Facultad de Informática
UNIVERSIDADE DA CORUÑA

TRABAJO FIN DE GRADO
GRADO EN INGENIERÍA INFORMÁTICA
MENCIÓN EN COMPUTACIÓN

La detección de actividades del sueño a través de la aplicación del Aprendizaje Profundo

Estudante: Sandra Gómez Gutiérrez
Dirección: Elena Mª Hernández Pereira

A Coruña, febrero de 2021.

A mi familia

Agradecimientos

Me gustaría agradecer a Elena María Hernández Pereira por su ayuda y paciencia a lo largo de la realización de este trabajo. A mis padres y mis hermanos por estar siempre ahí animándome y confiando en mí. También agradecer a mis compañeros y amigos por hacer estos años mucho más llevaderos.

Resumen

La detección de actividades del sueño en la señal del electroencefalograma (EEG) es una tarea fundamental para la clasificación de los estados de sueño. Esta tarea resulta costosa no solo por el elevado número de actividades que existen sino por la dificultad que entraña la identificación de sus características. La aplicación de métodos automáticos para detectar este tipo de actividades está totalmente justificada. La gran mayoría de estos métodos automáticos identifican características de las actividades del sueño y luego las clasifican. En este trabajo, se identifica una de estas actividades, denominada husos de sueño, a través de imágenes obtenidas a partir de la señal de EEG y con la ayuda del Aprendizaje Profundo mediante redes convolucionales. Esta aproximación se compara con la extracción de características sobre la señal de EEG, de nuevo a través de Aprendizaje Profundo mediante redes convolucionales.

Abstract

The detection of sleep activities in the electroencephalogram (EEG) signal is a fundamental task for the classification of sleep states. This task is costly not only due to the high number of activities that exist but also because of the difficulty involved in identifying their characteristics. The application of automatic methods to detect this type of activity is fully justified. The vast majority of these automated methods identify characteristics of sleep activities and then classify them. In this work, one of these activities, called sleep spindles, is identified through images obtained from the EEG signal and with the help of Deep Learning through convolutional networks. This approach is compared with the extraction of features on the EEG signal, again through Deep Learning using convolutional networks.

Palabras clave:

- Aprendizaje Profundo
- Clasificación
- Huso de sueño

Keywords:

- Deep Learning
- Classification
- Sleep Spindle

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

El sueño es una parte importante de la vida cotidiana, una necesidad biológica que permite restablecer las funciones físicas y psicológicas esenciales para un pleno rendimiento. Durante el sueño existe una ausencia o disminución de movimientos corporales voluntarios y se adopta una postura estereotipada de descanso, distinta en cada especie animal, existiendo una escasa respuesta a estímulos externos de baja intensidad que es reversible, a diferencia del coma.

El sueño es el estado de descanso natural del cerebro observado en los seres humanos y los animales; la falta del mismo influye seriamente en la capacidad de funcionamiento del cerebro humano. Los problemas relacionados con el sueño tienen consecuencias directas en la calidad de vida de las personas, afectando a áreas como la memoria o la capacidad de concentración. Por esta razón, existe un campo de la medicina centrado en su estudio, con el objetivo de comprender el funcionamiento de estos problemas y sus efectos.

El estudio del sueño implica identificar las distintas etapas de sueño y dentro de estas, una serie de actividades que proporcionan información relevante para determinar la existencia de problemas o más concretamente, trastornos de sueño. Dentro de estas actividades se encuentran los husos de sueño que juegan un papel importante en la consolidación de la memoria, y varios estudios demuestran una correlación positiva entre la densidad de los husos de sueño y el coeficiente intelectual [1]. Las anomalías en los patrones de los husos de sueño son biomarcadores de varios trastornos neurodegenerativos como el Alzheimer [2], y su identificación se utiliza en el diagnóstico de muchos trastornos del sueño [3].

Este trabajo se centra en la identificación de los husos de sueño, con el objetivo de mejorar la clasificación de las etapas de sueño y como consecuencia mejorar la precisión en el diagnóstico de ciertas enfermedades de sueño. Esta identificación se realizará a través de Aprendizaje Profundo siguiendo dos aproximaciones. La primera aproximación trabajará con imágenes y la segunda con características de la señal de electroencefalograma (EEG), como representación gráfica y analítica de los husos de sueño.

1.1 Objetivos

El principal objetivo de este trabajo es diseñar modelos de Aprendizaje Profundo mediante redes convolucionales para la identificación de los husos de sueño, utilizando tanto imágenes como una serie de características extraídas ambas de la señal de EEG. Este objetivo se descompone en los siguientes subobjetivos:

- Analizar y procesar la señal de EEG.
- Generar las imágenes y extraer las características que se utilizarán como entrada en los modelos.
- Desarrollar modelos basados en Aprendizaje Profundo que permitan la identificación de los husos de sueño.
- Comparar los modelos desarrollados entre sí y con otros modelos similares existentes.

1.2 Estructura de la memoria

Esta memoria se ha estructurado de la siguiente manera:

- En el capítulo 1 se describe la motivación del proyecto, sus objetivos y la estructura de la memoria.
- En el capítulo 2 se hace una breve introducción sobre la medicina del sueño, se describen los husos de sueño y los trabajos realizados hasta ahora sobre su detección automática.
- En el capítulo 3 se detallan tanto las tecnologías y las herramientas utilizadas para el desarrollo como el conjunto de datos usado en este trabajo.
- En el capítulo 4 se describen la metodología y planificación del proyecto.
- En el capítulo 5 se describe el desarrollo de las dos aproximaciones.
- En el capítulo 6 se describen los resultados obtenidos en ambas aproximaciones.
- En el capítulo 7 se muestran las conclusiones y el trabajo futuro.

Capítulo 2

Descripción del dominio y estado del arte

En este capítulo se hace una breve introducción del dominio de trabajo a través de la descripción de las distintas etapas del sueño y de una de las actividades que se identifican en ellas: los husos de sueño. El capítulo se completa con una descripción de las distintas aproximaciones existentes para la detección automática de esta actividad.

2.1 Medicina del sueño

La medicina del sueño es una especialidad médica dedicada al diagnóstico y tratamiento de las alteraciones y trastornos del sueño [4]. Desde mediados del siglo XX, las investigaciones sobre el sueño han proporcionado un conocimiento cada vez mayor y han respondido muchas preguntas sobre el funcionamiento del sueño y la vigilia.

Los médicos pueden diagnosticar patrones de sueño anormales o cualquier otro problema relacionado con el sueño, con los datos recogidos de un estudio de sueño. El objetivo de estos estudios es detectar los síntomas y diagnosticar los trastornos que puedan estar causando problemas de sueño e impactando en la vida diaria, puesto que un porcentaje alto de la población sufre problemas o patologías relacionadas con el sueño, entre los que se encuentran el insomnio, la narcolepsia o la apnea del sueño [5].

La técnica más utilizada para el estudio del sueño es la polisomnografía, que consiste en el registro simultáneo de señales neurofisiológicas y respiratorias que permiten evaluar la cantidad y calidad del sueño, así como identificar los diferentes eventos respiratorios. Esta prueba se lleva a cabo en laboratorios de sueño de los centros médicos, de forma vigilada por técnicos cualificados y se realiza en horario nocturno, o en el horario habitual del sueño del paciente. Mediante la polisomnografía es posible la clasificación de las etapas de sueño, así como la localización de eventos adicionales de importancia para el diagnóstico, como por

ejemplo los husos de sueño, los complejos K o los despertares.

La realización de la polisomnografía puede tener mayor o menor complejidad dependiendo del número de señales que se registren. En general en todo estudio polisomnográfico las señales principales son el electroencefalograma (EEG), el electrooculograma (EOG) y el electromiograma (EMG).

Electroencefalograma (EEG)

El EEG es la señal que monitoriza la actividad eléctrica del cerebro, a través de electrodos colocados sobre múltiples áreas del cuero cabelludo del paciente.

En el EEG se pueden encontrar diferentes tipos de ondas según el estado de vigilancia en el que se encuentre el sujeto, tales como las ondas alpha, beta, theta y delta, y diferentes actividades transitorias como los husos de sueño o los complejos K. Cada una de ellas va a ayudar a determinar las etapas de sueño, al mismo tiempo que permite diagnosticar los trastornos del sueño.

Las ondas cerebrales se definen como ondas regulares a lo largo del tiempo, caracterizadas por su frecuencia, localización y asociación con varios aspectos del funcionamiento y estado del cerebro. Normalmente las amplitudes de estas ondas están en un rango entre 5 y 200 μV, y un rango de frecuencia entre 1 y 30 Hz. En la Figura 2.1 se muestra un ejemplo de cada una de ellas.

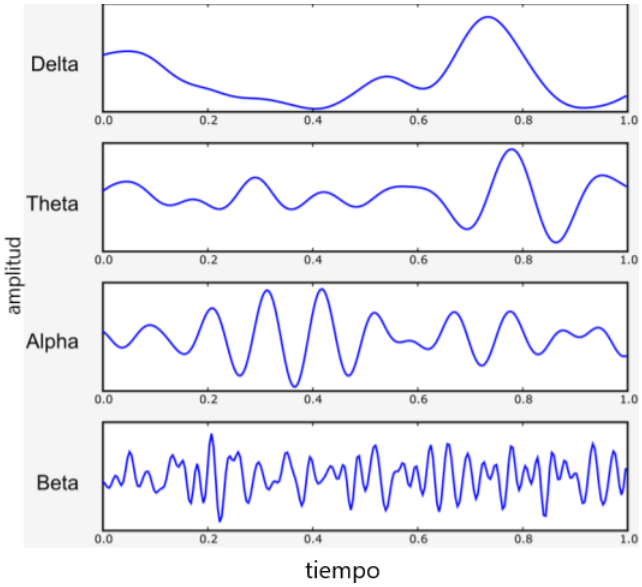

Figura 2.1: Representación de las ondas cerebrales del EEG

Electrooculograma (EOG)

El EOG es la señal que monitoriza los movimientos oculares utilizando la diferencia de potencial entre la córnea y la retina del ojo, caracterizando el ojo como un dipolo rotatorio. Esta señal permite distinguir los diferentes patrones de movimientos oculares que se producen durante algunos períodos del sueño.

Electromiograma (EMG)

El EMG es la señal que monitoriza la actividad neuromuscular asociada a la contracción de los músculos. Se registra mediante un electrodo colocado sobre la superficie de la piel del músculo que se pretende vigilar. En el ámbito del sueño la derivación más utilizada es el EMG submental, porque los músculos de esta zona proporcionan señales de buena calidad que reflejan los cambios producidos en la progresión normal del sueño. A mayores, se utilizan dos derivaciones suplementarias colocadas sobre la tibia para detectar los movimientos de las piernas, característicos del síndrome de piernas inquietas que se produce durante el sueño.

2.1.1 Etapas del sueño

En 1968, Rechtschanffen y Kales (R&K) [6] desarrollaron un método para identificar las distintas etapas del sueño, método que se convirtió en estándar. Esta identificación se basa en el análisis de las tres señales descritas previamente: EEG, EOG y EMG. Las etapas de sueño a las que hacen referencia son la etapa REM (*Rapid Eye Movement*), y las etapas 1, 2, 3, 4 que se corresponden con el sueño NREM (No REM).

En 2007, la Academia Americana de Medicina del Sueño (AASM) [7] publicó un estándar, que adopta los criterios originales de clasificación de R&K, con algunas excepciones notables. Entre ellas reduce las cuatro etapas del sueño NREM a tres, combinando las etapas del sueño 3 y 4 en una única etapa, y modificando su nomenclatura, que a partir de este momento será W, N1, N2, N3 y REM.

Siguiendo los estándares de la AASM y R&K, la clasificación de las diferentes etapas del sueño se realiza para cada intervalo de 30 segundos, denominado epoch. A continuación se describen las etapas mencionadas en el estándar de la AASM.

