



Universidad de Valladolid

Facultad de Ciencias

Trabajo de Fin de Grado

27 de junio de 2022

Grado en Física

Búsqueda de las relaciones entre la estructura de los materiales sólidos nanoporosos y sus capacidades de almacenamiento de hidrógeno, mediante redes neuronales.

Autor: Álvaro Baños Izquierdo

Tutores: Iván Cabria Álvaro y M^a Begoña Torres Cabrera

Resumen

La economía del hidrógeno es uno de los campos de investigación más importantes en la actualidad. La búsqueda de nuevos combustibles para sustituir a los hidrocarburos se ha hecho necesaria y urgente a causa del oligopolio, contaminación y limitación en el uso de los hidrocarburos más comunes como el petróleo. Uno de los principales candidatos como nueva fuente de energía es el hidrógeno, siendo los vehículos de hidrógeno una de las mejores alternativas como sustitutos de los vehículos de gasolina. Sin embargo, uno de los principales problemas del vehículo de hidrógeno es el almacenamiento del hidrógeno a bordo del vehículo. Una de las posibles soluciones a este complicado problema es el almacenamiento de hidrógeno mediante la adsorción en materiales sólidos nanoporosos. Este Trabajo de Fin de Grado se centra en el estudio de las capacidades volumétricas de almacenamiento de hidrógeno de materiales sólidos nanoporosos formados por poros plano-paralelos y cilíndricos o nanotubos, en función de su densidad y a temperatura ambiente y 25 MPa de presión. En estudios previos se obtuvieron estas capacidades mediante simulaciones GCMC, *Grand Canonical Monte Carlo* de poros plano-paralelos y cilíndricos, conllevando un coste operacional y temporal bastante elevado, lo que ralentiza el avance en este campo de investigación. Hemos usado los resultados de simulaciones GCMC de los citados poros y una red neuronal *BackPropagation* como herramienta para obtener las capacidades volumétricas correspondientes, demostrando la mejora en eficiencia frente a las simulaciones GCMC. Esto permitirá avanzar en el complicado problema de la economía del hidrógeno. En este Trabajo de Fin de Grado hemos tratado de relacionar la Física (capacidades de almacenamiento) con la Inteligencia Artificial (redes neuronales) y la Informática (implementación del código de la red neuronal).

Abstract

The hydrogen economy is one of the most important fields of research today. The search for new fuels to replace the hydrocarbons is necessary and urgent due to the oligopoly, pollution and limited availability of the most common hydrocarbons such as oil. One of the main candidates as new energy source is hydrogen, being the hydrogen vehicles one of the best alternatives as substitutes of the gasoline cars. However, one of the main problems of the hydrogen vehicle is the onboard hydrogen storage. One of the possible solutions to this complicated problem is the hydrogen storage by adsorption on solid porous materials. This Final Degree Project is focused in the study of the volumetric hydrogen storage capacities of solid nanoporous materials composed by slit and cylindrical pores or nanotubes, as a function of the density and at room temperature and at a pressure of 25 MPa. These capacities were obtained in previous studies by means of GCMC, *Grand Canonical Monte Carlo*, simulations of slit and cylindrical pores, implying a rather high computational cost, which slows the advancement on this research field. We have used the results of the GCMC simulations of the mentioned pores and a *BackPropagation* neural network as a tool to get the corresponding volumetric capacities, showing the improvement of the efficiency vs the GCMC simulations. This will allow to progress in the complicated problem of the hydrogen economy. In this Final Degree Project we have tried to link the Physics (storage capacities) with the Artificial Intelligence (neural networks) and the Informatics (implementation of the neural network code).

Resumen	1
1. La economía del hidrógeno y las redes neuronales.	1
2. El almacenamiento de hidrógeno.	4
3. Los materiales sólidos nanoporosos.	7
3.0.1. Características de los poros plano-paralelos y nanotubos.	8
4. Las Redes neuronales.	11
4.1. El aprendizaje automático y la inteligencia artificial	11
4.2. Acerca de las Redes neuronales.	12
4.2.1. Funcionamiento de la Red Neuronal	15
4.3. Entrenamiento, Validación, Codificación	16
4.3.1. Entrenamiento	17
4.3.2. Validación	17
4.3.3. Producción	18
5. La red neuronal <i>Backpropagation</i>	19
5.1. Descenso del gradiente	20
5.2. Algoritmo <i>Backpropagation</i>	21
5.3. Breve explicación de su implementación en R	25
6. Parámetros de la red y bases de datos utilizadas.	27
6.1. Bases de datos utilizadas.	27
6.1.1. Simulaciones GCMC.	28
6.2. Procedimiento realizado.	28
6.2.1. Comprobación de la red.	28
6.2.2. Aplicación de la red a las bases de datos poros plano-paralelos y nanotubos.	30
7. Presentación y análisis de la aplicación de la red neuronal.	34
7.1. Conjuntos de Entrenamiento, Validación y Producción	34

7.2. Nanotubos	35
7.3. Poros plano-paralelos	37
7.4. Conclusiones	39

CAPÍTULO 1

LA ECONOMÍA DEL HIDRÓGENO Y LAS REDES NEURONALES.

El término economía del hidrógeno hace referencia a la nueva estructura del negocio energético y el sistema de relaciones técnico-económicas y sociales que se generarán como consecuencia del reemplazo gradual de los combustibles fósiles por el hidrógeno. No se trata exclusivamente de los aspectos que engloban el uso del hidrógeno como combustible, sino del proceso de transición hacia una economía descentralizada y autosuficiente (Solano [1]). El hidrógeno se presenta como uno de los principales sustitutos de combustibles fósiles como el petróleo, siendo una de las principales soluciones en la búsqueda de energías alternativas a los hidrocarburos.

La principal ventaja del uso del hidrógeno como combustible sería la creación de una nueva red energética en la que el usuario final sea también el productor, lo que se conoce como generación distribuida (Solano [1]). La celda de hidrógeno es la alternativa con mayor número de posibles aplicaciones comerciales. De esta manera se solucionarían los principales problemas que han llevado a la búsqueda de nuevas energías alternativas: Impacto medioambiental, monopolio energético, y la escasez de las principales fuentes de hidrocarburos usadas actualmente. Sin embargo, a pesar de ser el elemento más abundante del Universo, no existe en grandes cantidades en la naturaleza de una forma aprovechable, ya que se encuentra en su mayor parte combinado en forma de agua, por lo que no se dispone de hidrógeno libre.

El hidrógeno es un elemento que se caracteriza por tener un bajo punto de ebullición y poca densidad en forma de gas y líquido. Como consecuencia de su baja densidad (0.0899 g/L), el hidrógeno es un gas difícil de almacenar. En la actualidad se están desarrollando diferentes técnicas para su almacenamiento. Sin embargo, el inconveniente que estas presentan está relacionado con el peso, volumen o coste de dichos sistemas, lo cual limita el uso de hidrógeno como fuente de energía en determinadas aplicaciones, como por ejemplo, los teléfonos móviles. En el ámbito automovilístico, el objetivo es poder almacenar hasta 5 kg de hidrógeno, lo cual equivale a una autonomía de unos 500 Km en un vehículo con pila de combustible. Es necesario además que el hidrógeno se libere a temperaturas entre 80 y 150 °C , que el tiempo y la energía necesaria para el llenado sean pequeños y que el coste del tanque sea asequible para el consumidor.

Una posible forma de evitar los inconvenientes derivados del almacenamiento de hidrógeno como

gas comprimido es utilizar materiales que sean capaces de adsorber y desorber grandes cantidades de este gas. (El significado de la adsorción se explica en el siguiente capítulo). Para diferenciar que materiales son mejores para el almacenamiento de hidrógeno se estudia la capacidad volumétrica de los mismos. Esta capacidad se define como la masa de hidrógeno almacenada por unidad de volumen, y se mide en kg/L en unidades de S.I.

El proceso para obtener la capacidad volumétrica para distintos materiales no es trivial. Uno de los métodos usados son las simulaciones por el Método de Monte Carlo, el cual hace referencia a un conjunto de métodos estadísticos que se basan en la generación de números pseudoaleatorios, una sucesión de números que se repiten periódicamente con un periodo, T . Por tanto, no son realmente números aleatorios, debido a su periodicidad. Esta secuencia es generada a partir de un número entero llamado semilla, que etiqueta uno de los números de la sucesión. Sin embargo, este método tiene el inconveniente del alto coste computacional y tiempo necesario que conlleva para obtener los resultados requeridos.

Una de las principales técnicas que han surgido para mejorar resultados obtenidos por el método anterior han sido las técnicas que engloba el Aprendizaje Automático (*Machine Learning*, en inglés), dentro de las cuales destacan las *Redes Neuronales*. Este trabajo se centra en el estudio de la aplicación de las Redes Neuronales en el estudio del almacenamiento de hidrógeno, evaluando las consecuencias físicas y mejoras frente al método de Monte Carlo usado en anteriores trabajos.

Las redes neuronales surgen de la búsqueda del ser humano de nuevas vías para mejorar sus condiciones de vida. Estos esfuerzos le han llevado a construir máquinas calculadoras que ayuden a resolver de forma automática y rápida determinadas operaciones que resultarían tediosas si se realizaran a mano. Uno de los primeros impulsores en acometer esta empresa fue Charles Babbage, (R. Lackes et al. [18]), quien infructuosamente intentó construir una máquina capaz de resolver problemas matemáticos. No fue hasta la Segunda Guerra Mundial, cuando se empezaron a recoger los primeros frutos. En 1946 se construyó la primera computadora electrónica, ENIAC, *Electronic Numerical Integrator And Computer*, (R. Lackes et al. [18]). Desde entonces los desarrollos en este campo han tenido un auge espectacular.

Las redes neuronales emulan ciertas características propias de los humanos, como capacidad de memorizar y de asociar hechos. Si se examinan con atención aquellos problemas que no pueden expresarse a través de un algoritmo, se observará que todos ellos tienen una característica en común: la experiencia. Una red neuronal es un nuevo sistema para el tratamiento de la información, cuya unidad básica de procesamiento está inspirada en la célula fundamental del sistema nervioso humano: la neurona (Matich [2]).

Lo que ocurre en una neurona es lo siguiente: la neurona es estimulada o excitada a través de sus entradas (*inputs*), y una vez alcanzado un cierto umbral, la neurona se dispara o activa, pasando una señal hacia el axón. Estos resultados son el resultado de eventos electroquímicos. Este proceso tiene lugar en el cerebro, que consta de billones de neuronas interconectadas. Una de las principales características de los humanos, es que somos capaces de aprender; es decir, tenemos la capacidad de resolver problemas que no podían resolverse inicialmente, después de obtener más información sobre los mismos. Algo similar sucede en las redes neuronales, que consisten en unidades de procesamiento que intercambian datos o información, lo que las con-

vierte en herramientas verdaderamente útiles para la obtención de relaciones entre diferentes variables inicialmente desconocidas. Como es el caso en el que se centra este trabajo, el estudio de la capacidad volumétrica de los materiales sólidos nanoporosos a determinadas condiciones de presión y temperatura.

CAPÍTULO 2

EL ALMACENAMIENTO DE HIDRÓGENO.

El Departamento de Energía de los EE.UU. (U.S Department of Energy, DOE) estableció en el 2018 unos objetivos tecnológicos concretos para conseguir un vehículo de hidrógeno con la misma autonomía que un vehículo promedio de gasolina, unos 500-600 km, para los años 2020 y 2025 (DOE [19, 20]): Una capacidad gravimétrica de hidrógeno del 4.5 % en peso y una capacidad o densidad volumétrica de 0.030 kg/L para el año 2020, y una capacidad gravimétrica del 5.5 % en peso y una capacidad volumétrica de 0.040 kg/L para el años 2025. Se trata de capacidades de almacenamiento reversible.

La densidad de energía por unidad de masa del hidrógeno es tan elevada que cinco kilos de hidrógeno son suficientes para alcanzar la mencionada autonomía de 500-600 km, teniendo también en cuenta la eficiencia de las actuales celdas de hidrógeno.

Uno de los mayores problemas sin resolver en la economía del hidrógeno es el almacenamiento del mismo. Para poder usar el hidrógeno como fuente de energía es necesario poder almacenarlo de una forma abundante y eficaz. Sin embargo, debido a su baja densidad a temperatura ambiente y bajas presiones, 0.0899 g/L, es un gas difícil de almacenar. Existen varias formas de almacenar hidrógeno:

- **Almacenamiento de hidrógeno gaseoso:** Es el método más utilizado. El hidrógeno a temperatura ambiente y bajas presiones se encuentra en estado gaseoso, pero debido a su baja densidad energética por unidad de volumen ($12 \text{ MJ}/\text{m}^3$), el hidrógeno en estado gaseoso es normalmente almacenado a altas presiones, 700-1200 bares (P.A. Lessing [3]). El hidrógeno se almacena a altas presiones en depósitos de acero o de materiales ligeros que resisten altas presiones, los cuales son caros y consumen mucha energía en las aplicaciones a gran escala. El objetivo de la industria es construir un cilindro capaz de soportar presiones de 70 MPa con una masa de 110 kg. En la Tabla 2.1 se muestra el volumen que debería tener un tanque para almacenar 1 kg de hidrógeno a 20 °C en función de la presión (Züttel [4]).

