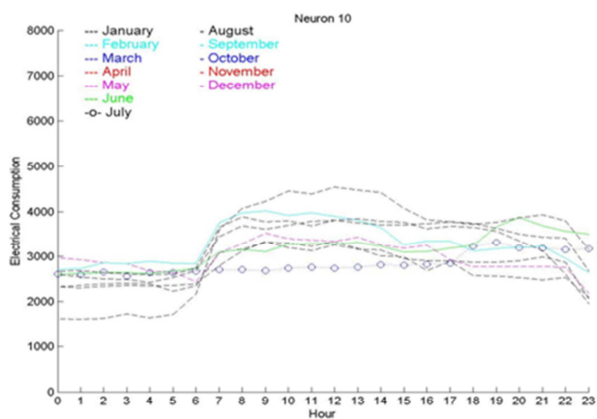
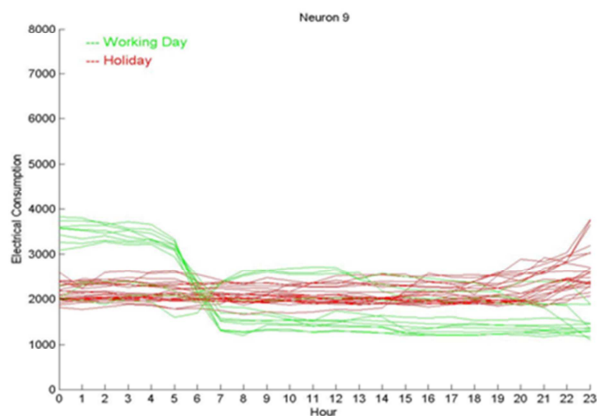
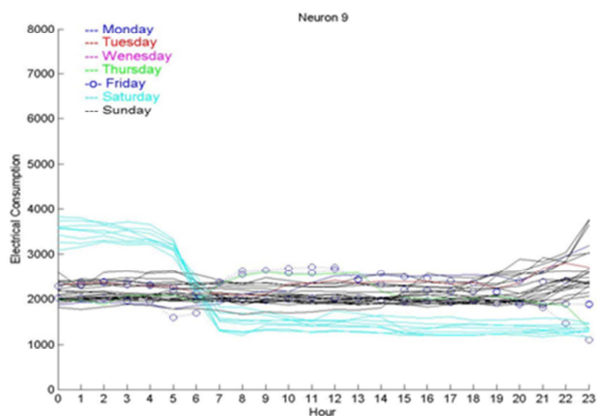
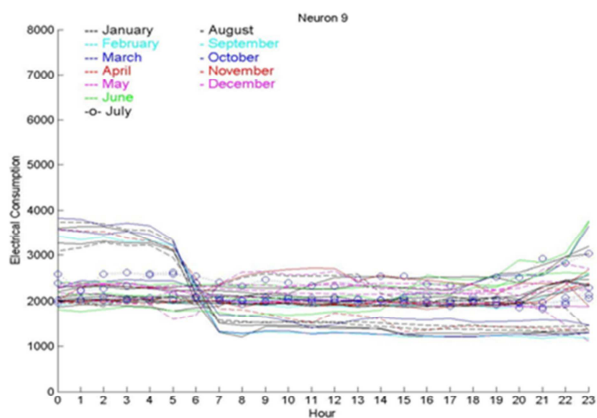


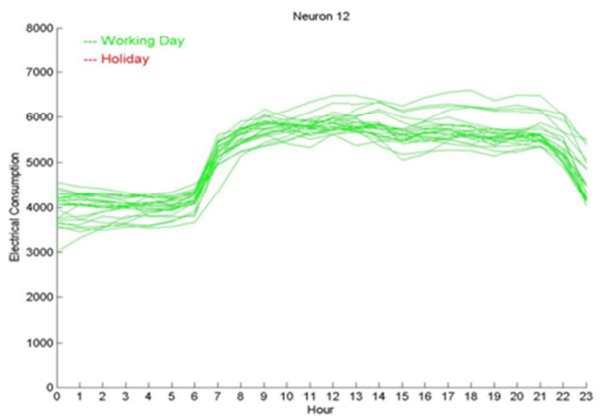
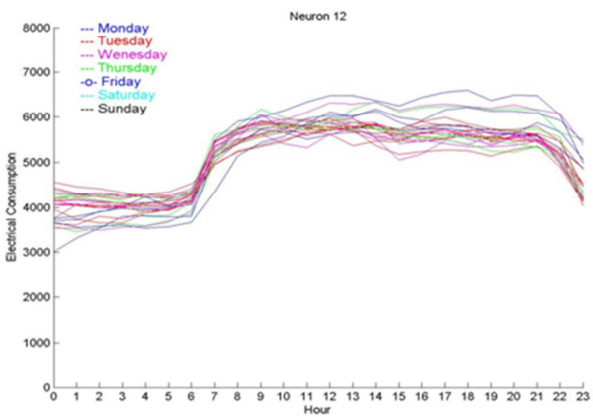
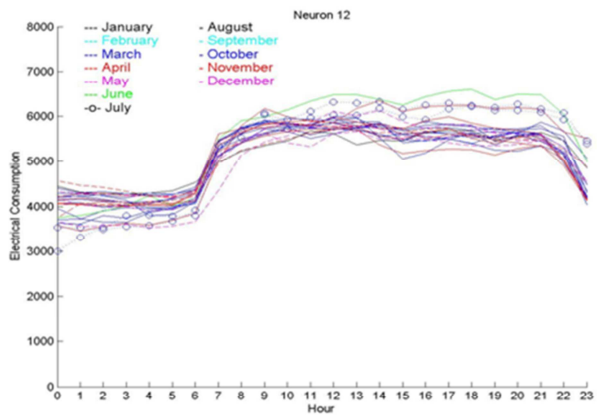