Etapa W

Representa el estado de vigilia. En esta etapa la señal de EEG revela la presencia de ritmos alfa con frecuencias de 8 a 13Hz, que se atenúa con los ojos abiertos. Esta señal puede mostrar parpadeos rápidos del ojo, a una frecuencia de entre 0.5 y 2 Hz y en caso de que los ojos estén abiertos, muestra movimientos oculares rápidos. La señal de EMG presenta amplitud variable que suele ser más alta que durante el resto de las etapas de sueño.

Etapa N1

Representa una transición de la vigilia al sueño. Es la etapa de sueño más ligera, en la que se perciben la mayoría de los estímulos que hay alrededor. En la señal de EEG predominan las ondas theta en más del 50% del epoch y las ondas alfa disminuyen o desaparecen. La señal de EOG suele mostrar movimientos oculares lentos y la amplitud muscular registrada en la señal de EMG es variable, pero a menudo inferior a la de la etapa W.

Etapa N2

Se caracteriza por la presencia de husos de sueño y complejos K en la señal de EEG, donde predominan las ondas theta y las ondas delta aumentan en comparación con la etapa N1. La señal de EOG no suele mostrar movimientos oculares pero en algunos individuos puede persistir el movimiento ocular lento. La señal de EMG muestra amplitudes variables pero menores que en las etapas anteriores.

Etapa N3

Es la etapa de sueño profundo, esencial para un sueño reparador. La señal de EEG muestra ondas lentas, con predominio de la onda delta, y elevada amplitud en al menos un 20% del epoch. Los husos de sueño y complejos K pueden persistir en esta etapa. Típicamente, la señal de EOG no muestra movimientos oculares. A menudo, la actividad muscular presente en la señal de EMG es menor que en la etapa N2 y a veces tan baja como en la etapa REM.

Etapa REM

Se describe comúnmente como etapa de sueño paradójico. La señal de EEG muestra actividad con ritmos múltiples, de bajo voltaje y alta frecuencia, similar a la etapa N1. Puede mostrar ondas en dientes de sierra de 2 a 6 Hz en regiones frontales. La señal de EOG muestra la presencia de movimientos oculares rápidos. La señal de EMG cae al nivel más bajo del registro y muestra pérdida del tono muscular.

2.1.2 Husos de sueño

Los husos de sueño consisten en un grupo de ondas con frecuencias entre 11 y 16 Hz (más comúnmente entre 12 y 14 Hz) con una duración mayor o igual a 0.5 segundos (normalmente entre 0.5 y 2 segundos) y que alcanza la máxima amplitud hacia la mitad del evento.

Los husos de sueño fueron descritos por primera vez por Berger [8], pero el término fue introducido por Loomis et al. en 1937[9]. Son una de las características clave que contribuyen a la evaluación de las etapas del sueño. Concretamente, son un claro sello distintivo de la etapa

N2, tanto en adultos como en niños. Los husos de sueño son también uno de los pocos eventos de la señal de EEG que únicamente están relacionados con el sueño y suelen ir ligados a los complejos K. En la Figura 2.2 se puede ver un ejemplo de estas dos actividades.

Las ondas de los husos de sueño se caracterizan por una amplitud progresivamente creciente y luego gradualmente decreciente. Se utilizan como biomarcadores de trastornos neurodegenerativos específicos y se consideran un acontecimiento neuronal de interés para comprender el mecanismo subyacente a la consolidación de la memoria mediada por el sueño.

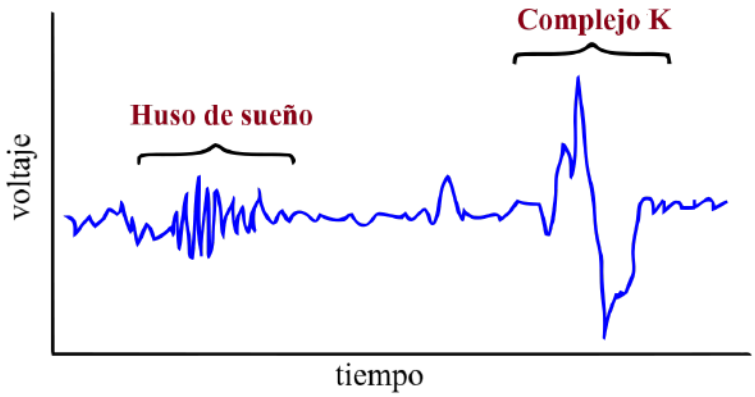

Figura 2.2: Representación de un huso de sueño

2.2 Estado del arte

La tarea de clasificación de las etapas de sueño se realiza manualmente por expertos en la medicina del sueño. Este proceso es bastante tedioso y complicado dado que los registros polisomnográficos suelen durar toda la noche, dando lugar a un volumen de datos elevado. En la identificación de la etapa N2, la dificultad estriba en la identificación correcta y precisa de los husos de sueño y los complejos K. Esta identificación es difícil, ya que se están buscando actividades del sueño que no suelen durar más que unos segundos en registros polisomnográficos de aproximadamente 8 horas. Otra dificultad es que la fiabilidad para la identificación visual sugiere una variabilidad entre expertos, posiblemente debida a la subjetividad o experiencia de los mismos [10]. Así pues, se han desarrollado detectores automatizados de husos del sueño, para reducir la carga de trabajo de los expertos y eliminar la subjetividad mencionada.

Los primeros detectores de husos de sueño dependían del hardware [11, 12], aunque en la actualidad las soluciones más habituales están basadas en software. Existen dos enfoques principales: los que utilizan el filtrado de paso banda y la detección de la amplitud, y los que aplican la extracción de características, seguidos ambos de la toma de decisiones para la detección. El primer enfoque, seguido en [13, 14], adolece de la variabilidad de los expertos y

esa es una de las razones por las que el segundo enfoque es una línea de investigación destacada. En cuanto a las propuestas basadas en la extracción de características seguido de una clasificación (presencia o ausencia de husos de sueño), la Transformada de Fourier de corto plazo (STFT, del inglés *Short Term Fourier Transform*) resulta una herramienta adecuada para identificar el contenido de frecuencia de los husos de sueño. En [15], se utilizan coeficientes de la STFT como entradas del clasificador. Se obtuvieron tasas de concordancia del 88.7% y del 95.4% con un perceptrón multicapa y una máquina de soporte vectorial respectivamente. Otro método utilizado para la extracción de características es el modelo autorregresivo adaptativo (AAR). En [16] los coeficientes de AAR se utilizan como entrada para diferentes clasificadores: un perceptrón discreto, un perceptrón multicapa y una máquina de soporte vectorial. Los resultados obtenidos se compararon en términos de sensibilidad, logrando valores de 99.2%, 89.1% y 94.6% respectivamente. En los últimos años se han aplicado herramientas avanzadas de análisis de tiempo y frecuencia, como las wavelets, para obtener vectores de características de husos de sueño mejoradas. En [17] se propuso un detector automático basado en el Operador de Energía Teager (TEO) y en la relación de energía de distintas familias de wavelets, alcanzando una precisión del 93.9%. En [18] se desarrolla una técnica de descomposición multi-resolución basada en wavelets y la transformada STFT, para detectar husos de sueño. Después de la detección, se aplica el operador TEO para determinar la duración del huso de sueño. Con este enfoque se logró una sensibilidad y especificidad del 96.17% y 95.54% respectivamente. El operador TEO también se emplea en [19], para aislar la zona con posibles husos de sueño candidatos y la frecuencia del borde espectral para confirmar su presencia. Con esta aproximación se consiguieron valores de sensibilidad y especificidad del 80% y 97.6% respectivamente. En [20] se propuso un método híbrido basado en las características del dominio del tiempo y la frecuencia. Se utilizó el análisis espectral de Welch para la extracción de características del dominio de la frecuencia y una red de neuronas para la clasificación. Las precisiones obtenidas para tres conjuntos de características diferentes fueron del 100%, 56.86% y 93.86%. En [21] se propone un algoritmo que modela la distribución del huso de sueño de amplitud-frecuencia con una distribución normal bivariante. La detección del huso no se basa directamente en los umbrales de amplitud y frecuencia, sino en un modelo de distribución del huso de sueño que se adapta automáticamente a cada sujeto individual. Se llegó a la conclusión de que el modelado normal mejoraba el rendimiento y la calidad de detección de husos de sueño.

En [22] se estudiaron las capacidades de varias técnicas de Aprendizaje Máquina para clasificar los husos de sueño, con el objetivo de obtener un método que lograra los mejores resultados de precisión en esta tarea de clasificación. La extracción de características se realiza mediante una descomposición de wavelets discretas de la señal de EEG. Para la clasificación se utilizaron dos modelos lineales - una Red Neuronal y una máquina de soporte vectorial

proximal-, y cinco modelos no lineales - una Red Neuronal multicapa, un Árbol de Decisión, un Random Forest, una Máquina de Soporte Vectorial (SVM) y un clasificador Naive Bayes-. El conjunto de datos utilizado es el mismo que se utilizará en este trabajo y aparece descrito en la Sección 3.2. En este caso, se han utilizado 289 ejemplos correspondientes a 8 registros. Los resultados informaron que el Random Forest era la mejor opción, con un valor de exactitud de $94.08 \pm 2.8\%$ y $94.08 \pm 2.4\%$ con las familias de wavelets Symlet y ortogonal respectivamente.

En [23] desarrollaron un detector de husos de sueño basado en la clasificación multivariante de epochs de EEG utilizando SVMs. Como entradas al clasificador se utilizaron tres señales: la señal de EEG sin procesar y dos señales basadas en esta última, obtenidas tras la aplicación de tres filtros lineales FIR. Sobre estas señales se identificaron 34 características que posteriormente fueron reducidas a 12, resultado de la aplicación del algoritmo CFS [24] de selección de características. El detector fue probado en dos bases de datos de sueño públicas – DREAMS [13] y MASS [25] – y se obtuvieron los siguientes resultados en términos de sensibilidad, precisión y especificidad respectivamente: 53%, 37% y 96% para la base de datos DREAMS, y 77%, 46% y 96%, para la base de datos MASS.

En [26] se presenta un método que aplica el análisis wavelet-Fourier sobre características estadísticas de la señal de EEG, dividida en segmentos de 0.5 s con solapamiento de 0.4 s. Se utilizaron dos conjuntos de datos: DREAMS y MASS. Se utilizaron cuatro clasificadores para identificar los husos de sueño con las características extraídas, después de aplicar la transformación de onda de Fourier: una SVM de mínimos cuadrados, el método k vecinos más cercanos, un algoritmo K-means y un árbol de decisión C4.5. Los resultados de evaluación muestran que la SVM de mínimos cuadrados es el mejor de todos los métodos en todas las medidas de rendimiento estudiadas. Se obtuvieron valores de exactitud del 97.9%, de sensibilidad del 98.5% y de especificidad del 97.8%.

En [27] se presenta un detector de husos de sueño basado en una representación del mismo en términos de duración, frecuencia, modulación y un conjunto de índices obtenidos a través de un modelo generativo. Esta representación se utiliza como entrada a un modelo no supervisado basado en la identificación de puntos temporales que activa elementos de la representación a lo largo del proceso y aplica técnicas de agrupamiento. El detector fue probado sobre la base de datos DREAMS, obteniendo un valor de sensibilidad global del 67.7%.

En [28] se desarrolló un modelo de clasificación automática de husos de sueño basado en redes neuronales convolucionales, con el objetivo de determinar la influencia de la profundidad de la red y de la proporción de ejemplos de cada clase. Para obtener los ejemplos de entrenamiento y validación, la señal de EEG se dividió en ventanas de duración 1.73 s sobre los intervalos con husos de sueño identificados por el experto y sobre el resto de la señal. Para corregir el desbalanceo entre clases, se realizó un submuestreo aleatorio con reemplazo. Los modelos que consiguieron el mejor rendimiento presentaban 2 y 3 capas y la siguiente com-

binación de parámetros: 0.005 de tasa de aprendizaje, 0.9 de tasa de dropout y 256 de tamaño de batch. El modelo fue entrenado sobre la base de datos MASS y probado sobre la base de datos DREAMS, obteniendo valores de 0.80 para el valor F en cualquiera de los dos modelos.

