

Trabajo de Fin de Máster

Máster en Ingeniería de Telecomunicación

Comparación de técnicas de aprendizaje profundo (deep learning) para detección de crisis epilépticas

Autor: Francisco Luis Sanmillán Vidoy

Tutor: Rubén Martín Clemente



Dep. Teoría de la Señal y Comunicaciones
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2020



Trabajo de Fin de Máster
Máster en Ingeniería de Telecomunicación

Comparación de técnicas de aprendizaje profundo (deep learning) para detección de crisis epilépticas

Autor:

Francisco Luis Sanmillán Vidoy

Tutor:

Rubén Martín Clemente

Profesor titular

Dep. de Teoría de la Señal y Comunicaciones
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla
Sevilla, 2020

Trabajo de Fin de Máster: Comparación de técnicas de aprendizaje profundo (deep learning) para
detección de crisis epilépticas

Autor: Francisco Luis Sanmillán Vidoy

Tutor: Rubén Martín Clemente

El tribunal nombrado para juzgar el Proyecto arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2020

El Secretario del Tribunal

A mi familia

A mis maestros

Agradecimientos

En primer lugar, quiero agradecer a Rubén, el tutor de este trabajo, haber confiado una vez más en mí. Al igual que en el TFG, agradezco su paciencia y que siempre haya sacado tiempo para resolver mis dudas y guiarme en el desarrollo de este trabajo.

Sin lugar a duda, no habría llegado hasta aquí de no ser por el apoyo constante de Daniela, quien ha convivido conmigo día a día y no ha dejado de animarme en ningún momento.

Desde la distancia, también han sido un gran apoyo, mis abuelos, siempre intentando que dé lo mejor de mí, mi tío Joaquín, guiándome en cómo afrontar cada asignatura, y mis hermanos Juan y Pedro, siempre dispuestos a ayudarme y aconsejarme.

Quiero agradecer a mis amigos el haber compartido conmigo tantos buenos momentos y, por último, a todas aquellas personas con las que he tenido el gusto de compartir la experiencia de estudiar el Máster de telecomunicaciones, quienes, en mayor o menor medida, también han contribuido a que esté donde estoy.

Francisco Luis Sanmillán Vidoy

Sevilla, 2020

Resumen

En este trabajo, se tratarán de aplicar varias técnicas de inteligencia artificial a señales electroencefalográficas recogidas directamente del cuero cabelludo de pacientes que padecen epilepsia intratable. Estas técnicas son específicas para cada paciente y estarán orientadas a investigar con qué técnicas y parámetros se pueden obtener las mayores precisiones en los menores tiempos posibles al detectar un ataque epiléptico. Además de comparar los resultados de las distintas técnicas entre sí, se tiene como objetivo analizar si estas se pueden aplicar en tiempo real, lo cual hace que se tengan que optimizar todos los recursos para conseguir obtener un resultado de alta precisión en el menor tiempo posible. Poder aplicar el sistema en tiempo real permitiría a los potenciales pacientes utilizarlo en su día a día, lo cual mejoraría su calidad de vida al disponer de un sistema de alertas que le avisa si va a padecer un ataque epiléptico hasta una hora antes o más de que ocurra, permitiendo así, que los pacientes tomen las medidas necesarias para mantenerse a salvo o, incluso, para evitar que el ataque suceda. Se aplicarán algunos cambios a cada una de las técnicas y se analizará como afectan al resultado en cada caso.

Abstract

In this work, we will try to apply several artificial intelligence techniques to electroencephalographic signals collected directly from the scalp of patients suffering from intractable epilepsy. These techniques are patient-specific and will be focused on investigating with which techniques and parameters the highest precision in the shortest possible time can be obtained when detecting an epileptic seizure. In addition to comparing the results of the different techniques with each other, the objective is to analyze whether they can be applied in real time, which means that all resources have to be optimized to achieve a high precision result in the shortest possible time. Being able to apply the system in real time would allow potential patients to use it in their day-to-day life, which would improve their quality of life by having an alert system that alerts them if they are going to suffer an epileptic attack up to an hour before or more than occur, thus allowing patients to take the necessary measures to stay safe or even to prevent the attack from happening. Some changes will be applied to each of the techniques and it will be analyzed how they affect the result in each case.