La densidad de energía por unidad de volumen del gas hidrógeno es reducida o muy

reducida, a temperatura ambiente y presiones moderadas o bajas. Para almacenar los mencionados cinco kilogramos de hidrógeno en forma de gas, es necesario presurizar el gas hidrógeno a altas presiones, de modo que el gas esté contenido en un volumen asequible para su transporte a bordo del vehículo de hidrógeno. Esas altas presiones implican usar recipientes con paredes de 600 kg de acero o 135 kg de fibra de carbono, lo cual aumenta considerablemente el coste y el peso del vehículo.

Presión /MPa	Volumen /L
0.101325	11934
200	68.4
350	42.7
700	25.7

Tabla 2.1: Volumen de un tanque para almacenar 1 kg

- **Almacenamiento de hidrógeno en microesferas de vidrio:** Empleado principalmente en prácticas de automoción, el hidrógeno se introduce en los huecos de las microesferas aprovechando la permeabilidad de estas. El principal inconveniente es la baja densidad volumétrica y las altas presiones que se tienen que usar para su llenado. Como aspecto positivo, destaca su seguridad.
- **Almacenamiento de hidrógeno líquido:** Este se obtiene a partir de la licuación del mismo. La licuación es el proceso de pasar un gas a líquido modificando sus condiciones de presión y temperatura. Es necesario un gran aporte energético, y usar depósitos que puedan mantener las bajas temperaturas, al ser almacenado el hidrógeno en tanques criogénicos a $T=21,2$ K y presión ambiental. La densidad volumétrica del hidrógeno líquido es 70.8 kg/m^3 t el coste para su mantenimiento es elevado (Züttel [4]). El hidrógeno almacenado presenta densidades volumétricas similares a las obtenidas en el almacenamiento de hidrógeno gaseoso a 700 bares.
- **Almacenamiento de hidrógeno en sólidos:** Consiste en almacenar el hidrógeno por sorción (adsorción o absorción) en materiales sólidos. Es importante clarificar el significado de la adsorción frente al de absorción, en el contexto del almacenamiento de gases en materiales sólidos.

 - **Adsorción:** La adsorción es el fenómeno de sorción en el que las moléculas o átomos quedan adheridos a la *superficie* sólida del material adsorbente. Las moléculas o átomos puede quedar fijados en la superficie por fuerzas de van der Waals. Este fenómeno se conoce como *fisisorción*. También puede quedar adheridos por formación de un enlace químico con los átomos de la superficie, lo que se conoce como *quimisorción*.

- **Absorción:** La absorción es el fenómeno de sorción en el que los átomos o moléculas quedan incorporados o adheridos al *volumen* de un material. Los átomos o moléculas pasan a formar parte de la estructura interna del material. La absorción puede ser física, si los átomos o moléculas no pierden su naturaleza física y química, o puede ser química, si se forman nuevos enlaces con los átomos del volumen del material. Los hidruros metálicos sólidos almacenan hidrógeno mediante la absorción química.

Los principales mecanismos que permiten almacenar el hidrógeno en materiales sólidos son (López [17]):

- **Quimisorción:** Las moléculas de hidrógeno se disocian en las superficies del material, formando un enlace químico fuerte, con una energía de enlace de 100 a 500 kJ/mol . El proceso de quimisorción implica la rotura de los enlaces químicos de la molécula de hidrógeno y la creación de nuevos enlaces químicos.

- **Absorción química:** Las moléculas de hidrógeno se disocian en el volumen del material, formando un enlace químico fuerte. Al igual que la quimisorción, este proceso implica la rotura y la creación de nuevos enlaces químicos.

- **Fisisorción:** Este método de almacenamiento se da en sólidos porosos en los que el hidrógeno se adsorbe molecularmente sobre la superficie por medio de fuerzas de Van der Waals. En este caso, la molécula fisisorbida conserva su identidad física y química, ya que la energía no es lo suficientemente grande como para poder romper el enlace químico, aunque es probable que su geometría pueda quedar distorsionada. La adsorción se lleva a cabo en multicapas.

Ninguna de las formas de almacenamiento de hidrógeno (gas, líquido, en microesferas de vidrio y en sólidos) alcanza los objetivos tecnológicos establecidos por el DOE, a temperatura ambiente y presiones moderadas (10-25 MPa) (DOE, [19, 20]). El almacenamiento por fisisorción en materiales sólidos es reversible a temperatura ambiente y rápido, pero es bajo o moderado y no alcanza los objetivos del DOE. El almacenamiento por quimisorción es elevado, pero es reversible (desorción) a altas temperaturas, no a temperatura ambiente, y la velocidad de desorción es muy lenta.

Una ventaja del almacenamiento por fisisorción en materiales sólidos es la siguiente. La densidad volumétrica de hidrógeno almacenado en materiales sólidos por fisisorción es bastante mayor que la densidad volumétrica por compresión (el almacenamiento de hidrógeno gaseoso) a temperatura ambiente y *presiones moderadas*, hasta unos 250 bares. Esto permite reducir la presión de operación de los tanques de almacenamiento de hidrógeno, usar tanques menos pesados, reducir los costes de llenado y de mantenimiento (menores presiones) de los tanques y reducir los costes de construcción y operación de las hidrogeneras. Actualmente, la investigación del almacenamiento por fisisorción en materiales sólidos, es una importante línea de investigación en el campo de las energías sostenibles que protejan el medio ambiente.

El capítulo siguiente se dedica a los materiales sólidos nanoporosos, y especialmente a los materiales formados por poros plano-paralelos y cilíndricos (nanotubos) de carbono, en los cuales se centra este Trabajo de Fin de Grado.

CAPÍTULO 3

LOS MATERIALES SÓLIDOS NANOPOROSOS.

El almacenamiento del hidrógeno en los materiales sólidos nanoporosos se produce mediante fisisorción. Estos materiales son sólidos que poseen poros de muy pequeño tamaño (del orden del nanómetro), gracias a los cuales la superficie interna es muy elevada, del orden de un campo de fútbol por gramo de masa o incluso mayor. Una de las principales características en el estudio de estos materiales es la ya mencionada capacidad volumétrica (kg/L):

$$Capacidad\ volumetrica = \frac{Masa\ de\ hidrogeno\ almacenado}{Volumen\ total\ del\ sistema} \quad (3.1)$$

Se conocen diversos grupos de materiales sólidos nanoporosos. Los grupos de materiales más prometedores para el almacenamiento de hidrógeno son los carbones nanoporosos, los MOFs (*Metal-Organic Frameworks*), los polímeros de microporosidad intrínseca (PIMs, *Polymers of Intrinsic Microporosity*), las redes porosas aromáticas (PAFs, *Porous Aromatic Frameworks*), los COFs (*Covalent Organic Frameworks*) y las redes de polímeros porosos (PPNs, *Porous Polymer Networks*).

Dentro de este tipo de materiales, destacan los MOFs. Son polímeros de coordinación cristalinos, microporosos (poros de diámetro inferior a 20 \AA) y formados por estructuras tridimensionales de iones metálicos unidos por puentes orgánicos. Recientemente, la Universidad de Michigan sintetizó nuevos MOFs, SNU-70, UMCM-9 y PCN-610/NU-100, siendo el PCN-610/NU-100 el de mayor capacidad volumétrica (Villajos Collado [5]). En la Figura 3.3 se muestra la estructura del PCN-610/NU-100:

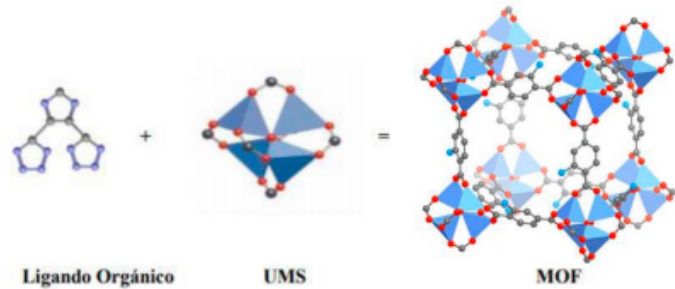


Figura 3.1: Estructura del PCN-610/NU-100 (Villajos Collado [5])

Se puede generar infinitud de MOFs con diferentes geometrías y conectividad, variando los centros metálicos y los grupos funcionales de los ligandos.

Los carbones nanoporosos y sus capacidades de almacenamiento de hidrógeno han sido extensamente estudiados durante las últimas tres décadas. Dentro de este grupo de materiales también existe una gran variedad. Estos materiales están formados por poros de diferentes formas y diámetros. Según los experimentos, muchas zonas de estos materiales están formadas por poros plano-paralelos (Park et al. [6]). Por ese motivo y por su simplicidad, los poros plano-paralelos son los modelos geométricos más usados para estudiar los carbones nanoporosos.

En este Trabajo de Fin de Grado nos hemos centrado en materiales basados en carbono y formados por poros plano-paralelos o poros rendija (*slit pores*, en inglés) y por poros cilíndricos o nanotubos.

3.0.1. Características de los poros plano-paralelos y nanotubos.

Vamos a explicar las características de estos materiales, así como una pequeña descripción de las simulaciones de las celdas de los poros a partir de los cuales se obtuvieron las bases de datos de entrada de la red neuronal.

Las dimensiones de las celdas de simulación de los sistemas estudiados se eligieron para obtener un volumen que no sea demasiado grande, evitando una carga computacional excesiva, y que pudiese contener suficientes moléculas de hidrógeno en su interior, para el rango de presión entre 0.1 y 25MPa y a la temperatura ambiente. A continuación describimos brevemente los poros plano-paralelos y los nanotubos:

- **Poros plano-paralelos:** Son poros que tienen forma de rendija plano-paralela. Se ilustra su geometría en la Figura 3.1.

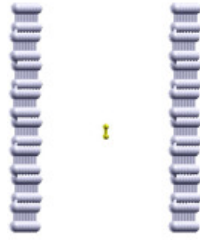


Figura 3.2: Geometría del poro plano-paralelo [15])

La celda de simulación de un poro plano-paralelo consiste en dos láminas planas de grafeno, paralelas entre sí y separadas una distancia w . Las láminas o capas de grafeno se encuentran en el plano xy . Se aplican condiciones de contorno periódicas a la celda de simulación de los poros plano-paralelos en las direcciones x e y . Cada capa o lámina de grafeno tenía longitudes de 88.54 y 89.46 Å a lo largo de los ejes x e y , respectivamente. Estas longitudes de las capas de grafeno son consistentes con las condiciones periódicas de los límites del grafeno. Los poros plano-paralelos simulados tenían una anchura w entre 4 y 15 Å. El volumen de la celda de simulación de un poro plano-paralelo es $S \cdot w$, donde S es la superficie de una capa de grafeno, y w es la anchura del poro. Los volúmenes de los poros plano-paralelos simulados se situaban entre 32000 y 120000 Å³, más una celda de simulación de 199000 Å³.

- **Nanotubos:** Se trata de poros que tienen forma de tubo, como se ilustra a continuación en la Figura 3.2:



Figura 3.3: Geometría del nanotubo [15])

Los nanotubos de carbono están formados por redes hexagonales de carbono curvadas y cerradas, formando tubos cilíndricos de carbono cuyo diámetro es del orden del nanómetro. Son sistemas ligeros, huecos y porosos que tienen alta resistencia mecánica. Para obtener sus capacidades, se simularon nanotubos con un diámetro entre 4 y 15 Å. En el caso de los nanotubos, las condiciones de contorno periódicas se aplicaron en la dirección del eje principal del nanotubo. La longitud L del del nanotubo de la celda de simulación era coherente con las condiciones periódicas. Las celdas de simulación de los nanotubos tenían una

longitud L tal que el volumen del nanotubo era aproximadamente 88000 \AA^3 . El volumen del nanotubo de la celda de simulación es $\pi r^2 L$, donde r es el radio del nanotubo y L su longitud. Los volúmenes de los nanotubos con un diámetro entre 4 y 15 \AA están entre 77000 y 88000 \AA^3 y sus longitudes L están entre 6770 y 470 \AA .

La densidad volumétrica de hidrógeno de los materiales sólidos porosos que se conocen en la actualidad, incluidos los MOFs y los diferentes tipos de carbones nanoporosos, es menor o igual que 0.01-0.02 kg/L a temperatura ambiente y presiones moderadas (10-25 MPa). Esa densidad es menor que los objetivos del DOE para los años 2020 y 2025, 0.030 y 0.040 kg/L, respectivamente (DOE, [19, 20]).

En este capítulo se describen las redes neuronales dentro del ámbito de la Inteligencia artificial y del Aprendizaje Automático, ya que serán la herramienta que usaremos para el estudio de la capacidad volumétrica de determinados materiales sólidos nanoporosos (poros plano-paralelos y nanotubos).

4.1. El aprendizaje automático y la inteligencia artificial

El término inteligencia artificial, (IA), se refiere a cualquier inteligencia similar a la humana exhibida por una computadora, robot u otra máquina. En el uso popular, la IA se refiere a la capacidad de un ordenador o máquina para imitar la mente humana.

Por lo tanto se entiende en general por IA, a cualquier comportamiento similar al humano mostrado por una máquina o sistema. En su forma más básica, los ordenadores se programan para imitar el comportamiento humano a partir de numerosos datos de ejemplos anteriores de comportamientos similares. Esto puede abarcar desde el reconocimiento de las diferencias entre un gato y un pájaro hasta la realización de actividades complejas en una planta de fabricación.

El Aprendizaje Automático es una aplicación de la inteligencia artificial que proporciona a los sistemas la capacidad de aprender y mejorar automáticamente de la experiencia sin requerir programación. El aprendizaje automático se enfoca en el desarrollo de programas informáticos que pueden acceder a los datos y los utiliza para aprender por sí mismos. El proceso de aprendizaje comienza con observaciones y experiencia directa o instrucción, con el fin de buscar patrones en los datos y tomar mejores decisiones en el futuro. Es en este ámbito surgen las *Redes Neuronales*.