Capítulo 3

Análisis de tecnologías y herramientas

En este capítulo de la memoria, se detallan tanto las herramientas y las tecnologías utilizadas para el desarrollo como el conjunto de datos usado en este trabajo.

3.1 Herramientas de desarrollo

El lenguaje de programación utilizado para el desarrollo de este trabajo es Python [29]. Fue creado por Guido Van Rossum y lanzado en 1991. Es un lenguaje de alto nivel, interpretado y orientado a objetos. Se utiliza tanto para desarrollo web, desarrollo software, matemáticas y scripting de sistemas [30]. Python dispone de una gran cantidad de librerías para visualización, cálculo numérico, análisis de datos y aprendizaje automático.

Como entorno de desarrollo se ha utilizado Spyder [31].

Las principales librerías de python utilizadas para el desarrollo de este trabajo son todas de código abierto y se describen a continuación:

- **NumPy** [32] es una librería que proporciona estructuras de datos y funciones matemáticas para trabajar con ellos, posibilitando su análisis.
- **pyEDFlib** [33] es una librería que permite trabajar con ficheros EDF (European Data Format) [34], diseñados para intercambio y almacenamiento de series de tiempo médicas.
- **Matplotlib** [35] es una librería de visualización de datos en 2D que trabaja con variedad de formatos.
- **Scikit-Learn** [36] es una librería de aprendizaje automático y análisis de datos, basada en NumPy, SciPy [37] y Matplotlib.

- **Pandas** [38] es una librería que proporciona herramientas de análisis y manipulación de datos de alto rendimiento.

- **OpenCV** [39] es una librería de visión artificial desarrollada por Intel en C++, con alta eficiencia computacional. Su API es C++ pero incluye conectores para varios lenguajes entre los que se encuentra Python.

- **TensorFlow** [40] es una librería de aprendizaje automático desarrollada por Google y que se caracteriza por su flexibilidad y escalabilidad.

- **Keras** [41] es una librería diseñada para posibilitar la experimentación rápida con redes de Aprendizaje Profundo.

El equipo en el que se realizará el proyecto tiene las siguientes características:

- Sistema Operativo: Windows 10 Home Edition

- Procesador: Intel core i7-9700T (2GB)

- Memoria RAM: 16 GB

- Tarjeta gráfica: NVIDIA GeForce GTX 1050 (4GB)

3.2 Conjunto de datos

El conjunto de datos utilizado en este trabajo es la Base de Datos de husos de sueño DREAMS (DREAMS Sleep Spindle Database, [13]) que forma parte del proyecto del mismo nombre en el que se recogieron un conjunto de registros polisomnográficos en un laboratorio de sueño de un hospital en Bélgica. Esta base de datos consiste en 8 registros de 30 minutos cada uno (extraídos de polisomnografías completas), del canal central del EEG. Las frecuencias de muestreo fueron 200 Hz, 100 Hz y 50 Hz. Los registros fueron anotados de forma independiente por dos expertos en husos de sueño. Para reflejar la realidad en lo posible, la selección de los epochs con husos de sueño ha sido realizada de forma independiente a la calidad de los mismos o del registro propiamente dicho.

En la Tabla 3.1, se presenta un resumen de los registros mencionados en donde se puede apreciar que el experto 2 solamente identificó husos de sueño en los seis primeros registros. En este trabajo se utilizarán los datos correspondientes a la clasificación del experto 2, ya que el número de husos de sueño identificados es superior al del experto 1.

Registro	Frecuencia de muestreo (Hz)	#Eventos clasificados experto 1	#Eventos clasificados experto 2	Total de eventos clasificados
#1	100	52	115	690
#2	200	60	52	312
#3	50	5	44	264
#4	200	44	25	150
#5	200	56	86	516
#6	200	72	87	522
#7	200	18	/	/
#8	200	48	/	/

Tabla 3.1: Resumen del conjunto de datos

3.3 Tecnologías de desarrollo

La Inteligencia Artificial (IA) es una ciencia que generalmente se relaciona con el desarrollo e investigación de sistemas que operan o actúan inteligentemente, siendo considerada una disciplina de las ciencias de la computación.

Una de las ramas de la IA es el Aprendizaje Automático, que tiene como objetivo desarrollar técnicas que permitan a los ordenadores aprender, utilizando algoritmos capaces de generalizar comportamientos y reconocer patrones a partir de una información suministrada.

Una Red Neuronal Artificial (RNA) es un modelo matemático inspirado en el comportamiento biológico de las neuronas. Las RNAs alimentadas hacia delante fueron las primeras que se desarrollaron y son el modelo más sencillo. Los principales exponentes de está arquitectura son el Perceptrón y el Perceptrón multicapa, utilizados normalmente en problemas de clasificación simple.

El Aprendizaje Profundo es un enfoque del Aprendizaje Automático que se ha basado en gran medida en el conocimiento del cerebro humano, la estadística y las matemáticas aplicadas, tal como se ha desarrollado en las últimas décadas. Los modelos computacionales del Aprendizaje Profundo imitan las características de la arquitectura del sistema nervioso, permitiendo que dentro del sistema global haya redes de unidades de proceso que se especialicen en la detección de determinadas características ocultas en los datos [42].

En los últimos años, el Aprendizaje Profundo ha experimentado un enorme crecimiento de popularidad y utilidad, en gran parte como resultado de computadoras más potentes, conjuntos de datos más grandes y técnicas de entrenamiento más complejas. En la historia del Aprendizaje Profundo, las Redes Neuronales Convolucionales han jugado un papel muy importante. En 1979, K. Fukushima propuso un modelo denominado neocognitrón [43], que es una RNA jerárquica y de múltiples capas. Se utilizó para el reconocimiento de caracteres

manuscritos y otras tareas de reconocimiento de patrones, y sirvió de inspiración para las redes neuronales convolucionales.

3.3.1 Redes Neuronales Convolucionales

Una Red Neuronal Convolucional (CNN, del ingés, *Convolutional Neural Network*), es un algoritmo de Aprendizaje Profundo que puede tomar una imagen de entrada, asignar pesos y sesgos a varios aspectos u objetos de la imagen, y ser capaz de diferenciar los objetos de dicha imagen entre ellos [44].

Fueron introducidas por primera vez en la década de 1980 por Yann LeCun, basándose en el neocognitrón [43]. Una de las primeras CNN fue LeNet [45], diseñada para el reconocimiento de caracteres escritos a mano. Este tipo de red es una variación de un Perceptrón multicapa, realizada sobre matrices bidimensionales, resultando muy efectivas para tareas de visión artificial, como la clasificación y segmentación de imágenes, entre otras aplicaciones. Se caracterizan por realizar operaciones de convolución entre una capa de entrada y un filtro o kernel, obteniendo como resultado un mapa de características. Este proceso se repite en varias capas intermedias, con el objetivo de extraer características de la imagen con distintos niveles de abstracción. Las CNNs también tienen una aplicación importante en el procesamiento del lenguaje. En este caso la entrada a la red son oraciones o documentos representados como una matriz. En cualquiera de estos casos, la CNN actúa a la vez de clasificador y de extractor de características. La arquitectura de una CNN se inspira en la organización de la corteza visual. Un ejemplo de esta arquitectura se muestra en la Figura 3.1.

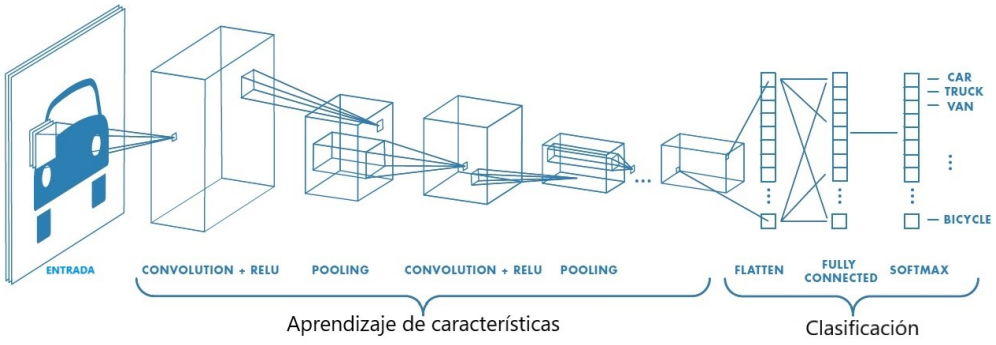

Figura 3.1: Arquitectura de una Red Neuronal Convolucional

A continuación se describen los componentes principales de una CNN típica.

Capas de convolución

Lo que distingue a las CNN de cualquier otra red neuronal es que utilizan la convolución en algunas de sus capas, en lugar de utilizar la multiplicación de matrices. El objetivo de la

operación de convolución es extraer las características de la imagen de entrada.

La convolución se define como:

$$(f * g)(n) = \sum_{m=-\infty}^{\infty} f(m)g(n-m)$$

siendo f una imagen de tamaño nxm y g un filtro o kernel de tamaño nxm

En la terminología de las CNN, la entrada de la capa convolucional es una imagen, a la que se le aplica un filtro y, finalmente, como resultado se obtiene una matriz de características que es conocida como mapa de características.

El tamaño del mapa de características viene definido por 3 parámetros que se deben establecer antes de realizar la convolución:

- Profundidad o depth, que corresponde al número de filtros que se usan para la operación de convolución.
- Paso o stride, que se corresponde con el número de píxeles por el que se desliza el kernel sobre la entrada.
- Relleno o padding, que permite controlar el tamaño del mapa de características, en los casos en los que la superposición del kernel no es posible debido al tamaño de la entrada.

Capas de pooling

La capa de *pooling* [46] se coloca generalmente después de la capa convolucional. Su función es simplificar progresivamente el tamaño espacial de la representación de los datos, para reducir también la cantidad de parámetros y la computación en la red, y como consecuencia controlar el sobreajuste.

Una de las ventajas de esta capa es que permite compactar los datos sin perder información relevante, y crea cierta invariabilidad al cambio traslacional en la imagen original. Esta capa no realiza ningún aprendizaje, solamente efectúa una operación de muestreo descendente.

El tipo de *pooling* más utilizado es el *max-pooling*, que devuelve el valor máximo de la porción de imagen cubierta por el kernel. En la figura 3.2 se muestra un ejemplo de aplicación de esta operación.

Los parámetros de configuración de esta capa son: la dimensión del elemento *pooling* (M), el paso o stride (S) y el relleno o padding (P). La salida de una capa de *pooling* es:

$$\frac{M-P}{S} + 1$$

siendo MxM el tamaño de entrada de los datos, P el tamaño del relleno y S el tamaño del paso.

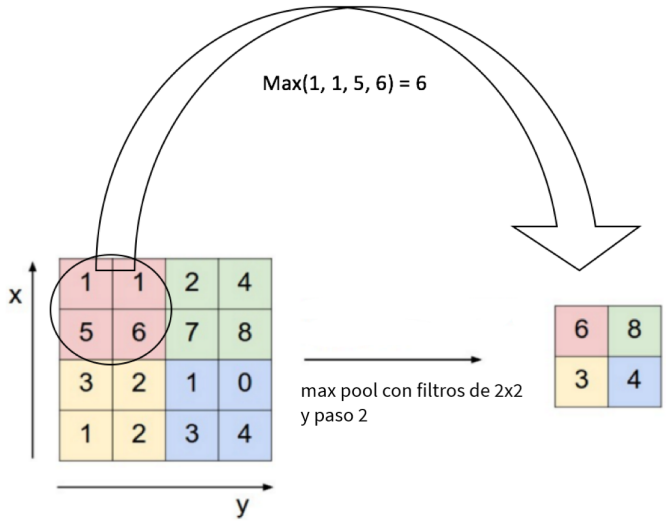

Figura 3.2: Ejemplo de aplicación de max-pooling

Capas Totalmente Conectadas

Las Capas Totalmente Conectadas (FC, del inglés *Fully Connected*) son capas que suelen constituir la parte final de las CNN. Todas las neuronas en una capa FC se conectan a las neuronas de la capa anterior, combinando todas las características aprendidas por las capas previas para identificar los patrones de mayor tamaño. Su objetivo es utilizar las características extraídas por las capas anteriores para clasificar la imagen de entrada en varias clases basadas en el conjunto de datos de entrenamiento.