Índice

Agradecimientos	ix
Resumen	xi
Abstract	xii
Índice	xiii
Índice de Tablas	xv
Índice de Figuras	xvi
1 Introducción	11
2 Conocimientos previos	13
2.1. <i>Epilepsia</i>	13
2.1.1. Síntomas	13
2.1.2. Causas	13
2.1.3. Diagnóstico	14
2.1.4. Tratamiento	14
2.1.5. Tipos	15
2.2. <i>Electroencefalograma (EEG)</i>	18
2.2.1. Introducción	18
2.2.2. Anatomía del encéfalo	18
2.2.2.1. Electrogénesis cerebral	21
2.2.2.2. Electrogénesis cortical	22
2.2.3. Señales EEG	23
2.2.4. Captación del EEG	24
2.2.4.1. Tipos de electrodos	24
2.2.4.2. Sistema de posicionamiento de electrodos superficiales	26
2.2.4.3. Montaje de un EEG	27
2.3. <i>Base de datos</i>	30
3 Inteligencia artificial	31
4 Introducción a técnicas de IA	37
4.1. <i>Perceptrón multicapa (MLP)</i>	37
4.2. <i>Red neuronal convolucional (CNN)</i>	47
4.3. <i>Redes neuronales recurrentes (RNN)</i>	50
5 RESULTADOS	56
5.1. <i>Metodología</i>	56
5.2. <i>Red MLP</i>	57
5.3. <i>Red CNN+MLP</i>	64
5.4. <i>Red CNN+Bi-LSTM</i>	68

5.5. Selección de canal (<i>Channel select</i>).....	72
6 Conclusiones	76
Referencias	78
Glosario	80
Anexo. Funciones de Python utilizadas	81

ÍNDICE DE TABLAS

Tabla 1 Resultados de la primera prueba del modelo MLP (R=Mejor. A=Peor)	59
Tabla 2 Resultados de la segunda prueba del modelo MLP (R=Mejor. A=Peor)	61
Tabla 3 Resultados de la tercera prueba del modelo MLP (R=Mejor. A=Peor)	62
Tabla 4 Resultados de la cuarta prueba del modelo MLP (R=Mejor. A=Peor)	63
Tabla 5 Resultados de la primera prueba del modelo CNN+MLP (R=Mejor. A=Peor)	65
Tabla 6 Resultados de la segunda prueba del modelo CNN+MLP (R=Mejor. A=Peor)	66
Tabla 7 Resultados de la tercera prueba del modelo CNN+MLP (R=Mejor. A=Peor)	67
Tabla 8 Resultados de la primera prueba del modelo CNN+Bi-LSTM (R=Mejor. A=Peor)	69
Tabla 9 Resultados de la segunda prueba del modelo CNN+Bi-LSTM (R=Mejor. A=Peor)	70
Tabla 10 Resultados de la tercera prueba del modelo CNN+Bi-LSTM (R=Mejor. A=Peor)	71
Tabla 11 Resultados medios obtenidos en la primera prueba de las tres técnicas	76

ÍNDICE DE FIGURAS

Figura 1 Estados cerebrales de pacientes con epilepsia [4]	17
Figura 2 Partes del encéfalo	18
Figura 3 Lóbulos de la corteza cerebral	20
Figura 4 Homúnculos motor y sensorial	21
Figura 5 Capas de la corteza cerebral	22
Figura 6 Principales tipos de ondas cerebrales	23
Figura 7 Electrodo en casco de malla	25
Figura 8 Electrodo quirúrgico	26
Figura 9 Sistema Internacional 10/20 de colocación de electrodos	27
Figura 10 Esquema de montaje para un registro monopolar	28
Figura 11 Esquema de montaje para un registro monopolar promediado	28
Figura 12 Esquema de montaje para un registro bipolar	28
Figura 13 Montajes bipolares longitudinal, o de doble banana, (A) y transversal (B)	29
Figura 14 Fichero SUBJECT-INFO con la edad y el género de los sujetos	30
Figura 15 Tipos de inteligencia artificial [9]	31
Figura 16 Asistentes inteligentes [11]	32
Figura 17 Esquema del test de Turing [12]	33
Figura 18 Categorías más importantes de inteligencia artificial [14]	34
Figura 19 MLP [18]	37
Figura 20 Función escalón	39
Figura 21 Softmax	39
Figura 22 Sigmoide	40
Figura 23 ReLU	41
Figura 24 Tangente hiperbólica	41
Figura 25 Arquitectura propuesta para el clasificador basado en MLP [4]	46
Figura 26 CNN [25]	47
Figura 27 Arquitectura CNN propuesta [4]	49
Figura 28 Diagrama de bloques de CNN+MLP [4]	50
Figura 29 Funcionamiento de una neurona recurrente en una red RNN [28]	51
Figura 30 Celda básica LSTM [4]	53
Figura 31 Diagrama de bloques CNN+Bi-LSTM [4]	54

Figura 32 La red LSTM bidireccional desenrollada [4]

55

Figura 33 Algoritmo de selección de canal (Channel select) [4]

72

1 INTRODUCCIÓN

El verdadero progreso es el que pone la tecnología al alcance de todos.

