



Universidad de Valladolid



ESCUELA DE INGENIERÍAS
INDUSTRIALES

UNIVERSIDAD DE VALLADOLID

ESCUELA DE INGENIERIAS INDUSTRIALES
Y FACULTAD DE MEDICINA

Grado en Ingeniería Biomédica

**Monitorización de la actividad de
personas mayores mediante técnicas de
aprendizaje profundo**

Autor:

Sánchez-Girón Coca, Celia

Tutor(es):

Pérez Turiel, Javier

Duque Domingo, Jaime

Departamento de Ingeniería de
Sistemas y Automática

Valladolid, Julio 2023

ÍNDICE DE CONTENIDO

ÍNDICE FIGURAS	4
ÍNDICE DE TABLAS	5
RESUMEN	6
ABSTRACT	7
CAPÍTULO 1: INTRODUCCIÓN.....	8
1.1 INTRODUCCIÓN.....	8
1.2 JUSTIFICACIÓN DEL PROYECTO	10
1.3 MOTIVACIÓN PERSONAL	11
1.4 OBJETIVOS.....	11
CAPÍTULO 2: ESTADO DEL ARTE.....	13
2.1 ESTUDIO DE MERCADO <i>WEARABLES</i> EN SALUD	13
2.2 ESTUDIO DE APPS DE SALUD Y ACTIVIDAD FÍSICA.....	22
2.3 ANTECEDENTES DEL APRENDIZAJE PROFUNDO.....	24
2.4 ESTUDIOS PREVIOS SOBRE HAR	26
2.5 BASES DE DATOS HAR	28
CAPITULO 3: MARCO TEÓRICO	31
3.1 SERIES TEMPORALES Y MONITOREO DE SALUD.....	31
3.2 REDES NEURONALES PARA EL APRENDIZAJE PROFUNDO.....	32
3.2.1 FUNDAMENTOS APRENDIZAJE PROFUNDO.....	32
3.2.2 CONCEPTOS DEL APRENDIZAJE PROFUNDO.....	34
3.3 EXTRACCIÓN DE DATOS	50
CAPÍTULO 4: DESARROLLO.....	55
4.1 EXTRACCIÓN DE DATOS	55
4.1.1 AUTENTICACIÓN EN PULSERA INTELIGENTE	55
4.1.2 PARÁMETROS DE SALUD	56
4.1.3 DATOS SENSORES INERCIALES	59
4.2 ELECCIÓN Y PREPARACIÓN DE <i>DATASETS</i>	64
4.3 ARQUITECTURA DE RED	67
4.3.1 DATOS DE ENTRADA.....	67
4.3.2 MODELO CNN	68
4.3.3 MODELO LSTM	69
4.3.4 MODELO HÍBRIDO CNN-LSTM	70
CAPÍTULO 5: RESULTADOS EXPERIMENTALES.....	72
5.1 ELECCIÓN DE MÉTRICAS DE CLASIFICACIÓN	72
5.2 EVALUACIÓN DEL ENTRENAMIENTO	73

5.3 MODIFICACIONES Y MEJORAS.....	79
CAPÍTULO 6: CONCLUSIONES.....	84
CAPÍTULO 7: LÍNEAS FUTURAS.....	85
BIBLIOGRAFÍA.....	87

ÍNDICE FIGURAS

Figura 1: Distribución de la población por edades en porcentaje [1]	8
Figura 2: Smartband Amazfit band 5.....	14
Figura 3: Smartband Amazfit Band 7.....	15
Figura 4: Smartband Mi Band 4	16
Figura 5: Smartband Xiaomi Mi Smart Band 6	16
Figura 6: Smartband Xiaomi Mi Smart Band 7	17
Figura 7: Smartband Fitbit Charge 5	18
Figura 8: Smartband Fitbit Inspire 3.....	19
Figura 9: Smartband Honor Band 5.....	20
Figura 10: Smartband Huawei Band 7	21
Figura 11: Esquema arquitectura Health Kit y servicios Huawei [7]	22
Figura 12: Carpetas extraídas del archivo .zip que recogen información de la app Zepp Life	23
Figura 13: Esquema de IA, ML y DL [1]	25
Figura 14: Esquema general HAR utilizando Machine Learning.....	27
Figura 15: Esquema general HAR utilizando Deep Learning	28
Figura 16: Modelo de perceptrón [16]	33
Figura 17: Representación función Relu en Geogebra	36
Figura 18: Representación función Sigmoide en GeoGebra	37
Figura 19: Representación función Tangencial Hiperbólica en GeoGebra	37
Figura 20: Arquitectura general modelo CNN	42
Figura 21: Modelo de neurona recurrente	43
Figura 22: Modelo neurona recurrente desenrollada en el tiempo.....	44
Figura 23: Arquitectura modelo LSTM [17].....	45
Figura 24: Arquitectura modelo residual	47
Figura 25: Arquitectura básica Bluetooth Low Energy [18].....	51
Figura 26: Interacciones GATT Cliente/Servidor [19].....	53
Figura 27: Captura aplicación nrfConnect.....	54
Figura 28: Modelo genérico de adquisición de datos.....	57
Figura 29: Datos obtenidos con Gadgetbridge en archivo .sqlite.....	58
Figura 30: Modelo de flujo de información de salud y bienestar	58
Figura 31: Datos obtenidos desde Gadgetbridge al ser exportados a python.....	59
Figura 32: Etiquetas estimadas de actividades obtenidas desde Gadgetbridge	59
Figura 33: UUID Oxfee0, página oficial de Bluetooth.....	61
Figura 34: Resultado al llamar CHARACTERISTIC_STEPS, CHARACTERISTIC_CURRENT_TIME, CHARACTERISTIC_ACTIVITY_DATA.....	62
Figura 35: Datos del giroscopio obtenidos a partir de CHARACTERISTIC_SENSOR_MEASURE.....	63
Figura 36: Señales acelerómetros para reconocer actividades en dataset WISDM ..	64
Figura 37: Distribución de las muestras por actividad en dataset UCI HAR	66
Figura 38: Distribución de las muestras por actividad en dataset WISDM.....	66
Figura 39: Arquitectura modelo CNN.....	69
Figura 40: Arquitectura modelo BiLSTM.....	70
Figura 41: Arquitectura del modelo híbrido.....	71
Figura 42: Representación de loss y accuracy para UCI HAR con modelo CNN-BiLSTM	76
Figura 43: Representación de los y accuracy para WISDM con modelo CNN-BiLSTM	76
Figura 44: Matriz de confusión CNN.....	77

Figura 45: Matriz de confusión LSTM.....	77
Figura 46: Matriz de confusión BiLSTM	77
Figura 47: Matriz de confusión CNN-BiLSTM	77
Figura 48: Matriz de confusión CNN.....	78
Figura 49: Matriz de confusión LSTM.....	78
Figura 50: Matriz de confusión BiLSTM	78
Figura 51: Matriz de confusión CNN-BiLSTM	78
Figura 52: Bloque y conexión residuales utilizada en el modelo.....	80
Figura 53: Arquitectura modelo residual con “bottleneck” planteado.....	81
Figura 54: Arquitectura modelo residual simple planteado.....	82
Figura 55: Representación loss y accuracy para WISDM con modelo residual con reducción de cuello de botella	83
Figura 56: Representación de loss y accuracy para WISDM con modelo residual simple	83

ÍNDICE DE TABLAS

Tabla 1: Comparativa de características en pulseras inteligentes.....	21
Tabla 2: Comparación datasets para el reconocimiento de actividades humanas	30
Tabla 3: Descripción de los datasets	65
Tabla 4: Cálculo F1-score de los modelos neuronales para cada dataset	73
Tabla 5: Métrica F1-score para las distintas actividades con UCI HAR dataset.....	73
Tabla 6: Métrica F1-score para las distintas actividades con WISDM dataset	74
Tabla 7: Métrica precision para las distintas actividades con UCI HAR dataset	75
Tabla 8: Exactitud de los modelos neuronales para cada dataset	75

RESUMEN

El presente proyecto busca utilizar técnicas de aprendizaje profundo para diseñar un modelo que pueda reconocer actividades humanas utilizando señales inerciales de una smartband. Mediante esta investigación se pretende integrar este modelo en un sistema de monitorización en el hogar para mejorar el bienestar y la seguridad de las personas mayores. Para ello se exploran diferentes modelos neuronales y se analizan sus resultados para seleccionar el enfoque más adecuado. Como idea complementaria, se busca no solo reconocer la actividad, sino también monitorizar los datos de salud de los usuarios para obtener más información relacionada con su bienestar.

Además se ha realizado un análisis del estado del arte de la tecnología actual, incluyendo las pulseras inteligentes disponibles en el mercado para la monitorización de la actividad humana, los conjuntos de datos HAR (Human Activity Recognition) y algunos de los modelos neuronales más utilizados con estos conjuntos de datos.

A través de este proyecto se espera brindar un apoyo a los cuidadores y a las personas mayores que tienen a cargo, que eligen vivir en la comodidad y familiaridad de sus propias casas, procurando su seguridad, bienestar y felicidad en cada etapa de sus vidas.

Palabras clave: aprendizaje profundo, red neuronal, Reconocimiento de Actividad Humana, smartband

ABSTRACT

This project aims to use deep learning techniques to design a model that can recognize human activities using inertial signals from a smart band. Through this research we aim to integrate this model into a home monitoring system to improve the well-being and safety of the elderly. For this purpose, different neural models are explored, and their results are analyzed to select the most appropriate approach. As a complementary idea, the aim is not only to recognize the activity, but also to monitor the health data of the users to obtain more information related to their well-being.

In addition, a state-of-the-art analysis of current technology has been carried out, including smart wristbands available in the market for human activity monitoring, HAR (Human Activity Recognition) datasets and some of the most used neural models with these datasets.

Through this project we hope to provide support to caregivers and older adults who choose to live in the comfort and familiarity of their own homes, ensuring their safety, well-being, and happiness at every stage of their lives.

Keywords: Deep Learning, neural network, Human Activity Recognition, smartband

CAPÍTULO 1: INTRODUCCIÓN

1.1 INTRODUCCIÓN

Gracias al auge en las tecnologías actuales, la esperanza de vida ha aumentado considerablemente en los últimos años. Debido a esto, el número de personas ancianas que viven solas en su casa que necesitan ser supervisadas continuamente ha crecido de forma exponencial. De hecho, en 2022 el porcentaje de población mayor de 65 años en nuestro país ha superado el 20%, algo que no había sucedido nunca antes en la historia de nuestro país. Si nos fijamos en las cifras de Castilla y León, la cifra asciende al 26.31%, por lo que en esta comunidad el envejecimiento de la población es un desafío urgente y crucial que requiere una atención prioritaria.

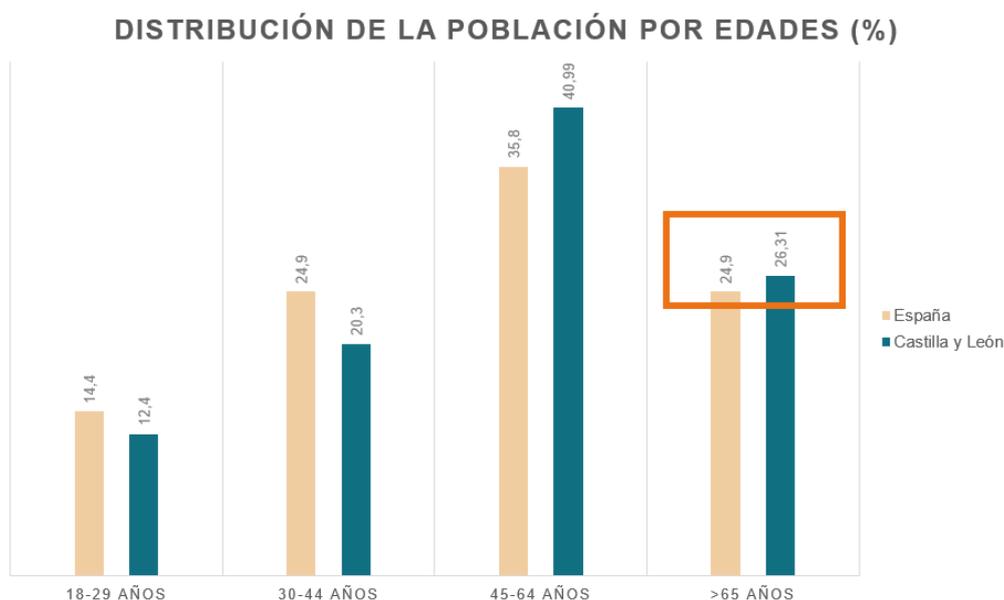


Figura 1: Distribución de la población por edades en porcentaje [1]

Según el Instituto Nacional de Estadística (INEA), se estima que en España 3 de cada 10 personas mayores de 75 años viven solas en su domicilio [1]. Como consecuencia, cada vez hay más líneas de investigación enfocadas a la monitorización y al seguimiento no invasivo de la salud y la actividad de las personas de tercera edad.

En los últimos años, una de las líneas de más populares ha sido el reconocimiento de la actividad humana o *Human Activity Recognition (HAR)*, que tiene como objetivo identificar actividades individuales como andar,

cocinar o sentarse. La investigación en HAR ha crecido recientemente debido a la necesidad de ayudar a las personas mayores y a otras personas con problemas cognitivos que necesitan una supervisión diaria.

Con la tecnología y los dispositivos inteligentes cada vez más presentes en nuestra vida diaria, la tecnología HAR se ha convertido en un área de investigación muy interesante para analizar los datos recogidos por estos dispositivos. Los sensores integrados en las pulseras inteligentes ayudan a registrar si el usuario está corriendo, andando e incluso nadando y es uno de los mayores atractivos para muchas personas que adquieren una pulsera inteligente. Además, como cada vez es más fácil y económico acceder a los datos de estos dispositivos, en los últimos años el número de *datasets* para el reconocimiento de actividades humanas ha aumentado considerablemente.

Varias investigaciones han demostrado la eficacia de sensores como el acelerómetro y el giroscopio para reconocer la actividad humana[2], [3]. Sin embargo, muchos de estos trabajos de investigación se centran únicamente en la exactitud como métrica para evaluar los resultados finales, sin tener en cuenta otras métricas necesarias para interpretar si esa exactitud es coherente o no. Además, muchas propuestas de modelos neuronales obtienen buenos resultados al entrenar durante largas épocas, lo que permite al modelo aprender patrones complejos en los datos de entrenamiento. Sin embargo, esto prolonga el tiempo de entrenamiento y puede haber un riesgo de sobreajuste en el modelo.[4]

En este estudio, se clasificarán actividades humanas diarias utilizando algoritmos de aprendizaje profundo. Los modelos utilizados serán redes neuronales convolucionales (CNN) y recurrentes (RNN), como el modelo Long-Short-Term-Memory (LSTM) y el modelo Bidireccional LSTM (BiLSTM) [5]. Para el entrenamiento del modelo, se emplearán dos conjuntos de datos públicos para el reconocimiento de actividades humanas: WISDM (Wireless Sensor Data Mining) y UCI-HAR. Estos conjuntos de datos contienen información proporcionada por acelerómetros y giroscopios.

El proyecto que se expone está enmarcado dentro del proyecto EIAROB, que busca crear un ecosistema de Inteligencia Ambiental para apoyar los cuidados de larga duración en el hogar mediante el uso de robots sociales [6]. Una de las propuestas consiste en utilizar dispositivos de adquisición de señales para controlar las actividades cotidianas de las personas mayores. El trabajo presentado será de gran utilidad para reconocer las actividades que realiza el usuario a lo largo del día utilizando las señales obtenidas por el dispositivo *wearable* utilizado.

Este documento se divide en siete partes fundamentales. En el CAPÍTULO 1 se hace una pequeña justificación del proyecto y se exponen los objetivos

principales; en el CAPÍTULO 2 se realiza una revisión bibliográfica sobre dispositivos inteligentes, redes neuronales, conjuntos de datos y técnicas utilizadas para el reconocimiento de actividades humanas; en el CAPITULO 3 se describen los diferentes procedimientos para la extracción de datos, la preparación de las bases de datos y la arquitectura de red a seguir; el CAPÍTULO 4 se plantea el desarrollo propuesto explicando la extracción de datos y el diseño del modelo neuronal; en el CAPÍTULO 5 se presentan los resultados obtenidos en el entrenamiento; en el CAPÍTULO 6 se exponen las conclusiones y se comenta cuál de los modelos ha obtenido un mejor rendimiento; finalmente en el CAPÍTULO 7 se plantean algunas alternativas a este proyecto que podrían desarrollarse en el futuro.

1.2 JUSTIFICACIÓN DEL PROYECTO

En un mundo cada vez más interconectado, la capacidad de capturar y analizar datos de manera eficiente se ha convertido en un factor clave para comprender y mejorar la vida cotidiana de las personas. El reconocimiento de actividades realizadas por los usuarios ha sido uno de los objetivos principales en el campo de las redes neuronales en los últimos años. El hecho de reconocer diferentes acciones supone un gran avance en diversos campos como en el mundo de la salud, seguridad o vigilancia. Gracias a los avances de la medicina en estos últimos años, la esperanza de vida ha aumentado de manera generalizada en todo el país. Este crecimiento en el número de personas ancianas supone un gran reto para la industria sanitaria, ya que cada vez son más las posibles personas ancianas dependientes que viven solas en sus casas. Debido a esto, la industria sanitaria recurre cada vez más a la tecnología para dar respuesta a estas necesidades.

Uno de los retos actuales con respecto a las personas mayores es el de mejorar la calidad de vida de aquellos que aún tienen cierta autonomía y viven en sus propias residencias para que puedan seguir viviendo de forma autónoma, al tiempo que reciben la asistencia de algunos servicios exteriores de monitorización y apoyo.

En este contexto, varias tecnologías basadas en sensores, como acelerómetros o sensores de vibración, han sido utilizadas para monitorizar las actividades diarias básicas de las personas de tercera edad.

Mediante el uso de técnicas de aprendizaje profundo y redes neuronales, se pretende desarrollar un enfoque innovador que permita identificar y clasificar las actividades realizadas por el usuario con alta precisión y eficacia. El objetivo final es obtener una visión más completa y detallada de las actividades diarias de las personas, lo que puede ser de gran utilidad en diversos ámbitos, como el diseño de sistemas de apoyo a los cuidados de larga duración en el hogar o

la creación de dispositivos portátiles inteligentes. Con este trabajo, se busca abrir nuevas posibilidades en el campo de la inteligencia ambiental y contribuir al desarrollo de soluciones que mejoren la calidad de vida de las personas.

1.3 MOTIVACIÓN PERSONAL

La elección de este tema como Trabajo de Fin de Grado está motivada por mi gran interés en las redes neuronales y su potencial para el futuro. Con la llegada del Chat GPT a nuestras vidas no pude evitar preguntarme la gran cantidad de aplicaciones que tienen las redes neuronales y el aprendizaje profundo que eran desconocidas para mí. Debido a esto, elegí este proyecto para conocer mejor esta herramienta tan útil para resolver problemas muy diferentes y complejos.

Mi motivación personal para llevar a cabo este proyecto también tiene relación con mi entorno familiar. He vivido una situación en la que tuvimos que cuidar de un familiar cercano, lo cual implicaba una vigilancia y seguimiento continuo para asegurarnos de que estuviera bien. No disponer de información detallada sobre sus actividades cotidianas podía generar frustración y preocupación, especialmente cuando no se contaba con un cuidador profesional que estuviera presente las 24 horas. La falta de visibilidad sobre las actividades de la persona limitaba la capacidad para garantizar su bienestar y responder de manera oportuna a sus necesidades.

Esta experiencia e interés personal me impulsó a elegir este proyecto y trabajar en el desarrollo de un sistema de monitorización de actividad basado en aprendizaje profundo. Reconocí la importancia de contar con herramientas tecnológicas que nos permitan obtener información precisa y detallada sobre las actividades diarias de las personas, especialmente aquellas que requieren cuidados especiales.

Realizar este proyecto me brindaría la oportunidad de aprender nuevos conceptos en un área que me motiva y aplicarlo a una situación de necesidad real, lo que considero una excelente oportunidad para mi crecimiento personal y profesional.

1.4 OBJETIVOS

En la realización de este proyecto se ha perseguido un objetivo principal: el diseño de un modelo de aprendizaje profundo que reconozca actividades humanas para su posterior uso en un sistema de monitorización en domicilio. Esta idea se engloba en el proyecto para crear un ecosistema inteligente que monitorice la actividad y el bienestar de las personas mayores. Además, como idea complementaria, en el presente trabajo se pretende no solo reconocer la actividad, si no también monitorizar los datos de salud de los usuarios para tener más información relacionada con su bienestar.

Teniendo en cuenta este enfoque, se han tenido en cuenta una serie de objetivos secundarios que es necesario cumplir para obtener los resultados finales:

- Diseñar un modelo neuronal de aprendizaje profundo para reconocer actividades cotidianas a partir de señales inerciales registradas por un dispositivo *wearable* de forma eficaz.
- Obtener datos de salud como frecuencia cardíaca o calidad de sueño para tener información sobre el bienestar del paciente.
- Obtener datos de los sensores inerciales del dispositivo *wearable* con el objetivo de reconocer las actividades que realiza el usuario mediante aprendizaje profundo.
- Realizar una búsqueda bibliográfica para comparar los resultados obtenidos con los de otras investigaciones que emplean bases de datos, dispositivos *wearables* y modelos neuronales similares o diferentes.

CAPÍTULO 2: ESTADO DEL ARTE

2.1 ESTUDIO DE MERCADO WEARABLES EN SALUD

Con el crecimiento en el mundo de los smartphones y los relojes inteligentes cada vez son más las marcas que apuestan por el reconocimiento de actividades a partir de los sensores integrados en sus dispositivos.

Un ejemplo muy conocido de ello es la marca Apple; el modelo de Apple Watch Series 4 y posteriores incorporan una función para la detección de caídas mientras que el usuario lleve puesto el reloj.

Otro ejemplo de ello es el modelo de reloj Versa de la marca Fitbit que con su función SmartTrack es capaz de reconocer ejercicios intensos automáticamente como ejercicio aeróbico, natación o bicicleta entre otros.

Ante la gran variedad de dispositivos *wearable* y los servicios que ofrecen, se ha realizado un estudio de mercado para escoger la pulsera más apropiada para nuestro proyecto. A la hora de elegir nuestra *smartband* tuvimos en cuenta algunos criterios básicos como el precio, los parámetros de salud adquiridos o la posibilidad de extraer los datos fácilmente, ya que a diferencia de otras marcas como Huawei no es necesario pedir permiso y esperar la aprobación de la empresa para usar la API que proporcionan u otras herramientas.

SMARTBANDS XIAOMI Y HUAMI

En primer lugar, uno de los grandes productores de pulseras inteligentes es la familia Xiaomi.

En el pasado, Xiaomi y Huami (ahora conocida como Zepp Health) trabajaban juntos pero debido a la gran popularidad de las pulseras de Huami (modelo Amazfit) hizo que Huami pasara a trabajar en solitario. De esta separación nacieron dos apps diferentes: Zepp Life (antes conocida como Mi Fit) y Mi Fitness (antes conocida como Xiaomi Wear). Debido a esta relación entre las marcas, todas las pulseras Amazfit y Mi Band pueden conectarse a las apps Zepp Life y Mi Fitness.

Tanto las MiBand de Xiaomi como las pulseras Huami Amazfit de Zepp Health se pueden conectar a la aplicación Zepp Life para registrar los datos de salud. Es necesario crear una cuenta e iniciar sesión en la app del móvil. Una vez que se haya iniciado sesión los datos de la pulsera se sincronizan con la app y se pueden descargar en cualquier momento desde la nube. Se enviará un mensaje al correo electrónico para descargar los datos en formato Excel (frecuencia cardiaca, paso, distancia y calorías a lo largo del día).

Igualmente, las pulseras Xiaomi y Huami, entre otras marcas, son compatibles con la aplicación Gadgetbridge de código abierto, que tiene la opción de exportar los datos de forma automática cada hora. Al exportar los datos obtenemos: fecha, batería, frecuencia cardiaca, número de pasos, intensidad de actividad, etiquetado de la actividad (las etiquetas las ha propuesto la comunidad de usuarios).