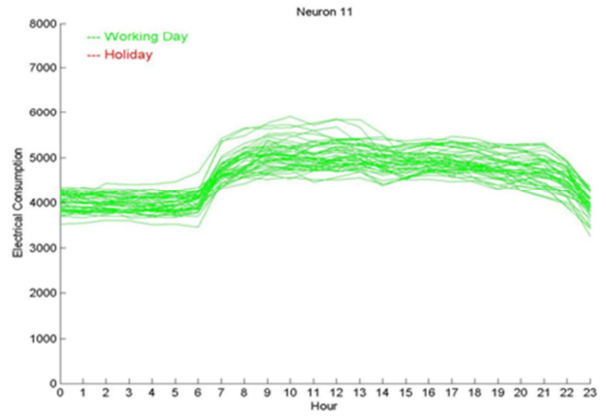
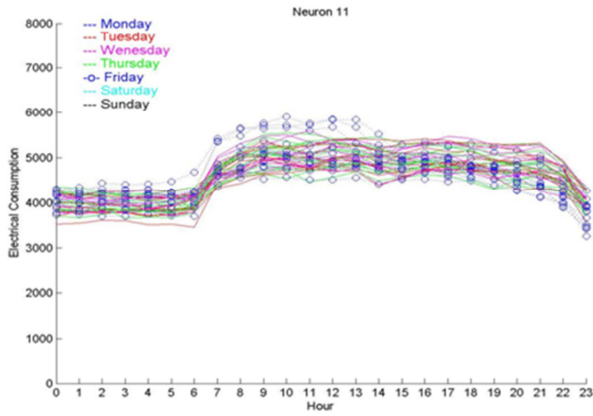
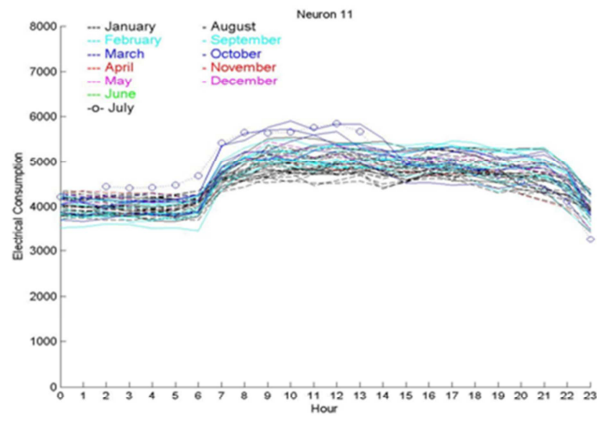


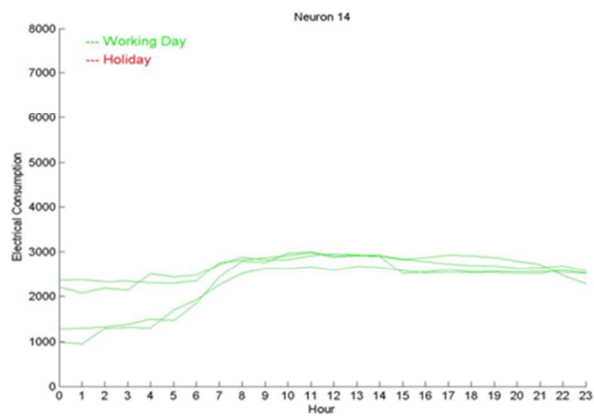
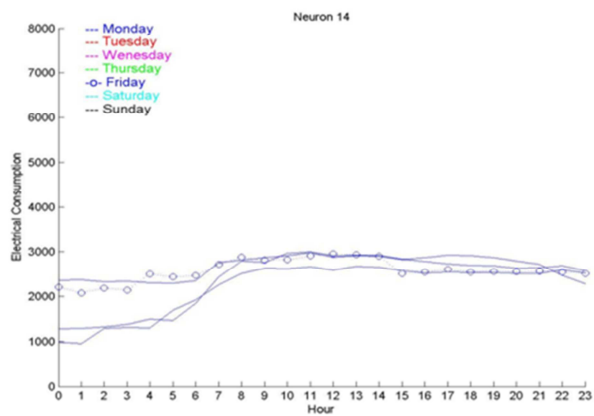
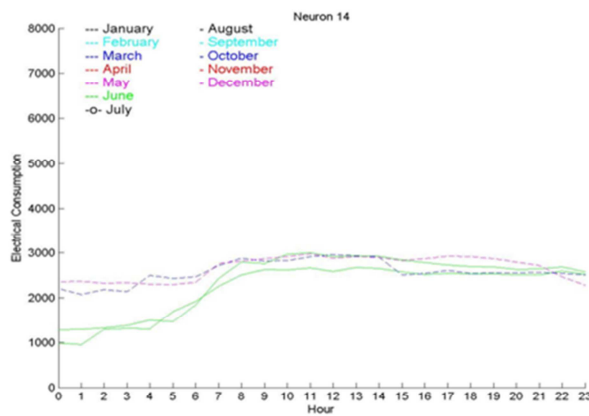
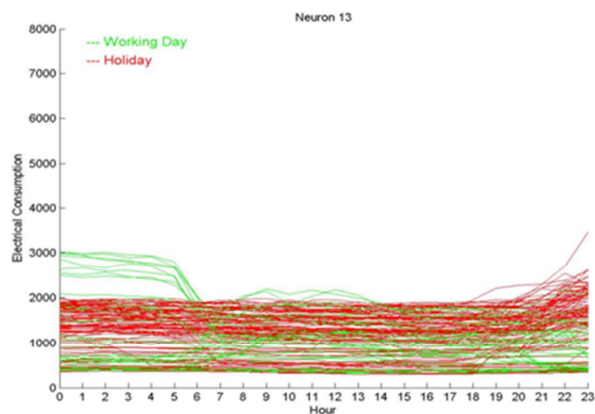