4.2. Acerca de las Redes neuronales.

Las redes neuronales surgen de la necesidad del ser humano de automatizar determinados procesos matemáticos para poder resolverlos mediante la experiencia. Las redes neuronales son un intento de emular artificialmente un comportamiento humano. En definitiva, una red neuronal es un nuevo sistema para el tratamiento de la información cuya unidad básica de procesamiento está inspirada en la célula fundamental del sistema nervioso humano: la neurona (Matich [2]).

Una neurona es una célula viva, conteniendo todos los elementos que componen una célula biológica. Sin embargo, contiene determinados elementos característicos que la diferencian del resto de células existentes. Consta de un cuerpo celular más o menos esférico, del que sale una rama principal llamada axón, y de varias ramas de menor longitud, llamadas dendritas. En la Figura 4.1 se ilustra la fisiología de una neurona.

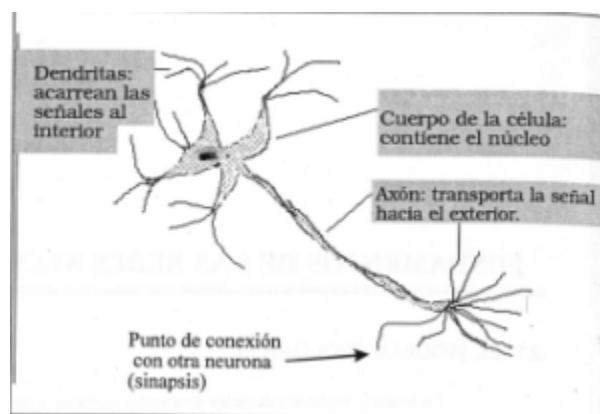


Figura 4.1: Biología de la neurona, (Matich [2])

La característica principal de las neuronas es su capacidad de comunicarse, las dendritas y el cuerpo celular reciben señales de entrada, que son procesadas por el cuerpo celular. El axón transporta estas señales a los terminales axónicos, los cuales distribuyen la información a un nuevo conjunto de neuronas. Esta cualidad dota a los seres humanos de nuestra capacidad más relevante: la capacidad de aprender. El intento de emular esta capacidad lleva a la creación de la *neurona artificial* y de la *red neuronal artificial*:

Neurona Artificial: Las neuronas artificiales son modelos que tratan de simular el comportamiento de las neuronas biológicas. Cada neurona se representa como una unidad de proceso que forma parte de una entidad mayor, la **red neuronal**. La neurona artificial es una unidad que cuenta con una serie de entradas x_i , que equivalen a las dendritas, en la neurona biológica, de donde reciben la estimulación, ponderadas por unos pesos w_i , que representan como los impulsos entrantes son evaluados y se combinan con la función de entrada, (combinación de las diferentes entradas), que dará el nivel de potencial de la neurona.

La salida de la función de entrada es evaluada en la función de activación que da lugar a la salida de la unidad de proceso.

En la figura 4.2, se muestra el funcionamiento de las neuronas artificiales en un esquema representativo (Matich [7]).

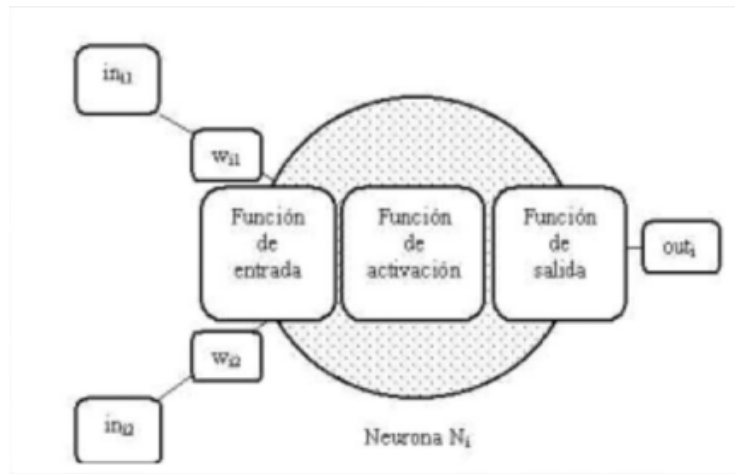


Figura 4.2: Esquema de una neurona (Matich [7]).

Mediante la combinación de múltiples neuronas podemos desarrollar una *red neuronal artificial*, las cuales están explicadas en mayor detalle en la sección 4.2.1. Las redes neuronales son modelos matemáticos multiparamétricos no-lineales, capaces de inducir una correspondencia entre conjuntos de patrones de información. El objetivo de las redes neuronales es emular el comportamiento de aprendizaje humano. Las principales ventajas que pueden ofrecer las redes neuronales son las siguientes (Matich [2]):

- **Aprendizaje adaptativo:** consiste en la capacidad de llevar a cabo tareas mediante un entrenamiento con ejemplos ilustrativos, sin necesidad de elaborar modelos a priori ni especificar funciones de probabilidad.
Las redes neuronales son sistemas adaptativos debido a la capacidad de autoajuste de los elementos procesales, las neuronas. El diseñador solo se tiene que preocupar de encontrar la arquitectura adecuada y de establecer un buen algoritmo de aprendizaje que le proporcione a la red la capacidad de discriminar.
- **Auto-organización:** Mientras que el aprendizaje es la modificación de cada elemento procesal, la auto-organización consiste en la modificación de la red neuronal completa para llevar a cabo un objetivo en específico. Esto dota a las redes neuronales de la facultad de responder apropiadamente cuando se les presenta datos o situaciones a las que no habían sido expuestas anteriormente, pudiendo estar estas incompletas. Por ejemplo, la red *Back-propagation* crea su propia característica, mediante la cual puede reconocer ciertos patrones. Esta red es la utilizada para el estudio de la capacidad volumétrica de los materiales nanoporosos, poros plano-paralelos y nanotubos, en este trabajo. Por ello, se profundizará más en ella en la próxima sección.
- **Tolerancia a fallos:** Las redes neuronales pueden aprender a reconocer patrones con ruido,

lo cual crea una tolerancia a fallos respecto a los datos, también, aun fallando ciertas neuronas de la red, esta puede seguir realizando su función, aunque con un cierto grado de degradación. Esto es debido a tener distribuida toda su información entre todas las neuronas que componen la red.

- Operación en tiempo real: al trabajar en paralelo en muchas neuronas, disminuye el tiempo operacional.
- Fácil inserción dentro de la tecnología existente: con las herramientas computacionales existentes, las redes pueden ser implementadas, validadas y verificadas de un manera rápida y eficaz.

Vamos a citar también, algunos los diferentes tipos de redes neuronales que existen actualmente y que se mencionan a lo largo del trabajo:

- Red Perceptrón Simple: es una red que consta de una capa de salida de n neuronas y otra de salida con m neuronas. Utiliza señales binarias, tanto de entrada como de salida de las neuronas y su función de activación es de tipo ReLU (Rectified Linear Unit).
- Red de Hopfield: red de tipo recurrente, es decir, en la que realimentación entre las neuronas.
- Red Perceptrón Multicapa: el perceptrón multicapa (MLP, *Multiple Layer Perceptron*), es una extensión y generalización del perceptrón simple en el que se añaden una o más capas ocultas y se permiten usar entradas continuas y diferentes funciones de activación como la sigmoidea.
- Redes convolucionales: una variación de un perceptrón multicapa, sin embargo, debido a que su aplicación es realizada en matrices bidimensionales, son muy efectivas para tareas de visión artificial, como en la clasificación y segmentación de imágenes.
- Redes Recurrentes: clase de redes para analizar datos de series temporales permitiendo tratar la dimensión de tiempo.

Solo se mencionan algunas de entre los muchos tipos de redes que existen, dando una breve explicación. No se entra en profundidad ni en las características de cada tipo ni en todos los tipos posibles ya que no es el objetivo del trabajo y se sale de las dimensiones del mismo.

También las podemos clasificar según su arquitectura y aprendizaje.

Según la arquitectura se clasifican en monocapas y multicapas, las monocapas son aquellas redes que solo cuentan con una capa, la más representativa es la red de Hopfield. Las redes multicapas están formadas por varias capas de neuronas, dentro de estas las más importantes son aquellas en las cuales las capas aparte del orden normal algunas están también unidas desde la salida hasta la entrada en el orden inverso en que viajan las señales de información. Las conexiones de este tipo se llaman conexiones retroalimentadas. La más importante se conoce como red neuronal Backpropagation, la cual está desarrollada en el capítulo 5. La clasificación según el aprendizaje se encuentra en la sección 4.3.1, debido a su influencia en la etapa de entrenamiento de las redes.

4.2.1. Funcionamiento de la Red Neuronal

Una vez introducidos los conceptos de neurona y red neuronal artificial, así como su clasificación y diferentes tipos de redes que existe, en esta sección explicaremos el funcionamiento de las redes comenzando con los elementos que conforman el esqueleto de la red, y las diferentes funciones de activación que dan forma a cada neurona.

Elementos de la red

Una red neuronal es un modelo simplificado que emula el modo en el que el cerebro humano procesa la información, y la estructura básica que posee se esquematiza en la Figura 4.3:

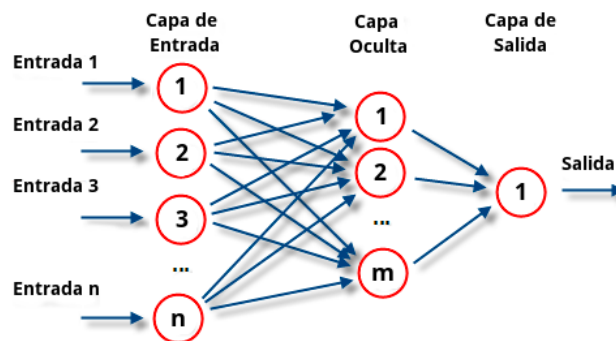


Figura 4.3: Estructura de una red neuronal (R.Rojas [8]).

Observamos que una red neuronal es una composición de unidades de procesamiento individuales, a las cuales denominamos exactamente igual que en el cerebro humano: neuronas. En la figura anterior cada círculo representa una única neurona. El conjunto de las neuronas de la primera columna sería la capa inicial, la última neurona compone la capa final, y todas las columnas de neuronas que hubiesen entre ellas compondrían las capas ocultas. En el caso de la Figura 4.2. solo habría una capa oculta.

Para el funcionamiento del cerebro humano es necesario que las neuronas reciban un estímulo inicial. En la red neuronal, este estímulo está compuesto por los datos o variables iniciales; además, como nuestro objetivo es obtener una serie de resultados en base a la experiencia, necesitamos otorgar a la red de un parámetro que posteriormente se pueda modificar para acercarse al valor deseado. Por ello, a cada neurona de la red le asignamos un peso w_i . La primera capa serán los mismos datos iniciales junto con el peso de cada neurona, sin realizar ninguna operación en ellos. A continuación estos datos son enviados a cada neurona de la capa siguiente. Para combinar estas simples entradas dentro de la entrada global, se utiliza la denominada *función de entrada*.

Supongamos que tenemos n variables iniciales (datos de partida), y ya se han asignado todos los pesos a cada neurona. Cada dato compondrá una neurona de la capa inicial, y por lo tanto, tendrá asignado un peso correspondiente. El *output* de cada neurona de esta primera capa es el dato d_i junto al peso asignado a la neurona n_i . Las salidas (*outputs*) de todas las neuronas de

la primera capa son enviadas a todas las neuronas de la siguiente capa. Aquí entra en juego la función de entrada (*input*):

$$f_{input} = (d_i \cdot w_i) * \dots * (d_n \cdot w_n),$$

donde el símbolo * representa el operador apropiado que hay designado (sumatorio, multiplicación, máximo, etc.). Los valores de entrada se están multiplicando por los pesos de la neurona correspondiente de la anterior capa. De esta manera, mediante el ajuste de estos pesos se puede variar la medida de la influencia que tienen los valores de entrada, de modo que un gran valor de entrada tenga una influencia pequeña escogiendo un peso suficientemente pequeño (R. Rojas [8]).

En el interior de la neurona, el *output* obtenido de la función de entrada, será el valor *input* de una función más, la denominada *función de activación*. Esta función se encarga de calcular el estado de activación de una neurona, transformando la entrada global en un valor de activación, que suele estar comprendido o bien entre 0 y 1, o bien entre -1 y 1, dado que una neurona puede estar totalmente inactiva, (0, o bien -1), o completamente activa (1). Estas funciones pueden ser de muchos tipos. Las más utilizadas son las siguientes:

- **Función lineal:** $f(x) = \begin{cases} -1 & \text{si } x \leq -1/a, a > 0 \\ a * x & \text{si } -1/a < x < 1/a, a > 0. \\ 1 & \text{si } x \geq 1/a, a > 0. \end{cases}$

- **Función sigmoidea:** $f(x) = \frac{1}{1+e^{-gx}}$ con $x = input - umbral_i$

- **Función tangente hiperbólica:** $f(x) = \frac{e^{gx} - e^{-gx}}{e^{gx} + e^{-gx}}$ con $x = in - u_i$

Finalmente, al resultado obtenido se le puede aplicar una función más (no es obligatorio) denominada *función de salida*, la cual determina el *output* que se transfiere a las neuronas vinculadas, es decir, a las neuronas de la siguiente capa. Esto puede ser necesario porque en determinados casos puede que no cualquier valor sea admitido como una entrada para una neurona, suelen estar comprendidos entre [0,1] o ser binarios, o bien 0,1 o bien -1,1. Se suele aplicar la función identidad, que sería equivalente a no aplicar ninguna función de salida, o la función binaria:

- Binaria: $f(x) = \begin{cases} 1 & \text{si } f_{act} \leq umbral_i \\ 0 & \text{si } f_{act} \geq umbral_i \end{cases}$

4.3. Entrenamiento, Validación, Codificación

Los pesos iniciales son generados aleatoriamente. Por tanto, dado un vector de datos iniciales, conseguimos un conjunto de salida, que obviamente, va a ser completamente disparato, sin relación alguna con los resultados que buscamos. La cualidad que intentan emular las redes neuronales es la del aprendizaje, como se describe a continuación.