Henry Ford

En este trabajo se llevan a cabo varias técnicas de detección de crisis epilépticas y se intenta analizar en qué condiciones y con qué parámetros se obtienen las detecciones más precisas y con la mayor anticipación. Para conseguir esto último se realizarán diversas pruebas utilizando señales electroencefalográficas (EEG) en crudo bajo distintos modelos de inteligencia artificial y variando sus parámetros, como pueden ser redes MLP, redes CNN o redes RNN y las múltiples variaciones de parámetros posibles.

Antes de comenzar con las pruebas se hace necesario introducir los distintos contenidos que se tratan en este trabajo, así como explicar la procedencia y características de las señales EEG utilizadas y presentar los distintos modelos utilizados. La estructura del trabajo será la siguiente:

1. Introducción: En la cual se explican los objetivos y el alcance del trabajo y se presenta la estructura del mismo.
2. Epilepsia: En este apartado se explicará con detalle en qué consiste la epilepsia y de qué diferentes formas pueden afectar a los pacientes. Entender bien en que consiste y cómo afecta la epilepsia a los distintos pacientes será fundamental a la hora de analizar resultados en este trabajo, puesto que es un trastorno muy singular.
3. Electroencefalogramas (EEG): Se explicará en qué consiste la encefalografía, como se obtienen las señales y las distintas formas estipuladas de posicionamiento de sensores para obtener dichas señales.
4. Procedencia de las señales EEG: Base de datos de la cual se han extraído las señales y explicación de dónde y cómo se obtuvieron las señales EEG que se utilizarán en este trabajo.
5. Pequeña introducción a la inteligencia artificial: Apartado importante en el cual se indican y describen las distintas categorías de IA y se introducen los campos más importantes en los que se divide actualmente.
6. Introducción a las distintas técnicas que se aplicarán: en este apartado se describirá como funciona un perceptrón multicapa y sus particularidades, así como, cómo funcionan las redes neuronales convolucionales y, por último, se describirán las redes neuronales recurrentes.
7. Especificaciones: Se detallarán los distintos modelos que se van a utilizar, así como, los parámetros concretos que se establecerán en cada uno de estos modelos inicialmente.

8. Tablas de resultados: Se incluirán varias tablas con los diferentes resultados obtenidos. Estas tablas constarán con las precisiones de detectar un ataque epiléptico obtenidas en las pruebas que se van a realizar.
9. Conclusiones: Para finalizar se analizarán los aspectos relevantes que se obtengan a partir de las tablas de resultados y se tratará de vislumbrar qué implicaciones pueden tener.

2 CONOCIMIENTOS PREVIOS

Este apartado es el mismo que el de mi trabajo fin de grado dado en [1], ya que los conocimientos previos coinciden al tratar ambos sobre la epilepsia.

2.1. Epilepsia

La epilepsia es un trastorno crónico del sistema nervioso central que origina una predisposición a sufrir crisis epilépticas repetidas. La epilepsia no es tan solo una enfermedad, sino una familia de trastornos neurológicos crónicos que tienen en común las convulsiones recurrentes o convulsiones epilépticas. Estas convulsiones epilépticas, que pueden aparecer en cualquier momento y duran unos segundos o minutos, son el resultado de una actividad transitoria, excesiva y anormal de las células nerviosas en la corteza cerebral [2] [3].

Cincuenta millones de personas en todo el mundo tienen epilepsia, cuatrocientas mil de ellas en España. Es el tercer trastorno neurológico más común tras el Alzheimer y el accidente cerebrovascular [2] [3].

Uno coma dos millones de individuos afectados sigue padeciendo convulsiones aun tomando medicación antiepiléptica. A las convulsiones padecidas por este subconjunto de pacientes se las conoce como convulsiones epilépticas intratables. En este trabajo, las señales que se usarán serán de pacientes con este tipo de convulsiones [2] [3].