La empresa Xiaomi no facilita ningún tipo de SDK ni API abierta para poder trabajar de forma libre y desarrollar aplicaciones de terceros. A pesar de esto, la comunidad de Github ha desarrollado posibles APIs para extraer los datos utilizando el usuario y contraseña de la cuenta Zepp Life.

A continuación se muestra una lista de modelos de la familia Xiaomi que se han considerado a la hora de elegir dispositivo *wearable* junto con algunas especificaciones técnicas:

AMAZFIT BAND 5

Smartband de la marca Zepp Health. Cuenta con la tecnología bluetooth 5.0 BLE. Algo que le diferencia de otras pulseras es la incorporación de Amazon Alexa, de forma que el usuario puede emplear el micrófono que incorpora la propia pulsera para hablar con el asistente, aunque esta característica aún tiene mucho que mejorar. Se trata de un modelo del año 2020 pero sigue siendo muy popular hoy en día.



Figura 2: Smartband Amazfit band 5

- Pasos, distancia, frecuencia cardiaca durante todo el día
- Saturación oxígeno en sangre
- Monitorización del sueño y el estrés
- Acelerómetro y giroscopio de 3 ejes
- Micrófono integrado, pero no altavoz
- Duración batería 15 días
- Tiempo de carga 2 horas
- Recordatorios y eventos
- Pantalla 1.1"
- Amazon Alexa integrado
- Compatible con dispositivos Android 5.0 o IOS 10.0
- Precio 21,90 € en página de Amazfit

AMAZFIT BAND 7

Smartband de la marca Zepp Health. Tecnología bluetooth 5.2. Amazon Alexa integrado. Es el siguiente modelo de Amazfit band después de Amazfit band 5 y salió al mercado en el año 2022. Cuenta con Sistema operativo Zepp OS para pulseras inteligentes. Gracias a esta pulsera se puede monitorizar la frecuencia cardiaca, saturación de oxígeno en sangre y nivel de estrés con un solo toque y obtener el resultado en 45 segundos.



Figura 3: Smartband Amazfit Band 7

Además, se pueden instalar miniaplicaciones adicionales desde la aplicación de Zepp como calculadora o linterna, o el usuario puede crear su propia miniaplicación.

- Pasos, distancia, frecuencia cardiaca durante todo el día
- Saturación oxígeno en sangre
- Monitorización del sueño y el estrés
- Acelerómetro y giroscopio de 3 ejes
- Micrófono integrado, pero no altavoz
- Duración batería 18 días
- Tiempo de carga 2 horas
- Recordatorios y eventos
- Pantalla 1.47''
- Amazon Alexa integrado
- Compatible con dispositivos Android 7.0 o IOS 12.0
- Precio 49.90€ en la página Amazfit

MI BAND 4

Pulsera inteligente de la marca Xiaomi con conectividad inalámbrica de tipo bluetooth 5.0 y BLE. Se trata de un modelo sencillo con una gran autonomía que ofrece al usuario información básica sobre su bienestar.

- Pasos, distancia, frecuencia cardíaca durante todo el día
- Monitorización del sueño
- Acelerómetro y giroscopio de 3 ejes
- Duración batería 20 días
- Tiempo de carga 2 horas
- Recordatorios y eventos
- Pantalla 0.95''
- Peso 22.1 gramos
- Precio 34.99€ en la página de Xiaomi



Figura 4: Smartband Mi Band 4

XIAOMI MI SMARTBAND 6

Smartband de la marca Xiaomi con *bluetooth low energy* 5.0. Modelo año 2021. Se han aumentado las dimensiones de la pantalla AMOLED respecto el modelo anterior. También ha incorporado la función de medida de la SpO2.

- Pasos, distancia, frecuencia cardíaca durante todo el día
- Saturación oxígeno en sangre
- Monitorización del sueño y el estrés
- Acelerómetro y giroscopio de 3 ejes
- Duración batería 14 días
- Tiempo de carga 2 horas
- Recordatorios y eventos
- Pantalla 1.56''
- Peso 12.8 gramos
- Precio 36.99€ en la página de Xiaomi



Figura 5: Smartband Xiaomi Mi Smart Band 6

XIAOMI MI SMART BAND 7

Smartband de la marca Xiaomi con bluetooth low energy 5.2 y NFC. Modelo año 2022. Tiene los mismos sensores básicos que el modelo Mi Band 6, pero tiene una autonomía algo mayor, una pantalla más grande y función NFC.

- Pasos, distancia, frecuencia cardiaca durante todo el día
- Saturación oxígeno en sangre
- Monitorización del sueño y el estrés
- Acelerómetro y giroscopio de 3 ejes
- Duración batería 15 días
- Tiempo de carga 2 horas
- Recordatorios y eventos
- Pantalla 1.62"
- Compatible con dispositivos Android 6.0 o IOS 10.0
- Peso 13.5 gramos
- Precio 49.99€ en la página de Xiaomi



Figura 6: Smartband Xiaomi Mi Smart Band 7

SMARTBANDS FITIBIT

Las pulseras de la marca Fitbit, de la compañía Google, tienen muy buenas reseñas en cuanto a la monitorización de los datos de actividad física. También es importante destacar que el precio es mayor en comparación con otras marcas y la duración de la batería es menor (sus modelos tienen una duración de batería entre 7 y 10 días). Los modelos de Fitbit tienen una precisión muy elevada a la hora de medir parámetros como la frecuencia cardiaca entre otros. También cabe destacar que los últimos modelos de Fitbit (como el Charge 5 o el Fitbit 3) miden la frecuencia cardiaca con un sensor de fotopletismografía, lo que permite detectar una posible fibrilación auricular. A pesar de esto en la propia app de Fitbit se indica que esta función no está diseñada para que la usen personas con antecedentes o diagnóstico de fibrilación auricular.

Fitbit proporciona "Device APIs" para acceder a datos técnicos de la pulsera (acelerómetro, giroscopio...), que solo son accesibles por aplicaciones que se ejecutan en dispositivos Fitbit, y "Web APIs" para acceder a los datos de salud entre otros.

FITIBIT CHARGE 5

Es una de las pulseras más completas respecto a funcionalidades, pero tiene una autonomía de tan solo 7 días. Fitbit ofrece un servicio Fitbit premium de pago que proporciona funcionalidades extra como herramientas del sueño, entrenamientos y ejercicios para reducir el estrés; a pesar de esto los datos básicos como pasos o frecuencia cardiaca se pueden consultar a través de la app Fitbit sin suscripción.



Figura 7: Smartband Fitbit Charge 5

El modelo Fitbit Charge 5 es la pulsera más grande que Fitbit ha hecho hasta la fecha, aunque tan solo pese 15 gramos. Cuenta con herramientas tan avanzadas como el registro de ECG mediante la pulsera o la detección de una posible fibrilación auricular.

- Pasos, distancia, frecuencia cardiaca durante todo el día y notificación si está alta o baja
- Saturación oxígeno en sangre
- ECG
- Monitorización del sueño y el estrés mediante sensores EDA (actividad electro dérmica) de gran precisión
- Acelerómetro y giroscopio de 3 ejes
- Posibilidad de que el usuario registre su nivel de glucosa en sangre (no mediante sensor, escribiéndolo en la app de forma manual)
- Detección de posible fibrilación auricular
- Duración batería 7 días
- Tiempo de carga 2 horas
- Recordatorios y eventos
- Pantalla 1.04''
- GPS integrado
- Compatible con dispositivos Android 8.0 o IOS 12.2
- Peso 15 gramos
- Precio 109.95€ en la página de Fitbit

FITBIT INSPIRE 3

Smartband que salió al mercado en septiembre de 2022. No cuenta con tantas funciones como el modelo Charge 5; por ejemplo, no tiene opción de mostrar ECG ni tampoco GPS integrado. A pesar de esto se trata de una pulsera que registra datos de actividad física y bienestar con gran precisión.



Figura 8: Smartband Fitbit Inspire 3

- Pasos, distancia, frecuencia cardiaca durante todo el día y notificación si está alta o baja
- Saturación oxígeno en sangre
- Monitorización del sueño y el estrés
- Acelerómetro y giroscopio de 3 ejes
- Posibilidad de que el usuario registre su nivel de glucosa en sangre (no mediante sensor, escribiéndolo en la app de forma manual)
- Posible detección de fibrilación auricular
- Duración batería 10 días
- Tiempo de carga 2 horas
- Recordatorios y eventos
- Pantalla 1.54"
- Compatible con dispositivos Android 10 o IOS 14
- Peso 15 gramos
- Precio 99.95€ en la página de Fitbit

SMARTBANDS HUAWEI

Las pulseras de Huawei destacan por una interfaz sencilla y una monitorización de la actividad física del paciente eficaz. También presumen de tener una batería de larga duración y de carga rápida (con 10 minutos de carga la *smartband* puede tener una autonomía de 3 días). La desventaja de Huawei es que no proporciona tantas herramientas para la adquisición de los datos de forma externa a la app oficial.

HONOR BAND 5

Smartband de la marca Honor, compañía Huawei. Es una pulsera de actividad económica y sencilla que hace un registro eficaz de la salud y el bienestar del usuario. En general se trata de un modelo sencillo e intuitivo capaz de monitorizar las actividades físicas básicas.

Esta pulsera ha tenido mejores reseñas que el modelo siguiente de Honor debido a su eficacia y su calidad precio.

A diferencia de Xiaomi, hay poco contenido relacionado con la extracción y el procesamiento de datos de esta pulsera por parte de la comunidad.



Figura 9: Smartband Honor Band 5

- Pasos, distancia, monitorización continua de frecuencia cardiaca.
- Saturación oxígeno en sangre
- Monitorización del sueño y el estrés
- Acelerómetro y giroscopio de 3 ejes
- Duración batería 14 días
- Tiempo de carga 2 horas
- Recordatorios y eventos
- Pantalla 1.47"
- Compatible con dispositivos Android 5.0 o IOS 9.0
- Bluetooth 5.0
- Peso 18 gramos
- Precio 38.95€ en la página de Amazon

HUAWEI BAND 7

Último modelo de *smartband* de la marca Huawei que salió al mercado en junio de 2022. Además de las funciones básicas como medir el pulso o el oxígeno en sangre puede detectar fibrilaciones y arritmias.

- Pasos, distancia, monitorización continua de frecuencia cardiaca, registro de sueño.
- Saturación oxígeno en sangre
- Monitorización del sueño y el estrés



- Acelerómetro y giroscopio
- Duración batería 14 días
- Tiempo de carga 2 horas
- Recordatorios y eventos
- Pantalla 1.47"
- Compatible con dispositivos Android 6.0 o IOS 9.0
- Bluetooth 5.0 BLE
- Peso 16 gramos
- Precio 49€ en la página de Huawei

Figura 10: Smartband Huawei Band 7

A continuación, se muestra una tabla comparativa con algunos de los modelos que se tuvieron en cuenta:

MODELO	AUTONOMÍA	ECG	DETECCIÓN CAÍDAS	DETECCIÓN ARRITMIAS	PANTALLA	PRECIO
Amazfit Band 7	18 días	No	No	No	1.47"	49.90
Amazfit Bip 3	14 días	No	No	No	1.69"	48.31
Xiaomi Smart Band 6	14 días	No	No	No	1.56"	34.99
Xiaomi Smart Band 7	15 días	No	No	No	1.62"	49.99
Fitbit Charge 5	7 días	Si	No	No	1.04"	109.95
Fitbit Inspire 3	10 días	No	No	Si	1.54"	99.95
Honor Band 5	14 días	No	No	No	1.47"	38.95
Huawei Band 7	14 días	No	No	Si	1.47"	49.90
AppleWatch SE Series 4,5,6...	24 horas	Si	Si	Si	1.6"	515.99
Samsung Galaxy Watch 5 Pro	48 horas	Si	Si	Si	1.4"	236.99

Tabla 1: Comparativa de características en pulseras inteligentes

Como hemos comprobado, todas las pulseras mencionadas hasta ahora cuentan con sensores internos como acelerómetros y giroscopios que están continuamente generando una cantidad enorme de datos. Los datos obtenidos se emplean para identificar patrones de comportamiento humano que se procesan con *Machine Learning* para distinguir la actividad que realiza el usuario.

2.2 ESTUDIO DE APPS DE SALUD Y ACTIVIDAD FÍSICA

Las diferentes compañías fabricantes de pulseras han desarrollado aplicaciones oficiales para registrar los datos de salud de su dispositivo. Igualmente, podemos encontrar apps de terceros que permiten acceder a los datos de uno o varios modelos de pulsera inteligente y que ofrecen ventajas en comparación con las aplicaciones oficiales. A continuación, se presentan algunas de las aplicaciones compatibles con los modelos de pulseras mencionados anteriormente:

- **Huawei Health:** Huawei Health es una aplicación de salud y estado físico desarrollada por Huawei Technologies para dispositivos Huawei y Honor. La aplicación se utiliza en combinación con los dispositivos portátiles de la marca Huawei, como los relojes inteligentes y las pulseras de actividad, para realizar un seguimiento y registrar la actividad física, el sueño, el ritmo cardíaco y otros datos de salud. Esta se puede sincronizar con otras aplicaciones y servicios de salud y fitness populares, como MyFitnessPal. Presenta una herramienta llamada Health Kit con la que se puede acceder a los datos de salud de la pulsera inteligente.

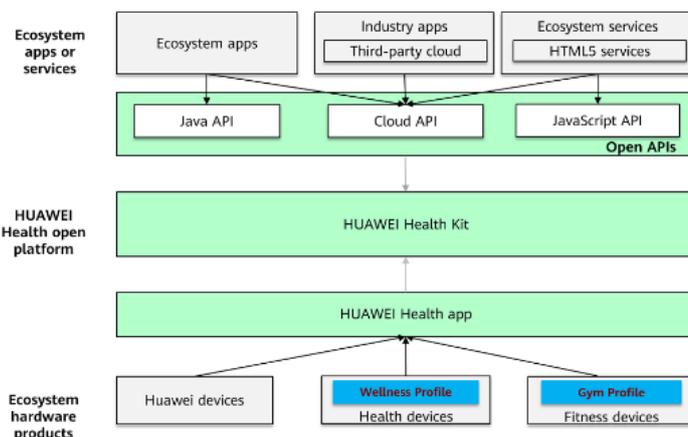
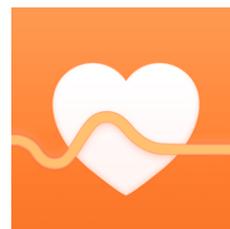
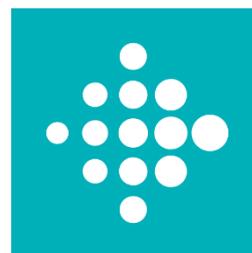


Figura 11: Esquema arquitectura Health Kit y servicios Huawei [7]

- **Fitbit App:** Fitbit ofrece una aplicación llamada "Fitbit App" (también conocida como "Fitbit Mobile App" o "Fitbit Tracker App") que se utiliza en conjunto con los dispositivos Fitbit. La aplicación Fitbit permite realizar un seguimiento de la actividad diaria, como pasos, distancia recorrida, calorías quemadas y escaleras subidas, al igual



que otros datos de interés como la calidad de sueño o los entrenamientos realizados. Además Fitbit App proporciona una variedad de gráficos y estadísticas que te ayudan a comprender tu actividad, es una app enfocada principalmente a las personas que hacen deporte y desean registrar su actividad de forma precisa. Según el fabricante, los datos se sincronizan automáticamente desde tu dispositivo Fitbit a la aplicación para que puedas ver tu progreso y establecer objetivos de actividad.

- Zepp Life** es una aplicación móvil desarrollada por Huami, empresa de la familia Amazfit, que se utiliza en conjunto con los dispositivos portátiles de Amazfit y Zepp para el seguimiento de la actividad física y el cuidado de la salud. Las pulseras Mi Band son fabricadas por la empresa Huami y comercializadas por la empresa Xiaomi; al ser del mismo fabricante las pulseras Mi Band mencionadas anteriormente son compatibles con esta app. La aplicación Zepp Life ofrece una amplia gama de características y funciones entre las que destacamos la monitorización del ritmo cardiaco o el registro de ejercicio y entrenamiento en tiempo real. Zepp Life se conecta con los dispositivos Amazfit y Zepp a través de Bluetooth para sincronizar automáticamente los datos de actividad y salud. Estos datos se pueden extraer desde la página de Xiaomi utilizando la cuenta oficial; el resultado es un archivo .zip con varias carpetas que recogen información del usuario:



ACTIVITY	14/02/2023 21:32	Carpeta de archivos
ACTIVITY_MINUTE	14/02/2023 21:32	Carpeta de archivos
ACTIVITY_STAGE	14/02/2023 21:32	Carpeta de archivos
BODY	14/02/2023 21:32	Carpeta de archivos
HEARTRATE	14/02/2023 21:32	Carpeta de archivos
HEARTRATE_AUTO	14/02/2023 21:32	Carpeta de archivos
SLEEP	14/02/2023 21:32	Carpeta de archivos
SPORT	14/02/2023 21:32	Carpeta de archivos
USER	14/02/2023 21:32	Carpeta de archivos

Figura 12: Carpetas extraídas del archivo .zip que recogen información de la app Zepp Life

La aplicación también se puede conectar con otros servicios y aplicaciones de salud y fitness, como Strava y Google Fit, para ofrecerte una visión más completa de tu estado físico.

- **Mi Fitness:** antes conocida como Xiaomi Wear, se trata de la app oficial para las pulseras inteligentes de Xiaomi y Amazfit de modelo Xiaomi Smart Band 6 o superior. La app nos permite obtener una serie de características básicas como frecuencia cardiaca, registro de entrenamientos... y otras más específicas como calidad de sueño o nivel de estrés que dependerá de la capacidad que tiene la pulsera para registrar estos parámetros.
- **Gadgetbridge:** se trata de una app móvil de código abierto que es compatible con un gran número de modelos de monitoreo de salud (Pebble, Amazfit, Mi band, HPplus...). Gadgetbridge proporciona datos como el número de pasos, frecuencia cardiaca, tipo de actividad o datos relacionados con el sueño exportándolos en un archivo SQLite en la memoria del teléfono. La información en el archivo .sqlite se organiza en diferentes tablas, una por cada modelo de pulsera compatible, y cada una de estas tablas proporciona información como frecuencia cardiaca, pasos, calidad de sueño... que varía de acuerdo con las funciones que tenga la pulsera inteligente vinculada. Una ventaja de la aplicación Gadgetbridge es que en este fichero ofrece una etiqueta de la actividad que realiza el usuario cada minuto, por lo que podemos obtener una aproximación de las acciones que realiza la persona de forma automática. Otra de las ventajas de esta app es que no es necesario registrarse y los datos no se comparten en la nube (tan solo se guardan a nivel local) por lo que el usuario no tendrá que compartir sus datos personales con las grandes empresas tecnológicas.



2.3 ANTECEDENTES DEL APRENDIZAJE PROFUNDO

El aprendizaje profundo o *Deep Learning* es un subgénero del aprendizaje automático o *Machine Learning* que emplea redes neuronales artificiales para diseñar y resolver problemas complejos. La idea fundamental del aprendizaje profundo es clasificar o identificar distintos datos a partir de grandes conjuntos de datos etiquetados o sin etiquetar. Se trata de una técnica dentro de la Inteligencia Artificial que utiliza redes neuronales para resolver tareas complejas de forma autónoma.

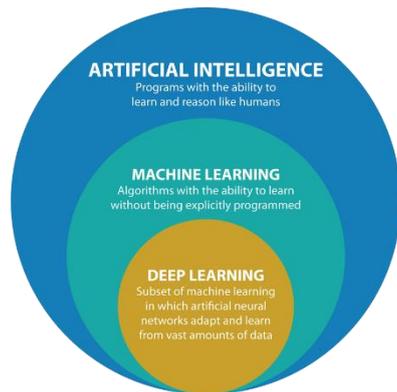


Figura 13: Esquema de IA, ML y DL [1]

El *Deep Learning* tiene sus orígenes en el año 1943, cuando los científicos Walter Pitts y Warren McCulloch crearon un modelo computacional basado en las redes neuronales propias del cerebro humano. Es aquí cuando se creó la neurona artificial, un modelo simple en el que la salida binaria se activaba cuando un determinado número de entradas, también binarias, estaban activas.

Más tarde en 1957 Frank Rosenblatt definió la unidad neuronal básica a día de hoy, el perceptrón. Aquí las entradas y salidas pasaron a ser números (en vez de variar en un rango de 0-1) y cada entrada tenía un peso asociado que determina la importancia que se le da a cada entrada. Desde el nacimiento del perceptrón las redes neuronales fueron evolucionando con el paso de los años para afinar la eficacia de las redes multicapa. Con el paso del tiempo numerosos investigadores, entre los que se destaca Geoffrey Hinton, demostraron el potencial que esta herramienta tendría en el futuro al realizar estudios relacionados con el reconocimiento de formas y la predicción de palabras.

Más tarde en la década de los 2000 se realizaron grandes avances en el procesamiento de datos y se desarrollaron las unidades de procesamiento gráfico, lo que supuso un gran progreso para el campo. Finalmente, fue en 2006 cuando Hinton acuñó a este conjunto de técnicas el término de *Deep Learning*.

Con el desarrollo de las nuevas tecnologías y el crecimiento del *Big Data*, las redes neuronales han evolucionado hasta convertirse en modelos eficaces y complejos. Esto ha permitido que en numerosas ocasiones estos modelos de aprendizaje profundo hayan sido capaces de aprender y responder ante situaciones complejas y desconocidas de forma más rápida que la mente humana. Las redes neuronales han sido utilizadas para procesar grandes cantidades de datos y extraer información relevante. Es este aprendizaje

profundo el que ha hecho posible que las máquinas aprendan y realicen tareas complejas de manera autónoma como la traducción automática o el reconocimiento de voz entre otras aplicaciones.

En los últimos años el aprendizaje profundo ha tenido un papel importante en el desarrollo de modelos basados en la experiencia a partir de datos preprocesados. Esto ha resultado de gran utilidad a la hora de automatizar procesos de toma de decisiones o de reconocer patrones para realizar predicciones. Gracias a esto se han desarrollado algoritmos de reconocimiento de imágenes, procesado de lenguaje natural o incluso otras aplicaciones como el reconocimiento de actividades a partir de señales.

En este último campo podemos destacar la aproximación HAR (*Human Activity Recognition*) que se basa en el reconocimiento de diferentes actividades humanas a partir de datos de entrada como imágenes, audio, señales inerciales...

Es importante destacar que, aunque el *Deep Learning* ha avanzado significativamente en los últimos años, sigue siendo una técnica dentro de la Inteligencia Artificial. A pesar de que las máquinas son capaces de procesar grandes cantidades de datos más rápidamente que los humanos, aún hay tareas que requieren la intervención humana y el razonamiento crítico.

2.4 ESTUDIOS PREVIOS SOBRE HAR

En los últimos años, las técnicas de reconocimiento de actividad humana han experimentado un gran auge. Hoy en día, la mayoría de las pulseras inteligentes presentan sensores inerciales cuyos datos pueden ser recogidos y empleados para reconocer las acciones que realice el usuario. Como consecuencia, cada vez son más los estudios que aplican técnicas de clasificación para reconocer una actividad a partir de los datos de sensores inerciales.

Varios estudios han utilizado los datos capturados por estos sensores para reconocer movimientos o actividades físicas mediante el uso de redes neuronales. Por ejemplo, en el estudio realizado por Xia K. [8], se entrena una red utilizando numerosos conjuntos de datos con señales de acelerómetros y otros sensores para clasificar actividades humanas previamente etiquetadas. Otros estudios, por ejemplo [2], integran datos etiquetados y no etiquetados para evaluar y predecir modelos mediante una clasificación semisupervisada, logrando una precisión de hasta el 95% en la base de datos UCI-HAR.