4.3.1. Entrenamiento

El aprendizaje o entrenamiento es el proceso por el cual una red neuronal modifica sus pesos en respuesta a una información de entrada (Matich [7]). La idea conceptual es la siguiente: para que la red nos resuelva la tarea correctamente y nos saque los *outputs* buscados, es necesario entrenarla. Un conjunto de datos de entrenamiento son necesarios para los cuales existan los *outputs* reales. La red compara su salida con estas soluciones reales, y en base a eso modifica los pesos de las neuronas y vuelve a ejecutar todo el proceso, hasta conseguir un resultado suficientemente bueno. Esa comparación mencionada, es lo que se denomina *función de coste*, que simplemente es una función que mide el error de la salida obtenida frente a la solución real. Durante este proceso de aprendizaje, los pesos de las conexiones de la red sufren modificaciones, y podremos afirmar que el proceso habrá terminado cuando los valores de los pesos permanecen estables.

Para modificar los pesos, de modo que la red se acerque a la solución buscada, se distinguen dos aprendizajes principales:

- Aprendizaje supervisado: se caracteriza por estar controlado por un agente externo, (maestro), que controla la salida de la red y en caso de diferir con la deseada, se procede a la modificación de los pesos. Esto se puede realizar por tres métodos.
 - Por corrección del error: consiste en ajustar los pesos de las neuronas en función del error cometido en la salida.
 - Por refuerzo: es el método más lento, pero a su vez el más eficaz. Se basa en un esquema de premios y castigos, en función de los comportamientos de cada conexión recibirá una recompensa o un castigo, según el efecto creado.
 - Estocástico: cambios aleatorios en los pesos y evaluar su efecto a partir del objetivo deseado.
- Aprendizaje no supervisado: no se requiere una influencia externa. Estas redes no reciben ninguna información de si la salida a una entrada es correcta o no. En su lugar, se encargan de encontrar las correlaciones, regularidades o categorías que se puedan deducir de los datos de entrada. Estas redes son muy usadas en problemas de clusterización.

4.3.2. Validación

Una vez finalizado el proceso de entrenamiento, los pesos de las neuronas se quedan fijos. El siguiente paso es saber si nuestra red puede resolver ya nuestro problema, para lo que necesitaremos un nuevo conjunto de datos denominado conjunto de validación, con sus respectivas soluciones. En este caso, se suministran a la red los datos de entrada, sin darle las soluciones como se hacía en el entrenamiento. Una vez tenemos las salidas de la red, las comparamos con las soluciones reales.

Dentro del aprendizaje, hay dos posibles problemas que tendremos que evitar:

- Sobreentrenamiento: Este es un fenómeno que puede ocurrir al entrenar en exceso la red, lo que podría hacer que la red aprendiese las particularidades del grupo de entrenamiento,

ajustando muy bien para ese grupo de datos de entrada en concreto, pero perdiendo la habilidad de poder generalizar su aprendizaje a conjuntos nuevos. Para evitarlo, se selecciona un conjunto de validación. Comprobando de manera paralela el error que comete la red con ese conjunto de validación, observaremos cómo este error va disminuyendo hasta que llega a un punto que empieza a aumentar. Este es el momento en el que se ha producido el sobreentrenamiento: la red está aprendiendo las particularidades del conjunto de entrenamiento concreto. Debemos detener el entrenamiento en el momento que detectemos este mínimo (R. Flórez López [9]).

- Sobreajuste: Un aspecto importante de la red, es el número de neuronas, conexiones y capas que componen el esqueleto de la red, lo cual determinará el éxito de la red. Si la red no tiene suficientes conexiones entre neuronas no convergerá nunca. Si esto se detecta, la solución es sencilla: aumentar las conexiones. El caso contrario es una red densamente poblada. Al ocurrir esto, la red puede llegar a memorizar el conjunto de entrenamiento, en vez de aprender, y la red no podría generalizar. Se trata del sobreajuste (en inglés *overfitting*).

Para evitarlo, se usa el método de Validación cruzada (Fiszlew [10]), el cual permite ajustar los hiperparámetros de la red (su esqueleto) para así poder elegir la estructura óptima (R. Flórez López [9]).

4.3.3. Producción

Una vez que la red está totalmente diseñada, y ha sido entrenada y validada, verificando su correcto funcionamiento, está preparada para su utilización. Esta etapa consiste en aplicar la red a un conjunto de datos (suele ser más grande que los anteriores), para los cuales obtenemos la salida, la cual sería directamente la buscada.

CAPÍTULO 5

LA RED NEURONAL *BACKPROPAGATION*

Esta red tiene sus orígenes en el año 1986, cuando Rumelhart, Hinton y Williams (Rumelhart, Hinton y Williams [11]), basándose en los trabajos previos de otros investigadores (Werbos [12]), formalizaron el algoritmo para que la red neuronal aprendiera la asociación entre las clases de entrada y las categorías correspondientes haciendo uso de un mayor número de niveles de neuronas que los que se usaron previamente para la red Perceptrón, este método es el conocido actualmente como *Backpropagation*.

La idea del algoritmo de *Backpropagation* es sencilla. Se basa en aplicar el estímulo inicial a las neuronas de entrada, (los datos *input*), los cuales se van propagando hacia adelante por los diferentes niveles de capas, generando una salida que se compara con la muestra de entrenamiento y calculando el error mediante la función de coste previamente definida. A continuación, este error se propaga hacia atrás, haciendo que todas las neuronas reciban el porcentaje de error aproximado que le corresponde a cada neurona. Este proceso se repite capa por capa hasta completar todo el esqueleto de la red. En función de este porcentaje se reajustan los pesos para que en la siguiente salida el error haya disminuido y la solución esté más cerca de la deseada.

Una vez establecida la idea genérica, la primera necesidad que nos surge es cómo se va a medir el error entre la salida y la solución de nuestro conjunto de entrenamiento, esto como ya se ha comentado se realiza mediante la función de coste, la cual podría ser muchos tipos de funciones diferentes, la función usada en este estudio, y así se tendrá en cuenta en el próximo desarrollo, es la función de error cuadrático medio:

$$f_{\text{coste}} = \frac{1}{n} \cdot \sum_{i=1}^n (y_i - \bar{y}_i)^2 \quad (5.1)$$

El objetivo de toda red es reducir ese error al mínimo, para encontrar estos mínimos y ajustar correctamente los pesos el algoritmo de *Backpropagation* usa el método conocido como descenso del gradiente.

5.1. Descenso del gradiente

El método del descenso del gradiente es un algoritmo de optimización que permite converger hacia el valor mínimo de una función mediante un proceso iterativo. Este método se basa en calcular la derivada parcial respecto a los parámetros de los cuales depende el error en cada punto de evaluación, (los valores de salida que saca nuestra red), es decir, el gradiente de la función.

Explicaremos esto en mayor profundidad, nuestro objetivo es encontrar un mínimo de una función no-convexa, es decir, una función que pueda tener varios mínimos locales, ya que este sería el caso más general. Como ya se ha mencionado, la derivada de la función, (en R , o gradiente en R^n), evaluado en un punto nos indica la pendiente de esa función en ese mismo punto, nos da la inclinación en cada punto de la función, por lo tanto igualamos la derivada a 0, es decir, estaremos buscando los puntos donde la pendiente es nula. Si fuese una función convexa, sabemos que existe un único punto mínimo, sin embargo, en una función no-convexa pueden existir más mínimos locales, puntos de silla, puntos de inflexión, o máximos locales, por lo que no sería eficiente resolver esta ecuación para todos los posibles puntos, o conseguir diferenciar, si el resultado es un tipo de punto u otro.

No debemos perder de vista que nuestro objetivo es buscar un mínimo de esa función de coste que nos indica el error cometido en el entrenamiento. Por lo tanto, si eligiésemos una función de coste que fuese convexa el problema se podría resolver como se ha indicado en el anterior párrafo de una manera sencilla y directa. Sin embargo, esto limitaría el uso de nuestra red a casos muy sencillos muy lejos de la realidad que se nos presenta, lo que nos obliga a encontrar una solución para resolver este problema para funciones convexas. Para resolverlo, imaginemos que nos encontramos en R^3 , es decir, 2 únicos parámetros evaluados en la función de coste como se muestra a continuación en la Figura 5.1:

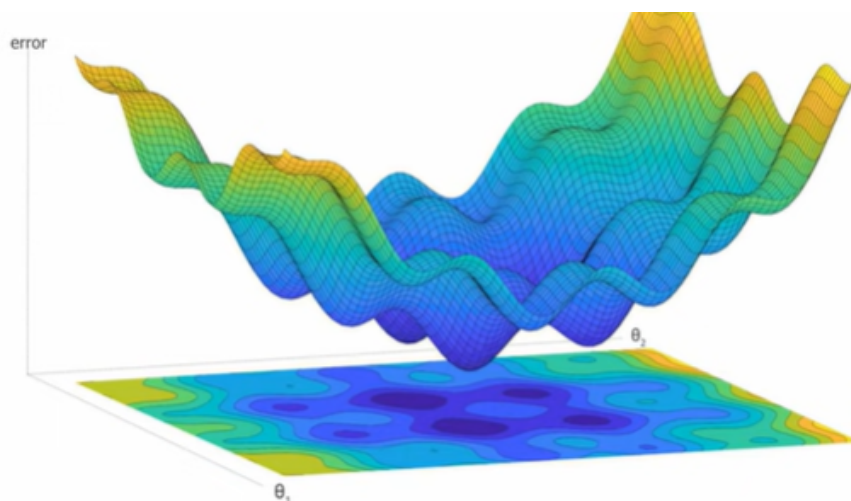


Figura 5.1: Descenso del gradiente, (R. Flórez y J. M. Fernández [9])

En una red usual, habrá miles de parámetros, pero generalizarlo a un caso general es totalmente directo. En un inicio, la red neuronal establece los parámetros que condicionarán el error de salida de una manera aleatoria, por lo que iniciemos en un punto cualquiera del esquema de la figura, ahora deberemos calcular la derivada parcial respecto de cada parámetro, lo cual nos indicará la pendiente de la función en cada eje, juntando ambas direcciones, conformarán un vector conocido como gradiente de la función, ∇f_{coste} , que evaluado en ese punto nos marcará la dirección en el espacio hacia donde la pendiente asciende, pero nosotros no queremos ascender en la función, lo que queremos es buscar un mínimo, por lo que tendremos que movernos en la sentido contrario. El siguiente paso sería decidir cuánto nos movemos antes de reiterar el proceso, esto es una operación importante, ya que si elegimos un factor de movimiento muy pequeño podría tardar mucho en converger, pero si lo elegimos demasiado grande, podría saltarse los mínimos y de esta manera no converger nunca, existen métodos para optimizar la elección de este parámetro como el de Adagrad, Newton y SGD (*Stochastic Gradient Descent*).

Generalizamos el caso a una dimensión y función cualesquiera. Sea f_θ un función en R^n , (depende de n parámetros), sea X la variable inicial, e Y la final. Queremos que $f_\theta(X) \approx Y$. Usaremos la función de error cuadrático medio ya mencionada:

$$f_{coste} = \frac{1}{N} \cdot \sum_{i=1}^n (f_\theta(X) - \bar{y}_i)^2 \quad (5.2)$$

Los datos iniciales X son fijos, por lo que la función f solo dependerá de los parámetros θ , esta función representa las operaciones que se realizan en el interior de la red, en las capas ocultas. Por lo que la función de coste solo dependerá también de estos n parámetros. Como ya se ha explicado, para hallar la pendiente en ese punto aleatorio inicial, (los parámetros en la primera iteración se generan de manera aleatoria), hallamos el gradiente, ∇f_{coste} , y nos movemos en sentido contrario, para avanzar hacia ese mínimo buscado. Primero elegimos como l_m el parámetro de movimiento, por lo que los próximos parámetros serán:

$$\theta = \theta - \nabla f_{coste}(\theta) \quad (5.3)$$

De esta manera iterativamente nos vamos acercando al punto en el que el gradiente será nulo y habremos encontrado lo que buscamos, asegurándonos de que es un mínimo y un máximo local. Sin embargo, este mínimo no podemos asegurar que sea el mínimo global de la función. Esta mejora aún está en estudio. De momento, el método del descenso del gradiente, muy utilizado para la implementación de redes neuronales, solo nos asegura encontrar un mínimo local, en el cual deberemos valorar si la red es suficientemente buena con los resultados obtenidos en ese punto encontrado. De no serlo, sería necesario volver a entrenarla o cambiar su estructura. Una de las múltiples ventajas de usar este tipo de algoritmos es el gran abanico de posibilidades que se pueden variar para encontrar resultados más precisos.

5.2. Algoritmo *Backpropagation*

Mediante el descenso del gradiente hemos encontrado un método para ir reduciendo el error del modelo mediante la variación de los parámetros de nuestra red, sin embargo, su aplicación en

una red neuronal no es directa.