2.1.1. Síntomas

Durante un ataque epiléptico el afectado puede tener sensaciones y emociones extrañas o comportamientos poco comunes, tales como espasmos musculares violentos, pérdida del conocimiento o alucinaciones sensoriales.

Estos ataques epilépticos, que en cada persona son diferentes y que dependen de la zona del cerebro afectada y las causas de la epilepsia, son intermitentes y constituyen el único síntoma de este trastorno.

Las convulsiones suponen un gran riesgo de lesión y limitan la independencia y la movilidad del afectado.

La mayor parte de los afectados son plenamente capaces en los periodos en los que no están padeciendo un ataque. Aun así, existe un mayor riesgo de muerte o discapacidad grave asociado con la epilepsia.

2.1.2. Causas

La epilepsia puede desarrollarse por una mutación heredada del comportamiento neuronal, es decir, por factores genéticos, o por algún traumatismo en la cabeza que dañe el tejido cerebral, o, incluso,

podría desarrollarse a partir de otra enfermedad que dañe, destruya o afecte al tejido cerebral, como en el caso de un tumor cerebral o de una infección, como la meningitis.

También se dan casos de epilepsia en niños que tienen un mal desarrollo cerebral debido a problemas antes del nacimiento.

En la mitad de los casos la causa de la epilepsia es desconocida.

2.1.3. Diagnóstico

Para diagnosticar la epilepsia se realizan pruebas como electroencefalogramas, resonancias magnéticas craneales, TACs y análisis de sangre. Todas las pruebas anteriores junto a la historia clínica del paciente permiten no solo diagnosticar la epilepsia, sino también saber de qué tipo es (se explicarán más adelante los distintos tipos de epilepsia), y, en ciertos casos, conocer la causa de la epilepsia.

En algunos afectados, no se detecta nada en las pruebas anteriores, teniendo resultados completamente normales. En estos casos el diagnóstico se basará en la historia clínica del afectado.

2.1.4. Tratamiento

No hay cura para la epilepsia, pero los medicamentos pueden controlar las convulsiones para la mayoría de las personas. Cuando los medicamentos no funcionan bien, la cirugía o los dispositivos implantados, como los estimuladores del nervio vago, pueden ayudar. También dietas especiales pueden tratar con eficacia ciertos tipos de epilepsia. Es importante comenzar el tratamiento de inmediato, tras ser diagnosticada la epilepsia.

En el caso de que los medicamentos no funcionen bien y el paciente tenga que someterse a cirugía, se pueden distinguir distintos tipos de intervenciones quirúrgicas, como los siguientes:

- Cirugía para extirpar el foco epiléptico, en la cual se extrae la zona del cerebro donde se produce la crisis epiléptica. Se conoce como lobectomía o lesionectomía.
- Otro tipo de cirugía se da cuando el foco epiléptico se encuentra en una zona del cerebro que no se puede extirpar. En este caso, lo que se hace es realizar una serie de cortes diseñados para impedir que las crisis se propaguen a otras partes del cerebro, dejando intactas las capacidades normales de la persona.
- La callosotomía, que es el corte de las conexiones neuronales entre hemisferios. Se hace en niños con crisis graves que se propagan de un hemisferio a otro. Puede eliminar crisis como las crisis de caída.
- Una cirugía muy radical, que se usa como última opción, es la hemisferectomía y la hemisferotomía, que consisten en retirar la mitad de la corteza cerebral. Este tipo de cirugía se usa para casos como el de la encefalitis de Rasmussen.

Para los afectados que son resistentes a la medicación y no pueden ser sometidos a cirugía, puede resultar interesante el uso de dispositivos estimuladores, como, por ejemplo, los siguientes:

- El estimulador del nervio vago, que se implanta bajo la piel del pecho y se adhiere al nervio vago en la parte inferior del pecho, envía descargas de energía eléctrica al cerebro a través del nervio vago, reduciendo así entre un 20 y un 40 por ciento las crisis.
- El estimulador para la neuroestimulación reactiva, que se implanta para analizar los patrones de la actividad cerebral y detectar una próxima crisis. Una vez detectada una crisis, el dispositivo manda una descarga eléctrica o administra un medicamento, para impedir la crisis. Analizar las detecciones de una próxima crisis de forma eficiente es el objetivo de este trabajo.