Además, existen numerosas investigaciones centradas en encontrar la arquitectura de red óptima para un conjunto de datos específico. Por ejemplo,

en su estudio [9], Agarwal propone un modelo combinado de red LSTM-CNN para el reconocimiento de actividad humana, utilizando señales de acelerómetro de un teléfono móvil para identificar 6 actividades realizadas por 29 sujetos. Otros autores optan por utilizar redes convolucionales de 1 dimensión (Conv1D) para analizar los datos de estos sensores, como se muestra en el trabajo [3]. A diferencia de las redes convolucionales de 2 dimensiones, que desplazan filtros a lo largo de la altura y el ancho de una imagen, los filtros en las redes Conv1D se mueven a lo largo de la dimensión de una secuencia (series de tiempo, señales de audio, etc.) para extraer características relevantes.

También se han implementado muchos métodos de clasificación utilizando algoritmos de aprendizaje automático como Random Forest, k-vecinos más cercanos, Support Vector Machines o Naive Bayes, como se menciona en los trabajos [10], [11]. Sin embargo, muchos autores prefieren aplicar técnicas de aprendizaje profundo porque no requieren una extracción previa de características, a diferencia de las técnicas de aprendizaje automático tradicionales ([12]). En particular, estudios recientes de HAR han revelado ciertos problemas asociados con las técnicas convencionales de *Machine Learning*. La principal limitación que se aborda es la elección de las características de forma manual, ya que esta selección es dependiente de las habilidades y el conocimiento de la persona que toma las decisiones.



Figura 14: Esquema general HAR utilizando Machine Learning

Después de realizar una revisión exhaustiva de la literatura existente, se ha comprobado que las redes neuronales, como las *Convolutional Neural Networks* (CNN) y las *Recurrent Neural Networks* (RNN), han demostrado buenos resultados en la clasificación de actividades humanas en conjuntos de datos específicos ([8], [13], [14]). Estos estudios han destacado la eficacia de

estas arquitecturas de redes neuronales en la tarea de reconocimiento de actividades. En la Figura 15 observamos cómo los modelos *Deep Learning* eliminan la necesidad de extraer las características para reconocer actividades humanas.



Figura 15: Esquema general HAR utilizando Deep Learning

No obstante, a pesar de la abundante cantidad de artículos publicados, todavía no se ha determinado de manera definitiva cuál es la mejor arquitectura de red para el reconocimiento de actividades humanas ([4], [15]). Además, la mayoría de los modelos propuestos tienen estructuras de red bastante complejas y requieren un alto número de parámetros, lo que implica un costo computacional elevado. Esto puede plantear problemas a la hora de realizar clasificaciones en tiempo real.

2.5 BASES DE DATOS HAR

A partir de los datos extraídos de pulseras y teléfonos inteligentes, se han podido crear *datasets* destinados al reconocimiento de actividades humanas. A continuación, se mencionarán algunas de las bases de datos HAR más conocidas:

Existen varias bases de datos reconocidas en el campo del reconocimiento de actividades humanas. A continuación, describiré algunas de las más conocidas:

1. UCI-HAR (University of California, Irvine - Human Activity Recognition): Esta base de datos (Anguita et al., 2013) recopila datos de acelerómetros y giroscopios de 30 individuos. Se obtienen registros de actividades como caminar, subir escaleras, bajar escaleras, sentarse, estar de pie y tumbarse.

Los datos se adquieren utilizando un teléfono inteligente Samsung Galaxy S II que se coloca en la cintura de los participantes. Se registran la aceleración lineal en 3 ejes y la velocidad angular en 3 ejes a una frecuencia de muestreo de 50Hz. Los datos del *dataset* están repartidos en diferentes archivos, de forma que los datos del conjunto de test se obtienen de un archivo que contiene información de 9 sujetos, y los de entrenamiento de un fichero que ofrece información sobre 21 sujetos diferentes a los anteriores. Con esto nos aseguramos de que el conjunto de test prueba datos desconocidos que provengan de personas distintas a las del entrenamiento.

2. WISDM (*Wireless Sensor Data Mining*): Esta base de datos (Kwapisz et al., 2011) recoge datos de acelerómetros de smartphones utilizados por 51 individuos. Los datos abarcan actividades como caminar, correr, subir escaleras, bajar escaleras y estar quieto. Se registran la aceleración lineal en 3 ejes a una frecuencia de muestreo de 20Hz.

3. OPPORTUNITY: Esta base de datos (Chavarriaga et al., 2013) recopila datos de múltiples sensores, incluyendo acelerómetros, giroscopios y magnetómetros, de 4 sujetos. Los datos se adquieren mientras los sujetos realizan actividades diarias en un entorno de vida asistida. Las actividades incluyen caminar, sentarse, levantar objetos, entre otras. La frecuencia de muestreo varía dependiendo del sensor y oscila entre 30Hz y 100Hz.

4. PAMAP (*Physical Activity Monitoring for Aging People*): Esta base de datos (Reiss et al., 2012) se centra en actividades físicas realizadas por personas mayores. Se utilizan sensores inerciales, como acelerómetros, giroscopios y magnetómetros, así como sensores fisiológicos, como un monitor de frecuencia cardíaca. Los datos se adquieren de 9 sujetos realizando una variedad de actividades físicas, como caminar, subir escaleras, ciclismo y tareas domésticas. La frecuencia de muestreo varía entre 20Hz y 100Hz.

Además de estas bases de datos, se valoraron otros conjuntos de datos menos conocidos por la comunidad. Se realizó una búsqueda avanzada en Google Scholar para comparar el número de artículos que citaban la página oficial de cada *dataset*. Se acotó la búsqueda especificando que los artículos hayan sido publicados en los últimos 10 años.

<i>Dataset</i>	<i>Sensor</i>	<i>Nº clases</i>	<i>Frecuencia muestreo</i>	<i>Nº citas</i>
UCI HAR	9D IMU	6	50 Hz	230
WISDM	3D Accel.	6	20 Hz	50
OPPORTUNITY	9D IMU	16	50 Hz	26
MHEALTH	9D IMU, ECG	12	50 Hz	23
PAMP2	9D IMU, Heart rate	18	100 Hz	39
Daphnet	3D Accel.	2	64 Hz	109
HHAR	3D Accel.	6	50-200Hz	2
SDHAR-HOME	9D IMU	18	-	1

Tabla 2: Comparación datasets para el reconocimiento de actividades humanas

Cada una de estas bases de datos proporciona un conjunto único de datos para el reconocimiento de actividades humanas, con diferentes frecuencias de muestreo, actividades registradas, número de sujetos participantes y dispositivos utilizados para la recopilación de datos. Estas bases de datos han sido ampliamente utilizadas en la investigación y desarrollo de algoritmos de reconocimiento de actividades.

CAPITULO 3: MARCO TEÓRICO

3.1 SERIES TEMPORALES Y MONITOREO DE SALUD

A lo largo del presente documento se trabajará con datos secuenciales, que se presentan en un orden específico y tienen una dependencia temporal entre ellos. Las series temporales son un tipo de datos secuenciales en los que cada punto de datos está asociado con un momento en el tiempo. Estos datos se utilizan comúnmente en una amplia variedad de aplicaciones, desde la predicción del clima hasta el análisis de datos de ventas y marketing. Igualmente, en la gran mayoría de smartphones y pulseras inteligentes actuales encontramos este tipo de datos recogido por sensores como los acelerómetros o los giroscopios.

Los sensores de tipo acelerómetro son responsables de detectar movimiento lineal y fuerzas gravitacionales midiendo la aceleración en 3 ejes (x, y, z) mientras que los giroscopios miden el grado de rotación (roll, yaw, pitch). En los teléfonos inteligentes solemos encontrar un dispositivo que integra ambas mediciones llamado IMU (Unidad de medición Inercial) de 6 ejes (3 ejes del acelerómetro y 3 del giroscopio). Las señales triaxiales de acelerómetros y giroscopios se utilizan comúnmente en aplicaciones de monitoreo de la salud, como en las pulseras de actividad, donde se utilizan para medir el movimiento del cuerpo y calcular el número de pasos dados, las calorías quemadas, entre otros.

Además de estos sensores, en la mayoría de las pulseras inteligentes actuales encontramos otros sensores específicos que recogen varios parámetros secuenciales relacionados con la salud:

- **Frecuencia cardiaca:** la amplia mayoría de las *smartband* actuales emplean sensores de fotoplestimografía para medir la frecuencia cardiaca. Este sensor mide variaciones del volumen de la sangre en los capilares de la muñeca para determinar el pulso cardiaco.
- **Registro de sueño:** se suelen emplear acelerómetros que miden el movimiento mientras estamos dormidos. Con esto se puede determinar durante cuánto tiempo ha dormido el usuario y cómo ha sido su calidad de sueño.
- **Seguimiento del estrés:** para definir este aspecto se utiliza la variabilidad de la frecuencia cardiaca (*heart rate variability* o HRV) que es adquirida mediante un sensor de fotoplestimografía. La HRV es una medida de la variación de tiempo entre dos latidos sucesivos (intervalo R-R) y puede estar influenciada por factores como una patología o estrés.

- **Registro de pasos:** se utiliza un acelerómetro para determinar cuántos pasos se han realizado a lo largo del día. Este conteo de pasos se realiza en momentos específicos del día y los datos obtenidos pueden ser organizados para analizar cambios en el número de pasos a lo largo del tiempo y trabajar con ellos como un dato secuencial.
- **Registro de ejercicio:** aquí suelen intervenir diferentes sensores incluyendo el acelerómetro o el sensor de frecuencia cardiaca, para realizar un seguimiento de ejercicios como correr, nadar o montar en bicicleta. Los sensores miden el movimiento del usuario y la frecuencia cardiaca proporciona datos sobre la duración, distancia o calorías quemadas durante el ejercicio.

En el ámbito de las señales biomédicas, los datos secuenciales y las series temporales son de gran importancia para el diagnóstico y seguimiento de enfermedades. Por ejemplo, en el análisis de la actividad cardíaca, se pueden medir los cambios en la señal eléctrica del corazón a lo largo del tiempo para determinar si hay alguna irregularidad en el ritmo cardíaco.

La adquisición de estos parámetros mediante sensores integrados en móviles y pulseras inteligentes ha supuesto un gran avance en el ámbito de la salud. No solo se consigue monitorizar la salud del usuario de forma continua, sino que además se consigue de forma no intrusiva, ya que no es necesario la integración de cámaras que invadan la privacidad de las personas. Esto puede ser de vital importancia especialmente para las personas de tercera edad, ya que muchas de ellas viven solas y necesitan alguna medida control sobre su estado de salud.

3.2 REDES NEURONALES PARA EL APRENDIZAJE PROFUNDO

3.2.1 FUNDAMENTOS APRENDIZAJE PROFUNDO

Como se ha comentado en apartados anteriores, los algoritmos de aprendizaje profundo son capaces de aprender características más complejas a medida que se va avanzando por las capas, lo que les permite modelar patrones complejos en datos de entrada de alta dimensionalidad.

En comparación con otros métodos de clasificación tradicionales, como los árboles de decisión y el algoritmo K-Nearest Neighbor (KNN), las redes neuronales profundas presentan varias ventajas, entre las que podemos destacar su capacidad de aprender representaciones de características sin

necesidad de una ingeniería de características manual o la posibilidad de trabajar con grandes conjuntos de datos con alta dimensionalidad, lo que las hace ideales para tareas de reconocimiento de actividades humanas en entornos de vigilancia y monitoreo.

Podemos definir red neuronal como un conjunto de algoritmos que están diseñados para reconocer patrones. Estos patrones que son capaces de reconocer se introducen a la red en forma de vectores numéricos que representan datos de la vida real, ya sean imágenes, sonidos, texto o datos temporales entre otros. Gracias a las redes neuronales es agrupar datos no etiquetados en función de su grado de similitud o clasificar datos desconocidos a partir de un conjunto de entrenamiento etiquetado.

La unidad básica de red neuronal es el perceptrón. Este cuenta con unas entradas “ x_n ” que son multiplicadas por sus respectivos pesos “ w_n ”. Los productos resultantes se suman y se añade un “bias” que funciona como offset (ayuda al modelo a ajustar la función para que la predicción esté próxima a los valores reales). Todo esto se pasa a una función de activación para determinar en qué medida esa señal seguirá avanzando por la red y afectará al resultado final. Esta función nos mostrará la salida de la neurona y su estado de activación. En resumen, el perceptrón es una neurona artificial con una capa de entrada y una capa de salida, por ello se considera la red neuronal más simple.

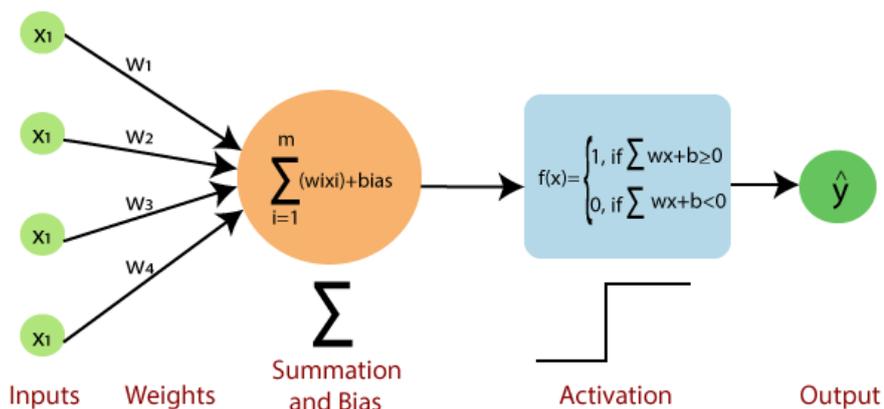


Figura 16: Modelo de perceptrón [16]

De esta forma, la red neuronal se forma a partir de una superposición de capa, cada una de las cuales está construida a base de neuronas. La primera capa se conoce como capa de entrada o *input layer*, la última capa es la capa de

salida u *output layer* y el resto de capas intermedias se denominan capas ocultas o *hidden layers*, ya que no se sabe la información que fluye entre ellas.

Teniendo en cuenta lo explicado en apartados anteriores, podemos definir *Deep Learning* como un método de inteligencia artificial, en particular de *Machine Learning*, que se basa en una red neuronal con 3 o más capas. Mientras que con una red neuronal básica de una sola capa se pueden hacer predicciones aproximadas, al incorporar capas ocultas se consigue optimizar el modelo e incrementar la precisión. En *Deep Learning* cada capa en el entrenamiento es la responsable de extraer una característica concreta a partir de la salida de la capa anterior. De esta forma, cuantas más capas tenga la red, lo que es lo mismo, cuanto más profunda sea, las características que se podrán reconocer serán cada vez más complejas, ya que las características de las capas anteriores se van recomblando. Así se consigue una extracción automática de características con una abstracción cada vez mayor.

3.2.2 CONCEPTOS DEL APRENDIZAJE PROFUNDO

3.2.2.1 FEEDFORWARD

Las redes *feedforward* o *forward propagation* son aquellos modelos en los que la información circula de izquierda a derecha, desde la entrada a la salida. En el proceso de *forward propagation* cada neurona recibe como entradas las salidas de la capa anterior, aplica los pesos y bias correspondientes a estas entradas y pasa el resultado a la función de activación. Este proceso se repite para cada capa hasta que la capa de salida genera la predicción final. Como se ha mencionado anteriormente, se obtendría un valor z a partir de los pesos y bias que se pasa a una función de activación $f(z)$:

$$z = x * w + b$$

$$y = f(z) = f(x * w + b)$$

3.2.2.2 BACKPROPAGATION

Se trata de un mecanismo por el cual la red neuronal ajusta sus pesos y bias durante el entrenamiento. Esto implica calcular la función de coste o error del gradiente de la red con respecto a sus parámetros y utilizar esta información para actualizar los parámetros en dirección opuesta al gradiente. Gracias a esto, la red mejora de forma iterativa sus predicciones minimizando la diferencia entre la salida predicha y la deseada. Según investigaciones recientes enfocadas al aprendizaje automático [5], los pasos que sigue este algoritmo de retropropagación son los siguientes:

1. Se inicializan los pesos empleando unos valores al principio.
2. Se toma una muestra de datos de la entrada y se determina la salida aplicando el proceso de *feedforward* o paso hacia adelante.

3. Obtenemos el error cuadrático medio de la salida al realizar una comparación con los valores deseados y los valores conseguidos.
4. Se calcula la parte proporcional del error de cada unidad neuronal a la salida empleando el error que se ha conseguido en el punto anterior. Con el error de cada neurona se puede calcular el error de las neuronas de la capa anterior, aplicando un cálculo del gradiente.
5. A continuación, se ajustan los nuevos pesos de cada neurona
6. Si se llega a un error mínimo determinado previamente o a algún criterio de parada entonces el proceso se detiene.
7. Se retorna al paso 2.

La fórmula general del *backpropagation* para actualizar los pesos de las conexiones se puede expresar de la siguiente manera:

$$\Delta w_{ij} = -\eta * \frac{\partial J(\theta)}{\partial w_{ij}}$$

Donde Δw_{ij} es el cambio que se debe realizar en el peso de la conexión entre la neurona i y la neurona j ; η es la tasa de aprendizaje que determina la magnitud de l ajuste realizado en cada iteración; y $\frac{\partial J(\theta)}{\partial w_{ij}}$ es la derivada parcial de la función de coste respecto al peso. Esta derivada indica cómo el cambio en el peso afecta al error de la red. Este último término se calcula mediante el algoritmo de *backpropagation*, que implica el cálculo de gradientes en cada capa de la red neuronal utilizando la regla de la cadena. Luego, se utilizan estos gradientes para actualizar los pesos en dirección al mínimo global de la función de error.

3.2.2.3 FUNCIONES DE ACTIVACIÓN

Las funciones de activación se aplican a la salida de cada neurona para introducir no linealidad en el modelo. Estas funciones determinan si una neurona se activa o no en respuesta a los estímulos que recibe. Encontramos una gran variedad de funciones de activación pero podemos destacar las siguientes:

- **ReLU (Rectified Linear Unit):**

ReLU es una función de activación popular en redes neuronales profundas. Su fórmula es:

$$f(x) = \max(0, x)$$

Esto que significa que cualquier valor negativo se establece en cero, mientras que los valores positivos se mantienen sin cambios. Esta función es simple y eficiente de calcular, y tiende a superar problemas de desvanecimiento del gradiente en comparación con funciones como la sigmoide y la tangente hiperbólica. ReLU se usa comúnmente en las capas ocultas de las redes neuronales.

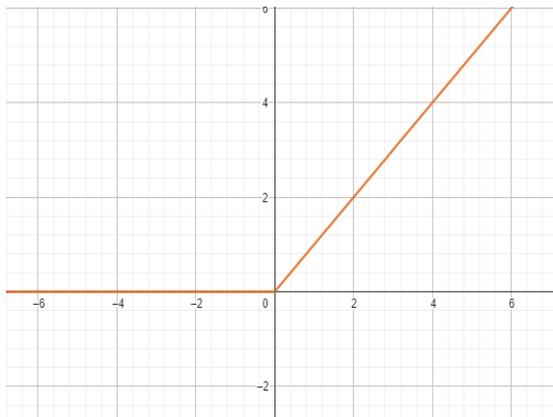


Figura 17: Representación función Relu en Geogebra

- **Sigmoide:**

La función sigmoide es una función de activación que produce una salida en el rango de 0 a 1. Su fórmula es:

$$f(x) = \frac{1}{1 + e^{-x}}$$

Mapea cualquier valor real a un valor entre 0 y 1, lo que se interpreta como una probabilidad. En el pasado, la función sigmoide se usaba ampliamente en las redes neuronales, especialmente en problemas de clasificación binaria. Sin embargo, su uso ha disminuido en las capas ocultas debido a problemas de desvanecimiento del gradiente cuando se aplica en redes neuronales profundas.

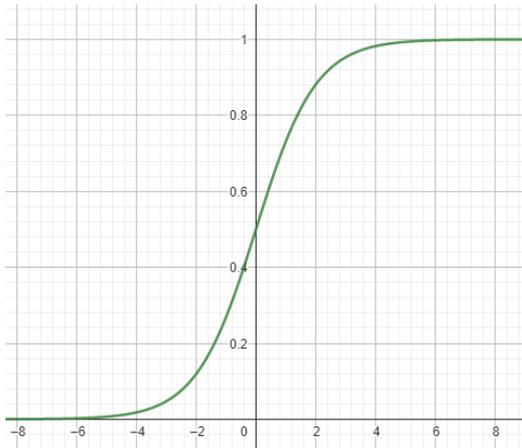


Figura 18: Representación función Sigmoide en GeoGebra

- **Tangente hiperbólica (tanh):**

La función tangente hiperbólica (tanh) es otra función de activación que produce una salida en el rango de -1 a 1. Su fórmula es:

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

Al igual que la función sigmoide, la función tangente hiperbólica también se usa en problemas de clasificación binaria y en capas ocultas de redes neuronales. A diferencia de la función sigmoide, la función tanh es simétrica alrededor del origen y tiene una salida centrada en cero, lo que puede ayudar al aprendizaje en ciertos casos.

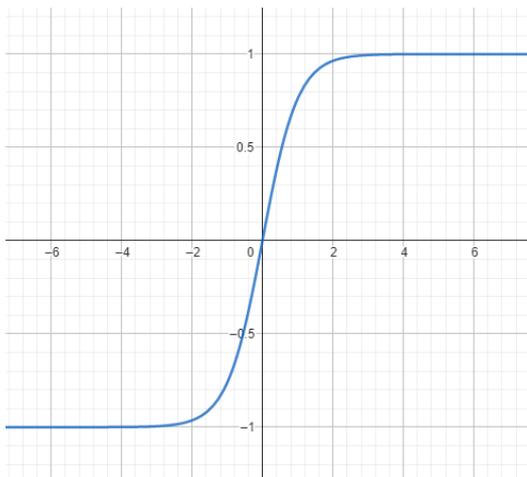


Figura 19: Representación función Tangencial Hiperbólica en GeoGebra

- **Softmax:**

La función softmax se utiliza comúnmente en la capa de salida de una red neuronal cuando se realiza una clasificación multiclase. La función softmax toma un vector de valores reales y produce un vector de probabilidades normalizado. Su fórmula es:

$$f(x_i) = \frac{e^{x_i}}{\sum(e^{x_j})}$$

para cada elemento x_i del vector de entrada x . La suma de las salidas softmax es 1, lo que permite interpretar la salida como una distribución de probabilidad sobre diferentes clases. Softmax es útil cuando se desea asignar una probabilidad a cada clase exclusiva en un problema de clasificación con múltiples clases. Tras el entrenamiento de la red, se genera una salida y en forma de vector de números, denominados logit, que corresponden con las posibilidades de cada salida de pertenecer a una clase. La función softmax transforma estos números en probabilidades entre 0 y 1, de forma que para cada posible resultado se consigue una probabilidad y la predicción se realizará en base a la probabilidad que sea más alta.

Más tarde, se aplica *one-hot encoding* para convertir el vector correspondiente con las etiquetas o *labels*, que es categórico, a un formato numérico binario. Así cada categoría se representa en un formato de 0s para las clases incorrectas y un 1 para la clase correcta.

Para comparar ambos vectores, el de etiquetas y el de probabilidades, se emplea la entropía cruzada o *cross-entropy* para medir cómo de bien se ajustan las predicciones del modelo a la distribución de probabilidad esperada. Esta medida compara la probabilidad predicha por el modelo para cada clase con la probabilidad real para esa clase. De esta forma si el modelo tiene una alta certeza en la clase correcta, la entropía cruzada será baja, mientras que si el modelo tiene una alta confusión en la clasificación la entropía cruzada será alta.