Para aplicar el método del gradiente, necesitamos el gradiente de la función de coste respecto a todos los parámetros de los que dependa nuestra función, lo que implica realizar las derivadas parciales respecto todos ellos. En una red neuronal, esto puede conllevar un coste operacional demasiado alto, en muchos casos inviable. Además, el gradiente de la función lo queremos porque con él respondemos a la pregunta de cómo varía nuestra función de coste en función de la variación de un parámetro. En el caso de una red neuronal, intuir la influencia de un parámetro en la función de coste de los datos finales, puede ser muy complicado, ya que la estructura de una red neuronal está compuesta en la mayoría de los casos por numerosas capas intermedias. Habría que tener en cuenta todas las conexiones que existen entre las neuronas de todas las capas. Para poder resolver este problema, y encontrar el grado de influencia de cada parámetro en la función de coste final, usaremos el algoritmo de *Backpropagation*.

Nuestro objetivo es encontrar el "grado de responsabilidad" que tiene una neurona en el coste final, para ello es importante tener en cuenta que el error de cada capa depende directamente del error de las capas posteriores, por lo que para hallar la responsabilidad de cada neurona recorreremos el esqueleto de la red neuronal en dirección inversa, hacia atrás, desde el error final, hacia la capa inicial, realizando el camino inverso al ejecutado al implementar nuestra red. De esta manera, si encontramos un elemento de la red que influye poco en el error final, esto significará que todas las conexiones posibles que llegan a esta neurona también influirán poco en el error final. Este proceso se le conoce como retropropagación de errores.

El algoritmo *Backpropagation* consiste en operar de forma recursiva capa tras capa trasladando el error hacia atrás, con la ventaja de que cada capa se puede considerar como la última capa de la red, ya que todas las posteriores ya han tenido que ser analizadas previamente, lo cual hace que este algoritmo sea muy eficiente.

Partimos de una red neuronal con L capas, como ya hemos visto en secciones anteriores, en una neurona, intervienen dos tipos de parámetros, los pesos w_i de las neuronas previas que se conectan a ella y el término de sesgo b , donde los pesos w_i participan en una primera operación, que en nuestro caso elegiremos la suma ponderada a la cual la llamaremos Z , posteriormente a esto se le aplica una función de activación, a la cual llamaremos a , por lo que finalmente tendremos que calcular dos tipos de derivadas parciales, con respecto a los pesos, $\frac{\partial C}{\partial w}$, y respecto al sesgo, $\frac{\partial C}{\partial b}$, (denominaremos de aquí en adelante la función de coste como C).

Comenzamos el algoritmo en la última capa, L , al realizar las derivadas parciales de la función C sobre las neuronas de esta última capa tendremos que hallar:

$$\frac{\partial C(a(Z^L))}{\partial w^L} \quad \frac{\partial C(a(Z^L))}{\partial b^L} \quad (5.4)$$

Por lo tanto aplicando la regla de la cadena:(Weisstein [13]).

$$\frac{\partial C}{\partial w^L} = \frac{\partial C}{\partial a^L} \cdot \frac{\partial a^L}{\partial Z^L} \cdot \frac{\partial Z^L}{\partial w^L} \quad (5.5)$$

$$\frac{\partial C}{\partial b^L} = \frac{\partial C}{\partial a^L} \cdot \frac{\partial a^L}{\partial Z^L} \cdot \frac{\partial Z^L}{\partial b^L} \quad (5.6)$$

Siendo nuestra función de coste la usada hasta ahora, la de error cuadrático medio, las derivadas que hay que hallar son de cálculo directo:

$$C(a_j^L) = \frac{1}{2} \cdot \sum_{j=1}^n (y_j - a_j^L)^2 \quad (5.7)$$

$$\frac{\partial C}{\partial a_j^L} = (a_j^L - y_j) , \quad (5.8)$$

donde n es el número de neuronas de esa capa, la derivada $\frac{\partial a^L}{\partial Z^L}$ dependerá de la función de activación de cada neurona, siendo una derivada directa, ponemos el ejemplo de que estemos usando la función sigmoide, entonces:

$$\frac{\partial a^L}{\partial Z^L} = a^L(Z^L) \cdot (1 - a^L(Z^L)) \quad (5.9)$$

Ahora teniendo en cuenta que hemos elegido Z como la suma ponderada:

$$\frac{\partial Z^L}{\partial w_i^L} = a_i^{L-1} \quad (5.10)$$

$$\frac{\partial Z^L}{\partial b^L} = 1 \quad (5.11)$$

Nos fijamos en las derivadas escritas anteriormente. En realidad la derivada $\delta^L = \frac{\partial C}{\partial Z^L} = \frac{\partial C}{\partial a^L} \cdot \frac{\partial a^L}{\partial Z^L}$ nos indica en qué grado se modifica el coste cuando variamos la suma de la neurona. Si la derivada es grande, un pequeño cambio afectará notablemente al error final, y si es pequeña al contrario. La responsabilidad que tiene la neurona en el resultado final, se denomina *error imputado* a la neurona, δ^L , resultando:

$$\frac{\partial C}{\partial w^L} = \delta^L \cdot a_i^{L-1} \quad (5.12)$$

$$\frac{\partial C}{\partial b^L} = \delta^L , \quad (5.13)$$

donde i hace referencia a los pesos de las neuronas conectadas con la que estamos calculando el error. Ya tendríamos los errores para esa última capa. Sin embargo, no es necesario iterar todo el proceso en las $L - 1$ capas restantes. Solo necesitamos una expresión más para poder hallar todas las derivadas parciales que nos interesa, ya que al repetir el proceso anterior tendríamos:

$$\frac{\partial C}{\partial b^{L-1}} = \overbrace{\frac{\partial C}{\partial a^L} \cdot \frac{\partial a^L}{\partial Z^L}}^{\delta^L} \cdot \frac{\partial Z^L}{\partial a^{L-1}} \cdot \overbrace{\frac{\partial a^{L-1}}{\partial Z^{L-1}} \cdot \frac{\partial Z^{L-1}}{\partial}}^{Derivada-fact} w^{L-1} \quad (5.14)$$

$$\frac{\partial C}{\partial b^{L-1}} = \overbrace{\frac{\partial C}{\partial a^L} \cdot \frac{\partial a^L}{\partial Z^L}}^{\delta^L} \cdot \frac{\partial Z^L}{\partial a^{L-1}} \cdot \overbrace{\frac{\partial a^{L-1}}{\partial Z^{L-1}} \cdot \frac{\partial Z^{L-1}}{\partial}}^{Derivada-fact} b^{L-1} \quad (5.15)$$

Sólo faltaría hallar la derivada $\frac{\partial Z^L}{\partial a^{L-1}}$. Esta derivada nos indica cómo varía la suma ponderada al variar el output de una neurona en la capa previa, y se corresponde con la matriz de parámetros W^L que conecta ambas capas. Por tanto, estaríamos consiguiendo mover el error de una capa a la capa anterior. Si se observa la expresión anterior se obtiene lo siguiente:

$$\begin{aligned} \delta^{L-1} &= \frac{\partial C}{\partial Z^{L-1}} = \frac{\partial C}{\partial a^L} \cdot \frac{\partial a^L}{\partial Z^L} \cdot \frac{\partial Z^L}{\partial a^{L-1}} \cdot \frac{\partial a^{L-1}}{\partial Z^{L-1}} \\ &= \delta^L W^L \frac{\partial a^{L-1}}{\partial Z^{L-1}} \end{aligned} \quad (5.16)$$

que es extensible al resto de capas. Desde la ecuación 5.16, las ecuaciones 5.14 y 5.15 se pueden reescribir de la siguiente manera:

$$\frac{\partial C}{\partial w^{L-1}} = \delta^{L-1} \cdot a^{L-2} \quad (5.17)$$

$$\frac{\partial C}{\partial b^{L-1}} = \delta^{L-1} \quad (5.18)$$

Finalmente, el algoritmo se muestra esquematizado de la siguiente manera:

- 1. Calculamos el error de la última capa:

$$\delta^L = \frac{\partial C}{\partial a^L} \cdot \frac{\partial a^L}{\partial Z^L} \quad (5.19)$$

- 2. Retropropagamos el error a la capa anterior:

$$\delta^{L-1} = W^L \delta^L \frac{\partial a^{L-1}}{\partial Z^{L-1}} \quad (5.20)$$

Donde L representa la capa en cuestión.

- 3. Calculamos las derivadas restantes usando el error:

$$\frac{\partial C}{\partial b^{L-1}} = \delta^{L-1} \quad \frac{\partial C}{\partial w^{L-1}} = \delta^{L-1} \cdot a^{L-2} \quad (5.21)$$

- 4. Una vez se obtiene el vector gradiente, aplicamos el método del descenso del gradiente. De esta manera se propaga el error y la variación a toda la red, por lo que se volvería a entrenarla hasta llegar a una solución suficientemente próxima a la requerida.

5.3. Breve explicación de su implementación en R

En este apartado se explicará brevemente, los pasos seguidos para su implementación en R, sin entrar en detalle ya que no es el objetivo del estudio. El código será añadido en el Anexo, junto con comentarios escritos en el propio R para facilitar su comprensión.

El programa consta de 3 subprogramas: `Preparaciondedatos.R`, `Funciones.R` y `RedneuronalBack-Propagation.R`.

El programa `Preparaciondedatos.R` es prepara los conjuntos de datos de entrenamiento, validación y producción. Lee las tablas de datos iniciales en formato csv mediante la función `'read.csv'`, y divide la muestra en entrenamiento (*training*) y validación (*validation*) mediante la función `'sample'`. Posteriormente, pone los datos en el formato correcto (*matrix*), y con el nombre correcto en las respectivas columnas.

En el segundo programa, `Funciones.R`, se encuentran todas las funciones necesarias que se usarán en la red. La principal función es la llamada *neurona*, programada como objeto `S4` con la función `setRefclass` (J. Chambers [14]), lo que nos permite crear clases y subclases que hereden las propiedades de las anteriores, pudiéndolas añadir más atributos. Se establece quién hereda los atributos. Dentro de este programa se encuentra la estructura de la neurona.

También encontramos en este programa las funciones de activación pertinentes, en este caso la sigmoide, la ReLU (Rectified Linear Unit), la lineal, la umbral o la softmax, entre otras muchas que se podrían añadir. También existe un bucle que va a ir guardando las características de las neuronas de cada capa en una lista. También definiremos la función de coste, para la cual usaremos la función de mínimos cuadrados.

Por último, el programa principal `RedneuronalBackPropagation`, en el cual encontramos la función `'redneuronal'`, la cual es una función con los siguientes parámetros de entrada: los datos iniciales junto con las soluciones correctas (para poder entrenar la red), la lista `red` que va guardando las características de las neuronas de cada capa, la cual nos permite que se tengan en cuenta los cambios producidos durante el entrenamiento y que los pesos se vayan actualizando adecuadamente, también la función de coste y por último el parámetro de paso en el descenso del gradiente, aquel que nos indica cuanto avanzamos en cada iteración. Antes de la programación de la propia red se encuentran definidos los parámetros que se usarán en ella, desde el número de neuronas de cada capa o las funciones de activación usadas en cada capa.

Dentro de esta función `red neuronal` se encuentra programada primero la propagación hacia adelante, mediante un bucle que simplemente va implementando las operaciones que se realizan dentro de la red según las especificaciones especificadas anteriormente. Y posteriormente se

encuentra la propagación hacia atrás, en la cual comparamos el resultado obtenido de la propagación hacia adelante con el verdadero resultado mediante la evaluación con la función de coste. Posteriormente, calculamos las derivadas que se indicaron en el desarrollo teórico, inicialmente multiplicando en la última capa la derivada de la función de coste por la función de activación y en las posteriores capas el error de la anterior capa por los pesos de la capa siguiente. De esta manera ya se tendrá implementada la propagación hacia atrás, y quedaría la implementación del descenso del gradiente para actualizar los valores de los parámetros de la red, una vez actualizados la red ya habrá obtenido su nueva estructura y habría que ejecutarla otra vez para que se siga entrenando, esto lo conseguimos simplemente implementando un bucle. Para ejecutarlo con el conjunto de datos de validación y producción, deberemos ejecutar únicamente una vez la red, para quedarnos solo con el resultado de la propagación hacia adelante.

Como ya se ha comentado, en esta sección no se entrará en mayor detalle ya que no es el objetivo del trabajo, la red neuronal solo es el medio mediante el cual estudiamos la capacidad volumétrica de los sólidos nanoporosos, siendo este el estudio en cuestión.

CAPÍTULO 6

PARÁMETROS DE LA RED Y BASES DE DATOS UTILIZADAS.

En los capítulos anteriores, hemos estudiado el funcionamiento de las redes neuronales y en concreto de la red neuronal *BackPropagation*, así como su aplicación al almacenamiento de hidrógeno en materiales sólidos nanoporosos. En este capítulo, expondremos el procedimiento llevado a cabo así como las explicaciones necesarias sobre las bases de datos correspondientes.

6.1. Bases de datos utilizadas.

Realizamos un estudio sobre las bases de datos utilizadas como datos de entrada de la red neuronal. Como ya se ha expuesto, el objetivo de este trabajo es el estudio del almacenamiento de hidrógeno en sólidos nanoporosos, entre los cuales nos centraremos en dos: los nanotubos y los poros plano-paralelos.

Dentro del estudio del almacenamiento de hidrógeno, nos centramos en el estudio de la capacidad volumétrica de estos materiales, propiedad muy importante en el almacenamiento de hidrógeno. Su capacidad volumétrica varía en función de la temperatura y la presión.