2.1.5. Tipos

Según el NINDS (National Institute of Neurological Disorders and Stroke), a grandes rasgos se distinguen dos tipos de crisis: las generalizadas, que afectan a toda la superficie del cerebro y provocan la pérdida de conocimiento y las crisis parciales o focales, donde la descarga comienza en una zona concreta que puede extenderse al resto de la corteza cerebral. Sin embargo, dentro de cada una de estas dos categorías hay muchos tipos de crisis. Los médicos han llegado a describir más de 30 tipos de crisis epilépticas [3].

Algunos tipos de epilepsia son los siguientes [2]:

- *Crisis generalizadas*: en las cuales, como se ha mencionado antes, la descarga epiléptica afecta al mismo tiempo a toda la superficie del cerebro. Los tipos de crisis epilépticas generalizadas más frecuentes son:
 1. *Crisis generalizadas tónicas*: causan rigidez muscular, sobre todo, en la espalda, en los brazos y en las piernas.
 2. *Crisis generalizadas clónicas*: causan movimientos bruscos repetidos de los músculos del cuerpo.
 3. *Crisis generalizadas tónico-clónica*: Se da cuando hay una pérdida brusca del conocimiento, cayendo el afectado al suelo si está de pie. Hay una fase tónica, que produce rigidez en el cuerpo en un primer momento, y, una fase clónica, que produce movimientos rítmicos por todo el cuerpo. Este tipo es una de las crisis epilépticas de mayor seriedad, ya que se pueden producir lesiones. No obstante, este tipo de epilepsia se controla bien con medicación.
 4. *Crisis generalizada de ausencia*: En las que, la persona permanece absorta e inmóvil durante diez o quince segundos. No revisten gravedad, pero puede ocasionar déficit de atención al afectado, puesto que, los ataques de este tipo son muy frecuentes.
 5. *Crisis generalizada mioclónica*: Se produce una sacudida brusca en todo el cuerpo o bien tan solo en una parte, durante un instante. Suele ocasionar caída al suelo del

afectado o la caída de los objetos que tenga en las manos. No son graves y suelen pasar desapercibidas.

6. *Crisis generalizadas atónicas*: El afectado cae de forma brusca al perder la conciencia. Dura un instante y la recuperación también es muy rápida. Suele ir acompañada de otros tipos de epilepsia.
 - *Crisis focales o parciales*: Según el sitio de la superficie del cerebro donde se produzcan las convulsiones tendremos un tipo de crisis epiléptica local u otro, habiendo muchas variantes. El 60 por ciento de las personas afectadas por epilepsia padecen crisis focales. Las crisis focales más importantes son las siguientes:
 1. *Sin alteración del conocimiento (crisis parciales simples)*: algunas de estas crisis producen sacudidas rítmicas de una parte del cuerpo durante segundos o minutos. Otras producen sensaciones anormales, tales como hormigueos que se extienden de forma brusca o alucinaciones visuales o, incluso, pueden producirse sensaciones de déjà vu, que solo nota quien padece la crisis
 2. *Con alteraciones del conocimiento (crisis epiléptica parcial compleja)*: el afectado unas veces pierde el conocimiento, quedándose inmóvil sin responder durante varios segundos o pocos minutos, pero sin perder el tono muscular. Otras veces sí que se mueve, pero tiene una actuación extraña y no responde o lo hace de forma incoherente. Recupera el conocimiento poco a poco
 3. *Evolucionando a crisis convulsiva bilateral (crisis bilateral con generalización secundaria)*: se da cuando una crisis focal simple o compleja se extiende desde el foco epiléptico a toda la superficie cerebral.
 4. *Espasmos epilépticos*: Se producen en niños de meses de edad y es raro que se produzcan en niños mayores de dos años. Consisten en una flexión o extensión brusca del cuerpo y de la cabeza durante varios segundos.