3.2.2.4 HIPERPARÁMETROS

En el contexto de las redes neuronales, los hiperparámetros son parámetros externos al modelo que se configuran antes de iniciar el proceso de entrenamiento y afectan el comportamiento y rendimiento del modelo. A diferencia de los parámetros del modelo, que se aprenden durante el entrenamiento, los hiperparámetros se establecen antes de comenzar el proceso de optimización y no se actualizan durante el entrenamiento. Algunos de los hiperparámetros más comunes en aprendizaje profundo son los siguientes:

- **Tasa de aprendizaje (*learning rate*):** Es la tasa a la que los pesos del modelo se actualizan durante el entrenamiento. Durante el entrenamiento de un modelo, se utiliza un algoritmo de optimización, como el descenso del gradiente, para ajustar los pesos y sesgos con el objetivo de minimizar la función de pérdida o error del modelo. La tasa de aprendizaje determina la velocidad a la que estos ajustes se realizan. Una tasa de aprendizaje alta implica realizar ajustes más grandes en los parámetros del modelo en cada iteración del algoritmo de optimización. Por otro lado, una tasa de aprendizaje baja implica realizar ajustes más pequeños en los parámetros del modelo en cada iteración.
- **Tamaño del lote (*batch size*):** Determina la cantidad de ejemplos de entrenamiento que se utilizan en cada iteración del entrenamiento. Cuando entrenamos un modelo, los datos de entrenamiento se dividen en lotes más pequeños para procesarlos de manera más eficiente. En lugar de alimentar todos los ejemplos de entrenamiento a la vez, se selecciona un lote de datos y se realiza un paso de optimización utilizando esos ejemplos. Luego, se toma otro lote y se repite el proceso hasta que todos los ejemplos de entrenamiento hayan sido procesados. Un tamaño de lote más grande puede acelerar el entrenamiento, pero también requiere más memoria.
- **Número de épocas (*epochs*):** Es el número total de veces que el modelo pasa por todo el conjunto de entrenamiento. Cada época consiste en iteraciones sobre los lotes de entrenamiento. Aumentar el número de épocas puede permitir que el modelo mejore su rendimiento a medida que ve más ejemplos de entrenamiento.
- **Inicialización de pesos:** Los hiperparámetros de inicialización de pesos determinan cómo se establecen inicialmente los pesos en la red neuronal antes del entrenamiento. Algunas opciones comunes incluyen la inicialización aleatoria y la inicialización con valores preentrenados

3.2.2.5 OPTIMIZADORES

Los optimizadores en redes neuronales son algoritmos que se utilizan para ajustar los pesos y los sesgos de una red neuronal durante el proceso de entrenamiento. Su objetivo principal es minimizar una función de pérdida, lo que implica encontrar los valores óptimos de los parámetros de la red que minimicen el error entre las predicciones y los valores reales. Es decir, tras aplicar la entropía cruzada y obtener el error entre los valores predichos y los deseados, el procedimiento de optimización intenta reducir la distancia que hay

entre el vector de probabilidades y el vector real para la clase correcta y realizar el proceso inverso para la clase errónea.

El proceso de determinación y optimización del error se llama *Gradient Descent* o descenso de gradiente. Como el cálculo del error de todos los datos es muy costoso computacionalmente se suele utilizar una técnica derivada de esta llamada *Stochastic Gradient Descent*, en el que el error se calcula a partir de un conjunto de datos en vez de tomar todos los datos en su totalidad. SGD es el optimizador más básico y comúnmente utilizado en el entrenamiento de redes neuronales. Actualiza los parámetros en función del promedio de los gradientes calculados el conjunto de datos seleccionados. SGD puede ser ruidoso debido a la variabilidad en este conjunto de datos, pero es computacionalmente eficiente y generalmente converge a una solución aceptable.

Igualmente, otros de los optimizadores más populares utilizados en redes neuronales son:

- **Adam (*Adaptive Moment Estimation*):** este combina el descenso de gradiente estocástico (SGD) con técnicas adicionales basadas en momentos. Utiliza tanto el promedio móvil de los gradientes pasados como el promedio móvil de los cuadrados de los gradientes pasados para adaptar la tasa de aprendizaje de cada parámetro de manera individual.
- **RMSprop (*Root Mean Square Propagation*):** RMSprop es otro optimizador que también se basa en SGD. Su enfoque principal es resolver el problema de la tasa de aprendizaje adaptativa. En lugar de utilizar el promedio de todos los gradientes pasados, RMSprop utiliza un promedio móvil de los cuadrados de los gradientes pasados. Esto ayuda a atenuar la magnitud de los gradientes para los parámetros que tienen una alta varianza en los gradientes.
- **Adagrad (*Adaptive Gradient Algorithm*):** se trata de un optimizador que ajusta la tasa de aprendizaje de cada parámetro de forma adaptativa según la frecuencia de actualización de ese parámetro en particular. Esto significa que los parámetros que tienen una alta frecuencia de actualización tienen una tasa de aprendizaje más baja, mientras que los parámetros con una baja frecuencia de actualización tienen una tasa de aprendizaje más alta. Adagrad es útil en situaciones en las que diferentes parámetros requieren tasas de aprendizaje muy diferentes.

Cabe destacar que existen muchos otros optimizadores disponibles, cada uno con sus propias características y variaciones. La elección del optimizador depende del problema específico, la arquitectura de la red y la disponibilidad de datos.

3.2.2.6 MODELOS NEURONALES

Dentro de las redes neuronales profundas podemos diferenciar diversos tipos en función de su arquitectura de red. A la hora de reconocer actividades humanas como es el caso, suelen emplearse las siguientes arquitecturas:

REDES CONVOLUCIONALES

Una red convolucional, también conocida como CNN (*Convolutional Neural Network*) en inglés, es un tipo de arquitectura de red neuronal especialmente diseñada para procesar datos con una estructura de cuadrícula, como imágenes, videos.

Las redes convolucionales se caracterizan por presentar capas capaces de realizar una extracción de características a partir de los datos de entrada. Las capas de la CNN realizan esta extracción mediante operaciones de convolución en la que utilizan filtros (*kernels*). Los filtros de convolución se van desplazando sobre la imagen, multiplican cada píxel por los valores determinados en el filtro y se calcula el valor central del mismo mediante un promedio. Gracias a la aplicación de estos filtros conseguimos entrenar la red con muchos menos parámetros en comparación con una capa densa y además reducimos la dimensionalidad tratando de conservar las características esenciales de la entrada.

La arquitectura de una red convolucional se compone principalmente de capas de convolución, capas de pooling y capas totalmente conectadas (*Dense Connected Layers*) al final:

***Capas de convolución:** Las capas de convolución son la piedra angular de una CNN. Estas capas realizan operaciones de convolución aplicando filtros o kernels a la entrada para extraer características locales, como se ha comentado anteriormente. La operación de convolución captura patrones y características importantes presentes en la imagen.

***Capas de pooling:** Después de una capa de convolución, a menudo se aplica una capa de pooling para reducir el tamaño espacial de la representación y disminuir la cantidad de parámetros en la red. La operación de pooling se

encarga de resumir la información localmente, generalmente mediante la selección del máximo valor (max pooling) o el promedio (average pooling) dentro de una ventana.

***Capas totalmente conectadas:** Al final de la arquitectura de la CNN, se suelen agregar capas totalmente conectadas para realizar la clasificación o predicción final. Estas capas están conectadas a todas las unidades de la capa anterior y aplican transformaciones lineales y no lineales a los datos. En el caso de la clasificación, la última capa suele ir acompañada de una función de activación softmax para obtener una distribución de probabilidades sobre las diferentes clases.

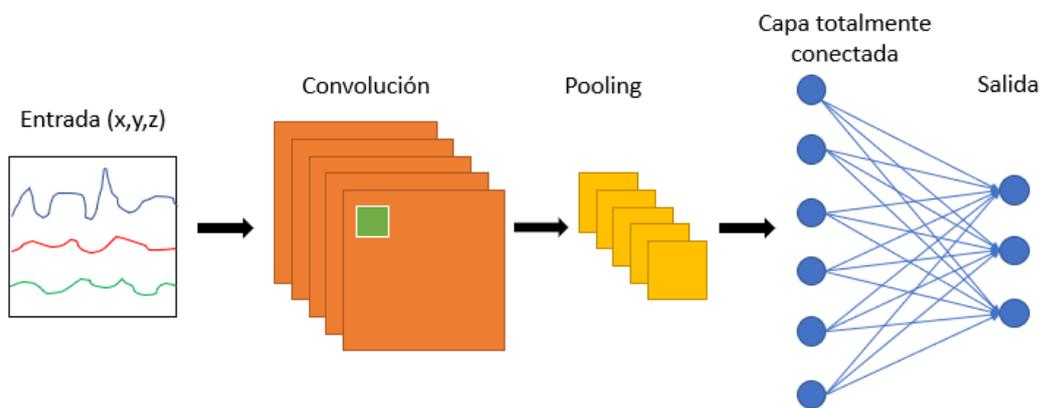


Figura 20: Arquitectura general modelo CNN

Además de estas capas básicas, una CNN puede contener otras capas adicionales, como capas de normalización para regularizar la salida de las capas anteriores y capas de activación no lineal, como la función de activación ReLU (Rectified Linear Unit), para introducir no linealidad en la red.

Las redes convolucionales más conocidas suelen utilizarse para la extracción de características de imágenes (2 dimensiones), pero es común utilizar redes convolucionales de una dimensión (Conv1D) para trabajar con series temporales. La capacidad de las redes convolucionales para extraer características relevantes de los datos de entrada las hace ideales para el reconocimiento de actividades humanas, ya que pueden identificar patrones específicos en las imágenes o videos que representan diferentes actividades. Aunque las redes convolucionales son más comúnmente utilizadas en la clasificación de imágenes y videos, también son capaces de procesar datos de series temporales como los datos de acelerómetro y extraer sus características más importantes.

REDES LSTM

Las redes LSTM (Long Short-Term Memory) son un tipo de redes recurrentes que es capaz de “recordar” un dato relevante en una secuencia temporal y preservarlo por varios instantes de tiempo.

Una red recurrente es un tipo de red neuronal donde las conexiones son recurrentes, lo que significa que la salida de una capa se retroalimenta como entrada a la capa siguiente. Esta retroalimentación permite que las RNN tengan memoria o estados internos, lo que las hace adecuadas para capturar dependencias a largo plazo en los datos secuenciales. Así en las redes recurrentes a partir de la entrada se obtiene como salida un estado oculto que encapsula información relevante de capas anteriores y una predicción del resultado.

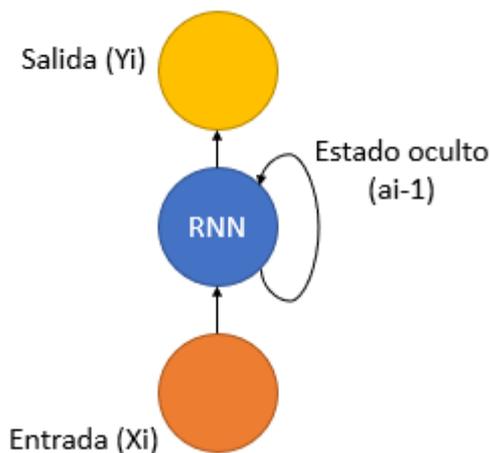


Figura 21: Modelo de neurona recurrente

El problema de las redes recurrentes tradicionales es que, al entrenar con secuencias muy largas, el gradiente se debilita de forma exponencial. Esto provoca que los pesos de las capas anteriores se actualicen de manera insignificante, por lo que la red tendrá más dificultades para recordar información a largo plazo, es decir, la red tendrá dificultades para memorizar patrones de secuencias largas.

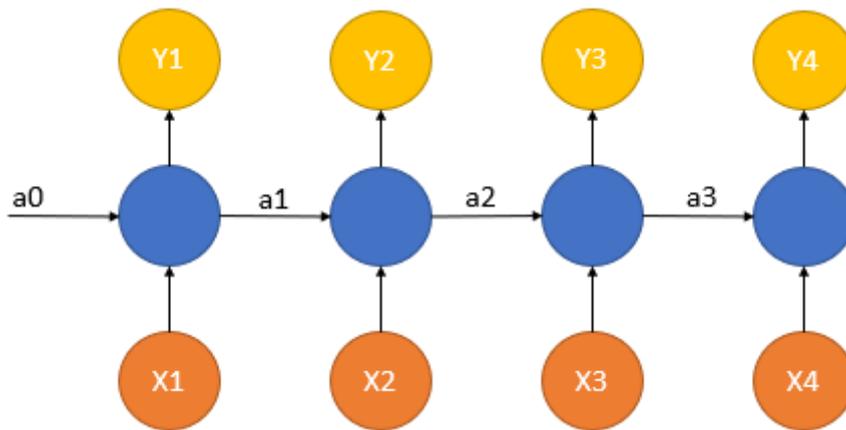


Figura 22: Modelo neurona recurrente desenrollada en el tiempo

La salida y los diferentes estados ocultos se pueden definir con las siguientes ecuaciones, donde y es la salida, a los estados ocultos y W la matriz de pesos:

$$y_4 = \text{softmax}(W_{ya} * a_4)$$

$$a_1 = \tanh(W_{aa} * a_0)$$

$$a_2 = \tanh(W_{aa} * a_1) = \tanh(W_{aa} * \tanh(W_{aa} * a_0))$$

$$a_3 = \tanh(W_{aa} * a_2) = \tanh(W_{aa} * \tanh(W_{aa} * \tanh(W_{aa} * a_0)))$$

Esto se puede explicar debido a su arquitectura de red. Si nos fijamos en la figura superior, observamos que en cada paso la red toma la entrada x y el estado oculto anterior a_{i-1} para generar un nuevo estado oculto a_i utilizando una función de activación tangente hiperbólica (\tanh). La predicción en cada etapa se calcula aplicando una transformación al estado oculto mediante una función de tipo softmax. El estado oculto 3, por ejemplo, se ha calculado aplicando la función \tanh al estado oculto 2, que se ha obtenido al aplicar una \tanh al estado oculto 1 y así sucesivamente. Esto provoca que el estado oculto inicial a_0 está siendo multiplicado por un número muy pequeño cuando llega a la salida; las funciones \tanh están anidadas y su valor máximo es cercano a 1.

Como resultado el efecto que tiene el estado oculto inicial en el cálculo de la salida final será mínimo por lo que la influencia de este primer estado en la predicción será muy pequeña. Si consideramos una secuencia más larga, el valor del estado oculto se irá multiplicando cada vez más veces por un número menor que 1 y su influencia se verá aún más reducida. Las redes neuronales recurrentes simples entonces tan solo tendrán una memoria a corto plazo.

Para solucionar este inconveniente una alternativa son las redes LSTM. Se trata de una variante específica de la red neuronal recurrente que fue diseñada para abordar el problema del desvanecimiento del gradiente, que ocurre en las RNN tradicionales al entrenar con secuencias largas.

Las redes LSTM se caracterizan por tener unidades de memoria que les permiten recordar información relevante de la secuencia de entrada durante un período de tiempo prolongado.

Cada unidad de memoria tiene una estructura interna que le permite retener información durante largos períodos de tiempo y decidir qué información es relevante para recordar y qué información debe descartarse. Esto se consigue gracias a la incorporación de un estado de celda (*cell state*) que es la memoria a largo plazo de una celda de memoria en un LSTM. El estado de la celda se actualiza mediante tres puertas (*gates*) que controlan el flujo de información dentro de la celda de memoria. Las tres puertas principales en una LSTM son:

***Puerta de olvido (*forget gate*):** define que información almacenada en la celda de memoria se debe descartar y cuánta información nueva se debe agregar. Esta puerta contiene una función sigmoide cuya salida puede ser 0 o 1 que servirá para olvidar o mantener la información respectivamente.

***Puerta de entrada (*input gate*):** controla la cantidad de nueva información que se debe agregar a la celda de memoria. Aquí encontramos una función sigmoide que decide qué valores serán actualizados y una función tanh que asigna un peso a estos valores para que sean añadidos al estado de la celda.

***Puerta de salida (*output gate*):** determina cuánta información de la celda de memoria se debe utilizar para calcular la salida actual. Presenta una función sigmoide que decide que parte del estado de la célula se utiliza para la salida y una tangente que asigna pesos a los valores elegidos.

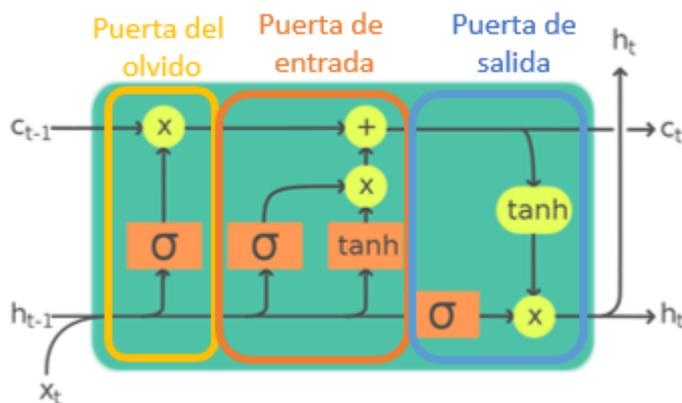


Figura 23: Arquitectura modelo LSTM [17]

Estas compuertas permiten que la red LSTM aprenda qué información almacenar, olvidar y generar como salida.

Las redes LSTM presentan una arquitectura más complicada en comparación con las redes recurrentes simples, pero gracias a esto son capaces de aprender patrones más complejos en secuencias de datos con tan solo un conjunto de parámetros que se aprenden durante el entrenamiento, igual que el resto de redes neuronales. Esta cualidad de su arquitectura les ayuda a aprender los parámetros de forma más eficiente y por ello se han utilizado con éxito para modelar la secuencia temporal de estos datos de acelerómetro y reconocer diferentes actividades humanas, como caminar, correr, sentarse y levantarse.

Una variante de la arquitectura LSTM es la utilización de redes LSTM Bidireccionales o BiLSTM. Este tipo de redes que utilizan dos capas de unidades LSTM: una que procesa la secuencia de entrada en orden y otra que la procesa en orden inverso. La combinación de ambas capas permite a la red capturar tanto las dependencias temporales hacia adelante como hacia atrás en la secuencia de entrada. Se trata de dos capas LSTM que procesan la secuencia de izquierda a derecha y de derecha a izquierda para fusionar los resultados y obtener una salida final. Gracias a esta red se puede obtener información sobre eventos pasados y futuros, algo que no puede conseguir la red LSTM simple.

MODELOS HÍBRIDOS

Varios autores han propuesto implementar en la arquitectura de red un **modelo híbrido** entre red convolucional y recurrente, como es el modelo CNN-LSTM. La combinación de CNN y LSTM da al modelo la capacidad de extraer características locales y de tener en cuenta aspectos de los datos secuenciales a largo plazo, lo que puede ser útil para los datos HAR de series temporales.

Esta arquitectura se utiliza para capturar tanto la dependencia espacial como temporal en los datos de acelerómetro, lo que permite a la red extraer características relevantes de la serie temporal de datos de acelerómetro y utilizarlas para clasificar diferentes actividades.

ARQUITECTURAS ESPECÍFICAS

A partir de estos modelos neuronales básicos es posible realizar combinaciones y modificaciones en las capas para obtener variantes que integren las ventajas de los modelos individuales.

Igualmente, en numerosos artículos se ha decidió utilizar un **modelo residual** para conseguir el reconocimiento de actividades humanas.

Un bloque residual se compone de una ruta residual y una conexión atajo o identidad. Esta última actúa como una ruta alternativa que permite que la información original fluya directamente hacia adelante en la red, sin pasar por las capas intermedias que podrían estar afectando negativamente el rendimiento. Esto significa que los datos originales, que han demostrado ser útiles para lograr un rendimiento óptimo, se preservan y se utilizan para tomar decisiones en las capas posteriores.

La conexión por atajo actúa como un atajo que ayuda a preservar y transmitir la información original sin alteraciones a través de las capas de la red. Esto es útil cuando se alcanza un rendimiento óptimo y se quiere evitar cualquier degradación en el rendimiento de la red. Además, al agregar más parámetros a la red, se aumenta el potencial de ajuste del modelo. Esto significa que la red tiene la capacidad de adaptarse y capturar patrones más complejos y sutiles en los datos de entrada, lo que puede mejorar aún más su rendimiento.

Los modelos residuales suelen tener en la salida una función de activación ReLU que se utiliza para mejorar la eficiencia computacional y evitar problemas de saturación.

3.2.2.7 CONTROL OVERFITTING Y UNDERFITTING

El sobreajuste (*overfitting*) y el subajuste (*underfitting*) son fenómenos comunes que pueden ocurrir durante el entrenamiento de un modelo de aprendizaje automático, incluyendo el aprendizaje profundo. Ambos problemas afectan negativamente el rendimiento del modelo y la capacidad de generalización. A continuación, se explica cada uno de ellos y se mencionan algunos mecanismos utilizados para controlarlos:

OVERFITTING

Ocurre cuando el modelo se ajusta demasiado a los datos de entrenamiento y no generaliza bien en datos nuevos. El modelo puede memorizar los ejemplos de entrenamiento en lugar de aprender patrones y características generales

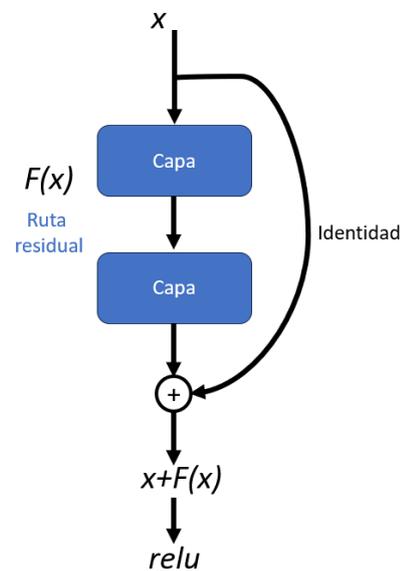


Figura 24: Arquitectura modelo residual

que se apliquen a datos no vistos. Esto puede conducir a un rendimiento deficiente cuando se enfrenta a nuevos datos. Algunos síntomas del sobreajuste incluyen un bajo error con los datos de entrenamiento, pero un alto error en el conjunto de prueba. Para controlar el sobreajuste, se utilizan técnicas como:

- **Regularización L1 y L2:** Introducen una penalización en la función de pérdida durante el entrenamiento para evitar que los pesos de la red neuronal se vuelvan demasiado grandes. La regularización L1 agrega una penalización proporcional al valor absoluto de los pesos, mientras que la regularización L2 agrega una penalización proporcional al cuadrado de los pesos. Estas técnicas promueven la simplicidad y evitan el sobreajuste.
- **Dropout:** Consiste en aleatoriamente desactivar un porcentaje de las neuronas durante el entrenamiento. Esto ayuda a prevenir la dependencia excesiva entre las neuronas y fomenta que cada neurona aprenda características útiles de manera independiente.
- **Early stopping:** Implica detener el entrenamiento del modelo antes de que alcance el número máximo de épocas si no se observa una mejora significativa en el rendimiento en el conjunto de validación. Esto ayuda a evitar que el modelo se sobreajuste a medida que continúa entrenando.

UNDERFITTING

Ocurre cuando el modelo es demasiado simple y no puede capturar patrones complejos en los datos de entrenamiento. En este caso, el modelo no se ajusta adecuadamente a los datos y su rendimiento tanto en el conjunto de entrenamiento como en el de prueba puede ser deficiente. Algunos métodos para controlar el subajuste incluyen:

- **Aumento de datos (*data augmentation*):** Consiste en aplicar transformaciones o modificaciones a los datos de entrenamiento existentes para aumentar su diversidad. Esto proporciona al modelo más ejemplos de entrenamiento y lo ayuda a aprender patrones más generales.
- **Aumento de la complejidad del modelo:** Aumentar la capacidad del modelo agregando más capas o neuronas puede ayudar a capturar

patrones más complejos en los datos. Sin embargo, se debe tener cuidado de no caer en el sobreajuste.

- **Batch Normalization:** Es una técnica que normaliza las activaciones de cada capa de la red neuronal. Esto ayuda a estabilizar el proceso de entrenamiento, evita la degradación del gradiente y facilita el aprendizaje en redes más profundas.

El objetivo principal al controlar el sobreajuste y el subajuste es encontrar un equilibrio para que el modelo pueda generalizar bien en nuevos datos. Estas técnicas son especialmente útiles cuando se rompe el equilibrio entre la optimización, es decir, la capacidad del modelo para ajustarse a los datos de entrenamiento, y la generalización, capacidad del modelo para hacer una clasificación de datos desconocidos que no se han utilizado durante el entrenamiento.