Para problemas de clasificación, el tamaño de salida en la última capa FC es igual al número de clases a predecir. Y para problemas de regresión, el tamaño de salida debe ser igual al número de variables de respuesta.

Flattening

El proceso de Flattening consiste en convertir los datos de salida de las capas convolucionales a una matriz unidimensional para introducirla en la siguiente capa.

Dropout

La técnica de Dropout consiste en desactivar de forma aleatoria un porcentaje de las neuronas de una capa. Las neuronas que se desactivan se mezclan constantemente al azar durante el entrenamiento. El efecto que se persigue es reducir la tendencia de la red a depender excesivamente de algunas neuronas, obligando así a la red a aprender una representación más

equilibrada, y ayudar a combatir el sobreajuste.

Funciones de activación

Las funciones de activación [47] son funciones matemáticas que determinan la salida de una capa de una red neuronal, y se utilizan tanto en las capas convolucionales como en las capas FC. Se utilizan para asignar los valores de la red entre valores acotados, facilitando que el modelo se adapte a una gran variedad de datos.

Históricamente, tanto la función sigmoide como la función tangente hiperbólica [48] se han utilizado como funciones de activación en las redes neuronales, pero debido al problema de desvanecimiento del gradiente, su uso en el Aprendizaje Profundo se ha reducido.

La función de Unidad Rectificada Lineal (ReLU) [49] se utiliza como función de activación estándar para las CNN. En la figura 3.3 se observa que los gradientes en los valores positivos se vuelven constantes y no se desvanecen, solucionando así el problema de desvanecimiento del gradiente.

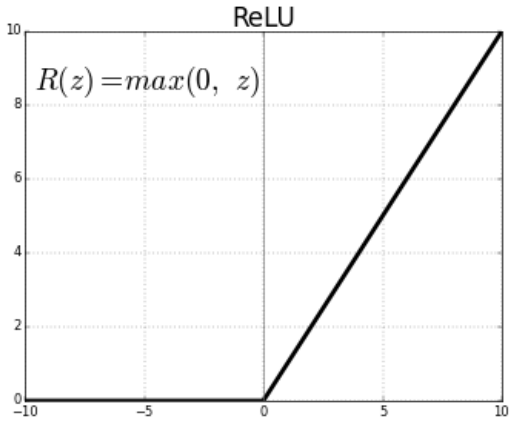

Figura 3.3: Función de activación ReLU

La función ReLU tiene un problema importante. Cuando las entradas se acercan a 0 o son negativas, el gradiente se vuelve cero, la red no puede realizar una propagación hacia atrás y no puede aprender. Existe una variante de la función ReLU, denominada Leaky ReLU, que previene este problema introduciendo una pequeña pendiente α (α = 0.001).

Optimizador Adam

Los optimizadores son algoritmos o métodos utilizados para cambiar los valores de los parámetros de la red neuronal con el fin de reducir las pérdidas.

El optimizador Adam [50] incorpora las ventajas de otros dos algoritmos de optimización: AdaGrad y RMSProp. Entre sus características están la existencia de diferentes tasas de

aprendizaje para diferentes parámetros de la red, la adaptación de la tasa de aprendizaje a partir del cálculo de dos estimaciones de momentos, un número reducido de parámetros a ajustar y unos valores por defecto que suelen dar buen resultado en la mayoría de problemas. La configuración de este optimizador se realiza a través de sus tasas de decadencia (β_1 y β_2).

Capítulo 4

Metodología de desarrollo y planificación

En este capítulo se describe la metodología de desarrollo utilizada para la elaboración del proyecto, así como la planificación del mismo, su evaluación de costes y las medidas de rendimiento utilizadas.

4.1 Metodología de desarrollo

En este trabajo se ha utilizado la metodología de desarrollo en espiral [51], que se materializa en una serie de versiones incrementales [52], siendo las primeras iteraciones un prototipo y las últimas unas versiones más completas del sistema diseñado.

Las fases por las que pasa cada ciclo en espiral son:

1. Planificación: se determinan los objetivos y el alcance del ciclo, teniendo en cuenta los ciclos anteriores.
2. Análisis de riesgos: se evalúa todo lo que pueda afectar al proyecto, tanto los riesgos como alternativas.
3. Implementación: consiste en desarrollar y validar el software según lo acordado en la planificación, teniendo en cuenta el análisis de riesgos previo.
4. Evaluación: consiste en analizar los resultados durante la implementación y en determinar si es necesaria o no una nueva iteración.

4.2 Planificación

La planificación de este trabajo se divide en fases que deben seguir el siguiente orden:

1. Adquisición de conocimientos teóricos sobre la medicina del sueño y los husos de sueño.
2. Adquisición de conocimientos teóricos sobre el Aprendizaje Profundo y las redes convolucionales.
3. Análisis y preprocesamiento del conjunto de datos.
4. Desarrollo y análisis de resultados de la aproximación basada en imágenes.
5. Desarrollo y análisis de resultados de la aproximación basada en características.
6. Comparación de los resultados y conclusiones.

Tanto para la fase 4 como para la fase 5, se llevaron a cabo las siguientes subfases:

1. Diseño e implementación del modelo de Aprendizaje Profundo.
2. Validación y pruebas del modelo.
3. Análisis de resultados.

En la figura 4.1 se puede observar el diagrama de Gantt de la planificación detallada por semanas de este trabajo.

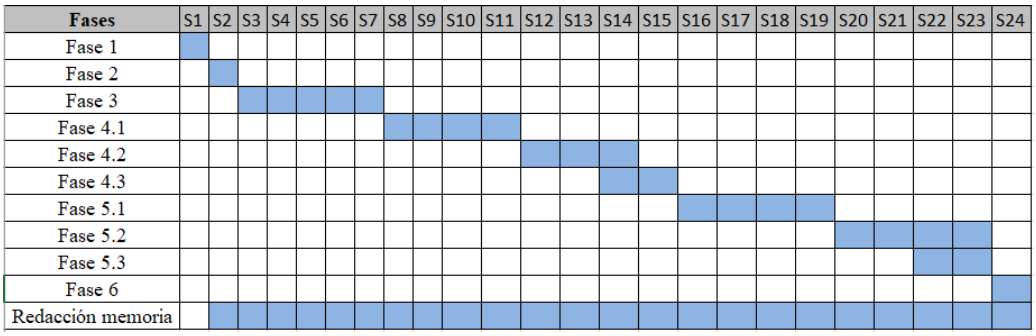

Figura 4.1: Diagrama de Gantt

4.3 Estimación de costes

En este apartado se muestra la estimación de los costes para el desarrollo de este proyecto. Se ha considerado la participación de un desarrollador junior, con un salario de 12€/hora. También es necesario disponer del coste estimado del material de trabajo, que se muestra en la Tabla 4.2. Tanto los conjuntos de datos como las herramientas son de uso libre por lo que no influyen en el coste del proyecto. En la Tabla 4.1 se detalla el coste estimado para cada fase para el ingeniero. El coste total estimado del proyecto es de 6.330€.

Fases	Horas	Coste (€)
1	15	180
2	15	180
3	75	900
4.1	60	720
4.2	45	540
4.3	30	360
5.1	60	720
5.2	45	540
5.3	30	360
6	15	180
Total:	375	4.680

Tabla 4.1: Coste estimado del trabajo del ingeniero

Material	Coste (€)
Internet	150
Equipo informático	1.500
Total:	1.650

Tabla 4.2: Coste estimado del material

4.4 Métricas de evaluación

Para evaluar el rendimiento de los modelos desarrollados se utilizan las siguientes métricas:

- **Exactitud:** La exactitud (accuracy) mide el porcentaje de casos que el modelo ha acertado.

$$exactitud = \frac{VP + VN}{VP + FP + VN + FN}$$

  donde VP se corresponde con los verdaderos positivos, VN con los verdaderos negativos, FP con los falsos positivos y FN con los falsos negativos.

- **Precisión:** La precisión es el número de elementos identificados correctamente como positivos del total de elementos identificados como positivos.

$$precisión = \frac{VP}{VP + FP}$$

- **Sensibilidad:** La sensibilidad (recall) es la proporción de casos positivos que fueron clasificados correctamente.

$$sensibilidad = \frac{VP}{VP + FN}$$

- **Medida-F1:** La medida-F1 (F1-score) es la media armónica entre la precisión y la sensibilidad.

$$medidaF1 = 2 * \frac{precision * sensibilidad}{precision + sensibilidad}$$

- **Índice kappa:** El índice kappa es la medida de acuerdo entre dos clasificadores que tienen en cuenta la posibilidad de acuerdo casual.

$$kappa = \frac{p_0 - p_e}{1 - p_e}$$

  donde p_0 es el acuerdo observado relativo entre los clasificadores, y p_e es la probabilidad hipotética de acuerdo por azar entre estos.

Capítulo 5

Desarrollo

En este capítulo de la memoria se describen dos aproximaciones para la clasificación de husos de sueño basadas en Aprendizaje Profundo: una primera aproximación basada en imágenes y una segunda aproximación basada en características, ambas extraídas a partir de la señal de EEG para clasificar la presencia o ausencia de husos de sueño.

5.1 Optimización de los hiperparámetros

La optimización o búsqueda de los hiperparámetros en el Aprendizaje Automático tiene como objetivo encontrar los mejores hiperparámetros que ofrezcan el mejor rendimiento del modelo. Los hiperparámetros, a diferencia de los parámetros, son establecidos por el ingeniero antes del entrenamiento.

Existen dos tipos principales de métodos de optimización: búsqueda manual y búsqueda automática. La búsqueda manual, a medida que aumenta el número de hiperparámetros y el rango de valores, se vuelve bastante difícil de manejar. El problema con la optimización de hiperparámetros es que evaluar la función objetivo para encontrar el mejor rendimiento es extremadamente costoso. Para evitar los inconvenientes de la búsqueda manual, se propusieron algoritmos de búsqueda automática. Entre ellos, destaca la búsqueda en cuadrícula que entrena un modelo de Aprendizaje Automático con cada combinación posible de valores de hiperparámetros, evaluando cada modelo y seleccionando la arquitectura que produce los mejores resultados. Otro método de búsqueda automática es la búsqueda aleatoria, que establece una cuadrícula de valores de hiperparámetros y selecciona combinaciones aleatorias para entrenar y puntuar el modelo, permitiendo controlar explícitamente el número de combinaciones de parámetros que se intentan. Ambas técnicas no poseen información alguna acerca de evaluaciones pasadas y, como resultado, a menudo emplean una cantidad significativa de tiempo evaluando hiperparámetros que no aportan una mejora al rendimiento.

Una alternativa a estas técnicas es la optimización bayesiana, que pertenece a una clase

de algoritmos de optimización basados en modelos secuenciales (SMBO), que permiten utilizar los resultados de la iteración anterior para mejorar el método de muestreo del siguiente experimento.

En este trabajo, para las dos aproximaciones se hace una búsqueda de hiperparámetros optimizada utilizando la técnica Tree-Structured Parzen Estimator (TPE) [53]. La técnica TPE utiliza el razonamiento bayesiano para construir el modelo sustitutivo y puede seleccionar los siguientes hiperparámetros utilizando la mejora esperada. Los valores que se obtienen en esta búsqueda son el número de bloques convolucionales, el tamaño del kernel, el número de filtros y el ratio de aprendizaje. Las distribuciones utilizadas para cada uno de los hiperparámetros se muestran en la tabla 5.1.

Hiperparámetro	Distribución
Bloques convolucionales	Uniforme entre 1 y 6
Tamaño del kernel	Uniforme entre 3 y 50
Filtros iniciales	Elección entre 8, 16, 32, 64 o 128
Ratio de aprendizaje	Log-uniforme entre -10 y -1

Tabla 5.1: Distribución para los distintos hiperparámetros de una red basada en Aprendizaje Profundo

5.2 Aproximación basada en imágenes

El objetivo de está aproximación es identificar husos de sueño representados mediante imágenes utilizando Aprendizaje Profundo. Estas imágenes se generarán a partir de intervalos seleccionados sobre la señal de EEG en los que los expertos han identificado un huso de sueño. El primer paso de esta aproximación consiste en generar dichas imágenes, que constituirán el conjunto de datos del modelo basado en Aprendizaje Profundo.