Así como hay distintos tipos de crisis epilépticas, también hay muchos tipos diferentes de epilepsia. Se han identificado cientos de síndromes epilépticos diferentes. Algunos de ellos son los siguientes [2]:

- 1 *La epilepsia de ausencia*: en la que se producen crisis repetidas que causan pérdidas momentáneas del conocimiento. Suelen ser causadas por factores genéticos. Algunas personas pueden realizar movimientos sin sentido, otras no muestran síntomas evidentes, excepto momentos breves en los que parecen estar perdidos en el espacio. Inmediatamente después de pasar la crisis el afectado puede seguir con lo que estaba haciendo. No obstante, estas crisis pueden ocurrir tan frecuentemente que la persona no se pueda concentrar en lo que esté haciendo.
- 2 *La epilepsia del lóbulo frontal*: se da cuando se producen crisis focales breves que pueden ocurrir en serie. Involucra las crisis que pueden causar movimientos anormales descontrolados. Normalmente ocurren cuando la persona está dormida, pero también pueden aparecer cuando está despierta.
- 3 *La epilepsia del lóbulo temporal*: es el más común entre los que tienen crisis focales. Estas

crisis se asocian con sensaciones de náuseas, de miedo, de déjà vu o con cambios inusuales del gusto y del olfato. Esta alteración puede aparecer como un episodio de mirar fijamente al vacío, un estado de ensueño o automatismos repetidos.

- 4 *La epilepsia neocortical*: se caracteriza por crisis originadas en la capa exterior del cerebro, en la corteza cerebral. Pueden ser tanto focales como generalizadas. Dependiendo del lugar donde se originen las crisis en el cerebro tendrá distintos síntomas.

Hay muchos otros tipos de epilepsia, como, por ejemplo, los siguientes: el síndrome de Lennox-Gastaut, la encefalitis de Rasmussen, la epilepsia mioclónica juvenil, el síndrome de Dravet o el hamartoma hipotalámico.

Por otra parte, los médicos clasifican la actividad cerebral de los pacientes con epilepsia mediante la utilización de electroencefalogramas, explicados con detalle en la siguiente sección, en cuatro estados:

- *Estado “ictal”*, que es el estado cerebral que se da durante las convulsiones.
- *Estado “preictal”*, que es el estado cerebral en el que se encuentra el paciente justo antes de la primera convulsión. En este trabajo se considerará “preictal” el estado en el que está el paciente la hora antes de que ocurra la primera convulsión.
- *Estado “postictal”*, que es el estado cerebral en el que se encuentra el paciente una vez que terminan las convulsiones.
- *Estado “interictal”*, que es el estado cerebral en el que se encuentra el paciente cuando no se encuentra en ninguno de los estados anteriores. En este trabajo se considerará que el paciente está en un estado “interictal” cuando esté cuatro horas antes y después de cualquier convulsión.

Esta división en cuatro estados se representa en la Figura 1, en la que se puede observar una señal de electroencefalograma con convulsiones y la división en los distintos estados. También se observa, como se verá más adelante, que la detección se realizará comparando entre los estados “preictal” e “interictal”, ya que interesa detectar un ataque epiléptico antes de que suceda, con el objetivo de evitarlo o de tomar precauciones.

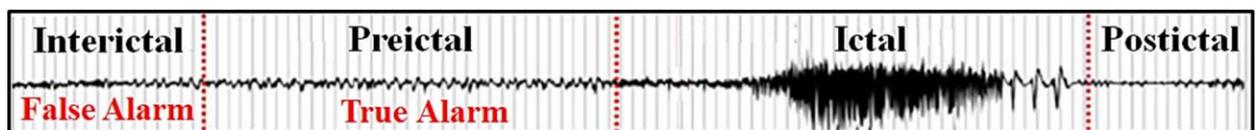


Figura 1 Estados cerebrales de pacientes con epilepsia [4]

2.2. Electroencefalograma (EEG)

2.2.1. Introducción

El electroencefalograma es una técnica de monitoreo electrofísico que permite registrar la actividad bioeléctrica cerebral usando el electroencefalógrafo, un equipo especial formado por un conjunto de electrodos capaces de medir la actividad eléctrica procedente de la corteza cerebral. Las señales captadas por los electrodos son mínimas, por lo que tienen que ser ampliadas para que puedan ser representadas y fácilmente observables [5].

El electroencefalograma es una técnica diagnóstica fácil de realizar, barata y no invasiva, al colocarse los electrodos sobre el cuero cabelludo. Se utiliza en pacientes en condiciones de reposo, en vigilia o sueño, o bajo algunas estimulaciones, como estimulaciones luminosas [5].

El electroencefalograma es especialmente útil en el diagnóstico de epilepsia, al hacer muy evidente la visualización de una actividad eléctrica cerebral anormal y brusca. Con esta técnica también es posible ver el tipo de crisis que tiene el afectado, lo que facilita el tratamiento de la epilepsia.

Para este *apartado 2.2* se han utilizado las referencias [6] y [7].