3.2.2.8 TRAIN, TEST Y VALIDATION

En el contexto del aprendizaje profundo, los conjuntos de entrenamiento, prueba y validación son divisiones de un conjunto de datos utilizadas para entrenar, evaluar y ajustar un modelo de red neuronal:

- **Conjunto de entrenamiento:** Es el conjunto de datos utilizado para entrenar el modelo de red neuronal. Los datos de este conjunto se utilizan para ajustar los pesos y los parámetros de la red durante el proceso de entrenamiento. La red aprende a través de la retropropagación del error y la optimización del gradiente descendente en función de estos datos de entrenamiento. Este conjunto de datos debe ser representativo de los casos reales para que el modelo aprenda correctamente.
- **Conjunto de prueba:** Una vez que el modelo ha sido entrenado en el conjunto de entrenamiento, se utiliza el conjunto de prueba o test para evaluar su rendimiento y medir su capacidad de generalización. Estos datos de prueba son independientes de los datos utilizados en el entrenamiento y se utilizan para estimar el rendimiento real del modelo en datos nuevos y no vistos previamente. La precisión o el error del modelo se calcula utilizando este conjunto de prueba, que nos indica cómo de bien se comporta el modelo ante datos diferentes de los del entrenamiento.

- **Conjunto de validación:** En algunos casos, se utiliza un tercer conjunto llamado conjunto de validación. Este conjunto se utiliza para ajustar los hiperparámetros del modelo, como la tasa de aprendizaje, el tamaño del lote o el número de capas. Durante el entrenamiento, se evalúa el rendimiento del modelo en el conjunto de validación para tomar decisiones sobre los hiperparámetros y ajustar el modelo. El conjunto de validación ayuda a evitar el *overfitting* del modelo y contribuye a la selección de la mejor configuración de hiperparámetros.

Es importante tener conjuntos de entrenamiento, prueba y validación separados y adecuadamente etiquetados para garantizar una evaluación justa y precisa del modelo de red neuronal. Estos conjuntos permiten medir el rendimiento del modelo en datos no vistos y ayudan a seleccionar y ajustar los hiperparámetros para mejorar la capacidad de generalización del modelo.

3.3 EXTRACCIÓN DE DATOS

Como se ha comentado anteriormente el proyecto EIAROB, sobre el que se estructura este trabajo, buscaba por una parte extraer los datos de salud de la *smartband* y por otro lado conseguir los datos en crudo de los sensores inerciales como acelerómetros y giroscopios. Para conseguir esta adquisición de datos se valoraron diferentes herramientas y métodos para extraer estos datos de la pulsera elegida. A pesar de esto, la obtención de estos datos tiene un denominador común: la tecnología BLE (*Bluetooth Low Energy*).

La tecnología bluetooth permite la transmisión de datos entre dos dispositivos localizados a una cierta distancia. Sus orígenes se remontan a 1994 cuando la compañía Ericsson comenzó a investigar en un nuevo método para mandar archivos entre teléfonos intentando reducir el coste y el consumo de batería.

El nombre de Bluetooth tiene su origen en un rey danés llamado Harald Blatand, que traducido al inglés resulta Harald Bluetooth. Uno de sus grandes logros fue la unificación de varias tribus danesas y su conversión a la religión cristiana. Uno de los ingenieros de Intel leyó la historia de este rey en el libro "*The Long Ships*" y pensó que sería buen nombre para una tecnología que lo que pretende es unificar diferentes dispositivos. Fue en el año 2000 cuando el primer móvil con bluetooth apareció en el mercado. Desde entonces la

utilización de esta tecnología ha sido clave en el desarrollo de la gran mayoría de dispositivos inteligentes.

Más tarde en 2011 nació el *Bluetooth Low Energy* (BLE) o Bluetooth de baja energía, que tenía como objetivo reducir la tasa de consumo de batería en comparación con los dispositivos que utilizaban la tecnología bluetooth tradicional. La idea de esto es que al reducir el consumo de la batería los dispositivos podrían mantenerse activos durante un periodo de tiempo mayor sin necesidad de cargarlos. A pesar de esto, uno de los contras de este tipo de tecnología es su rango de alcance, ya que los dispositivos bluetooth normalmente son capaces de comunicarse en un rango de 100 metros mientras que los de baja energía alcanzan un rango de 30 metros. Los dispositivos BLE utilizan el rango de radio frecuencia 2.4 GHz al igual que los teléfonos bluetooth tradicionales.

La estructura del protocolo de BLE es similar a la arquitectura de los protocolos TCP/IP. Las capas inferiores están diseñadas para hardware, emparejamiento y encriptación. Las cuatro capas superiores (Host y Aplicación) están encapsuladas para aplicaciones u otros sistemas operativos.

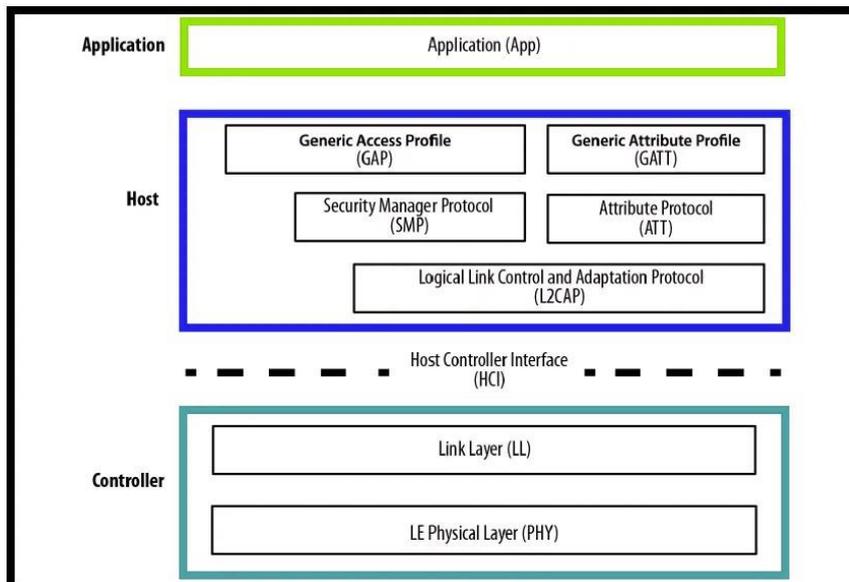


Figura 25: Arquitectura básica Bluetooth Low Energy [18]

Perfil de Acceso genérico (GAP): Se encarga de regular la búsqueda y conexión con otros dispositivos BLE

Perfil de atributos genéricos: Si nos fijamos en la capa de Host, encontramos un servidor llamado GATT; se trata de un perfil que define cómo los dispositivos BLE se comunican y cómo organizan y exponen sus servicios y características.

Los atributos de GATT se agrupan en servicios, cada uno de los cuales contiene características. Estas características a su vez pueden contener descriptores. Todo esto en el protocolo BLE, tanto servicios como características y descriptores, se localiza en el servidor GATT gracias a un UUID, un identificador de 16 bits o 128 bits. Así, por ingeniería inversa se puede estudiar que característica representa cada uno de los UUID y obtener información sobre el dispositivo.

Por lo que a la hora de trabajar con Bluetooth Low Energy es necesario tener en cuenta los siguientes conceptos:

- **Generic Attribute Profile (GATT):** especificación para mandar y recibir pequeños fragmentos, llamados atributos, entre un servidor y un cliente.
- **Attribute Protocol (ATT):** los atributos se identifican mediante un código universalmente único (UUID), que nos permite identificar la información correspondiente. Los atributos contenidos en ATT son formateados como características y servicios.
- **Characteristic:** cada característica presenta un valor único y varios descriptores que definen el valor de la característica.
- **Descriptor:** describen un valor de una característica
- **Service:** un servicio es un conjunto de características.

En el contexto de BLE los dispositivos se dividen en dos roles: el periférico o servidor, que es el dispositivo que ofrece los servicios y características, y el central o cliente, que es el dispositivo que accede y utiliza estos servicios. GATT define la estructura de datos y las reglas de intercambio de información entre el periférico y el central, como se ha comentado anteriormente. En el caso de la monitorización de actividad, el servidor GATT es el dispositivo que monitoriza la actividad (pulsera) mientras que el cliente es la aplicación integrada en el smartphone. El servidor expone los datos de actividad, como frecuencia cardíaca o calorías, mientras que el cliente puede consultar estos datos periódicamente.

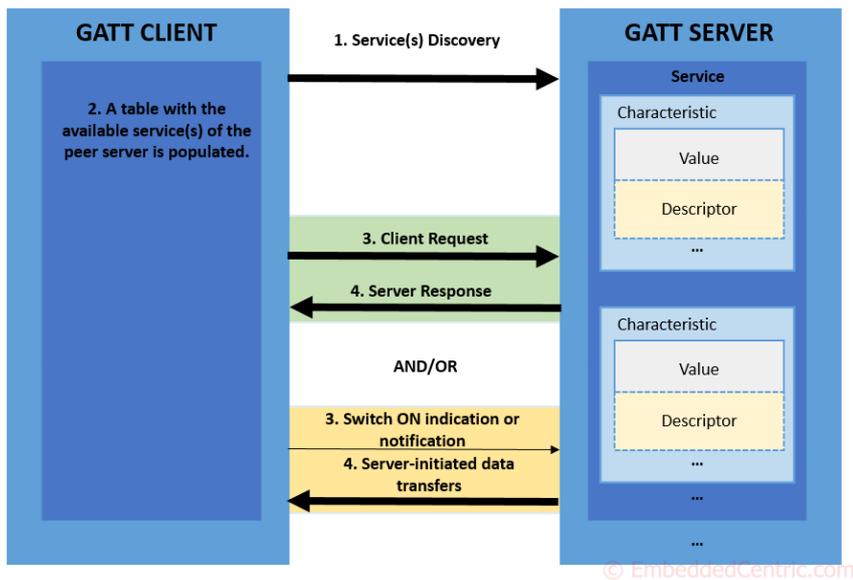


Figura 26: Interacciones GATT Cliente/Servidor [19]

Muchas de las aplicaciones mencionadas anteriormente utilizan BLE y el protocolo GATT para establecer la comunicación entre el teléfono inteligente Android y la pulsera escogida. Aquí se utilizan las capacidades de BLE y el protocolo GATT para descubrir servicios y características en las *smartband*, intercambiar datos, controlar funciones y recibir notificaciones de eventos como mensajes o datos de actividad física.

Para conocer los servicios y características a los que se puede acceder para la pulsera elegida se pueden utilizar herramientas de depuración, como por ejemplo nrfConnect. Esta nos permite explorar, configurar, probar y depurar dispositivos BLE. Proporciona una amplia gama de características y funcionalidades que facilitan el desarrollo de aplicaciones y servicios basados en BLE. Igualmente también se podría acudir a la documentación oficial de Bluetooth, donde se especifican algunas de los servicios y características básicos como frecuencia cardiaca o fecha.

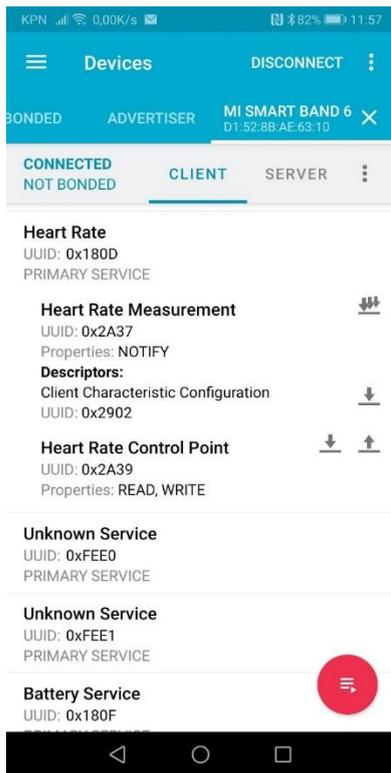


Figura 27: Captura aplicación nrfConnect

CAPÍTULO 4: DESARROLLO

Como se ha mencionado anteriormente, en este trabajo se pretende plantear una metodología adecuada para controlar la salud y las condiciones físicas de una persona anciana.

El primer paso para ello fue la elección de la pulsera de actividad que registrará esta información. Tras comparar las características de las diferentes pulseras se decidió escoger el último modelo de la marca Xiaomi, Xiaomi Smart Band 7, por ser una pulsera económica, precisa y con una batería de larga duración. Además, esta es capaz de adquirir una amplia gama de datos relacionados con la salud (como frecuencia cardiaca, pasos, saturación de oxígeno en sangre...) que pueden ser útiles para futuras etapas del proyecto. Igualmente, uno de los aspectos que fue esencial a la hora de elegir la pulsera fue la posibilidad de exportar los datos desde la pulsera o desde la app sin necesidad de requerir del permiso del fabricante, por lo que se eligió una pulsera de la familia Xiaomi ya que nos brinda más libertad en ese ámbito. Por último, al valorar diferentes modelos se observó que las pulseras inteligentes de la familia Xiaomi eran compatibles con un gran número de aplicaciones móviles que ofrecían numerosas ventajas relacionadas con la sincronización precisa y automática de los datos de la pulsera, por lo que finalmente se escogió para el proyecto.

Para estudiar el monitoreo de la actividad y salud de personas mayores se han tenido en cuenta dos propuestas diferentes: obtener los datos relacionados con la salud de la persona monitorizada, como frecuencia cardiaca, para que el cuidador pueda comprobarlo regularmente, y conseguir identificar las acciones que realiza el usuario aplicando aprendizaje profundo a partir de los sensores inerciales de su pulsera.

A continuación, se explicarán los pasos que se han llevado a cabo para extraer los datos correspondientes de la pulsera de actividad, así como las técnicas de clasificación empleadas para reconocer diferentes actividades a partir de las señales obtenidas de los acelerómetros y giroscopios de la *smartband*.

4.1 EXTRACCIÓN DE DATOS

4.1.1 AUTENTICACIÓN EN PULSERA INTELIGENTE

Para acceder algunas características determinadas de la pulsera inteligente, es necesario llevar a cabo una serie de pasos para lograr la autenticación de la misma. En el protocolo GATT, el servicio orientado a la autenticación es el que tiene una UUID que comienza con 0xfee1.

En el modelo de Xiaomi Band elegido el proceso de autenticación está basado en servidor, lo que significa que previamente hay que vincular la pulsera con la app oficial del fabricante (en este caso Zepp Life o Xiaomi Wear). Una vez que la *smartband* esté emparejada, nos dirigimos al directorio “freemyband” de la memoria local del teléfono y abrimos el fichero que contiene. Aquí se encuentra la dirección MAC de la pulsera inteligente y el “authkey” necesario para la autenticación.

En la app de Gadgetbridge, esta clave de autenticación se pide al emparejar la pulsera elegida con la aplicación.

4.1.2 PARÁMETROS DE SALUD

Como se ha explicado anteriormente, existen varias metodologías para extraer esta información, sin embargo estamos buscando un método de adquisición de datos que cumpla unos requisitos mínimos:

- A la hora de extraer los datos relacionados con el bienestar y la salud del usuario (frecuencia cardiaca, pasos, calidad de sueño...), la obtención de la información se debe realizar de forma automática, sin que el usuario intervenga en ningún momento (no podemos obligar a una persona dependiente a encender o sincronizar el dispositivo manualmente).
- Se debe garantizar la exportación de los datos para que el cuidador tenga acceso a ellos de forma remota.

Existen varias aplicaciones de actividad compatibles con la pulsera Xiaomi Smart Band 7 que registran la frecuencia cardiaca, calorías, estrés y otros parámetros que son medidos a través de la pulsera inteligente, sin embargo algunas de estas apps no consiguen sincronizar los datos en tiempo real con la pulsera o incluso no consiguen actualizar la información si no se indica de forma manual. Tras barajar varias alternativas se decidió emplear por un lado la app Gadgetbridge para obtener los datos relacionados con la salud del paciente y por otro lado los servicios y características del protocolo BLE para acceder a los datos en bruto de los sensores inerciales de la pulsera. Para conseguir esto se empleará un *smartphone* con sistema operativo Android 5.0 o superior, compatible con la app de actividad elegida, y una Raspberry Pi 4 modelo B que cuenta con conectividad Bluetooth 5.0.

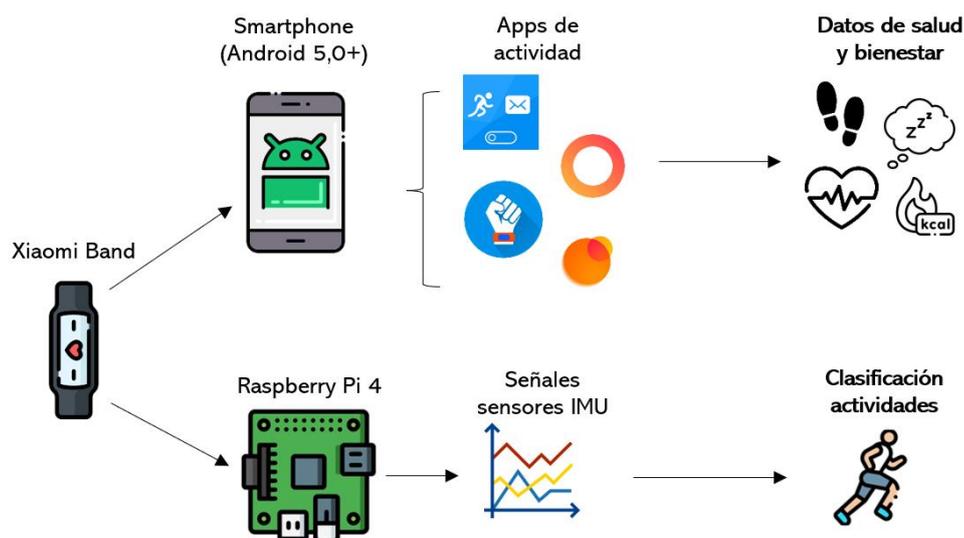


Figura 28: Modelo genérico de adquisición de datos

Como se ha comentado anteriormente, Gadgetbridge proporciona datos como el número de pasos, frecuencia cardiaca, tipo de actividad o datos relacionados con el sueño exportándolos en un archivo .sqlite. Este archivo contiene múltiples tablas, una por cada dispositivo compatible, por lo que para encontrar los datos de nuestro modelo Mi Band tendremos que desplazarnos hasta la tabla correspondiente. En la tabla “MI_BAND_ACTIVITY_SAMPLES” obtenemos las siguientes variables:

- TIMESTAMP**: representa el momento temporal en el que se han recogido los datos, utilizando formato Unix.
- DEVICE_ID**: código identificador que indica de qué dispositivo de monitorización se está tomando la información, en caso de que haya múltiples usuarios conectados a una misma app de Gadgetbridge.
- USER_ID**: código identificador que indica de qué usuario se está tomando la información, en caso de que haya múltiples usuarios conectados a una misma aplicación de Gadgetbridge.
- RAW_INTENSITY**: variable relacionada con la intensidad de movimiento del usuario, útil para la clasificación de actividades.
- STEPS**: representa el número de pasos registrados en un momento determinado.
- RAW_KIND**: código de actividades aproximado con el que se pueden diferenciar actividades como andar, estar sentado, dormir...
- HEART_RATE**: medida de la frecuencia cardiaca en un momento determinado.

A continuación, se muestran las variables obtenidas para el modelo Xiaomi Smart Band 7:

	TIMESTAMP	DEVICE_ID	USER_ID	RAW_INTENSITY	STEPS	RAW_KIND	HEART_RATE	UNKNOWN1	SLEEP	DEEP_SLEEP	REM_SLEEP
	Filtro	Filtro	Filtro	Filtro	Filtro	Filtro	Filtro	Filtro	Filtro	Filtro	Filtro
43	1683571860	1	1	41	0	80	75	5	0	128	0
44	1683571920	1	1	103	68	80	91	5	0	128	0
45	1683571980	1	1	79	39	80	106	15	0	128	0
46	1683572040	1	1	76	20	64	90	7	0	128	0
47	1683572100	1	1	69	0	64	89	7	0	128	0
48	1683572160	1	1	22	0	96	74	7	0	128	0
49	1683572220	1	1	75	26	96	81	7	0	128	0
50	1683572280	1	1	80	9	96	114	7	0	128	0
51	1683572340	1	1	66	10	64	119	7	0	128	0

Figura 29: Datos obtenidos con Gadgetbridge en archivo .sqlite

Los datos registrados están distanciados en intervalos de 1 minuto, por lo que tendremos un registro exhaustivo de la actividad del usuario. Gadgetbridge ofrece ventajas respecto a otras apps de bienestar como su sincronización automática con los datos de la pulsera inteligente y la opción de “Auto exportar datos” cada hora, que nos permite adquirir los datos de la app fácilmente y de forma automática. Estos datos, almacenados en la memoria local del teléfono inteligente, podrán sincronizarse con Google Drive u otras plataformas online para que los cuidadores y familiares del usuario tengan acceso a ellos en cualquier momento y de forma transparente.

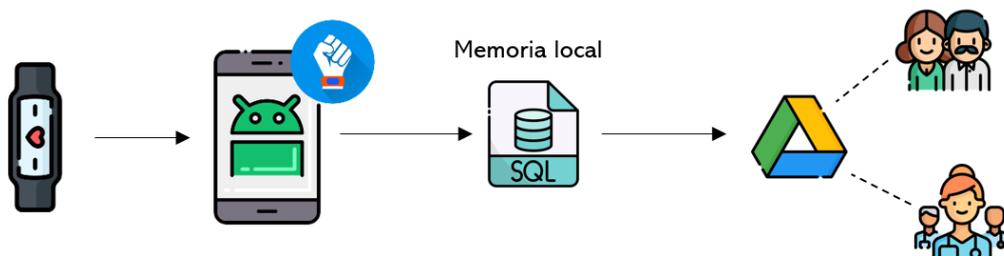


Figura 30: Modelo de flujo de información de salud y bienestar

Si nos fijamos en la Figura 29 observamos que una de las columnas se refiere a la variable “RAW_KIND” (tipo en bruto). Tras investigar en diversos artículos y foros, se llegó a la conclusión que la columna “RAW_KIND” del archivo .sqlite nos proporcionaba una etiqueta aproximada de la actividad que está realizando el usuario mediante un código de números (80: cargando, 96: estar sentado, 60: andar...), por lo que mediante este método podemos obtener información no solo información sobre las métricas de salud del paciente, como frecuencia

cardiaca o número de pasos, si no también podemos reconocer la actividad que realiza minuto a minuto de forma automática.

Ind	Type	Size	Value
42	tuple	11	(2023-05-08 20:51:00, 1, 1, 41, 0, 80, 75, 5, 0, 128, ...)
43	tuple	11	(2023-05-08 20:52:00, 1, 1, 103, 68, 80, 91, 5, 0, 128, ...)
44	tuple	11	(2023-05-08 20:53:00, 1, 1, 79, 39, 80, 106, 15, 0, 128, ...)
45	tuple	11	(2023-05-08 20:54:00, 1, 1, 76, 20, 64, 90, 7, 0, 128, ...)
46	tuple	11	(2023-05-08 20:55:00, 1, 1, 69, 0, 64, 89, 7, 0, 128, ...)
47	tuple	11	(2023-05-08 20:56:00, 1, 1, 22, 0, 96, 74, 7, 0, 128, ...)
48	tuple	11	(2023-05-08 20:57:00, 1, 1, 75, 26, 96, 81, 7, 0, 128, ...)
49	tuple	11	(2023-05-08 20:58:00, 1, 1, 80, 9, 96, 114, 7, 0, 128, ...)

Figura 31: Datos obtenidos desde Gadgetbridge al ser exportados a python

[2023-05-08 20:11:00, 'nopuesta']
[2023-05-08 20:12:00, 'nopuesta']
[2023-05-08 20:13:00, 'nopuesta']
[2023-05-08 20:14:00, 'andar']
[2023-05-08 20:15:00, 'sentar']
[2023-05-08 20:16:00, 'sentar']
[2023-05-08 20:17:00, 'sentar']
[2023-05-08 20:18:00, 'sentar']

Figura 32: Etiquetas estimadas de actividades obtenidas desde Gadgetbridge

A pesar de esto, las etiquetas obtenidas tan solo eran capaces de diferenciar unas pocas acciones, por lo que se decidió estudiar directamente las señales de los sensores inerciales en crudo para tener la posibilidad de realizar un etiquetado de las señales para cualquier actividad deseada.