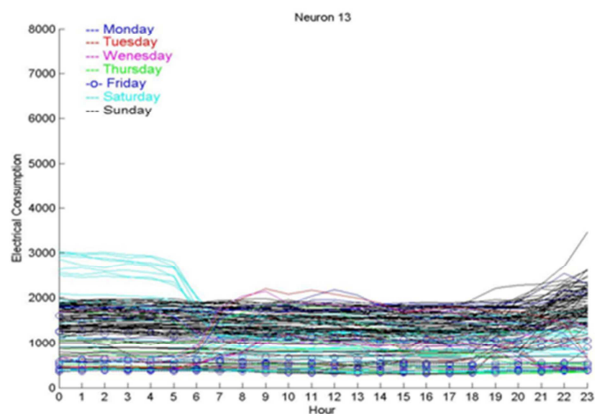
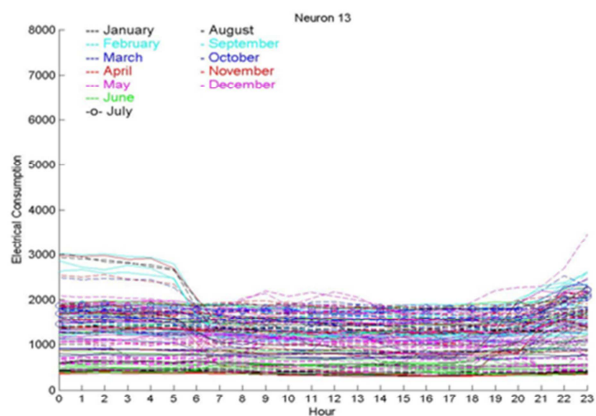


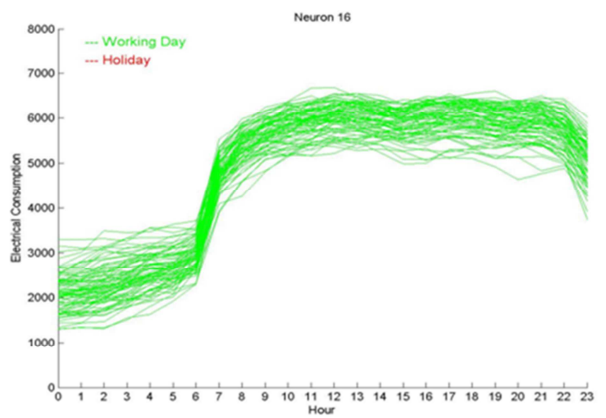
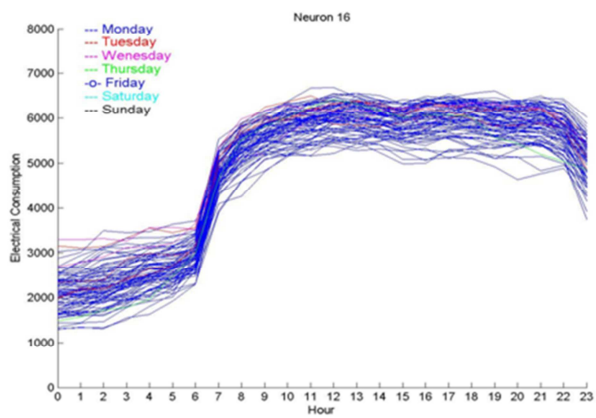
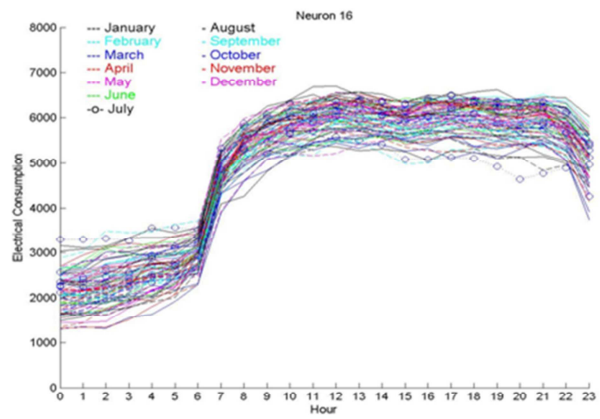
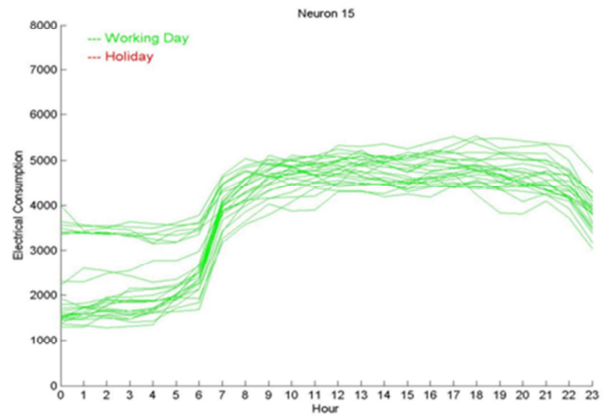
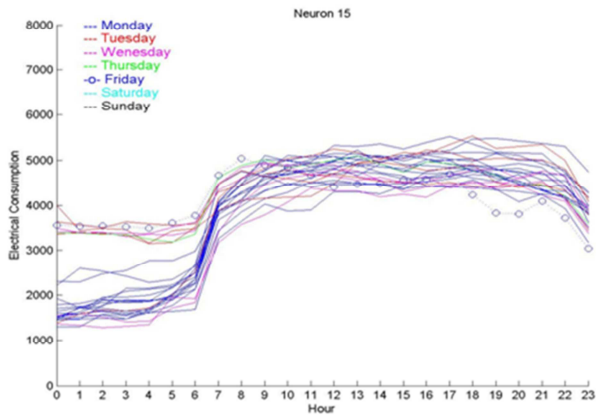
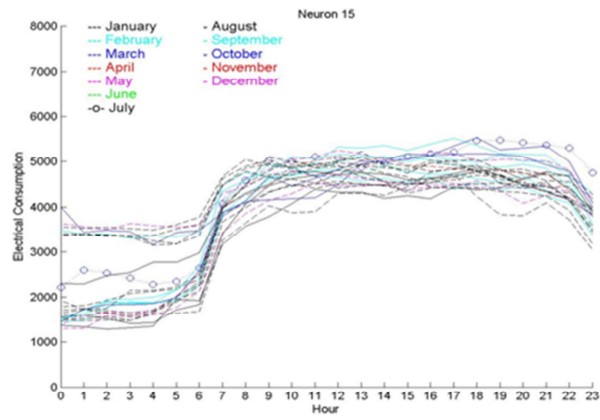












## ANEXO E. CLASIFICACIÓN Y CLUSTERIZADO DE PATRONES DE DEMANDA ELÉCTRICA EN MICROGRIDS. SECCIÓN 6.4.

A continuación se mostrarán los patrones de las curvas de carga que han sido asociadas a cada neurona una vez terminado el reconocimiento de patrones por medio de SOM.