2.2.2. Anatomía del encéfalo

El encéfalo es la parte superior y la de mayor masa del sistema nervioso central, formado también por la médula espinal. Se encuentra en el interior del cráneo. Es el centro de control de casi todas las actividades vitales necesarias para la supervivencia. También controla las emociones y se encarga de recibir e interpretar numerosas señales que llegan del propio organismo o del exterior.

El encéfalo está dividido en tres partes: el tronco del encéfalo, el cerebelo y el cerebro. Estas partes se muestran en la Figura 2.

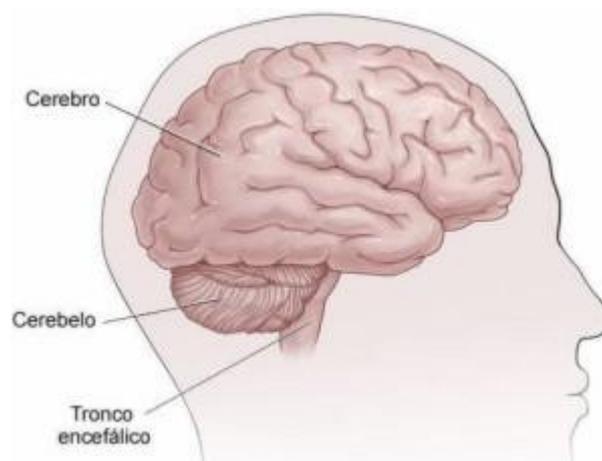


Figura 2 Partes del encéfalo

El tronco del encéfalo limita por debajo con la médula espinal, por detrás tiene el cerebelo y por encima tiene el diencefalo, conectándolos entre sí. Es la parte evolutiva más antigua del encéfalo. Tiene funciones sensoriales, recibiendo impulsos eléctricos de las extremidades, del tronco, de los órganos y del cráneo. También tiene funciones motoras, como controlar los actos motores reflejos y controlar los ritmos cardíaco y respiratorio. El tronco del encéfalo está formado por las siguientes partes [6]:

- Mesencéfalo: es la parte del tronco situada más cerca del centro del encéfalo, por lo que se la conoce como “cerebro medio”, y establece comunicación directa con muchas de las partes del sistema nervioso central. Une el tronco del encéfalo con el diencefalo y partes de la corteza cerebral y en su parte inferior tiene la protuberancia. Está formado principalmente por el tálamo y el hipotálamo. Consta de un conducto por donde circula el líquido cefalorraquídeo, encargado de proporcionar estabilidad mecánica al organismo. Contiene el nervio oculomotor y también se encarga de transmitir los impulsos eléctricos en los dos sentidos.
- Protuberancia o puente troncoencefálico: se encuentra posicionado entre el bulbo raquídeo y el mesencéfalo, conectándolos. En la protuberancia podemos encontrar una serie núcleos centrales, que se dividen en núcleos propios y núcleos asociados a nervios craneales, así como la línea trigémino-facial. Contiene en su núcleo una porción de la formación reticular, teniendo así una función importante en el sueño y en el estado de alerta. Actúa como puente de transmisión de impulsos cerebrales y trabaja en conjunto con el cerebelo y otras estructuras para permitir la coordinación automática e inconsciente de grupos de músculos, permitiendo así, por ejemplo, mantener el equilibrio en posición erguida.
- Bulbo raquídeo: interconecta la médula espinal con el puente troncoencefálico. Presenta la forma de un cono truncado de vértice inferior. Esta estructura neuronal controla funciones cardíacas, respiratorias, gastrointestinales y vasoconstrictoras, y constituye una parte fundamental para conectar el cerebro con la médula espinal.

El cerebelo se encuentra en la parte posterior del encéfalo. La función principal del cerebelo es integrar las vías sensitivas y las vías motoras. Actúa como coordinador de las actividades motoras que se inician en otras partes del sistema nervioso y hacen que los movimientos sean uniformes.