4.1.3 DATOS SENSORES INERCIALES

Para obtener los datos en bruto provenientes de los sensores inerciales de la pulsera, se decidió estudiar los servicios y características específicos siguiendo el protocolo BLE. Se puede conocer cada parámetro que mide la pulsera inteligente si se conoce cada identificador universalmente único (UUID).

Podemos conocer a qué UUID se corresponden diferentes servicios acudiendo al manual oficial de Bluetooth que recoge todos los nombres de los servicios y características con su UUID asociado. [20]

Los identificadores universalmente únicos para las *smartband* de Xiaomi se construyen a partir de una cadena base de caracteres:

`Cadena_base = "0000%s-0000-1000-8000-00805f9b34fb"`

A esta cadena se le añade una secuencia de números que servirá para identificar diferentes parámetros recogidos por la pulsera. A continuación se disponen algunos de los códigos UUID que se refieren a varios servicios de una pulsera Xiaomi Band:

- **Ritmo cardiaco:** el servicio correspondiente al ritmo cardiaco es `SERVICE_HEART_RATE="0000180d-0000-1000-8000-00805f9b34fb"`
Las características contenidas en este servicio para medir el ritmo cardiaco son las siguientes:

`CHARACTERISTIC_HEART_RATE_MEASURE = "00002a37-0000-1000-8000-00805f9b34fb"`

`CHARACTERISTIC_HEART_RATE_CONTROL = "00002a39-0000-1000-8000-00805f9b34fb"`

- **Alertas y notificaciones:** el servicio correspondiente con la funcionalidad de alertas y notificaciones es

`SERVICE_ALERT_NOTIFICATION="00001811-0000-1000-8000-00805f9b34fb"`

Las características contenidas en este servicio para medir las alertas y notificaciones son las siguientes:

`CHARACTERISTIC_ALERT="00002a06-0000-1000-8000-00805f9b34fb"`

`CHARACTERISTIC_CUSTOM_ALERT = "00002a46-0000-1000-8000-00805f9b34fb"`

SERVICIO 0XFEE0

SERVICE_MIBAND = "0000fee0-0000-1000-8000-00805f9b34fb"

El servicio "0xfee0" se trata de un servicio asignado a la empresa de Xiaomi, por lo que las características obtenidas a partir de este servicio serán únicas para esta familia.

0xFEE0	Anhui Huami Information Technology Co., Ltd.
--------	--

Figura 33: UUID 0xfee0, página oficial de Bluetooth

Este servicio está asociado con distintas características que se definen de una manera específica en los dispositivos Xiaomi; entre ellas se puede destacar el tiempo actual, el nivel de batería, los pasos que realiza el usuario... [21]

CHARACTERISTIC_BATTERY="00000006-0000-3512-2118-0009af100700"

CHARACTERISTIC_STEPS = "00000007-0000-3512-2118-0009af100700"

CHARACTERISTIC_CURRENT_TIME = BASE % '2A2B'

CHARACTERISTIC_ACTIVITY_DATA = "00000005-0000-3512-2118-0009af100700"

De hecho, si estudiamos el código abierto que ofrece Gadgetbridge, podremos identificar algunas de estas características si se utiliza la UUID adecuada, ya que la app se basa en el protocolo GATT para la extracción de los datos de la pulsera inteligente. Estas características son las mismas que se obtienen al exportar los datos de la app Gadgetbridge a la memoria del teléfono.

```

Archivo  Editar  Pestañas  Ayuda
09.05 - 15:01: category: 80; intensity 71; steps 0; heart rate 92;
09.05 - 15:02: category: 80; intensity 25; steps 0; heart rate 82;
09.05 - 15:03: category: 90; intensity 0; steps 0; heart rate 77;
09.05 - 15:04: category: 90; intensity 0; steps 0; heart rate 76;
09.05 - 15:05: category: 90; intensity 0; steps 0; heart rate 83;
09.05 - 15:06: category: 80; intensity 31; steps 0; heart rate 82;
09.05 - 15:07: category: 80; intensity 36; steps 0; heart rate 77;
09.05 - 15:08: category: 80; intensity 18; steps 0; heart rate 78;
09.05 - 15:09: category: 80; intensity 39; steps 0; heart rate 79;
09.05 - 15:10: category: 90; intensity 0; steps 0; heart rate 73;
09.05 - 15:11: category: 1; intensity 87; steps 17; heart rate 75;
Finished fetching

```

Figura 34: Resultado al llamar CHARACTERISTIC_STEPS, CHARACTERISTIC_CURRENT_TIME, CHARACTERISTIC_ACTIVITY_DATA

- Giroscopio y acelerómetro:** Uno de los objetivos de este proyecto es extraer los datos de los sensores inerciales de la pulsera, acelerómetro y giroscopio. Para conseguirlo debemos llamar al servicio y las características adecuadas. Tras investigar en diferentes artículos y foros la comunidad estipuló que los datos de estos sensores se obtenían al llamar al último servicio mencionado ("0xfee0) y especificando las siguientes características [22]:

CHARACTERISTIC_SENSOR_CONTROL : "00000001-0000-3512-2118-0009af100700"

CHARACTERISTIC_SENSOR_MEASURE : "00000002-0000-3512-2118-0009af100700"

Esto nos devolverá 10 conjuntos de datos con el siguiente formato:

0x0100 0500 8200 0b00 0400 8000 0b00 0300 8100 0b00

Donde el primer conjunto de valores corresponde al tipo de cabecera, y los siguientes bytes corresponden con las medidas del acelerómetro y el giroscopio en valores hexadecimales para los tres ejes del espacio.

```
Archivo  Editar  Pestañas  Ayuda
Realtime gyro: X: -36, Y: 16, Z: 249
Realtime gyro: X: -23, Y: 7, Z: 252
Realtime gyro: X: -31, Y: 9, Z: 251
Realtime gyro: X: -35, Y: 20, Z: 252
Realtime gyro: X: -21, Y: -19, Z: 258
Realtime gyro: X: -32, Y: 7, Z: 252
Realtime gyro: X: -23, Y: 4, Z: 253
Realtime gyro: X: -28, Y: -5, Z: 254
Realtime gyro: X: -25, Y: -6, Z: 255
Realtime gyro: X: -26, Y: 8, Z: 256
Realtime gyro: X: -20, Y: -7, Z: 254
Realtime gyro: X: -32, Y: 3, Z: 253
Realtime gyro: X: -24, Y: 6, Z: 256
Realtime gyro: X: -25, Y: -3, Z: 254
Realtime gyro: X: -27, Y: 4, Z: 251
Realtime gyro: X: -29, Y: 9, Z: 253
Realtime gyro: X: -24, Y: 1, Z: 253
Realtime gyro: X: -25, Y: 5, Z: 255
Realtime gyro: X: -25, Y: 4, Z: 254
Realtime gyro: X: -24, Y: -2, Z: 254
Realtime gyro: X: -26, Y: 4, Z: 255
Realtime gyro: X: -27, Y: 2, Z: 251
Realtime gyro: X: -28, Y: 1, Z: 255
```

Figura 35: Datos del giroscopio obtenidos a partir de CHARACTERISTIC_SENSOR_MEASURE

A partir de los datos extraídos se podría realizar un etiquetado de datos en función de la actividad que esté realizando el usuario. Al asociar los diferentes datos obtenidos del acelerómetro y giroscopio con la acción que está realizando el usuario en cada momento se podrá construir una base de datos de reconocimiento de actividades humanas. Más tarde, diferentes modelos de redes neuronales podrían utilizar esta base de datos para ser entrenadas y realizar una clasificación óptima de las acciones recogidas.

Recopilar y etiquetar un conjunto de datos desde cero puede ser un proceso laborioso y costoso en términos de tiempo. Para generar un *dataset* coherente es necesario registrar una cantidad considerable de señales de diferentes sujetos realizando acciones diferentes, lo que supondría mucho tiempo tan solo para la etapa de adquisición. Además, para crear un *dataset* etiquetado y de alta calidad es necesario unos conocimientos especializados en etiquetado y procesado de señal. Otra desventaja de generar un *dataset* propio, es que puede resultar más difícil comparar los resultados con los de otros investigadores. Cada *dataset* puede tener sus propias características y particularidades, lo que dificulta la comparación directa.

Por todo ello se decidió elegir *datasets* orientados al reconocimiento de actividades humanas preexistentes y ampliamente utilizados en la comunidad científica. Al utilizar un *dataset* ya creado se aprovechará el trabajo realizado por otros investigadores o desarrolladores y será más fácil concentrarse en

otros aspectos de tu proyecto. Además, se reducirán los problemas por comparación y reproducibilidad ya que se podrá comparar los resultados directamente con los de otros investigadores y se podrá evaluar la efectividad de los modelos empleados en relación con unos resultados publicados previamente.

4.2 ELECCIÓN Y PREPARACIÓN DE DATASETS

En la etapa de elección de los *datasets* se tuvieron en cuenta algunos aspectos relevantes para esta investigación. En primer lugar, se tuvo en cuenta qué *datasets* eran más empleados en artículos e investigaciones para entrenar modelos neuronales; esto es clave para comparar los resultados obtenidos con las métricas que se han conseguido en la actualidad con este tipo de *datasets*. También se decidió elegir bases de datos que reconociesen actividades más o menos similares para que sea posible realizar una comparación interna de la clasificación de cada actividad. Igualmente se buscó que para recoger los datos inerciales se hubieran utilizado sensores *ponibles*, en vez de teléfonos inteligentes, a ser posible unidos a la muñeca, ya que es la idea principal del proyecto.

Teniendo en cuenta todo lo expuesto anteriormente, se eligieron como *datasets* las bases UCI HAR y WISDM. Se eligió UCI HAR y WISDM porque son dos de los *datasets* más utilizados por la comunidad, por lo que se tendrá acceso a muchos artículos e investigaciones para comparar las métricas obtenidas frente a los resultados de otros investigadores. Además, los *datasets* nos proporcionan datos preprocesados fiables que permiten clasificar 6 acciones más o menos similares.

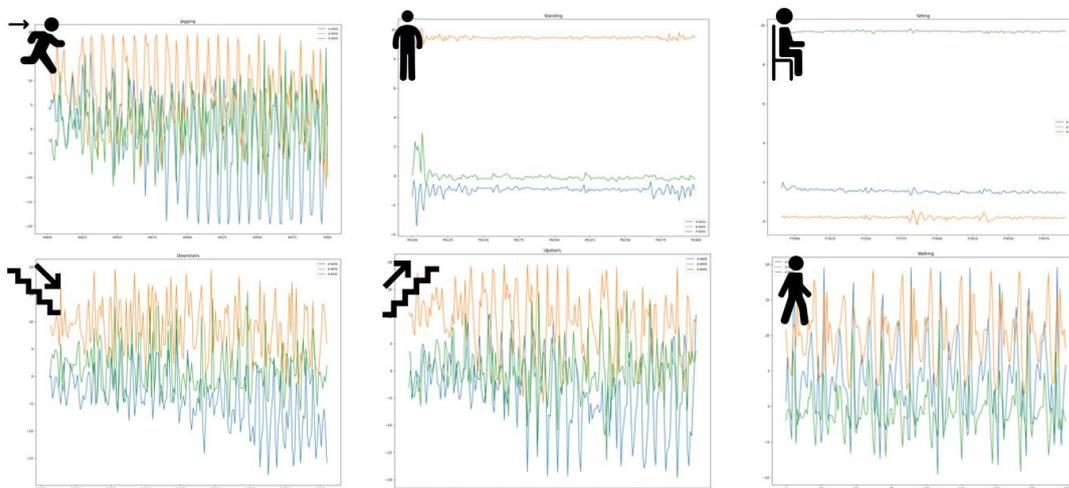


Figura 36: Señales acelerómetros para reconocer actividades en dataset WISDM

Para el *dataset* UCI HAR las señales de los sensores (acelerómetro y giroscopio) se preprocesaron aplicando filtros de ruido y, a continuación, se muestrearon en ventanas deslizantes de anchura fija de 2,56 segundos y solapamiento del 50% (128 lecturas/ventana). La señal de aceleración del sensor se preprocesó mediante un filtro Butterworth de paso bajo en aceleración del cuerpo y gravedad. Se supone que la fuerza gravitatoria sólo tiene componentes de baja frecuencia, por lo que se utilizó un filtro con una frecuencia de corte de 0,3 Hz. A partir de cada ventana, se obtuvo un vector de características calculando variables del dominio temporal y frecuencial.

En cuanto al *dataset* WISDM los sensores toman muestras a 20 Hz y se utiliza un tamaño de ventana de 10 segundos, por lo que la mayoría de las estadísticas se basan en la agregación de 200 lecturas de los sensores. Los datos brutos del acelerómetro fueron preprocesados para reducir el ruido utilizando un filtro de mediana de orden 5 en cada dimensión por separado. Igualmente se aplicó un filtro de paso bajo Butterworth de 3er orden con una frecuencia de corte de 0,3 Hz para separar la señal de aceleración en señales de aceleración de la gravedad y del cuerpo.

	UCI HAR	WISDM
<i>Sensores</i>	Acel. y giro.	Acel
<i>Frecuencia de muestreo</i>	50 Hz	20 Hz
<i>Nº muestras</i>	10299	1098207
<i>Nº clases</i>	6	6

Tabla 3: Descripción de los datasets

En Figura 37 y Figura 38 se representa la distribución de las muestras en cada base de datos escogida. Ambos *datasets* se dividieron en los conjuntos de entrenamiento y test en una proporción de 70% y 30% respectivamente.

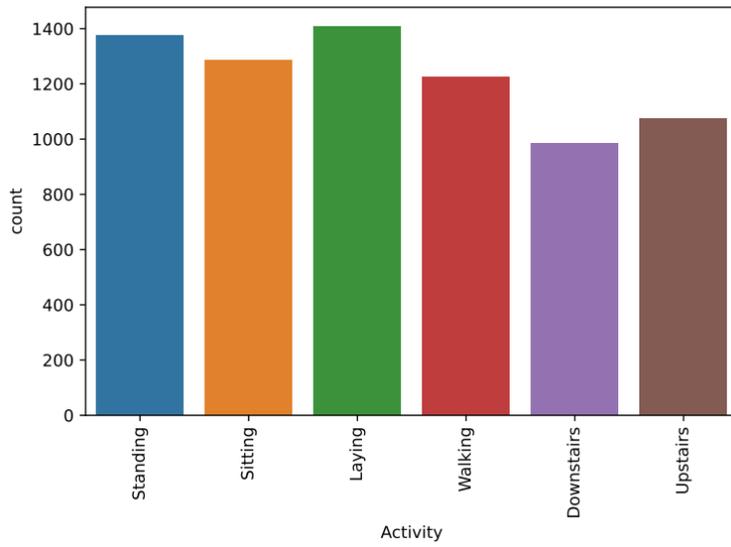


Figura 37: Distribución de las muestras por actividad en dataset UCI HAR

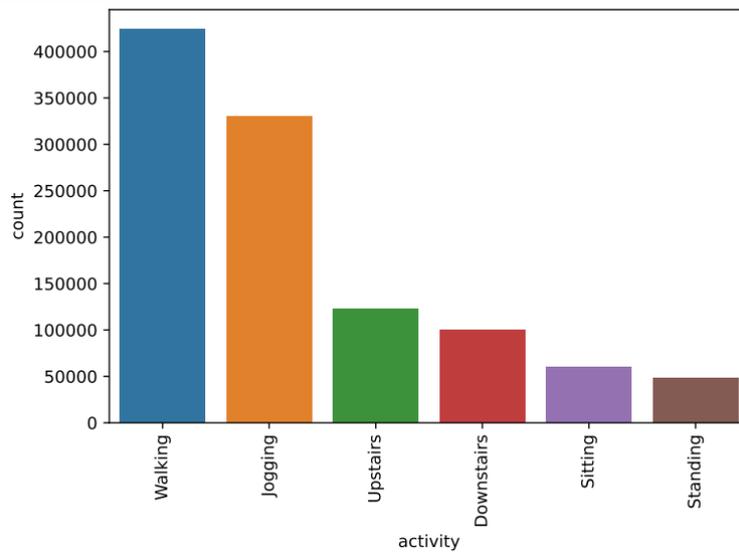


Figura 38: Distribución de las muestras por actividad en dataset WISDM

4.3 ARQUITECTURA DE RED

Como se mencionó anteriormente, el objetivo de este artículo es encontrar un modelo de red neuronal que reconozca de forma eficaz las actividades que realiza el usuario a partir de las señales de pulsera o teléfono inteligente. Tras hacer una lectura de varios artículos, se observó que los modelos más empleados eran CNN y LSTM, por lo que se evaluará el rendimiento de una red LSTM, una BiLSTM, una CNN y una red que será combinación de las dos últimas. Comprobaremos si la unión de estas dos redes tiene un efecto significativo. Más tarde se valorará si es posible realizar alguna alteración en el modelo neuronal para conseguir mejores resultados.

Al compilar los modelos, empleamos en todos los casos un optimizador Adam y una función de pérdida *categorical_crossentropy*.

4.3.1 DATOS DE ENTRADA

Cuando se trabaja con señales inerciales de tres ejes encontramos diferentes enfoques a la hora de pasar los datos como entrada. En algunos artículos se han desarrollado diversos modelos en los que cada uno de los ejes de la señal triaxial se pasa como entrada a una de las capas. Esto se suele plantear principalmente al trabajar con redes convolucionales, según la revisión bibliográfica realizada. [23]

La principal diferencia entre pasar los tres componentes de la entrada a una sola capa convolucional y pasar cada eje a una capa convolucional separada radica en la forma en que la red aprende las relaciones espaciales dentro de los datos de entrada.

Cuando se pasa cada eje (X, Y, Z) a capas convolucionales separadas, la red puede aprender características separadas de cada eje individualmente. Esto permite a la red captar patrones y correlaciones específicos de cada eje que podrían ser importantes para reconocer la actividad humana. A continuación, las operaciones de agrupación y aplanamiento de máximos se aplican por separado a la salida de la capa convolucional de cada eje antes de concatenarlas.

Por otro lado, cuando se pasan los componentes X, Y y Z juntos a una única capa convolucional, la red trata los tres componentes como un único canal de entrada. En este caso, la capa convolucional aprende filtros que se aplican a toda la entrada, capturando las relaciones espaciales que comparten los tres

componentes. A continuación, las operaciones de agrupamiento y aplanamiento de máximos se aplican a la salida de la capa convolucional única.

Igualmente, otras investigaciones realizan una transformación de los datos de entrada, de forma que convierten los datos triaxiales x , y , z en un vector unidimensional [24]. El principal objetivo de esta transformación es reducir las interferencias debidas a la rotación cuando el usuario está en movimiento. En ocasiones esta componente puede tener impacto sobre la señal del acelerómetro, por lo que se reduce transformando la aceleración bruta x , y , z en datos de magnitud vectorial. Esto se puede conseguir calculando la norma Euclídea:

$$\|a\| = \sqrt{x^2 + y^2 + z^2}$$

Al transformar las señales triaxiales en señales de una sola dimensión se reduce el posible error que puede generar la componente de rotación.

Tras comparar diferentes métricas utilizando un modelo de red neuronal simple se obtuvo mejor rendimiento cuando se pasa como entrada las señales directamente sin realizar una transformación o agrupación previa para ambos *datasets*.

4.3.2 MODELO CNN

Las redes convolucionales se caracterizan por presentar capas capaces de extraer características a partir de los datos de entrada. Las capas de la CNN realizan esta extracción mediante operaciones de convolución utilizando filtros (*kernels*). Las redes convolucionales más conocidas suelen utilizarse para extraer características de imágenes en 2 dimensiones, pero en nuestro caso emplearemos una red convolucional de una dimensión Conv1D, donde la capa de entrada estará compuesta por datos de series temporales.

El time step y las características de los datos de entrenamiento se pasarán a una capa densa de 64 neuronas. A continuación, se aplicará una capa de Batch Normalization para normalizar los datos antes de pasar a las capas convolucionales. La normalización se aplica con el objetivo de aumentar la estabilidad del modelo y acelerar la convergencia. Introducimos dos capas Conv1D de 64 filtros con una función de activación ReLu. Estas capas realizarán el mapeo y la representación de las entradas de los sensores para la extracción de características. La capa Conv1D desplaza el filtro en una única dirección, lo que resulta en una salida de 2 dimensiones (time step y número de filtros).

A continuación, se incorpora una capa Dropout con un valor de 0.5 y un MaxPooling unidimensional que reducirá el tamaño de las características aprendidas sin perder exactitud. Dado que la salida resultante también es de 2 dimensiones, aplicamos una capa Flatten cuya salida tendrá un tamaño igual al producto del time step por el número de filtros.

Finalmente, introducimos una capa densa a la que se le pasa como argumento el número de actividades que se desean etiquetar. Para esta última capa, aplicamos la función softmax, ya que es muy útil para la clasificación multiclase.

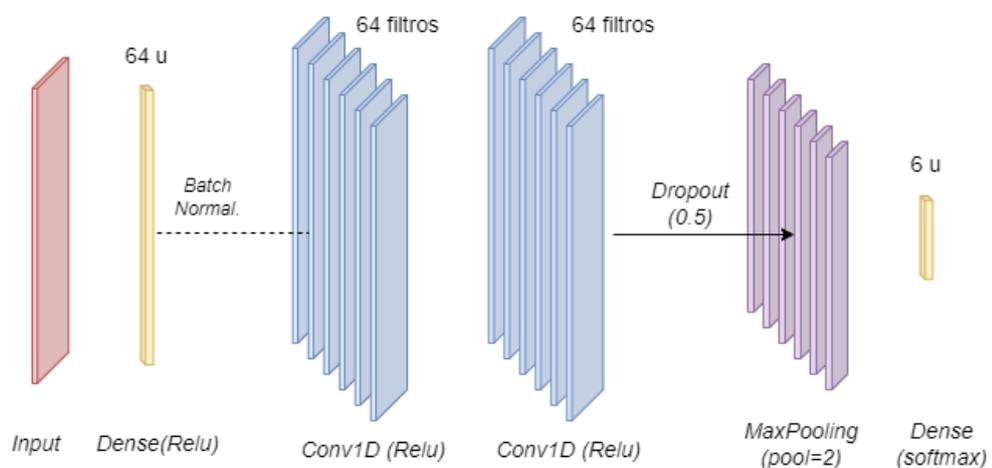


Figura 39: Arquitectura modelo CNN

4.3.3 MODELO LSTM

LSTM (*Long Short Term Memory*) son un tipo de redes neuronales recurrentes que pueden retener información de estados previos y utilizarla para tomar decisiones en el siguiente paso. Esta capacidad de su arquitectura les ayuda a aprender de manera más eficiente los parámetros. Una variante de este modelo es la red LSTM Bidireccional, que procesa secuencias de datos tanto hacia adelante como hacia atrás. Consiste en dos capas LSTM que procesan la secuencia de izquierda a derecha y de derecha a izquierda, fusionando los resultados para obtener una salida final. Gracias a esta red, es posible obtener información sobre eventos pasados y futuros, algo que no es posible con una red LSTM simple.

En este modelo, los datos de entrada se introducen en una capa densa con 64 neuronas. A continuación, se agrega una capa LSTM con 256 unidades. La

salida de la capa LSTM se pasa a una capa Dropout que desactiva el 50% de las neuronas para evitar el sobreajuste. Luego, se agrega una capa densa con 100 neuronas utilizando la función de activación ReLU. Por último, se agrega la capa de salida que tiene el número de actividades como salida y se aplica la función softmax.

Para obtener el modelo BiLSTM, simplemente se agrega el término "Bidirectional" antes de la capa LSTM, mientras que el resto de las capas permanecen iguales.