Actualmente, la obtención de la capacidad volumétrica de estos materiales se obtiene mediante el Método de Monte Carlo, siendo de alto coste computacional debido al tiempo necesitado para la obtención de las capacidades de almacenamiento. Por lo tanto, resulta difícil clasificar los materiales en función de su capacidad volumétrica debido a la dificultad de obtener información sobre la misma, dificultando también posibles estudios que se podrían realizar para comprobar la relación de la capacidad volumétrica de estos materiales en función de sus condiciones de presión y temperatura, es en este punto donde entra en juego la red neuronal.

Partiremos de dos bases de datos, una para los poros plano-paralelos y otra para los nanotubos. Las bases tienen cuatro columnas: la densidad ρ (kg/L), la temperatura T (K), la presión P (MPa) y la capacidad volumétrica v_c (kg/L). Los datos de estas bases provienen de simulaciones *Grand Canonical Monte Carlo*, GCMC, de las capacidades de almacenamiento de hidrógeno en poros plano-paralelos, torusenos y nanotubos a 298.15 K y presiones entre 0.1 y 25 MPa, publicadas por otros autores [15]. En este trabajo solo resumiremos brevemente en qué consiste el método

GCMC.

6.1.1. Simulaciones GCMC.

Los métodos o simulaciones de Monte Carlo son un conjunto de técnicas matemáticas que se utilizan para estimar los posibles resultados de un suceso incierto. Estos métodos predicen un conjunto de resultados basándose en un rango estimado de valores frente a un conjunto de valores de entrada fijos. Una simulación de Monte Carlo crea un modelo de resultados posibles aprovechando una distribución de probabilidad. Posteriormente, vuelve a calcular los resultados repetidamente, en una sucesión de iteraciones, utilizando en cada iteración un conjunto diferente de números aleatorios entre los valores mínimo y máximo y generando un gran número de resultados probables. Dentro de sus principales desventajas, destaca el gran coste computacional y el tiempo necesario para la obtención de los resultados.

En el método de Monte Carlo-Metrópolis (Metropolis N. [16]) se utiliza el criterio o algoritmo de Metrópolis para simular los posibles resultados en cada iteración. Las simulaciones GCMC son una aplicación del método de Monte Carlo-Metrópolis al conjunto gran canónico y al almacenamiento de un gas dentro de un sólido nanoporoso. En nuestro caso se trata de gas hidrógeno dentro de poros plano-paralelos y nanotubos.

Las tablas de datos de entrada (*input*) usadas en este trabajo, provienen de simulaciones GCMC de 10 millones de iteraciones, estableciendo el equilibrio en 5 millones de iteraciones y utilizando las 5 millones restantes para calcular las capacidades [15]. El 20 % de los ensayos o intentos del algoritmo de Metrópolis consistió en el movimiento de una molécula, el 40 % en la eliminación de una molécula y el 40 % restante consistió en la adición de una molécula.

6.2. Procedimiento realizado.

En este apartado explicaremos el procedimiento que se ha llevado a cabo para la aplicación de la red neuronal en este estudio así como la obtención de los conjuntos de entrenamiento, validación, y producción.

En este trabajo nos restringimos a unas condiciones fijas de temperatura y presión: $T = 298.15$ K y $P = 25$ MPa, siendo estas las condiciones de estudio más relevantes y realistas para el almacenamiento de hidrógeno a bordo de un vehículo de hidrógeno.

Antes de ejecutar la red con los bases de datos de los materiales poros plano-paralelos y nanotubos se comprueba su correcto funcionamiento con dos funciones sencillas.

6.2.1. Comprobación de la red.

Comprobamos el correcto funcionamiento de la red con la función lineal y cuadrática.

Función lineal: $y = x$, se introduce la función en el código generando 250 valores aleatorios entre 0 y 1. Una vez obtenidos los datos de entrada, entrenamos la red con los parámetros de la red mostrados en la Tabla 6.1:

N° capas	Neuronas por capa	Funciones de activación	N° iteraciones	Fac.paso
2	2-1	Sigmoide-ReLU	1000000	0.01

Tabla 6.1: Estructura de la red neuronal para la función lineal

Red sin ninguna capa oculta, solo la inicial y final, con 2 y 1 neuronas respectivamente. Para medir el error, y poder contabilizar el grado de ajuste hemos usado el error de desviación estándar. Los resultados obtenidos se muestran en la Figura 6.1.

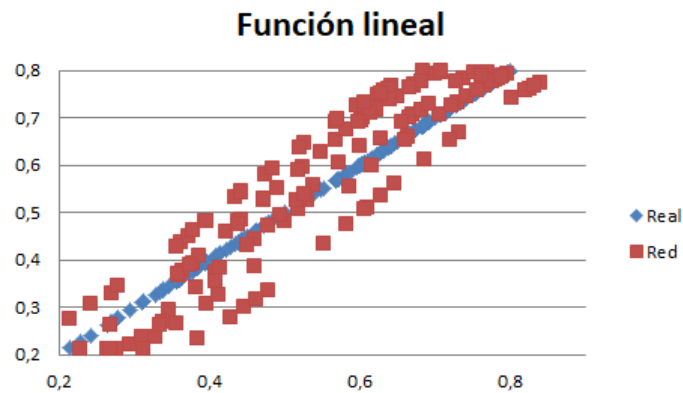


Figura 6.1: Ajuste de la red para la función lineal

La desviación estándar vale 0.0387 u, (unidad genérica), siendo una aproximación coherente con el número de iteraciones empleado. Para entrenar la red con las bases de datos de poros plano-paralelos y nanotubos elegiremos un factor de paso más pequeño para obtener resultados más precisos.

Función cuadrática: $y = x^2$, de la misma manera que en la función anterior se introduce la función en el código generando 250 valores aleatorios entre 0 y 1. Una vez obtenidos los datos de entrada, entrenamos la red con los parámetros mostrados en la Tabla 6.2.

N° capas	Neuronas por capa	Funciones de activación	N° iteraciones	Fac.paso
3	3-2-1	sigmoide-ReLU-sigmoide	10000000	0.01

Tabla 6.2: Estructura de la red neuronal para la función cuadrática

Red con 1 capa oculta. La capa inicial cuenta con 3 neuronas, la capa oculta con 2 y la capa final con una única neurona. Los resultados obtenidos se muestran en la Figura 6.2.

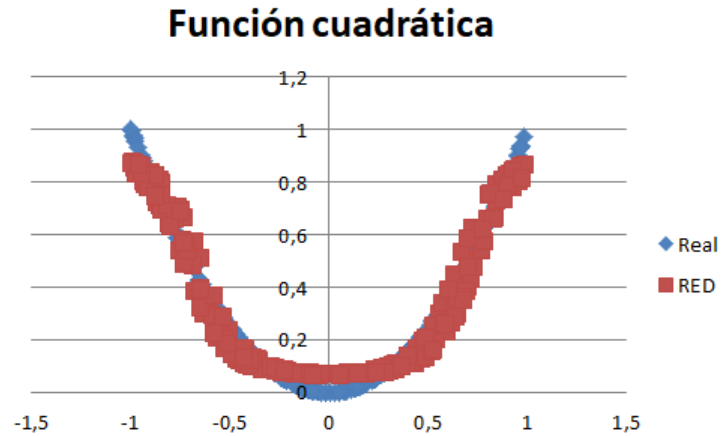


Figura 6.2: Ajuste de la red para la función cuadrática

Como se observa, la red ajusta peor para valores cercanos al 0 y al 1, al encontrarse en los límites de los datos normalizados, siendo este un comportamiento esperado. Para el correcto funcionamiento de la red, es importante normalizar los datos antes de introducirlos en la misma, ya que las funciones de activación introducidas en cada neurona no normalizan los datos por defecto. Una incorrecta normalización de los datos de entrada puede conllevar un ajuste incorrecto por parte de la red, provocando que esta no llegue a converger.

La desviación estándar vale 0.018 u, que es un valor coherente con el número de iteraciones empleado.

6.2.2. Aplicación de la red a las bases de datos poros plano-paralelos y nanotubos.

Antes de explicar el proceso realizado, es muy importante entender las bases de datos de las que partimos, para poder elaborar una estrategia acorde y entender los datos que estamos obteniendo. Por ello, vamos a ilustrar las gráficas correspondientes a los datos de cada material.

Las bases de datos usadas se encuentran expuestas en el Anexo.

- **Poros plano-paralelos:** En la Figura 6.3 se muestra las capacidades volumétricas correspondientes al material poros plano-paralelos.

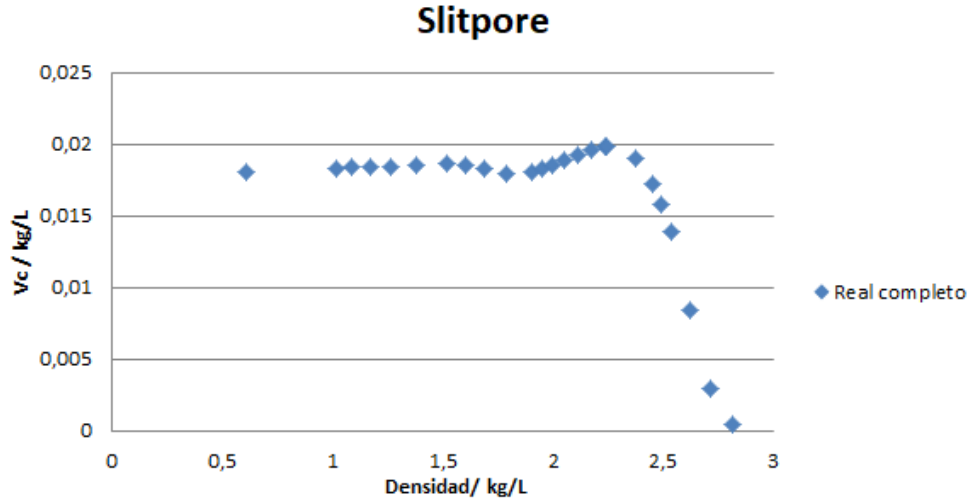


Figura 6.3: Capacidad volumétrica en función de la densidad de los poros plano-paralelos, a $T = 298.15$ K y $P = 25$ MPa.

- **Nanotubos:** En la Figura 6.4 se muestra las capacidades volumétricas correspondientes al material nanotubo.

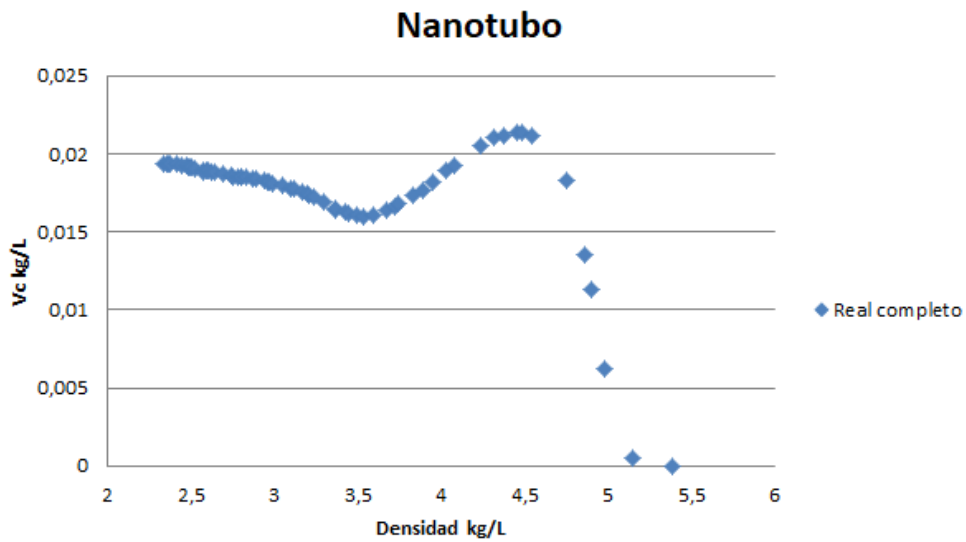


Figura 6.4: Capacidad volumétrica en función de la densidad de los nanotubos, a $T = 298.15$ K y $P = 25$ MPa.

Para entender las bases de datos de entrada, hay que tener en cuenta varios aspectos. Primero, la relación entre la capacidad o densidad volumétrica y la presión es prácticamente lineal a

temperatura ambiente. Además, la capacidad volumétrica tiene un comportamiento casi lineal a presiones bajas y moderadas, entre 0 y 10 MPa, pasando posteriormente a un comportamiento asintótico.

Se puede diferenciar dos tramos en las capacidades volumétricas de ambos tipos de poros, en función de la densidad del poro. Una parte del almacenamiento en los poros se debe a la fisorción y otra parte a la compresión. La contribución debida a la fisorción aumenta muy rápidamente con el diámetro del poro, alcanza un máximo para diámetros de 7-8 Å y luego disminuye. La contribución debida a la compresión es nula o casi nula para poros muy estrechos y es constante, no depende del diámetro del poro, para los demás poros. En el primer tramo tenemos los poros de menor densidad (los poros más anchos) y en el segundo tramo los de mayores densidades (los poros más estrechos). El almacenamiento en el primer tramo es debido a la fisorción y a la compresión. A menor densidad, menor será la contribución de la fisorción. El principal origen físico del almacenamiento en el segundo tramo es la fisorción. En las siguientes figuras mostramos los dos tramos.

- **Poros plano-paralelos:** En la Figura 6.5 se muestran los dos tramos correspondientes a los poros plano-paralelos.