5.2.1 Generación de imágenes

Para la obtención de las imágenes es necesario localizar los husos de sueño sobre la señal de EEG y realizar la captura de intervalos con presencia y ausencia de husos de sueño. Esto nos permitirá generar los ejemplos positivos y negativos con los que trabajará el modelo. Debido a la duración variable de los husos de sueño identificados por el experto y teniendo en cuenta la duración máxima de un huso de sueño (apartado 2.1.2) se utilizarán ventanas de duración 4 segundos, centradas en cada huso de sueño. En la figura 5.1 se muestra un ejemplo de una ventana, que dará lugar a una imagen, en donde el huso de sueño se ha localizado en el centro de dicha ventana.

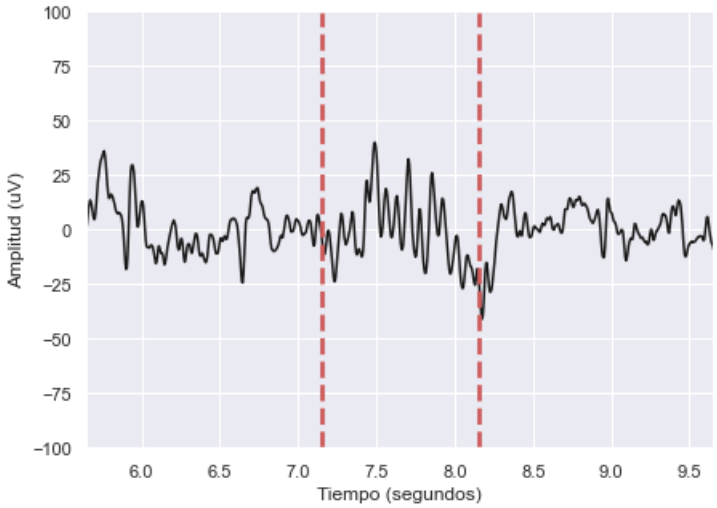

Figura 5.1: Imagen con huso de sueño generada a partir de la señal de EEG

La señal de EGG se representará en el dominio de tiempo, ya que es importante que la imagen resultante sea similar a la que los expertos suelen ver al analizar las polisomnografías. Para ello se utilizan los valores de magnitud de la señal en un rango de $\pm 100\mu$V, que son los valores de amplitud habituales del EEG [54]. El eje **x** representará la longitud en el tiempo de la ventana definida previamente y el eje **y** representará la magnitud de la señal de EEG. Las imágenes fueron generadas utilizando una resolución de 320x240 píxeles.

Para obtener las imágenes correspondientes a los ejemplos negativos (sin husos de sueño), se eligieron intervalos de la señal de EEG en los que el experto no había identificado un huso de sueño. El número de intervalos elegidos coincide con el número de intervalos seleccionados con huso de sueño.

Con la generación de imágenes se consiguieron 409 imágenes de ejemplos positivos y 409 imágenes de ejemplos negativos.

5.2.2 Preprocesado de datos

El Aprendizaje Profundo requiere una gran cantidad de datos etiquetados para obtener un rendimiento elevado. Debido a que el tamaño del conjunto de datos obtenido mediante la generación de imágenes no es suficientemente grande, se ha considerado utilizar técnicas de aumento de datos (*data augmentation*). Esta técnica consiste en modificar cada uno de los datos del conjunto inicial, de manera que cada una de estas modificaciones representen un nuevo dato para el conjunto. Estas transformaciones incluyen desde la rotación de la imagen un número de grados, aplicación de un efecto zoom aleatorio, o modificación de los valores RGB de la imagen. En este trabajo, se utilizó la transformación de volteo en el eje horizontal de la imagen, y el desplazamiento de la imagen, tanto hacia la izquierda como hacia la derecha,

para así incluir ejemplos de husos de sueño no centrados [55]. Finalmente, se obtuvieron 2454 imágenes de ejemplos positivos y 2454 imágenes de ejemplos negativos.

5.2.3 Modelos de Aprendizaje Profundo

En este apartado se describen los modelos de Aprendizaje Profundo desarrollados para llevar a cabo la tarea de clasificación de husos de sueño utilizando el conjunto de imágenes generado previamente. Se empezó realizando una prueba exhaustiva de modelos, en una primera iteración y luego, en una segunda iteración, se realizó un ajuste de la capa de max-pooling para la optimización de la red.

Iteración 1

En está primera iteración se propone una red convolucional que recibe como entrada las imágenes generadas y está formada por un bloque convolucional y dos capas totalmente conectadas. La figura 5.2 refleja el modelo propuesto.

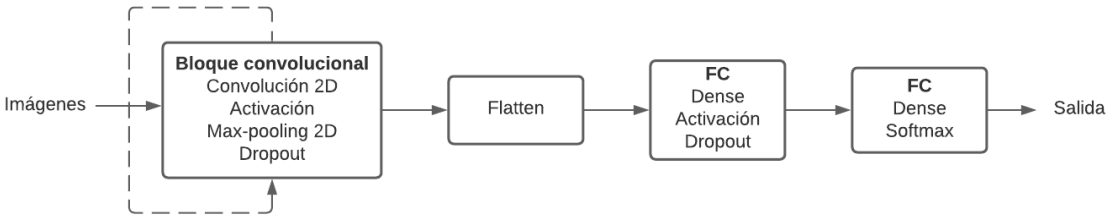

Figura 5.2: Red convolucional de la primera iteración

El bloque convolucional que se muestra es una sucesión de 4 capas incluyendo una capa de convolución 2D que guarda el tamaño de la entrada con padding, una capa de activación LeakyReLU con $\alpha = 0.1$, una capa max-pooling con un factor de 2 y una capa Dropout con un valor de 0.5. Este bloque se repite un número de veces en función de cada modelo y su salida es ”aplanada” mediante una capa Flatten.

La primera capa totalmente conectada está formada por una capa con 32 neuronas, una capa de activación LeakyReLU con $\alpha = 0.1$ y una capa Dropout con un valor de 0.5.

Finalmente, la última capa totalmente conectada, consta de una capa con 2 neuronas y utiliza una función de activación *softmax*.

El modelo emplea el algoritmo de optimización Adam [50], utilizando valores por defecto para los parámetros β ($\beta_1 = 0.9$ y $\beta_2 = 0.99$) [1].

[1] Valores por defecto de la librería Keras.

Con los valores obtenidos mediante la técnica TPE para los distintos hiperparámetros, se probaron un total de 40 modelos y de estos se escogieron los cinco mejores atendiendo al valor del índice kappa. En la tabla 5.2 se muestran los hiperparámetros de estos cinco modelos.

	Modelo 1	**Modelo 2**	**Modelo 3**	**Modelo 4**	**Modelo 5**
Bloques convolucionales	4	1	4	2	2
Tamaño de kernel	11	9	5	3	3
Filtros iniciales	128	8	128	64	8
Ratio de aprendizaje	1.57×10^{-4}	2.92×10^{-4}	3.54×10^{-3}	4.48×10^{-3}	12.79×10^{-3}

Tabla 5.2: Valores de los hiperparámetros de los cinco mejores modelos

Nº modelo	**Exactitud**	**Precisión**	**Sensibilidad**	**Medida F1**
Modelo 1	0.93	0.96	0.89	0.93
Modelo 2	0.82	0.88	0.74	0.80
Modelo 3	0.67	0.78	0.46	0.58
Modelo 4	0.93	0.98	0.87	0.92
Modelo 5	0.77	0.69	0.97	0.80

Tabla 5.3: Resultados de evaluación de los modelos seleccionados

Los datos disponibles se dividen de forma aleatoria en un conjunto de entrenamiento, otro de validación y otro de test con unos porcentajes de 80%, 10% y 10%, respectivamente. El entrenamiento se llevó a cabo durante 40 epochs, utilizando un tamaño de batch de 32. Este tamaño está condicionado por el hardware disponible para la ejecución de los modelos.

En la tabla 5.3 se muestran los resultados en términos de exactitud, precisión, sensibilidad y medida F1 de los 5 mejores modelos. El análisis de estos resultados muestra el buen comportamiento de la mayoría de los modelos, con valores de la medida F1 superiores o iguales al 80%. Dichos valores se corresponden con valores altos para el resto de las medidas, demostrando así la capacidad de aprendizaje de la red. Es necesario destacar el bajo rendimiento del modelo 3 cuyos resultados están muy por debajo de los conseguidos por el resto de los modelos. Por esta razón, este modelo se excluye en la segunda iteración.

Iteración 2

En está segunda iteración se intenta mejorar el rendimiento del sistema ajustando la capa de max-pooling.

Se utilizaron los cuatro mejores modelos de la iteración anterior, y se modificaron los valores de max-pooling para cada modelo. Se probaron diferentes valores, en un rango entre 2 y 9, para cada uno de los modelos y se escogieron los modelos con mayor índice kappa. En esta segunda iteración se mantuvieron los valores de los hiperparámetros de la iteración

previa, así como los valores del optimizador Adam, el número de epochs y el tamaño de batch. Se utilizó early stopping como medida para evitar el sobreajuste. En la tabla 5.4 se muestran los valores obtenidos tras las pruebas, demostrando que los valores de max-pooling elegidos en la iteración anterior para los modelos 2 y 5 eran los adecuados.

En la tabla 5.5 se pueden ver los resultados de evaluación de los cuatro modelos - modelos 1, 2, 4 y 5-, en la que comprobamos la mejora obtenida en cualquiera de los modelos estudiados. Esta mejora no es elevada ya que el punto de partida de está iteración ya era suficientemente bueno en cuanto a rendimiento.

Modelo 1	**Modelo 2**	**Modelo 4**	**Modelo 5**
8	2	8	2

Tabla 5.4: Valores de la dimensión max-pooling seleccionados para los modelos.

Nº modelo	**Exactitud**	**Precisión**	**Sensibilidad**	**Medida F1**
Modelo 1	0.95	0.98	0.92	0.95
Modelo 2	0.82	0.88	0.74	0.80
Modelo 4	0.94	1.00	0.88	0.94
Modelo 5	0.77	0.69	0.97	0.80

Tabla 5.5: Resultados de evaluación de los modelos seleccionados

5.3 Aproximación basada en características

El objetivo de esta aproximación es utilizar como entrada al modelo un conjunto de características, que permitan describir a los husos de sueño tanto en el dominio del tiempo como en el dominio de la frecuencia. En el dominio de la frecuencia se utilizó la transformada rápida de Fourier [56] y en el dominio del tiempo se utilizó un conjunto de características extraídas de la señal de EEG. El conjunto de datos utilizado está formado por los husos de sueño identificados por el experto 2, que constituyen un total de 409 ejemplos positivos. Al igual que en la aproximación anterior, se seleccionó la misma cantidad de ejemplos negativos. La extracción de características se realizó sobre ventanas de duración 2 segundos, para tener en cuenta la duración máxima de los husos de sueño (sección 2.1.2), aplicando la técnica de ventana deslizante de Hamming [57]. Las características consideradas para cada ventana fueron:

- Duración (f_1).
- Amplitud pico a pico (f_2).