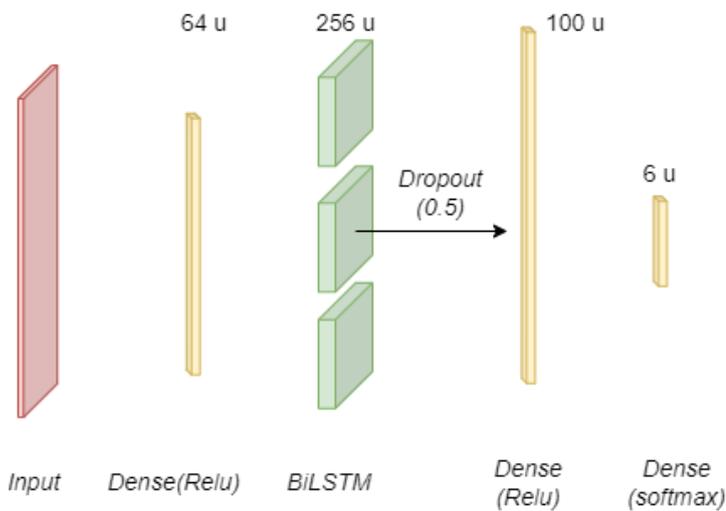


Figura 40: Arquitectura modelo BiLSTM

4.3.4 MODELO HÍBRIDO CNN-LSTM

Algunos autores han propuesto la implementación de un modelo híbrido que combina redes convolucionales y recurrentes en la arquitectura de la red [25]. La combinación de CNN y LSTM brinda al modelo la capacidad de extraer características locales y tener en cuenta aspectos de los datos secuenciales a largo plazo, lo cual es beneficioso para los datos de reconocimiento de actividades humanas (HAR) basados en series temporales.

El modelo híbrido propuesto combina la red Conv1D con el modelo BiLSTM. Se ha optado por utilizar el modelo BiLSTM debido a su mayor robustez y precisión en comparación con el modelo LSTM, según indican algunos autores [26]. Esto sugiere que el rendimiento de la clasificación podría ser mejor con un modelo bidireccional que con uno simple.

Por lo tanto, hemos procedido a concatenar los dos modelos mencionados anteriormente y analizar su rendimiento. La arquitectura de este modelo constará de una capa Dense con 64 neuronas, BatchNormalization, dos capas Conv1D, Dropout, MaxPooling, BiLSTM, Dropout, una capa Dense con 100

neuronas y la capa de salida. En la Figura 41 se muestra una representación aproximada de esta arquitectura.

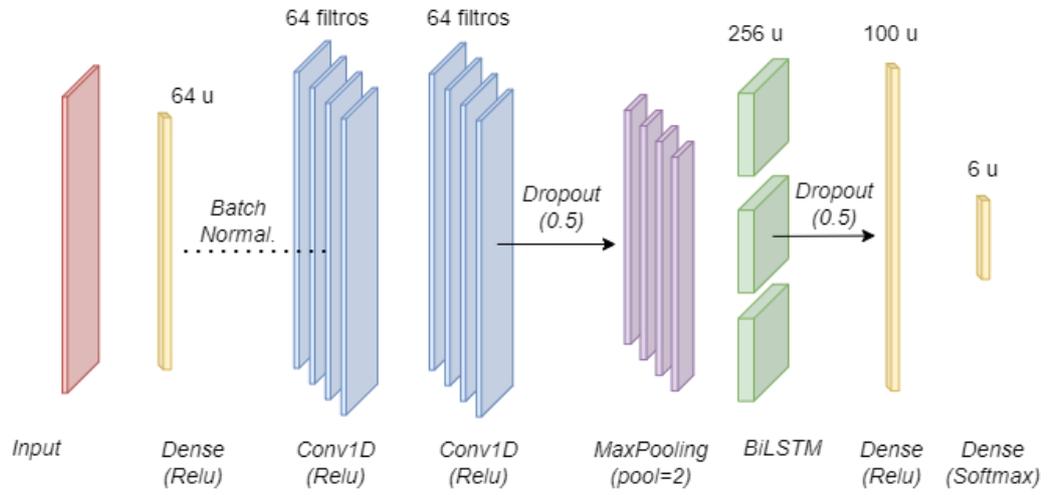


Figura 41: Arquitectura del modelo híbrido

CAPÍTULO 5: RESULTADOS EXPERIMENTALES

5.1 ELECCIÓN DE MÉTRICAS DE CLASIFICACIÓN

Como se ha observado en Figura 37 y Figura 38 , las muestras en el *dataset* WISDM no están distribuidas de manera equitativa entre las actividades en comparación con el *dataset* UCI-HAR. Si la clase con el mayor número de muestras se predice correctamente la mayor parte del tiempo, pero las clases minoritarias se clasifican incorrectamente, la exactitud (*accuracy*) seguirá siendo alta. Debido a esto, la exactitud puede no ser confiable en algunos casos, por lo que es importante tener en cuenta otras métricas adicionales al sacar conclusiones.

Entre las métricas más utilizadas, se destaca el F1-score, que considera los falsos positivos y falsos negativos. El F1-score combina dos medidas basadas en el número total de muestras correctamente etiquetadas, que son la precisión (*precision*) y la exhaustividad (*recall*):

$$Precision = \frac{VerdaderosPositivos}{VerdaderosPositivos + VerdaderosNegativos}$$

$$Recall = \frac{VerdaderosPositivos}{VerdaderosPositivos + FalsosPositivos}$$

$$F_1 = \sum_i 2 * w_i * \frac{precision_i * recall_i}{precision_i + recall_i}$$

La métrica F1-score tiene en cuenta el "peso" de cada clase en el modelo en función de su proporción de muestras. En esta métrica, el parámetro $w_i = n_i/N$ representa la proporción del número de muestras en la clase i , donde n_i es el número de muestras de la clase y N es el número total de muestras.

Otra medida derivada de esta última es el Macro average F1-score, donde todas las clases se tratan por igual independientemente del número de muestras de cada clase. Se obtiene calculando el promedio de los valores F1-score individuales de cada clase.

5.2 EVALUACIÓN DEL ENTRENAMIENTO

Para clasificar las distintas clases, utilizamos 4 modelos de red: CNN, LSTM, BiLSTM y CNN-BiLSTM. Todos los modelos fueron entrenados y ajustados durante 20 épocas.

En primer lugar, empleamos la métrica F1-score para evaluar el rendimiento de nuestros clasificadores.

	UCI HAR	WISDM
LSTM	0.89	0.93
BiLSTM	0.91	0.93
CNN	0.92	0.93
CNN-BiLSTM	0.94	0.96

Tabla 4: Cálculo F1-score de los modelos neuronales para cada dataset

Al comparar los valores de la Tabla 4, podemos observar que las redes diseñadas actúan mejor con el dataset de WISDM que con el UCI-HAR. En el caso del dataset WISDM, los resultados de la métrica F1-score fueron de 0.93 para las redes CNN, LSTM y BiLSTM, y de 0.96 para el modelo híbrido CNN-BiLSTM. Por otro lado, en el caso del dataset UCI-HAR, se obtuvieron métricas de 0.89, 0.91, 0.92 y 0.94 para las redes LSTM, BiLSTM, CNN y CNN-LSTM, respectivamente.

Si bien no se observan diferencias muy significativas entre los resultados, es notable que al incorporar una capa convolucional en la arquitectura de la red, la métrica F1-score evaluada aumenta en ambos datasets.

A continuación, procederemos a comparar los valores de F1-score obtenidos para cada actividad en cada dataset por separado.

	LSTM	BiLSTM	CNN	CNN-BiLSTM
Walking	0.90	0.96	0.97	0.97
Upstairs	0.86	0.93	0.95	0.97
Downstairs	0.97	0.94	0.94	0.94
Sitting	0.82	0.81	0.83	0.86
Standing	0.82	0.85	0.85	0.90
Laying	0.95	0.96	0.98	0.98

Tabla 5: Métrica F1-score para las distintas actividades con UCI HAR dataset

	LSTM	BiLSTM	CNN	CNN-BiLSTM
Walking	0,82	0,81	0,83	0,91
Jogging	0,99	0,99	0,97	0,99
Upstairs	0,99	0,99	0,99	0,99
Downstairs	0,98	0,99	0,99	0,99
Sitting	0,81	0,83	0,80	0,90
Standing	0,97	0,97	0,98	0,99

Tabla 6: Métrica F1-score para las distintas actividades con WISDM dataset

Tras comparar los valores obtenidos en las tablas anteriores, se comprueba que el modelo híbrido CNN-BiLSTM clasifica las diferentes actividades con una eficacia muy alta. Esto se puede distinguir claramente en la Tabla 6, donde el modelo híbrido ha obtenido una puntuación F1 de 0.99 para varias actividades, lo cual indica un desempeño casi perfecto. Los demás modelos también muestran puntuaciones bastante elevadas en el *dataset* WISDM, siendo la más baja 0.8 al clasificar la actividad de sentarse con la red convolucional.

Es importante destacar que las puntuaciones de F1 para todas las actividades en ambos *datasets* son iguales o superiores a 0.8, lo que indica que todos los modelos tienen un buen rendimiento en términos de precisión y exhaustividad para los *datasets* estudiados.

Resulta interesante observar que los resultados obtenidos con los modelos LSTM y BiLSTM son similares. En ambos *datasets*, la clasificación realizada por cada actividad es muy parecida, lo que indica que no se puede afirmar con certeza que la aplicación de un modelo LSTM bidireccional mejore el algoritmo de clasificación, como se suponía inicialmente.

Además, evaluaremos los resultados obtenidos utilizando la métrica de precisión. A continuación se muestran las métricas de precisión para cada modelo de red utilizando los datos del *dataset* UCI-HAR:

	<i>LSTM</i>	<i>BiLSTM</i>	<i>CNN</i>	<i>CNN-BiLSTM</i>
<i>Walking</i>	1,00	0,99	1,00	1,00
<i>Upstairs</i>	0,77	0,92	0,97	0,99
<i>Downstairs</i>	0,96	0,89	0,89	0,90
<i>Sitting</i>	0,79	0,80	0,87	0,89
<i>Standing</i>	0,86	0,81	0,85	0,90
<i>Laying</i>	0,99	1,00	0,99	0,96

Tabla 7: Métrica *precision* para las distintas actividades con UCI HAR dataset

Observando las etiquetas "*Laying*" y "*Walking*" en la Tabla 7, podemos ver que algunas de las etiquetas obtuvieron una precisión del 100% con cualquiera de las redes. Un valor de precisión del 100% implica que el modelo es muy eficiente al identificar los casos positivos (los casos en los que el usuario está tumbado). Los altos valores de precisión podrían deberse a que el acelerómetro y el giroscopio proporcionan valores muy específicos cuando alguien está tumbado o caminando.

La precisión por sí sola puede ser engañosa si no se tienen en cuenta otras métricas como *recall* o F1, ya que un modelo puede tener una alta precisión al clasificar correctamente las instancias positivas, pero puede tener un bajo valor de *recall* si no está capturando correctamente todas las instancias en el conjunto de datos. En nuestro caso, al analizar los valores obtenidos con la métrica F1, podemos afirmar que tanto la precisión como la exhaustividad son elevadas tanto para la clase "*Laying*" como para "*Walking*" (debido a los altos valores de F1), lo que nos permite estimar que el rendimiento de las redes neuronales es muy bueno para clasificar las actividades de tumbarse y andar. En cuanto a los modelos neuronales, el CNN-BiLSTM muestra la mayor precisión para la mayoría de las actividades, alcanzando resultados de 1.00 para "*Walking*" y 0.99 para "*Upstairs*". Las redes BiLSTM y CNN también obtienen puntuaciones sólidas en el rango de 0.80 a 1.00 para las diferentes actividades.

	<i>UCI HAR</i>	<i>WISDM</i>
<i>LSTM</i>	0.88	0.94
<i>BiLSTM</i>	0.90	0.94
<i>CNN</i>	0.92	0.94
<i>CNN-BiLSTM</i>	0.94	0.97

Tabla 8: Exactitud de los modelos neuronales para cada dataset

Por último, evaluaremos la exactitud del modelo al exponerlo a un conjunto de pruebas. Analizaremos la bondad de las matrices de confusión para cada red neuronal y la evolución de la exactitud a lo largo del entrenamiento.

Al observar las gráficas que representan la exactitud durante las épocas de entrenamiento, se puede notar que las gráficas de entrenamiento y validación no difieren significativamente entre sí. También se observa que la exactitud aumenta progresivamente con el paso de las épocas, lo que indica que el modelo no presenta sobreajuste (*overfitting*).

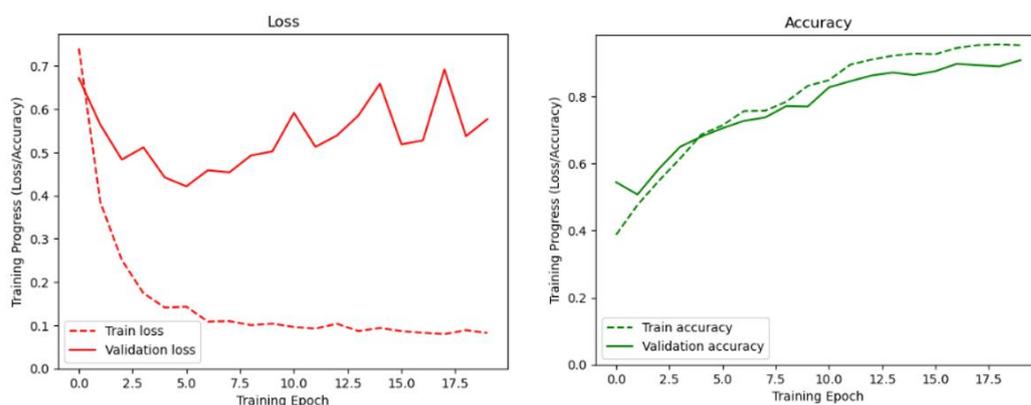


Figura 42: Representación de loss y accuracy para UCI HAR con modelo CNN-BiLSTM

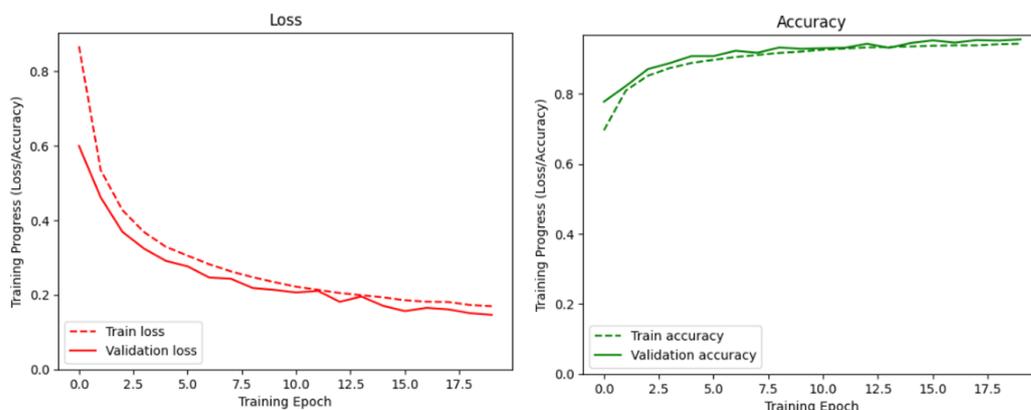


Figura 43: Representación de los y accuracy para WISDM con modelo CNN-BiLSTM

De acuerdo con la Tabla 8, el modelo CNN-BiLSTM muestra los valores más altos de exactitud, con un máximo de 0.94 en el *dataset* UCI-HAR y 0.97 en WISDM. Si comparamos las matrices de confusión para el *dataset* UCI-HAR, las actividades que la red predijo mejor fueron "Laying" y "Downstairs", mientras que las que presentaron una clasificación menos precisa fueron "Standing" y "Sitting". En la mayoría de los casos, los modelos confundieron la actividad de

estar de pie con la acción de sentarse, posiblemente debido a que ambas actividades implican posiciones relativamente estacionarias del cuerpo con poco o ningún movimiento. En el caso de WISDM, la red clasificó correctamente casi todas las instancias de las etiquetas "Standing", "Jogging", "Upstairs" y "Laying". Las dos clases restantes presentaron una exactitud que varió entre 0.71 y 0.92.

Los resultados de las matrices de confusión resaltan la capacidad de los modelos propuestos para reconocer diferentes actividades con una alta exactitud.

MATRICES DE CONFUSIÓN PARA EL DATASET UCI HAR



Figura 44: Matriz de confusión CNN

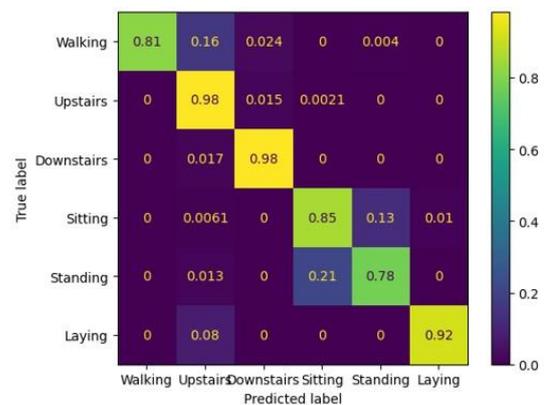


Figura 45: Matriz de confusión LSTM

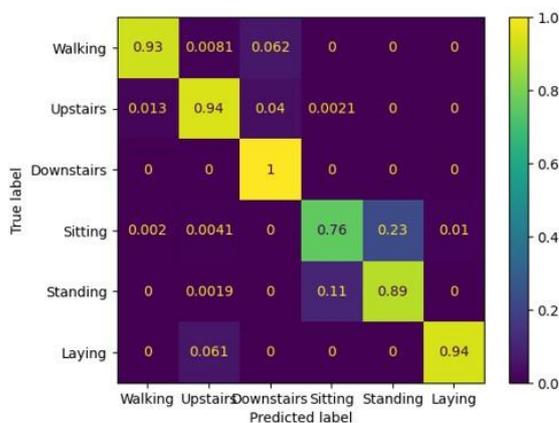


Figura 46: Matriz de confusión BiLSTM

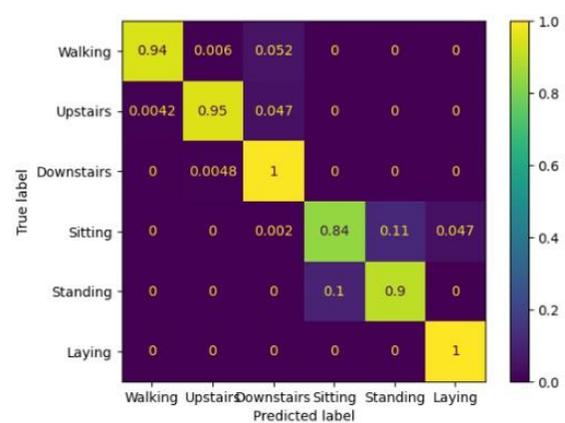


Figura 47: Matriz de confusión CNN-BiLSTM

MATRICES DE CONFUSIÓN PARA EL DATASET WISDM

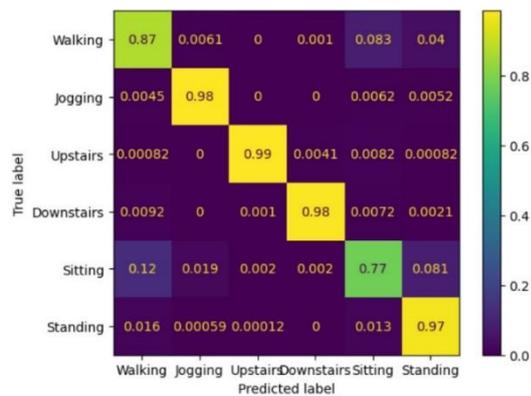
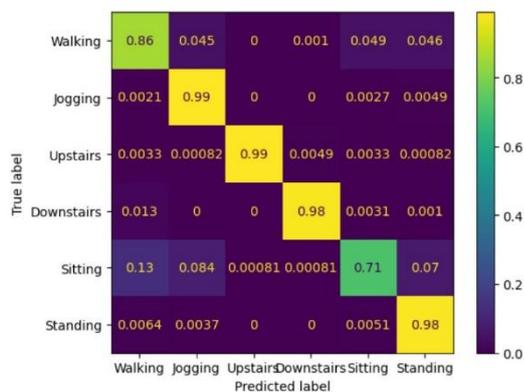


Figura 48: Matriz de confusión CNN Figura 49: Matriz de confusión LSTM

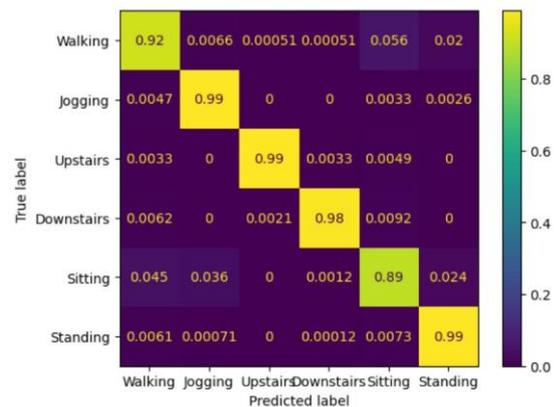
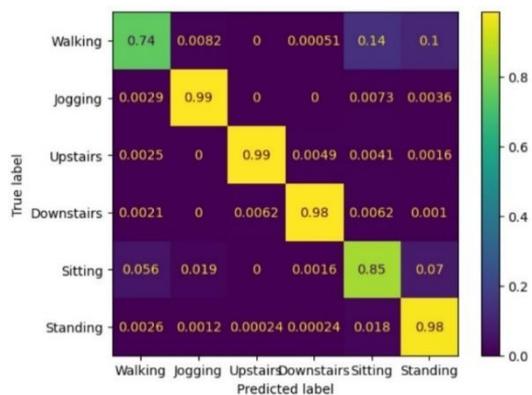


Figura 50: Matriz de confusión BiLSTM Figura 51: Matriz de confusión CNN-BiLSTM

Para confirmar la eficacia del modelo neuronal planteado, se decidió realizar una revisión de la literatura para los *datasets* utilizados en este estudio.

Al comparar estos resultados con estudios previos, se puede observar que las diferencias no son muy significativas. Por ejemplo, en un artículo [27], se reportó una exactitud del 92% para el modelo CNN-LSTM en el *dataset* UCI-HAR. En otro estudio [28] que comparó diferentes métodos de clasificación utilizando aprendizaje profundo y aprendizaje automático clásico en los *datasets* WISDM y UCI-HAR, se encontró que el modelo LSTM obtuvo los

mejores resultados de exactitud con un 95.45% y 95.13%, respectivamente. En [29], se propuso un modelo que combinaba capas convolucionales con LSTM para el reconocimiento de actividades humanas en 3 *datasets* diferentes, obteniendo puntuaciones de F1 de 95.78% para UCI-HAR y 95.85% para WISDM.

En la investigación realizada por Ihianle, se propuso una arquitectura multicanal CNN-BiLSTM y se evaluó su desempeño utilizando los conjuntos de datos WISDM y MHEALTH [14]. Después de completar el entrenamiento, se logró una puntuación media de F1 del 99% para el conjunto de datos WISDM en la época número 100. En otras investigaciones, como [30], se comparan diferentes técnicas de *Machine Learning* y *Deep Learning* con los mismos *datasets*. En este caso se consiguió la mejor exactitud con un modelo CNN dando como resultado un 93.32% de *accuracy* para el *dataset* UCI HAR y un 97.72% para WISDM.

Al realizar una revisión de la literatura, se encontró que el rendimiento de varios modelos propuestos en este trabajo era similar o incluso mejor en comparación con otros artículos que también utilizaban estos mismos conjuntos de datos. Además, es importante destacar que se obtuvieron resultados de precisión y puntuación F1 muy similares o incluso superiores a los informados en los artículos mencionados, y esto se logró con tan solo 20 épocas de entrenamiento. Estos resultados sugieren que la red neuronal tiene una buena capacidad de generalización y clasificación.