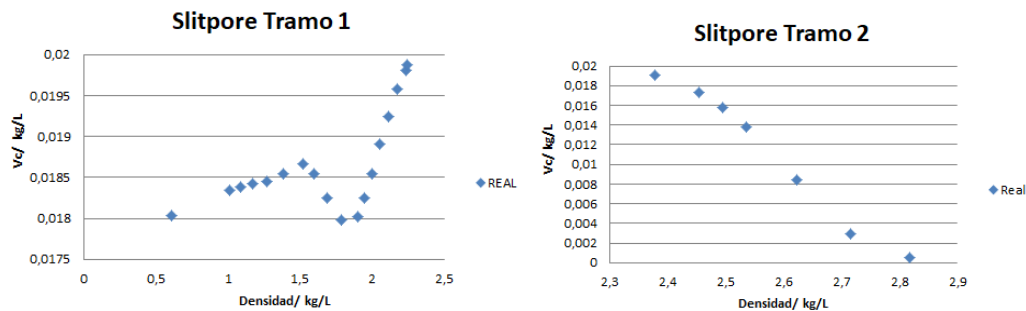


Figura 6.5: Tramos divididos de los poros plano-paralelos

- **Nanotubos:** En la Figura 6.6 se muestran los dos tramos correspondientes a los nanotubos.

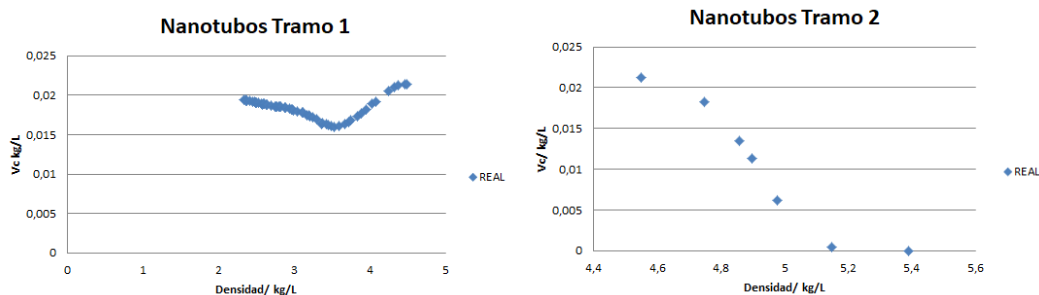


Figura 6.6: Tramos divididos de los nanotubos

Vamos a entrenar la red para cada material por separado. Sin embargo, tenemos que tener en cuenta dos aspectos: el primero es que el número de datos de entrada es limitado, lo cual dificulta el entrenamiento de la red, y el segundo es el hecho de que los dos tramos que hemos diferenciado en cada muestra tienen comportamientos y orígenes físicos diferentes, por lo que decidimos tratarles por separado. Tanto las particularidades de cada red, como el tratamiento y exposición de los resultados se expone en el capítulo siguiente.

CAPÍTULO 7

PRESENTACIÓN Y ANÁLISIS DE LA APLICACIÓN DE LA RED NEURONAL.

En este capítulo, expondremos los resultados obtenidos de la aplicación de la red neuronal al problema del almacenamiento de hidrógeno en materiales sólidos nanoporosos, así como las dificultades encontradas en su aplicación y las conclusiones obtenidas.

Inicialmente explicaremos la obtención de los conjuntos de entrenamiento, validación y producción, ya que esta parte del proceso es común para todos los ajustes que realicemos.

7.1. Conjuntos de Entrenamiento, Validación y Producción

Escogeremos los conjuntos de entrenamiento y validación de forma aleatoria, de tal manera que el conjunto de entrenamiento se corresponda con el 80 % y el conjunto de validación con el 20 % del total de la muestra o base de datos. Es importante que nos aseguremos que ambos conjuntos tengan sus puntos distribuidos a lo largo de toda la muestra, es decir, tenemos que evitar que tanto el conjunto de validación como el de entrenamiento estén concentrados en torno a un conjunto de puntos determinado. Para ello, leeremos la base de datos, que está en formato csv, con la función `read.table`. Posteriormente usaremos la función `sample`, la cual nos divide de forma aleatoria y distribuida la muestra según el porcentaje que elijamos, por lo que obtendríamos el 80% correspondiente al conjunto de entrenamiento. Una vez tenemos este conjunto, definiremos el conjunto de validación como la parte de la base de datos que no se corresponde con el conjunto de entrenamiento ya definido.

Por último, para hallar el conjunto de producción, es necesario que sea una muestra mayor a las dos anteriores, por lo que debido a la cantidad reducida de puntos de la que partimos (a causa del gran coste operacional y el tiempo que lleva obtenerlos mediante el método de Monte-Carlo), definiremos el conjunto de producción como el conjunto de validación más el de entrenamiento, es decir, toda la muestra o base de datos.

Una vez realizado el estudio sobre la base de datos, y dejando bien definidos los tres conjuntos, entrenamiento, validación y producción, podemos empezar con el entrenamiento de la red como se ha explicado en la sección 4.2, lo que detallaremos a continuación.

El código usado se expone en el Anexo, junto a explicaciones concretas de todas las sentencias usadas.

7.2. Nanotubos

En el anterior capítulo, hemos representado la capacidad volumétrica en función de la densidad, recordando que nos encontramos en las condiciones de temperatura de 298.15 K y presión de 25 MPa. En la figura 6.8 podemos ver los dos tramos en los que dividimos los puntos que se corresponden con los nanotubos. Ahora mostraremos los resultados del conjunto de producción (ya que es el que nos interesa) de cada tramo junto con las particularidades de la red usada en cada caso y el error cometido.

- **Tramo 1:** Se corresponde a las densidades entre 2.335 kg/L y 4.488 kg/L. Se usan los 58 valores correspondientes. Posteriormente entrenamos la red con las siguientes parámetros mostrados en la Tabla 7.1:

Nº capas	Neuronas por capa	Funciones de activación	Nº iteraciones	Fac.paso
3	4-2-1	sigmoide-ReLU-sigmoide	100000000	0.001

Tabla 7.1: Estructura de la red Tramo1-Nanotubos

Hemos elegido una red con una capa oculta formada por dos neuronas, también hemos aumentado el número de iteraciones y usado un factor de paso más pequeño para mejorar el ajuste. Se ha elegido la configuración de funciones de activación sigmoide-ReLU-sigmoide, debido al buen rendimiento de la función sigmoide en las capas iniciales y finales, y de la función ReLU en la capa oculta cuando esta capa no posee muchas neuronas, aumentando la velocidad de convergencia, (Chollet [21]). Los resultados obtenidos se muestran en la Figura 7.1:

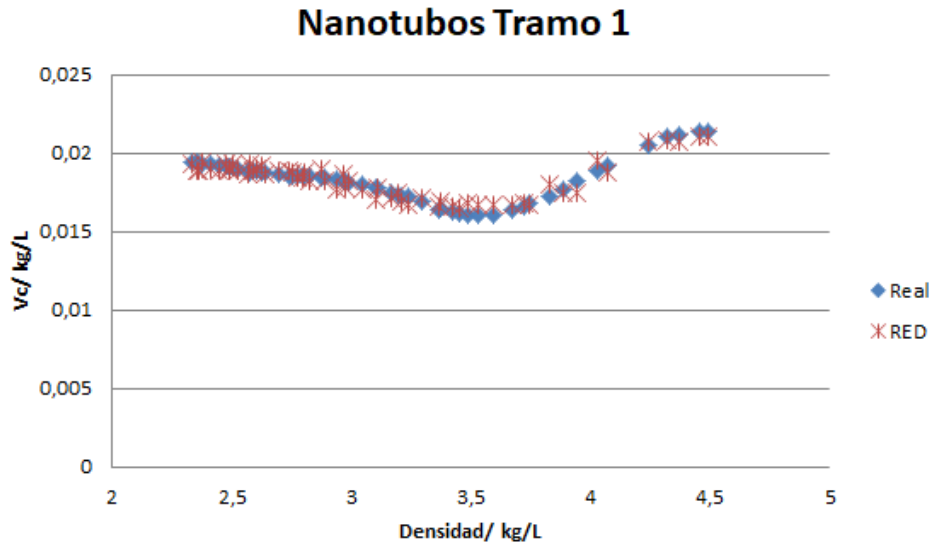


Figura 7.1: Ajuste de la red para el tramo 1 de los nanotubos.

Se observa un correcto ajuste por parte de la red, gracias a un aumento de las iteraciones y al haber usado un factor de paso más pequeño acorde con los datos que estamos tratando. Antes de entrenar la red hemos normalizado los datos. Una vez normalizadas las tres muestras, la de entrenamiento, validación y producción, hemos entrenado la red con las anteriores especificaciones. Después, hemos ejecutado la red para la muestra de validación comprobando el correcto funcionamiento y finalmente para la muestra de producción, cuyos resultados son los observados en la gráfica anterior. La desviación estándar toma el valor de 0.000203 kg/L.

- Tramo 2: Se corresponde con las densidades entre 4.488 kg/L y 5.39 kg/L. Este tramo solo está compuesto por 7 puntos, por lo que son insuficientes para poder entrenar correctamente la red. Debido a este inconveniente hemos procedido de la siguiente manera:

- Primero representamos los datos correspondientes a este tramo, realizando un ajuste a una función polinómica de grado 2, obteniendo un $R^2 = 0.92$ con la siguiente función:

$$0.0139x^2 - 0.1671x + 0.4962$$

- Una vez tenemos la función de ajuste, programamos en R una función que genere 250 puntos a través de este ajuste.
- Construimos las muestras de entrenamiento, validación y producción con los 250 puntos generados, procediendo al entrenamiento, validación y generación de los puntos de producción como en el resto de casos.
- Una vez realizado el usual proceso con la red, calculamos el error de los 7 puntos iniciales.

Hemos realizado el entrenamiento de la red con las especificaciones mostradas en la Tabla 7.2:

N° capas	Neuronas por capa	Funciones de activación	N° iteraciones	Fac.paso
3	5-3-1	sigmoide-ReLU-sigmoide	100000000	0.001

Tabla 7.2: Estructura de la red Tramo2-Nanotubos

Se ha usado una red con 5 neuronas en la capa inicial, 1 capa oculta con 3 neuronas y la capa final con una única neurona. La Figura 7.2 muestra los resultados correspondientes al ajuste realizado y particularizando a los 7 puntos correspondientes de nuestra base de datos inicial.

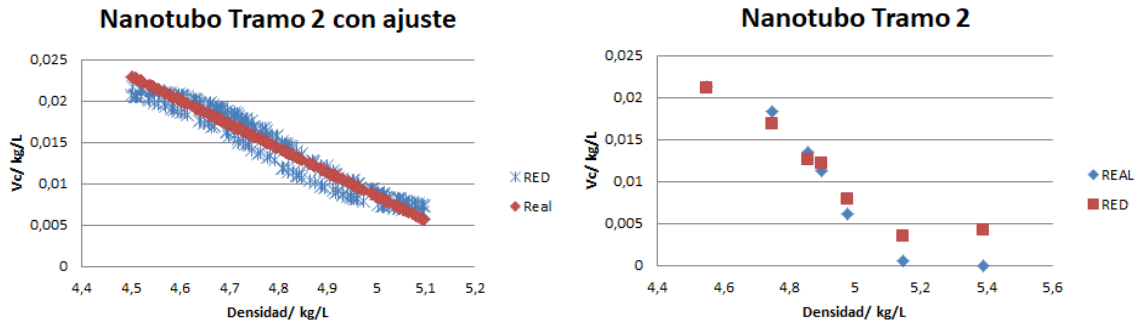


Figura 7.2: Ajuste de la red para el tramo 2 de los nanotubos.

Siguiendo el proceso explicado, obtenemos un correcto ajuste por parte de la red, con una desviación estándar de 0.0014 kg/L. El error calculado es el correspondiente a los 7 puntos de nuestra base de datos. Observamos que aunque se pueda considerar un buen ajuste, este es peor que en el tramo 1, lo cual se debe principalmente al error obtenido en el último punto, el cual es muy difícil de llegar a ajustar correctamente ya que no sigue la monotonía esperada del resto de puntos y es un valor que da exactamente 0. Al ser una muestra de únicamente 7 puntos, este punto influye considerablemente en el error total. Quitándole de la muestra, obtendríamos un error de 0.00097 kg/L, siendo más próximo al error obtenido en el primer tramo.

7.3. Poros plano-paralelos

Como en el caso de los Nanotubos, hemos representado la capacidad volumétrica en función de la densidad, recordando que nos encontramos en las condiciones de temperatura de 298.15 K y presión de 25 MPa. En la figura 6.7 podemos ver los dos tramos en los que dividimos los puntos que se corresponden con los poros plano-paralelos. A continuación, mostraremos los resultados del conjunto de producción, de cada tramo junto con las particularidades de la red usada en cada caso y el error cometido.

- Tramo 1: Se corresponde a las densidades menores a 2.4 kg/L. Se usan todos valores correspondientes, existiendo menos que en el primer tramo de los nanotubos. Posteriormente entrenamos la red con los parámetros mostrados en la Tabla 7.3:

N° capas	Neuronas por capa	Funciones de activación	N° iteraciones	Fac.paso
3	3-2-1	sigmoide-ReLU-sigmoide	50000000	0.001

Tabla 7.3: Estructura de la red Tramo1-poros plano-parallelos

Se ha usado una red con 3 neuronas en la capa inicial, 1 capa oculta con 2 neuronas y una capa final con una única neurona. Mismo factor con un número de iteraciones algo menor. El hecho de poseer una muestra menor, se ve reflejado en los resultados mostrados en la Figura 7.3.