- Número de oscilaciones (f_3), considerando el número de picos positivos del huso de sueño.
- Media cuadrática (f_4).
- Potencia absoluta media (f_5), calculada a partir de la transformada de Hilbert [58].
- Longitud de onda de la señal (f_6), obtenida a partir de la longitud acumulativa de la forma de la señal en un segmento particular.
- Entropía (f_7).
- Parámetros de Hjorth, considerados como una buena caracterización del EEG [59]. Estos parámetros son:
  - Actividad (f_8), que representa la potencia de la señal:

$$Actividad = var(X(n)) \tag{5.1}$$

    donde X es la señal y *var* es la varianza de la señal.
  - Movilidad (f_9), que representa la frecuencia media de la señal.

$$Movilidad = \sqrt{\frac{Actividad(X'(n))}{Actividad(X(n))}} \tag{5.2}$$

    donde X' es la primera derivada de la señal.
  - Complejidad (f_{10}), que representa los cambios de la señal en el dominio de la frecuencia.

$$Complejidad = \frac{Movilidad(X'(n))}{Movilidad(X(n))} \tag{5.3}$$

5.3.1 Selección de características

La selección de características es una técnica de reducción de la dimensionalidad que consiste en detectar las características relevantes y descartar las que no lo son [60]. Esta técnica presenta ventajas como:

- Mejora del rendimiento de los algoritmos de Aprendizaje Automático.
- Comprensión de datos, adquiriendo conocimiento sobre el proceso y ayudando a visualizarlo.
- Reducción de datos, limitando los requisitos de almacenamiento y los costes.

- Simplicidad, posibilitando el uso de modelos más sencillos lo que se traduce, en una disminución en tiempo de procesamiento.

Los métodos de selección de características se pueden dividir en métodos envoltorio (wrapper), de filtro o embebidos. Los métodos envoltorio se basan en la utilización de un algoritmo de aprendizaje que mide la eficacia de las características según el nivel de predicción que se obtiene en dicho algoritmo. Los métodos de filtro realizan una selección de características de manera independiente al algoritmo de aprendizaje que será utilizado y suponen un paso previo a la utilización del método de aprendizaje. Los métodos embebidos realizan la selección de características en el proceso de entrenamiento del algoritmo de aprendizaje y esta selección es propia y específica de cada algoritmo. A pesar de que los métodos envoltorio y los embebidos permiten obtener mejor rendimiento, su consumo en recursos computacionales es elevado y existe el riesgo de un sobreajuste cuando el tamaño de muestras es pequeño. Por otra parte, los métodos de filtro se adecuan mejor a grandes conjuntos de datos para los que ofrecen tiempos de ejecución reducidos.

En este trabajo, teniendo en cuenta el tamaño del conjunto de datos de trabajo y las limitaciones del hardware disponible, se ha decidido utilizar tanto métodos envoltorio como métodos de filtro. Dentro de los primeros, se utiliza el algoritmo BorutaShap [61], que combina el algoritmo de selección de características Boruta [62] con los valores de Shapley [63]. Este algoritmo se encarga de encontrar un conjunto mínimo de características óptimas, en lugar de encontrar todas las características relevantes para la variable objetivo, consiguiendo una selección estable de características importantes y no importantes.

Entre los métodos basados en filtro, se utiliza la selección de características basada en correlación (CFS, del inglés, *Correlation Features Selection*). Este algoritmo clasifica los subconjuntos de características según una función de evaluación heurística basada en la correlación [64]. El sesgo de la función es hacia subconjuntos que contienen características altamente correlacionadas con la clase y no correlacionadas entre sí. Las características irrelevantes deben ser ignoradas porque tienen poca correlación con la clase. Las características redundantes deben descartarse porque están muy relacionadas con una o más de las características restantes. La aceptación de una característica depende de la medida en que predice clases en áreas del espacio de instancias, que no han sido predichas por otras características.

5.3.2 Modelos de Aprendizaje Profundo

En este apartado se describen los modelos de Aprendizaje Profundo desarrollados para llevar a cabo la tarea de clasificación de husos de sueño utilizando el conjunto de características descrito previamente. Se empezó realizando una prueba exhaustiva de modelos, en una primera iteración, para posteriormente, en una segunda iteración, realizar una selección de características que permitiese eliminar las características redundantes.

Iteración 1

En está primera iteración se propone una red convolucional que recibe como entrada el conjunto de características descrito, y que está formada por un bloque convolucional y dos capas totalmente conectadas. La figura 5.3 refleja el modelo propuesto.

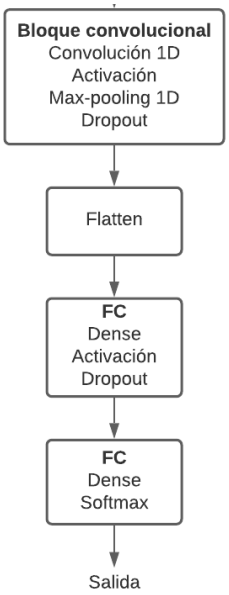

Figura 5.3: Red convolucional de la primera iteración

El bloque convolucional que se muestra está compuesto por 4 capas entre las que se encuentran una capa de convolución 1D, una capa de activación ReLU, una capa max-pooling 1D con un factor de 2 y una capa de Dropout con un valor de 0.5. Su salida es ”aplanada” mediante una capa Flatten.

La primera capa totalmente conectada está formada por una capa con 32 neuronas, una capa de activación ReLU y una capa Dropout con un valor de 0.5.

Por último, la última capa totalmente conectada, consta de una capa con 2 neuronas y una función de activación *softmax*.

El modelo emplea el algoritmo de optimización Adam, utilizando los valores por defecto para los parámetros β.

Con los valores obtenidos mediante la técnica TPE para los distintos hiperparámetros, se probaron un total de 40 modelos y de estos se escogieron los cinco mejores atendiendo al valor del índice kappa. En la tabla 5.6 se muestran los hiperparámetros de estos cinco modelos.

El entrenamiento se llevó a cabo durante 40 epochs, utilizando un tamaño de batch de 32. Este tamaño está condicionado por el hardware disponible para la ejecución de los modelos.

En la tabla 5.7 se muestran los resultados en términos de exactitud, precisión, sensibilidad y medida F1 de los 5 mejores modelos. El análisis de estos resultados muestra un comporta-

miento aceptable para la mayoría de los modelos, con valores de la medida F1 por encima del 70%. El modelo 4 es el que presenta las peores medidas de rendimiento. Por esta razón, este modelo se excluye en la segunda iteración.

	Modelo 1	**Modelo 2**	**Modelo 3**	**Modelo 4**	**Modelo 5**
Bloques convolucionales	1	1	1	1	1
Tamaño del kernel	5	3	7	5	5
Filtros iniciales	32	64	16	8	16
Ratio de aprendizaje	2.29×10^{-3}	6.47×10^{-5}	5.08×10^{-3}	1.28×10^{-3}	1.24×10^{-3}

Tabla 5.6: Valores de los hiperparámetros de los cinco mejores modelos

Nº modelo	**Exactitud**	**Precisión**	**Sensibilidad**	**Medida F1**
Modelo 1	0.89	1.00	0.78	0.87
Modelo 2	0.80	1.00	0.60	0.75
Modelo 3	0.79	1.00	0.57	0.73
Modelo 4	0.72	1.00	0.42	0.60
Modelo 5	0.70	0.62	1.00	0.76

Tabla 5.7: Resultados de evaluación de los modelos seleccionados

Iteración 2

En está segunda iteración se propone la aplicación de métodos de selección de características para determinar si existen características redundantes y comprobar si al descartar dichas características, se consigue mejorar el rendimiento de los modelos de la iteración anterior. Tal y como se detalla en la sección 5.3.1, se utilizarán los métodos BorutaShap y CFS para realizar la tarea de selección. En la tabla 5.8 se muestra el conjunto de características resultantes tras aplicar el método BorutaShap, en el que se puede ver que se han descartado 2 características, de las 10 originales; y tras aplicar el método CFS, con el que se han descartado 3 características del total. La tabla muestra las características por orden de relevancia para cada uno de los dos métodos.

Con estos dos conjuntos se entrenaron los mejores modelos obtenidos en la iteración anterior. Se mantuvieron los valores de los hiperparámetros de dicha iteración, así como los valores del optimizador Adam, el número de epochs y el tamaño del batch.

Las tablas 5.9, 5.10 muestran los resultados de evaluación de los cuatro modelos tras aplicar los métodos de selección de características BorutaShap y CFS respectivamente. En ellas se puede comprobar que la mejora en el rendimiento es elevada, independientemente del método de selección empleado.

	BorutaShap		**CFS**
f_9	movilidad	f_1	duración
f_5	potencia absoluta media	f_7	entropía
f_7	entropía	f_3	número de oscilaciones
f_2	amplitdu pico a pico	f_{10}	complejidad
f_{10}	complejidad	f_9	movilidad
f_8	actividad	f_8	actividad
f_1	duración	f_2	amplitud pico a pico
f_4	media cuadrática		

Tabla 5.8: Conjunto de características resultado de la selección

Nº modelo	**Exactitud**	**Precisión**	**Sensibilidad**	**Medida F1**
Modelo 1	0.90	1.00	0.80	0.89
Modelo 2	0.95	1.00	0.90	0.95
Modelo 3	1.00	1.00	1.00	1.00
Modelo 5	0.85	0.97	0.72	0.83

Tabla 5.9: Resultados de evaluación de los modelos seleccionados con el método BorutaShap

Nº modelo	**Exactitud**	**Precisión**	**Sensibilidad**	**Medida F1**
Modelo 1	1.00	1.00	1.00	1.00
Modelo 2	1.00	1.00	1.00	1.00
Modelo 3	0.98	1.00	0.95	0.97
Modelo 5	0.90	1.00	0.80	0.89

Tabla 5.10: Resultados de evaluación de los modelos seleccionados con el método CFS

Tal y como se puede comprobar en las tablas 5.9 y 5.10, el método CFS mejora los resultados de todos los modelos excepto del modelo 3. Este modelo ya ofrecía los segundos peores resultados en la primera iteración, lo que podría indicar que el conjunto completo de características bajaba considerablemente el rendimiento del modelo.

Capítulo 6

Resultados

Este capítulo resume y evalúa los resultados obtenidos con las dos aproximaciones desarrolladas y los compara con otros estudios.

6.1 Evaluación de resultados de la aproximación basada en imágenes

La arquitectura de partida de esta aproximación está formada por un bloque convolucional y dos capas totalmente conectadas (FC). El bloque convolucional es una sucesión de cuatro capas (una capa convolucional 2D, una capa de activación, una capa max-pooling y una capa Dropout), conectado con una capa Flatten. La primera capa FC está formada por una capa Dense, una capa de activación y una capa Dropout. La última capa FC está formada por una capa con 2 neuronas y una función *softmax*.

Sobre esta arquitectura se presentan dos iteraciones. La primera iteración tenía como objetivo encontrar los mejores hiperparámetros, a través de una prueba exhaustiva de modelos para determinar aquellos con mejor rendimiento. El tiempo de ejecución de está iteración fue de aproximadamente 140 minutos. Los resultados de la iteración permitieron identificar que los modelos obtenían mejor rendimiento con un número de filtros elevado (128) independientemente del tamaño de dicho filtro. Este tamaño varía entre 3, 9 y 11.

A partir de los mejores modelos obtenidos en esta iteración, la segunda iteración trató de mejorar el rendimiento de estos modelos mediante el ajuste de la capa de max-pooling. Los mejores resultados se obtuvieron para un factor de 8. En esta iteración el tiempo de ejecución mejoró aproximadamente 50 minutos en comparación con la iteración anterior.

El resultado final de estas iteraciones es un modelo cuya arquitectura aparece representada en la figura 6.1. Los valores de rendimiento obtenidos son de 95%, 98%, 92% y 95% para la exactitud, precisión, sensibilidad y medida F1 respectivamente.