5.3 MODIFICACIONES Y MEJORAS

Tras obtener los resultados anteriores se decidió evaluar el rendimiento del modelo aplicando una arquitectura alternativa: los bloques residuales. Como se ha comentado anteriormente un modelo de red residual o ResNet es una arquitectura de red neuronal profunda que utiliza conexiones residuales para facilitar el entrenamiento y mejorar el rendimiento a medida que se aumenta la profundidad de la red. La novedad de este tipo de arquitectura es la introducción de conexiones residuales que permiten que la información del flujo de datos se salte una o más capas, lo que facilita el aprendizaje de las diferencias o residuos. Al agregar el residuo al resultado de una capa anterior, la red tiene la capacidad de aprender funciones más complejas y profundas. Esto permite la construcción de redes neuronales extremadamente profundas sin sufrir una degradación significativa en el rendimiento.

Como se comentó anteriormente, los modelos neuronales residuales cuentan con un bloque residual con diferentes capas secuenciales y por otro lado una

conexión residual que sumará la entrada en un momento determinado del modelo. A la hora de diseñar esta arquitectura se decidió diseñar una estructura convolucional-batch_normalization-relu (CBR) que se repetirá de forma modular. Se ha integrado una capa batch normalization a este módulo para acelerar el proceso de entrenamiento y la función de activación relu evitará el problema de desvanecimiento del gradiente. En la última capa CBR la función relu se omite, ya que la función de activación se añadirá tras la adición de la conexión residual para evitar la pérdida de información relevante.

Para que la dimensión de salida de las dos capas convolucionales sea coherente con la dimensión de entrada original, se realiza la misma convolución bidimensional en el enlace residual.

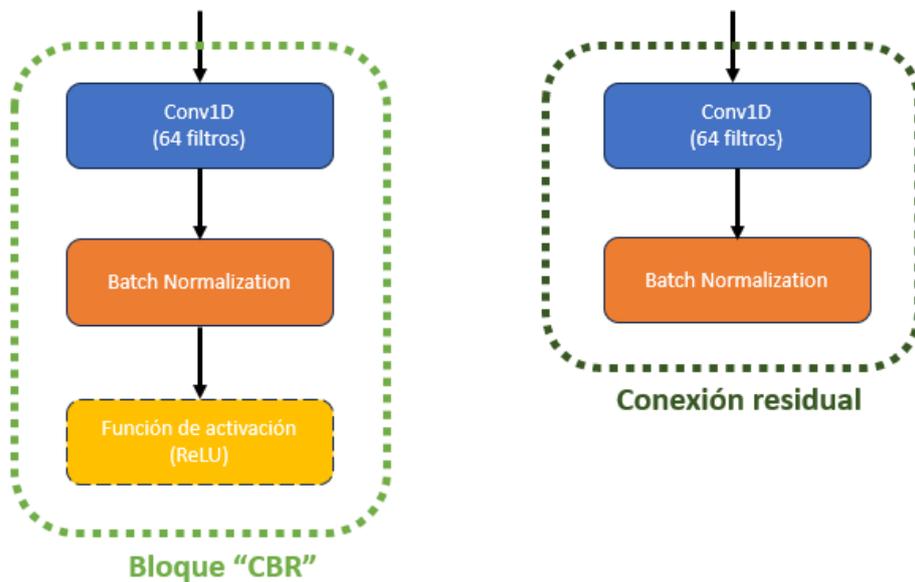


Figura 52: Bloque y conexión residuales utilizada en el modelo

La idea de la arquitectura planteada se basa en sustituir los bloques convolucionales de los modelos neuronales explicados anteriormente por bloques residuales junto con su conexión residual correspondiente. A continuación se incorporarán las capas recurrentes y optimizadores que se plantearon en un principio. Es decir, tras sumar la conexión residual con la salida del módulo residual y aplicar una función ReLU se integrará el modelo BiLSTM de apartados anteriores. Con esto se pretende capturar información de las secuencias temporales.

Varios artículos han estudiado el efecto de variar diferentes características de las redes residuales para comparar su eficacia. [31], [32] Entre estas

variaciones podemos destacar los modelos residuales con reducción de cuello de botella o *bottleneck*.

La idea principal detrás de las redes residuales con *bottleneck* es reducir la carga computacional y mejorar la eficiencia de los cálculos en redes profundas. Esto se logra mediante el uso de capas de reducción de cuello de botella que reducen la dimensión espacial de los mapas de características antes y después de las capas convolucionales. Estas capas generalmente consisten en una capa convolucional 1x1 seguida de una capa convolucional 3x3, y finalmente otra capa convolucional 1x1 para restaurar la dimensión original.

Para estudiar estas diferencias se entrenarán las bases de datos con un modelo residual simple donde las capas CBR presenten kernels convolucionales de dimensión 3 y por otro lado un modelo residual con reducción de cuello de botella cuyas capas convolucionales tendrán kernels de tamaño 1,3 y 1 en este orden.

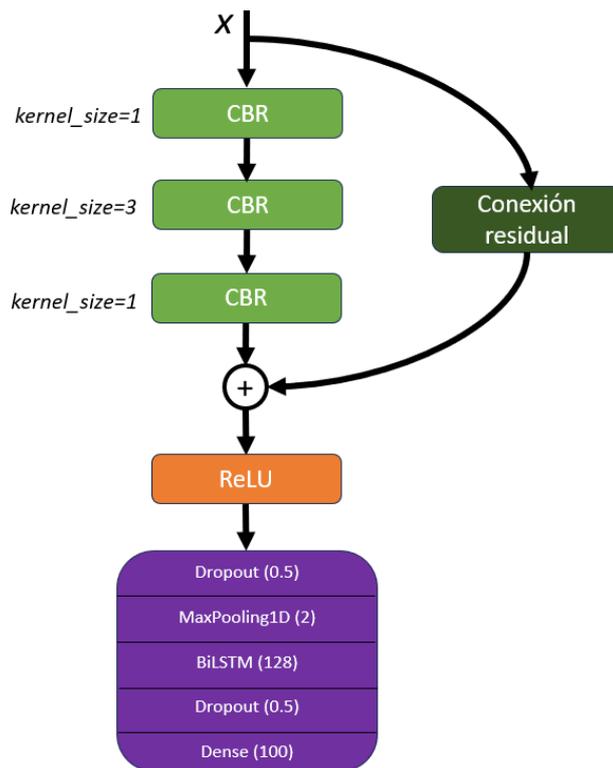


Figura 53: Arquitectura modelo residual con "bottleneck" planteado

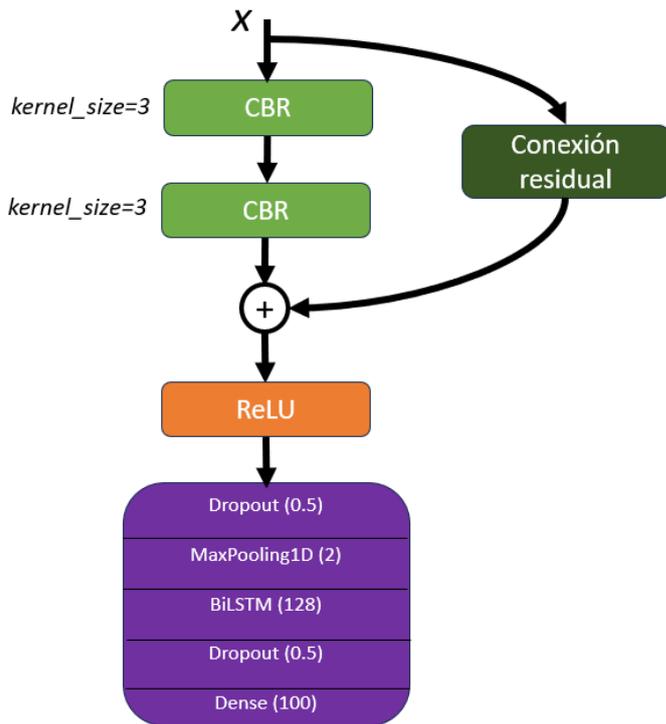


Figura 54: Arquitectura modelo residual simple planteado

La comparación de los dos modelos neuronales residuales para el reconocimiento de la actividad humana a partir de los *datasets* escogidos arrojó resultados interesantes.

Los resultados mostraron que el modelo simple alcanzó una precisión de prueba del 92.46% para el conjunto de datos UCI HAR y un 97.47% para WISDM. Por otro lado, el modelo con reducción de cuello de botella logró un 97.72% para WISDM y un 93.5% para UCI HAR.

Si se comparan los resultados obtenidos con los conseguidos a partir de los modelos convolucionales y recurrentes anteriores, comprobamos que en el caso del modelo residual con reducción de cuello de botella la exactitud es más o menos similar para los dos *datasets*. Los bloques residuales permiten el aprendizaje de conexiones directas entre capas, lo que ayuda a mitigar el problema de la desaparición del gradiente y facilita el entrenamiento de redes más profundas. Además, la utilización de la función de activación ReLU y la normalización por lotes (batch normalization) contribuyen a mejorar la estabilidad y la capacidad de generalización de los modelos, lo que se puede reflejar de forma positiva en la exactitud del modelo.

Al analizar las gráficas de pérdida (loss) y exactitud (accuracy), se observa que la pérdida es inicialmente alta en las primeras épocas y luego disminuye a valores normales. La alta pérdida inicial podría deberse a varios factores, como la inicialización aleatoria de los pesos, la complejidad del problema o la

dificultad para aprender patrones en las primeras etapas del entrenamiento. Sin embargo, a medida que el modelo se entrena, los pesos se ajustan y la pérdida disminuye.

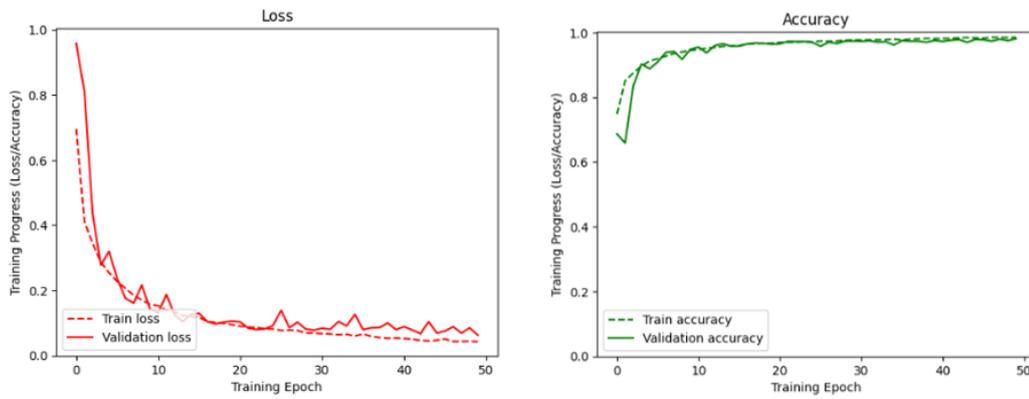


Figura 55: Representación loss y accuracy para WISDM con modelo residual con reducción de cuello de botella

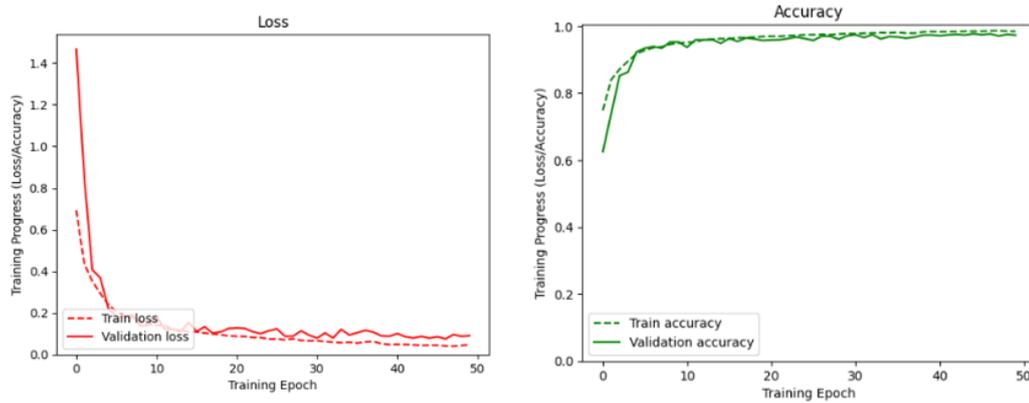


Figura 56: Representación de loss y accuracy para WISDM con modelo residual simple

CAPÍTULO 6: CONCLUSIONES

En el presente trabajo se propusieron diferentes redes neuronales para el reconocimiento de actividades humanas utilizando datos de acelerómetros y giroscopios presentes en dispositivos como smartphones y smartwatches. El reconocimiento de actividades humanas mediante técnicas de *Deep Learning* puede ser una herramienta útil para monitorear las acciones diarias de las personas, especialmente en el caso de personas de la tercera edad.

Con el objetivo de evaluar la capacidad de generalización y efectividad de los modelos propuestos, se utilizaron dos conjuntos de datos en este experimento: UCI-HAR y WISDM. Dado que la elección del mejor modelo de red depende de varios factores, se emplearon diferentes métricas para evaluar el rendimiento de las redes.

De acuerdo con los resultados expuestos en el CAPÍTULO 5, el modelo híbrido convolucional-bidireccional LSTM parece ser la mejor opción para clasificar actividades humanas en ambos conjuntos de datos. Este modelo proporciona el mayor valor de precisión y menos clasificaciones erróneas en comparación con otras arquitecturas de red. Además, el modelo logró los mejores resultados en términos de Macro average F1 y una puntuación de F1 de 0.99 en varias de las actividades. Además, se valoraron diferentes alternativas como los modelos residuales para mejorar la capacidad de generalización y exactitud del modelo.

Tras realizar el presente trabajo y extraer los resultados adecuados se concluyó que se han cumplido los objetivos planteados en un primer momento en el informe:

- ✓ Se ha diseñado un modelo de aprendizaje profundo que reconozca actividades humanas, como andar, subir las escaleras o estar tumbado.
- ✓ Se han obtenido los datos de los sensores inerciales de la pulsera inteligente elegida.
- ✓ Se han obtenido datos relacionados con parámetros de salud como calidad de sueño o pasos para conocer más aspectos sobre el bienestar del paciente.
- ✓ Se ha realizado una búsqueda exhaustiva sobre los modelos neuronales más utilizados hoy en día para la clasificación de actividades humanas, así como un estado del arte de diferentes pulseras inteligentes, aplicaciones de actividad y bases de datos orientadas al reconocimiento de actividades humanas.

CAPÍTULO 7: LÍNEAS FUTURAS

A pesar de los buenos resultados obtenidos, siempre se debe aspirar a la mejora continua. En este apartado, se presentan algunas líneas de investigación que podrían ser interesantes de cara a mejorar los sistemas de reconocimiento de actividades humanas:

- En futuras líneas de investigación, podría ser interesante explorar el efecto de combinar diferentes arquitecturas de redes o utilizar conjuntos de datos con un mayor número de actividades. Aunque las redes de aprendizaje profundo son la técnica dominante para el reconocimiento de actividades humanas, es importante continuar investigando y realizando más estudios para evaluar diferentes métodos y lograr resultados óptimos en la clasificación.
- Como se ha comentado, la aplicación Gadgetbridge proporciona, además de varios datos relacionados con los parámetros de salud del individuo, una etiqueta aproximada de la actividad que ha realizado el individuo a lo largo del día. Una posible línea podría ser emplear los datos recogidos por los sensores inerciales de la pulsera, etiquetarlos y comparar con los resultados obtenidos con la app Gadgetbridge, ya que es una herramienta que tiene un gran potencial pero no es del todo fiable a la hora de clasificar actividades diferentes de andar o estar sentado.
- Considerar la inclusión de otros sensores complementarios, como el sensor de frecuencia cardíaca, para enriquecer la información capturada y mejorar la precisión en la detección de actividades físicas y el monitoreo de la salud, de forma que el reconocimiento de actividades pudiera darse teniendo en cuenta las señales inerciales y las fisiológicas.
- Realizar estudios de validación y evaluación del desempeño de los modelos propuestos en poblaciones específicas, como adultos mayores o personas con condiciones de salud particulares, para evaluar su utilidad en entornos clínicos y de atención médica.

En resumen, aunque los resultados obtenidos son alentadores, es importante tener en cuenta que siempre hay espacio para la mejora y la exploración de nuevas áreas de investigación. La búsqueda constante de la excelencia y el progreso continuo son esenciales para avanzar en el reconocimiento de actividades humanas y contribuir al bienestar de las personas en su rutina diaria.

BIBLIOGRAFÍA

- [1] "INEbase / Demografía y población /Cifras de población y Censos demográficos /Encuesta continua de hogares / Últimos datos." https://www.ine.es/dyngs/INEbase/es/operacion.htm?c=Estadistica_C&cid=1254736176952&menu=ultiDatos&idp=1254735572981 (accessed jun. 27, 2023).
- [2] S. Hurtado, J. García-Nieto, A. Popov, and I. Navas-Delgado, "Human Activity Recognition From Sensorised Patient's Data in Healthcare: A Streaming Deep Learning-Based Approach," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 8, no. 1, p. 23, 2023, doi: 10.9781/IJIMAI.2022.05.004.
- [3] M. K. A. Ramesh, R. G. S. Prem, R. A. A., and D. M. P. Gopinath, "1D Convolution approach to human activity recognition using sensor data and comparison with machine learning algorithms," *International Journal of Cognitive Computing in Engineering*, vol. 2, pp. 130–143, Jun. 2021, doi: 10.1016/J.IJCCCE.2021.09.001.
- [4] S. Benavidez and D. Mccreight, "A Deep Learning Approach for Human Activity Recognition Project Category: Other (Time-Series Classification)".
- [5] J. Duque Domingo, J. Gómez García-Bermejo, and E. Zalama Casanova, *Visión artificial mediante Aprendizaje Automático con TensorFlow y Python*, 1st ed., vol. 1. Ra-Ma S.A., 2023.
- [6] David García, "EIAROB CARTIF," 2022.
- [7] Huawei, "Huawei Health Kit."
- [8] K. Xia, J. Huang, and H. Wang, "LSTM-CNN Architecture for Human Activity Recognition", doi: 10.1109/ACCESS.2020.2982225.
- [9] P. Agarwal and M. Alam, "A Lightweight Deep Learning Model for Human Activity Recognition on Edge Devices," *Procedia Comput Sci*, vol. 167, pp. 2364–2373, 2020, doi: 10.1016/J.PROCS.2020.03.289.
- [10] F. Demrozi, G. Pravadelli, S. Member, A. Bihorac, and P. Rashidi, "Human Activity Recognition Using Inertial, Physiological and Environmental Sensors: A Comprehensive Survey", doi: 10.1109/ACCESS.2020.3037715.
- [11] A. Jain and V. Kanhangad, "Human Activity Classification in Smartphones Using Accelerometer and Gyroscope Sensors," *IEEE Sens J*, vol. 18, no. 3, pp. 1169–1177, Feb. 2018, doi: 10.1109/JSEN.2017.2782492.
- [12] Y. Bengio, "Deep Learning of Representations: Looking Forward," 2013, Accessed: May 07, 2023. [Online]. Available: <http://oreillynet.com/pub/e/2538>

- [13] B. Oluwalade, S. Neela, J. Wawira, T. Adejumo, and S. Purkayastha, "Human Activity Recognition using Deep Learning Models on Smartphones and Smartwatches Sensor Data".
- [14] I. K. Ihianle, A. O. Nwajana, S. H. Ebenuwa, R. I. Otuka, K. Owa, and M. O. Orisatoki, "A deep learning approach for human activities recognition from multimodal sensing devices," *IEEE Access*, vol. 8, pp. 179028–179038, 2020, doi: 10.1109/ACCESS.2020.3027979.
- [15] D. Burns, P. Boyer, C. Arrowsmith, and C. Whyne, "Personalized Activity Recognition with Deep Triplet Embeddings," *Sensors (Basel)*, vol. 22, no. 14, Jul. 2022, doi: 10.3390/S22145222.
- [16] "Target Prediction using Single-layer Perceptron and Multilayer Perceptron | by Ananda Hange | Nerd For Tech | Medium." <https://medium.com/nerd-for-tech/flux-prediction-using-single-layer-perceptron-and-multilayer-perceptron-cf82c1341c33> (accessed May 01, 2023).
- [17] "Comprehensive guide to LSTM & RNNs." <https://www.turing.com/kb/comprehensive-guide-to-lstm-rnn> (accessed May 07, 2023).
- [18] "Bluetooth de baja energía (BLE): Introducción, (2ª parte)." <https://www.electrodaddy.com/bluetooth-baja-energia-ble-introduccion-2/> (accessed Jun. 27, 2023).
- [19] "Bluetooth Low Energy, théorie et pratique | hackimalement." <https://hackimalement.wordpress.com/2015/06/13/bluetooth-low-energy-theorie-et-pratique/> (accessed Jun. 27, 2023).
- [20] "Assigned Numbers | Bluetooth® Technology Website." <https://www.bluetooth.com/specifications/assigned-numbers/> (accessed May 28, 2023).
- [21] "miband4/constants.py at master · BeanCounterTop/miband4 · GitHub." <https://github.com/BeanCounterTop/miband4/blob/master/constants.py> (accessed Jun. 05, 2023).
- [22] "4lhc/MiBand_HRX: Library to work with Xiaomi MiBand HRX Accelerometer Data." https://github.com/4lhc/MiBand_HRX (accessed Jun. 05, 2023).
- [23] M. Zeng *et al.*, "Convolutional Neural Networks for human activity recognition using mobile sensors," *Proceedings of the 2014 6th International Conference on Mobile Computing, Applications and Services, MobiCASE 2014*, pp. 197–205, Jan. 2015, doi: 10.4108/ICST.MOBICASE.2014.257786.
- [24] S. M. Lee, S. M. Yoon, and H. Cho, "Human activity recognition from accelerometer data using Convolutional Neural Network," in *2017 IEEE*

International Conference on Big Data and Smart Computing, BigComp 2017, 2017. doi: 10.1109/BIGCOMP.2017.7881728.

- [25] S. Zhang *et al.*, “Deep Learning in Human Activity Recognition with Wearable Sensors: A Review on Advances,” *Sensors* 2022, Vol. 22, Page 1476, vol. 22, no. 4, p. 1476, Feb. 2022, doi: 10.3390/S22041476.
- [26] S. Siami-Namini, N. Tavakoli, and A. S. Namin, “The Performance of LSTM and BiLSTM in Forecasting Time Series,” *Proceedings - 2019 IEEE International Conference on Big Data, Big Data 2019*, pp. 3285–3292, Dec. 2019, doi: 10.1109/BIGDATA47090.2019.9005997.
- [27] Y. J. Luwe, C. P. Lee, and K. M. Lim, “Wearable Sensor-Based Human Activity Recognition with Hybrid Deep Learning Model,” *Informatics*, vol. 9, no. 3, 2022, doi: 10.3390/informatics9030056.
- [28] S. Khare, S. Sarkar, and M. Totaro, “Comparison of Sensor-Based Datasets for Human Activity Recognition in Wearable IoT,” in *IEEE World Forum on Internet of Things, WF-IoT 2020 - Symposium Proceedings*, 2020. doi: 10.1109/WF-IoT48130.2020.9221408.
- [29] K. Xia, J. Huang, and H. Wang, “LSTM-CNN Architecture for Human Activity Recognition,” *IEEE Access*, vol. 8, 2020, doi: 10.1109/ACCESS.2020.2982225.
- [30] A. Ignatov, “Real-time human activity recognition from accelerometer data using Convolutional Neural Networks,” *Appl Soft Comput*, vol. 62, pp. 915–922, Jan. 2018, doi: 10.1016/J.ASOC.2017.09.027.
- [31] S. Mekruksavanich, A. Jitpattanakul, K. Sitthithakerngkiet, P. Youplao, and P. Yupapin, “ResNet-SE: Channel Attention-Based Deep Residual Network for Complex Activity Recognition Using Wrist-Worn Wearable Sensors”, doi: 10.1109/ACCESS.2022.3174124.
- [32] “MDPI - Publisher of Open Access Journals.”
<https://www.mdpi.com/1424-8220/22/2/635/pdf?version=1642167102>
(accessed Jun. 27, 2023).