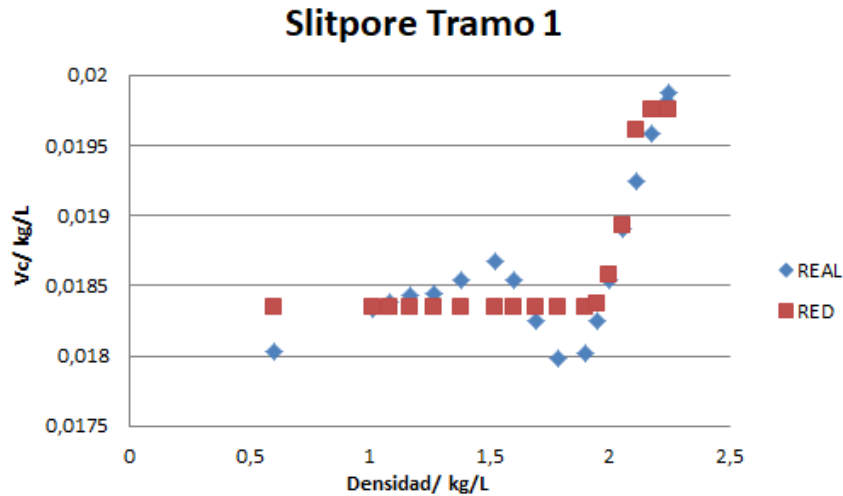


Figura 7.3: Ajuste de la red para el tramo 1 de los poros plano-parallelos.

El procedimiento seguido es el mismo que el explicado para los nanotubos. Observando la gráfica, se observa cómo ajusta correctamente la parte final ascendente y como le resulta más complicado ajustar la parte inicial. Este resultado es el esperado, debido al cambio de tendencia y los pocos datos disponibles que existen para captarlo. Sin embargo, calculando el error de desviación estándar del valor absoluto de los errores producidos en el ajuste, 0.000125 kg/L, nos damos cuenta que el resultado es bastante bueno y preciso. Obviamente, el error en la parte inicial de la gráfica es mayor que en el tramo ascendente. Una de las posibles mejoras que se podría realizar es dividir este tramo también en dos subtramos, facilitando así el entrenamiento y ajuste de la red.

- **Tramo 2:** Se corresponde a las densidades mayores de 2.4 kg/L. Este tramo solo está compuesto por 7 puntos, como en el caso de los nanotubos, son insuficientes para poder entrenar correctamente la red. Se ha realizado el mismo procedimiento, obteniendo $R^2 = 0.98$ con la función de ajuste siguiente:

$$y = -0.0466x + 0.1309$$

El entrenamiento de la red le hemos realizado con las especificaciones de la Tabla 7.4:

N° capas	Neuronas por capa	Funciones de activación	N° iteraciones	Fac.paso
3	5-3-1	sigmoide-ReLU-sigmoide	100000000	0.001

Tabla 7.4: Estructura de la red Tramo2-poros plano-parallelos

Red con 5 neuronas en la capa inicial, 3 neuronas en la única capa oculta y 1 neurona en la capa final. Mostramos ahora los resultados correspondientes al ajuste realizado y particularizando a los 7 puntos correspondientes. La figura 7.4 se muestran los resultados obtenidos.

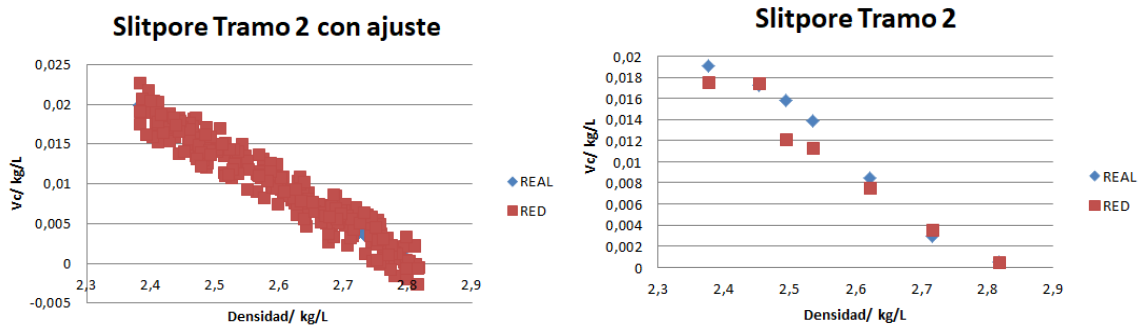


Figura 7.4: Ajuste de la red para el tramo 2 de los poros plano-paralelos.

Obtenemos un correcto ajuste por parte de la red, con un error de desviación estándar del 0.00134 kg/L. El error calculado es el correspondiente a los 7 puntos de nuestra base de datos. Únicamente hay que comentar el error es algo mayor que en el tramo 1, lo que se debe principalmente al error del tercer punto. El hecho de que este error sea algo mayor se debe a la ligera curva que experimenta la muestra a esa densidad, cambiando la tendencia, aún así el error cometido en ese punto sigue siendo menor de 0.01 kg/L, lo cual es valorado positivamente.

De esta manera, se comprueba el correcto funcionamiento de la red. Una vez entrenada, el tiempo y coste computacional para la obtención de las capacidades volumétricas sería mucho menor que el obtenido mediante las simulaciones de Monte Carlo-Metrópolis, pudiendo avanzar en la solución del problema del almacenamiento de hidrógeno.

7.4. Conclusiones

En este Trabajo de Fin de Grado hemos tratado de relacionar la Física (capacidades de almacenamiento de hidrógeno) con la Inteligencia Artificial (redes neuronales) y la Informática o más bien la Programación Informática (implementación y pruebas del código en R de una red neuronal). Hemos usado la red neuronal *Backpropagation* para la obtención de las capacidades volumétricas de los poros plano-paralelos y nanotubos. Este tipo de poros son importantes en el estudio del almacenamiento de hidrógeno en carbones nanoporosos.

El método usado hasta el momento era el método de simulación de Monte Carlo. La obtención de los resultados mediante estas simulaciones conllevaba un alto coste computacional. Mediante la aplicación de la red neuronal hemos obtenido resultados muy similares en un tiempo mucho menor. Por tanto, hemos verificado que la aplicación de métodos de Aprendizaje Automático al problema físico del almacenamiento de hidrógeno, proporciona un método altamente eficiente para el estudio de las capacidades de almacenamiento de hidrógeno en materiales sólidos nanoporosos.

Dentro de los posibles estudios futuros se encuentra la aplicación de esta u otras redes neuronales a más materiales sólidos nanoporosos. Esto permitiría, finalmente, encontrar los mejores materiales para almacenar hidrógeno. De esta manera, nos aproximaríamos algo más a la solución del problema del almacenamiento de hidrógeno y contribuiríamos a conseguir el objetivo final de implantar el ecosistema que conlleva la economía del hidrógeno.

- [1] J. R. SOLANO, *Hacia una economía basada en el hidrógeno*, Anales de la Universidad Metropolitana 4, 165-180, 2004. 1
- [2] D. J. MATICH, *Redes Neuronales: Conceptos Básicos y Aplicaciones*, Tesis de Grado en Ingeniería Química, Universidad Tecnológica Nacional, Facultad Regional Rosario, Departamento de Ingeniería Química, Argentina, 55 pp., marzo de 2001. 2, 12, 13, 43
- [3] P. A. LESSING, *Low Permeation Liner for Hydrogen Gas Storage Tanks*, Idaho National Engineering and Environmental Laboratory and University of California at Los Angeles (UCLA), 2003. 4
- [4] A. ZÜTTEL, *Hydrogen storage methods*, Naturwissenschaften 91, 157-172. doi:<https://link.springer.com/article/10.1007/s10450-019-00065-x>, 2004. 4, 5
- [5] J. A. VILLAJOS COLLADO, *Adsorción de hidrógeno en materiales MOF: estrategia para la inclusión de centros de adsorción preferente*, Tesis Doctoral en Ingeniería Química, Universidad Rey Juan Carlos, Escuela Superior de Ciencias Experimentales y Tecnología, Móstoles, 2017. 7, 8, 43
- [6] M-S PARK, S-E LEE, M I KIM and Y-S LEE, *CO₂ adsorption characteristics of slit-pore shaped activated carbon prepared from cokes with high crystallinity*, Carbon Lett. 16, 45-50, 2015. 8
- [7] E. VARELA ARREGOCÉS, EDWIN CAMPBELLS SÁNCHEZ and U. SIMÓN BOLÍVAR, *Redes Neuronales Artificiales: Una Revisión del Estado del Arte, Aplicaciones Y Tendencias Futuras Artificial Neural Networks: A Brief Review*, Barranquilla -Atlántico, Investig. y Desarro. en TIC 2, 18-27, 2011. 13, 17, 43
- [8] R. ROJAS, *A Systematic Introduction: Neural Networks*, Springer Verlag, Berlin and Heidelberg, 1996. 15, 16, 43
- [9] R. FLÓREZ-LÓPEZ and J. M. FERNÁNDEZ FERNÁNDEZ, *Las redes neuronales artificiales, fundamentos teóricos y aplicaciones prácticas*, Ed. Netbiblo, Oleiros (La Coruña), 2008. 18, 20, 43
- [10] A. FISZLEW, *Generación automática de redes neuronales con ajuste de parámetros basado en algoritmos genéticos*, Tesis de Grado en Ingeniería Informática, Facultad de Ingeniería, Universidad de Buenos Aires, Argentina, febrero de 2002. 18

- [11] D. E. RUMELHART, G. E. HINTON and R. J. WILLIAMS, *Learning representations by back-propagating errors*, *Nature* 323, 533–536, 1986. 19
- [12] P. WERBOS, *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Science. Thesis (Ph. D.). Appl. Math. Harvard University*, 1974. 19
- [13] E. W. WEISSTEIN, *Chain Rule*, from MathWorld—A Wolfram Web Resource. <https://mathworld.wolfram.com/ChainRule.html>. Accessed, 25 June 2022. 22
- [14] J. CHAMBERS, *Software for Data Analysis: Programming with R*, Springer Verlag, Berlin and Heidelberg, January 2007. doi = 10.1007/978-0-387-75936-4 25
- [15] D. CAVIEDES and I. CABRIA, *Grand Canonical Monte Carlo simulations of the hydrogen storage capacities of slit-shaped pores, nanotubes and torusenes*, *Int. J. Hydrogen Energy* 47, 11916-11928, 2022. 9, 27, 28, 43
- [16] N. METROPOLIS, *The Beginning of the Monte Carlo Method (Special Issue dedicated to Stanislaw Ulam)*, *Los Alamos Sci* 15, 125-130, 1987. 28
- [17] E. LÓPEZ, *Simulaciones de Monte Carlo-Metrópolis del almacenamiento de hidrógeno en PIMs (Polymers of Intrinsic Microporosity).*, Trabajo de Fin de Grado, Grado en Física, Universidad de Valladolid, junio de 2021. 6
- [18] R. LACKES, J. ZIOLA and K. AHERN., *Neuronal Networks: Basics and Applications*, CBT (Computer Based Training) Springer Verlag, Berlin and Heidelberg, 1998. 2
- [19] OFFICE OF ENERGY EFFICIENCY & RENEWABLE ENERGY, FUEL CELL TECHNOLOGIES OFFICE, *DOE Technical Targets for Onboard Hydrogen Storage for Light-Duty Vehicles*, <https://www.energy.gov/eere/fuelcells/doe-technical-targets-onboard-hydrogen-storage-light-duty-vehicles>, 2018. Accessed, 25 June 2022. 4, 6, 10
- [20] OFFICE OF ENERGY EFFICIENCY & RENEWABLE ENERGY, FUEL CELL TECHNOLOGIES OFFICE, *Materials-Based Hydrogen Storage*, <https://www.energy.gov/eere/fuelcells/materials-based-hydrogen-storage>, 2018. Accessed, 25 June 2022. 4, 6, 10
- [21] FRANCOIS CHOLLET, *Deep Learning with Python*, Manning Publications, Shelter Island, New York, November 2017, 504 pp. 35

ÍNDICE DE FIGURAS

3.1. Estructura del PCN-610/NU-100 (Villajos Collado [5])	8
3.2. Geometría del poro plano-paralelo [15])	9
3.3. Geometría del nanotubo [15])	9
4.1. Biología de la neurona, (Matich [2])	12
4.2. Esquema de una neurona (Matich [7]).	13
4.3. Estructura de una red neuronal (R.Rojas [8]).	15
5.1. Descenso del gradiente, (R. Flórez y J. M. Fernández [9])	20
6.1. Ajuste de la red para la función lineal	29
6.2. Ajuste de la red para la función cuadrática	30
6.3. Capacidad volumétrica en función de la densidad de los poros plano-paralelos, a $T = 298.15$ K y $P = 25$ MPa.	31
6.4. Capacidad volumétrica en función de la densidad de los nanotubos, a $T = 298.15$ K y $P = 25$ MPa.	31
6.5. Tramos divididos de los poros plano-paralelos	32
6.6. Tramos divididos de los nanotubos	32
7.1. Ajuste de la red para el tramo 1 de los nanotubos.	36
7.2. Ajuste de la red para el tramo 2 de los nanotubos.	37
7.3. Ajuste de la red para el tramo 1 de los poros plano-paralelos.	38
7.4. Ajuste de la red para el tramo 2 de los poros plano-paralelos.	39