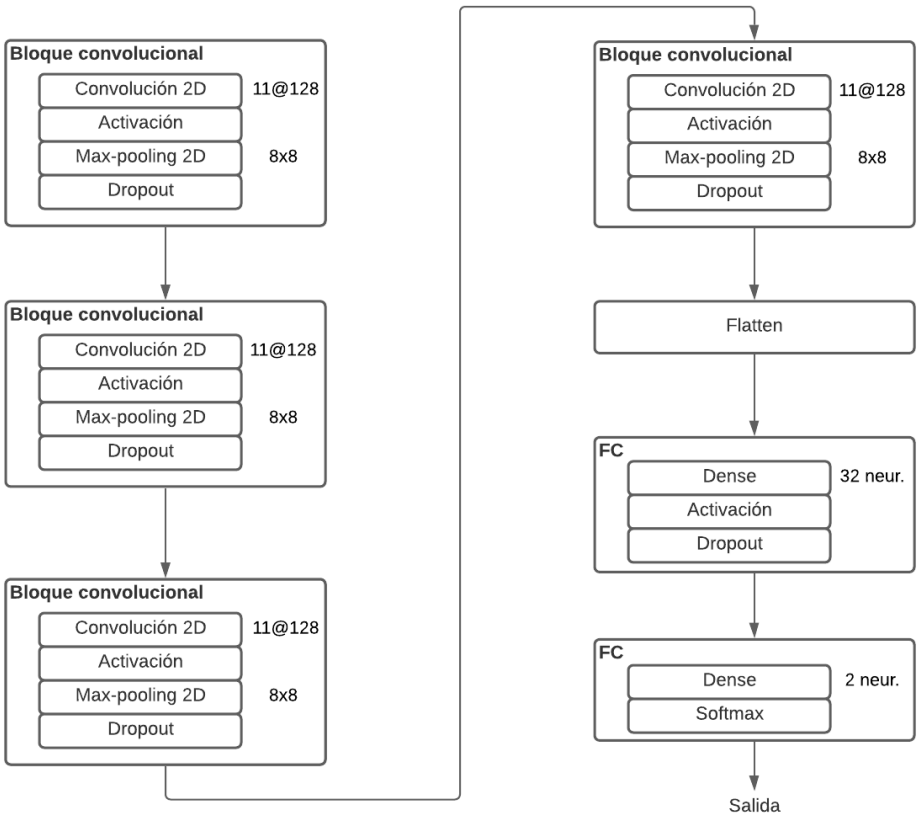

Figura 6.1: Arquitectura del mejor modelo de la aproximación basada en imágenes.

6.2 Evaluación de resultados de la aproximación basada en características

La arquitectura de partida de esta aproximación está formada por un bloque convolucional y dos capas totalmente conectadas (FC). El bloque convolucional es una sucesión de cuatro capas (una capa convolucional 1D, una capa de activación, una capa max-pooling y una capa Dropout), conectado con una capa Flatten. La primera capa FC está formada por una capa Dense, una capa de activación y una capa Dropout. La última capa FC está formada por una capa con 2 neuronas y una función *softmax*.

Sobre esta arquitectura se presentan dos iteraciones. La primera iteración tenía como objetivo encontrar los mejores hiperparámetros a través de una prueba exhaustiva de modelos, con el fin de obtener el mejor rendimiento. Los resultados de la iteración permitieron identificar que los modelos obtenían mejor rendimiento con un número de filtros no demasiado elevado (16 y 32) para un tamaño de filtro igual a 5. Estos valores, más pequeños que los

obtenidos en la aproximación anterior, pueden ser debidos al reducido número de ejemplos disponibles.

A partir de los mejores modelos de esta iteración, la segunda iteración trató de mejorar el rendimiento de estos modelos a través de la selección de características. Se probó con dos métodos de selección de características (BorutaShap y CFS). Ambos métodos mejoran el rendimiento de los modelos considerados, consiguiendo en algún caso medidas del 100%. El método CFS es el que permite que el modelo 2, cuya arquitectura se muestra en la figura 6.2, obtenga el mejor rendimiento.

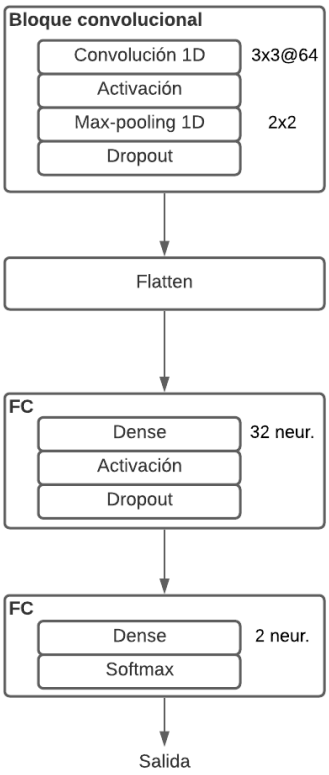

Figura 6.2: Arquitectura del mejor modelo de la aproximación basada en características.

6.3 Comparación con otros estudios

En este apartado se intenta establecer una comparación de los resultados obtenidos con las dos aproximaciones desarrolladas en este trabajo, con los resultados de algunos de los trabajos descritos en la sección 2.2.

Los trabajos que utilizamos en está comparación utilizan el mismo conjunto de datos que se utilizó en este trabajo con algunas diferencias. Hernandez-Pereira [22], Al-Salman [26]

y Loza y Principe [27] utilizan los datos del experto 1 y el número de husos de sueño es menor que el utilizado en cualquiera de las dos aproximaciones. Lachner-Piza [23] y Usai y Trappenberg [28] utilizan los datos de los dos expertos, con un número de husos de sueño mayor que en la aproximación basada en características.

La tabla 6.1 muestra los resultados obtenidos por los distintos trabajos del estado del arte y los mejores modelos de las dos aproximaciones desarrolladas. Para aquellos modelos en los que se dispone la medida F1, podemos concluir que ambas aproximaciones presentan un rendimiento superior a ellos. Ocurre lo mismo para el trabajo de Loza and Principe [27] para el que solamente se dispone de la medida de sensibilidad. En este caso, ambos desarrollos presentados son mucho más exhaustivos. La comparación con el trabajo que motivó este desarrollo [22], demuestra que ambas aproximaciones consiguen el objetivo perseguido en la identificación de husos de sueño.

Trabajo	**Número de husos de sueño**	**Exactitud**	**Precisión**	**Sensibilidad**	**Medida F1**
Al-Salman et al. [26]	355	0.97	-	0.98	0.71
Loza y Principe [27]	355	-	-	0.67	-
Lachner-Piza et al. [23]	1002	0.95	0.37	0.53	0.43
Usai y Trappenberg [28]	1822	0.78	0.75	0.86	0.80
Hernandez-Pereira et al. [22]	289	0.94	-	0.95	-
Aproximación basada en imágenes	2454	0.95	0.98	0.92	0.95
Aproximación basada en características	409	1.00	1.00	1.00	1.00

Tabla 6.1: Comparación entre distintos estudios y las aproximaciones desarrolladas.

Capítulo 7

Conclusiones

En este capítulo se comentan las conclusiones generales del trabajo, así como los conocimientos aprendidos y posibles trabajos futuros.

7.1 Conclusiones

El principal objetivo de este trabajo era el desarrollo de modelos de Aprendizaje Profundo mediante redes convolucionales capaces de identificar los husos de sueño. Para ello se desarrollaron dos aproximaciones: una basada en imágenes y otra basada en características de la señal de EEG.

Para la aproximación basada en imágenes, primero fue necesario la generación de estas a partir del conjunto de datos dado. Este conjunto de datos no era lo suficientemente grande, por lo que fue necesario utilizar técnicas de aumento de datos para obtener un conjunto de datos más grande. Se obtuvieron así 2454 imágenes de ejemplos positivos (con husos de sueño) y 2454 imágenes de ejemplos negativos. Para llevar a cabo la clasificación, se realizó una prueba exhaustiva de modelos mediante una técnica de optimización de hiperparámetros, consiguiendo unos valores de rendimiento bastante buenos. En una segunda iteración, con los mejores modelos de la iteración anterior se hizo un ajuste de la capa max-pooling, en la cual la mejora no fue elevada, aunque se ha conseguido un modelo con un rendimiento del 95% para la medida F1. Con esta aproximación hemos demostrado que la utilización de imágenes para la identificación de husos de sueño resulta una opción adecuada. Los tiempos de ejecución obtenidos durante el entrenamiento y validación de los modelos son asumibles para el tamaño del conjunto de datos con el que se trabaja.

Para la identificación de husos de sueño a través de un conjunto de características, fue necesario primero extraer este conjunto de características de la señal de EEG. De nuevo, para llevar a cabo la clasificación, se realizó una prueba exhaustiva de modelos mediante la técnica de optimización de hiperparámetros, obteniendo unos resultados que no mejoraban los

conseguidos con la aproximación anterior. Con vistas a mejorar estos resultados, se realizó un proceso de selección de características que demostró la existencia de características redundantes. Los mejores resultados se consiguen con el método CFS para el que se obtiene un rendimiento del 100%. Estos resultados aunque son excelentes, pueden ser debidos al tamaño del conjunto de datos. Es necesario recordar que el Aprendizaje Profundo consigue buenos resultados cuando el conjunto de trabajo es de gran tamaño y en este caso, se podría pensar que el modelo presenta un sobreajuste que lo lleva a obtener un rendimiento perfecto.

7.2 Conocimientos aprendidos

Los conocimientos adquiridos en este trabajo han sido:

- Conocimientos teóricos sobre los husos de sueño y la medicina del sueño.
- Estudio y desarrollo de modelos de Aprendizaje Profundo.
- Utilización de técnicas de optimización de hiperparámetros.
- Conocimiento de técnicas de selección de características y su aplicación al aprendizaje máquina.

7.3 Trabajo futuro

En cuanto al trabajo futuro, los resultados obtenidos con el desarrollo realizado, dejan abiertas algunas líneas de mejora:

- Buscar conjuntos de datos de mayor tamaño para poder confirmar si los resultados obtenidos se mantienen.
- Realizar transferencia de aprendizaje, utilizando los modelos seleccionados sobre otros conjuntos de datos.
- Evaluar los modelos desarrollados sobre los datos del primer experto, para el que se disponía un número de ejemplos más reducido.
- Estudiar otros métodos de selección de características para intentar reducir aún más el conjunto de características.

Apéndices

Lista de acrónimos

AAR Adaptive Autoregressive Modelling. 8

AASM Academia Americana de Medicina del Sueño. 5

AdaGrad Adaptative Gradient Algorithm. 17

API Application Programming Interface. 12

CFS Correlation Features Selection. 9, 30

CNN Convolutional Neural Network. 14

EDF European Data Format. 11

EEG Electroencefalograma. 1, 4

EMG Electromiograma. 4, 5

EOG Electrooculograma. 4, 5

FC Fully Connected. 16, 35, 36

FIR Finite Impulse Response. 9

IA Inteligencia Artificial. 13

NREM No Rapid Eye Movement. 5

R&K Rechtschaffen y Kales. 5

ReLU Rectifier Linear Unit. 17

REM Rapid Eye Movement. 5

RMSProp Root Mean Square Propagation. 17

RNA Red Neuronal Artificial. 13

SMBO Sequential model-based optimization. 24

STFT Short Term Fourier Transform. 8

SVM Support Vector Machine. 9

TEO Teager Energy Operator. 8

TPE Tree-Structured Parzen Estimator. 24

Glosario

complejos K Una actividad del EEG que consiste en una onda aguda negativa bien delineada seguida inmediatamente por un componente positivo que se destaca del fondo del EEG con una duración total ≥ 0.5 segundos, generalmente de amplitud máxima sobre las regiones frontales. 7

hiperparámetros Parámetros ajustables que permiten controlar el proceso de entrenamiento de un modelo. 23

Bibliografía

[1] S. Fogel and C. Smith, "The function of the sleep spindle: A physiological index of intelligence and a mechanism for sleep-dependent memory consolidation," *Neuroscience and biobehavioral reviews*, vol. 35, pp. 1154–65, 12 2010.

[2] D. Petit, J.-F. Gagnon, M. L. Fantini, L. Ferini-Strambi, and J. Montplaisir, "Sleep and quantitative EEG in neurodegenerative disorders," *Journal of psychosomatic research*, vol. 56, no. 5, p. 487—496, May 2004. [En línea]. Disponible en: https://doi.org/10.1016/j.jpsychores.2004.02.001

[3] F. Espa, B. Ondze, P. Deglise, M. Billiard, and A. Besset, "Sleep architecture, slow wave activity, and sleep spindles in adult patients with sleepwalking and sleep terrors," *Clinical neurophysiology : official journal of the International Federation of Clinical Neurophysiology*, vol. 111, pp. 929–39, 05 2000.

[4] Wikipedia, "Medicina del sueño." [En línea]. Disponible en: https://es.wikipedia.org/wiki/Medicina_del_sue%C3%B1o

[5] J. Gallego Pérez-Larraya, J. Toledo, E. Urrestarazu, and J. Iriarte, "Clasificación de los trastornos del sueño," *Anales del Sistema Sanitario de Navarra*, vol. 30, pp. 19 – 36, 00 2007. [En línea]. Disponible en: http://scielo.isciii.es/scielo.php?script=sci_arttext&pid=S1137-66272007000200003&nrm=iso

[6] A. Rechtschaffen and A. Kales, "A manual of standardized terminology, technique and scoring system for sleep stages of human sleep," *Brain Information Service, Los Angeles*, 1968.

[7] C. Iber, S. Ancoli-Israel, A. Chesson, and S. Quan, "The AASM Manual for the Scoring of Sleep and Associated Events: Rules, Terminology and Technical Specifications," *Westchester, IL: American Academy of Sleep Medicine*, 01 2007.

[8] Berger, "H. Über das Elektrenkephalogramm des Menschen," *Archiv f. Psychiatrie 87*, pp. 527–570, 1929. [En línea]. Disponible en: https://doi.org/10.1007/BF01797193

[9] A. L. Loomis, E. N. Harvey, and G. Hobart, "Potential Rhythms of the cerebral cortex during sleep," *Science*, vol. 81, no. 2111, pp. 597–598, 1935. [En línea]. Disponible en: https://science.sciencemag.org/content/81/2111/597

[10] L. B. Ray, S. M. Fogel, C. T. Smith, and K. R. Peters, "Validating an automated sleep spindle detection algorithm using an individualized approach," *Journal of sleep research*, vol. 19, no. 2, pp. 374–378, 2010.

[11] A. Kumar, W. Hofman, and K. Campbell, "An automatic spindle analysis and detection system based on the evaluation of human ratings of the spindle quality." *Waking & Sleeping*, 1979.

[12] D. Fish, P. Allen, and J. Blackie, "A new method for the quantitative analysis of sleep spindles during continuous overnight eeg recordings," *Electroencephalography and clinical neurophysiology*, vol. 70, no. 3, pp. 273–277, 1988.

[13] S. Devuyst, T. Dutoit, P. Stenuit, and M. Kerkhofs, "Automatic sleep spindles detection — Overview and development of a standard proposal assessment method," in *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2011, pp. 1713–1716.

[14] E. M. Ventouras, E. A. Monoyiou, P. Y. Ktonas, T. Paparrigopoulos, D. G. Dikeos, N. K. Uzunoglu, and C. R. Soldatos, "Sleep spindle detection using artificial neural networks trained with filtered time-domain EEG: a feasibility study," *Computer methods and programs in biomedicine*, vol. 78, no. 3, pp. 191–207, 2005.

[15] D. Görür, "Automated detection of sleep spindles," *Middle EastTechnical University*, 2003.

[16] N. Acır and C. Güzeliş, "Automatic recognition of sleep spindles in EEG by using artificial neural networks," *Expert Systems with Applications*, vol. 27, no. 3, pp. 451–458, 2004.

[17] B. Ahmed, A. Redissi, and R. Tafreshi, "An automatic sleep spindle detector based on wavelets and the teager energy operator," in *2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE, 2009, pp. 2596–2599.

[18] F. Duman, A. Erdamar, O. Erogul, Z. Telatar, and S. Yetkin, "Efficient sleep spindle detection algorithm with decision tree," *Expert Systems with Applications*, vol. 36, no. 6, pp. 9980–9985, 2009.

[19] S. A. Imtiaz, S. Saremi-Yarahmadi, and E. Rodriguez-Villegas, "Automatic detection of sleep spindles using Teager energy and spectral edge frequency," in *2013 IEEE Biomedical Circuits and Systems Conference (BioCAS)*. IEEE, 2013, pp. 262–265.

[20] S. Güneş, M. Dursun, K. Polat, and Ş. Yosunkaya, "Sleep spindles recognition system based on time and frequency domain features," *Expert Systems with Applications*, vol. 38, no. 3, pp. 2455–2461, 2011.

[21] A. Nonclercq, C. Urbain, D. Verheulpen, C. Decaestecker, P. Van Bogaert, and P. Peigneux, "Sleep spindle detection through amplitude-frequency normal modelling," *Journal of neuroscience methods*, vol. 214, 01 2013.

[22] E. Hernandez-Pereira, I. Fernandez-Varela, and V. Moret-Bonillo, "A Comparison of Performance of Sleep Spindle Classification Methods Using Wavelets," in *International Conference on Innovation in Medicine and Healthcare*. Springer, 2016, pp. 61–70.

[23] D. Lachner-Piza, N. Epitashvili, A. Schulze-Bonhage, T. Stieglitz, J. Jacobs, and M. Dümpelmann, "A single channel sleep-spindle detector based on multivariate classification of EEG epochs: MUSSDET," *Journal of Neuroscience Methods*, vol. 297, pp. 31 – 43, 2018. [En línea]. Disponible en: http://www.sciencedirect.com/science/article/pii/S0165027017304417

[24] R. Pérez-Rubido, "Una revisión a algoritmos de selección de atributos que tratan la redundancia en datos microarreglos," *Revista Cubana de Ciencias Informáticas*, vol. 7, no. 4, pp. 16–30, 2013.

[25] C. O'Reilly, N. Gosselin, J. Carrier, and T. Nielsen, "Montreal Archive of Sleep Studies: an open-access resource for instrument benchmarking and exploratory research," *Journal of Sleep Research*, vol. 23, no. 6, p. 628–635, 2014. [En línea]. Disponible en: https://dx.doi.org/10.1111/jsr.12169

[26] W. Al-Salman, Y. Li, and P. Wen, "Detecting sleep spindles in EEGs using wavelet fourier analysis and statistical features," *Biomedical Signal Processing and Control*, vol. 48, pp. 80 – 92, 2019. [En línea]. Disponible en: http://www.sciencedirect.com/science/article/pii/S1746809418302672

[27] C. Loza and J. Principe, "The Generalized Sleep Spindles Detector: A Generative Model Approach on Single-Channel EEGs," in *Advances in Computational Intelligence*, 05 2019, pp. 127–138.

[28] F. Usai and T. Trappenberg, "Using a Deep CNN for Automatic Classification of Sleep Spindles: A Preliminary Study," in *Advances in Artificial Intelligence*, M.-J. Meurs and F. Rudzicz, Eds., 04 2019, pp. 570–575.

[29] G. Rossum, "Python Reference Manual," NLD, 1995.

[30] "Python Introduction." [En línea]. Disponible en: https://www.w3schools.com/python/python_intro.asp

[31] P. Raybaut, "Spyder-Documentation," *Available online at: pythonhosted. org*, 2009.

[32] T. E. Oliphant, *A guide to NumPy*. Trelgol Publishing USA, 2006, vol. 1.

[33] "pyEDFlib." [En línea]. Disponible en: https://pypi.org/project/pyEDFlib/

[34] B. Kemp, A. Värri, A. C. Rosa, K. D. Nielsen, and J. Gade, "A simple format for exchange of digitized polygraphic recordings," *Electroencephalography and Clinical Neurophysiology*, vol. 82, no. 5, pp. 391 – 393, 1992. [En línea]. Disponible en: http://www.sciencedirect.com/science/article/pii/0013469492900097

[35] S. Tosi, *Matplotlib for Python developers*. Packt Publishing Ltd, 2009.

[36] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in Python," *the Journal of machine Learning research*, vol. 12, pp. 2825–2830, 2011.

[37] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright *et al.*, "Scipy 1.0: fundamental algorithms for scientific computing in python," *Nature methods*, vol. 17, no. 3, pp. 261–272, 2020.

[38] W. McKinney and P. Team, "Pandas: powerful Python data analysis toolkit," *Pandas—Powerful Python Data Analysis Toolkit*, p. 1625, 2015.

[39] A. Mordvintsev and K. Abid, "Opencv-python tutorials documentation," *Obtenido de https://media. readthedocs. org/pdf/opencv-python-tutroals/latest/opencv-python-tutroals. pdf*, 2014.

[40] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "Tensorflow: A system for large-scale machine learning," in *12th symposium on operating systems design and implementation 16)*, 2016, pp. 265–283.

[41] N. Ketkar, "Introduction to keras," in *Deep learning with Python*. Springer, 2017, pp. 97–111.

[42] R. A. Moreno, "Deep Learning: qué es y por qué va a ser una tecnología clave en el futuro de la inteligencia artificial," 2016.

[43] K. Fukushima, "Neural network model for a mechanism of pattern recognition unaffected by shift in position-neocognitron," *IEICE Technical Report, A*, vol. 62, no. 10, pp. 658–665, 1979.

[44] S. Saha, "A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way." [En línea]. Disponible en: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

[45] Y. LeCun *et al.*, "Lenet-5, convolutional neural networks," *URL: http://yann. lecun. com/exdb/lenet*, vol. 20, no. 5, p. 14, 2015.

[46] D. Yu, H. Wang, P. Chen, and Z. Wei, "Mixed pooling for convolutional neural networks," in *International conference on rough sets and knowledge technology*. Springer, 2014, pp. 364–375.

[47] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[48] J. Han and C. Moraga, "The influence of the sigmoid function parameters on the speed of backpropagation learning," in *International Workshop on Artificial Neural Networks*. Springer, 1995, pp. 195–201.

[49] H. Ide and T. Kurita, "Improvement of learning for CNN with ReLU activation by sparse regularization," in *2017 International Joint Conference on Neural Networks (IJCNN)*, 2017, pp. 2684–2691.

[50] S. Ruder, "An overview of gradient descent optimization algorithms," *ArXiv*, vol. abs/1609.04747, 2016.

[51] "Spiral Model - What is SDLC Spiral Model?" Aug. 2020. [En línea]. Disponible en: https://www.softwaretestinghelp.com/spiral-model-what-is-sdlc-spiral-model/

[52] "Modelo espiral." [En línea]. Disponible en: https://www.ecured.cu/Modelo_espiral

[53] J. S. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for hyper-parameter optimization," in *Advances in neural information processing systems*, 2011, pp. 2546–2554.

[54] T. Harmony, "Origen del electroencefalograma," *Salud Mental*, vol. 13, no. 3, pp. 27–34, 1990. [En línea]. Disponible en: http://www.revistasaludmental.mx/index.php/salud_mental/article/view/415

[55] L. Perez and J. Wang, "The effectiveness of data augmentation in image classification using deep learning," *arXiv preprint arXiv:1712.04621*, 2017.

[56] A. L. Schmidt, "Fft: Transformada rápida de fourier," *Estudiante de Ingenieria en Sistemas de Computación. Universidad Nacional de Sur. Bahia Blanca, Argentina. Marzo 2013*, 2013.

[57] N. Pielawski and C. Wählby, "Introducing Hann windows for reducing edge-effects in patch-based image segmentation," *PLOS ONE*, vol. 15, p. e0229839, 03 2020.

[58] W. J. Freeman, "Hilbert transform for brain waves," *Scholarpedia*, vol. 2, no. 1, p. 1338, 2007.

[59] G. Rodríguez-Bermúdez, P. García-Laencina, J. Roca-González, J. Roca-González, and J. Roca-Dorda, "Diseño de sistemas BCI adaptativos mediante una selección eficiente de características para discriminantes lineales," 2012.

[60] I. Guyon, S. Gunn, M. Nikravesh, and L. A. Zadeh, *Feature extraction: foundations and applications*. Springer, 2008, vol. 207.

[61] "Borutashap." [En línea]. Disponible en: https://pypi.org/project/BorutaShap/

[62] M. B. Kursa, W. R. Rudnicki *et al.*, "Feature selection with the boruta package," *J Stat Softw*, vol. 36, no. 11, pp. 1–13, 2010.

[63] M. V. García and J. L. Aznarte, "Shapley additive explanations for no2 forecasting," *Ecological Informatics*, vol. 56, p. 101039, 2020.

[64] M. A. Hall, "Correlation-based feature selection for machine learning," 1999.