El cerebro es la parte más voluminosa del encéfalo. Está dividido en dos hemisferios separados y en su mayor parte está constituido por la corteza cerebral, la cual se encuentra plegada. Dentro del cráneo, el cerebro flota en el líquido cefalorraquídeo, que tiene funciones de protección tanto físicas como inmunológicas. Está formado por millones de neuronas que permiten regular todas las funciones del cuerpo y de la mente. El cerebro está formado a su vez por estructuras corticales, del córtex cerebral, y por estructuras subcorticales, bajo el córtex. Entre las estructuras subcorticales podemos encontrar, por ejemplo, el cuerpo caloso que une los dos hemisferios, los ganglios basales, la amígdala, el hipocampo o los cuerpos mamilares. El córtex cerebral aparece dividido en los siguientes lóbulos (como se muestra en la Figura 3) [6]:

- Lóbulo frontal: Este lóbulo controla varias funciones como la resolución de problemas, pensamiento creativo, juicio, intelecto, atención, comportamiento, reacciones físicas, pensamiento abstracto, movimientos coordinados, músculos coordinados y personalidad.
- Lóbulo parietal: En este lóbulo se pueden encontrar el córtex motor, que permite controlar el movimiento del cuerpo, y el córtex sensorial, el cual recibe información de la postura y como se mueven distintas partes del cuerpo y transmite información del sentido del tacto. Debido a

esto último, este lóbulo, se centra en el movimiento, cálculo, orientación y ciertos tipos de reconocimiento.

- **Lóbulo occipital:** es la parte del cerebro que controla la visión y donde se procesan las imágenes. Es uno de los lóbulos más pequeños y se encuentra en la parte trasera del encéfalo. Hay dos estructuras occipitales, una en cada hemisferio. Algunas de sus funciones son: interpretación de las imágenes, controlar la visión, reconocimiento espacial y discriminación del movimiento y los colores.
- **Lóbulo temporal:** controla la memoria visual, la auditiva y la comprensión del habla y la escucha. Contiene el área de Wernicke que se encuentra alrededor del córtex auditivo y formula y entiende el habla.

Existen otras zonas como el giro cingulado, el cual envuelve parcialmente al cuerpo calloso y se encuentra involucrado en la formación de emociones, en el procesamiento de datos básicos referidos a la conducta, el aprendizaje y la memoria, o, como, la corteza insular, que se encuentra ubicada profundamente en la superficie lateral del cerebro y entre algunas de sus funciones, se encarga de los estímulos gustativos, el peristaltismo, la presión arterial, de la información somatosensorial y de algunas funciones del lenguaje.

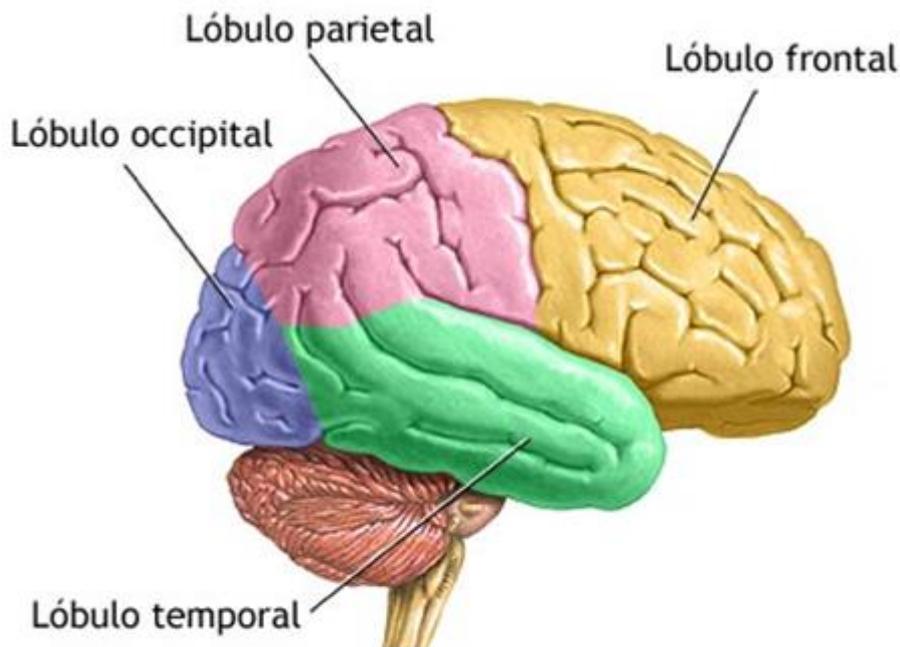


Figura 3 Lóbulos de la corteza cerebral

Existe una forma de representación de las zonas del cerebro en función de las partes del cuerpo que controla. Esta representación se conoce como el homúnculo cortical y representa las partes del cuerpo correspondiéndolas con la zona de la corteza motora primaria, homúnculo motor, y con la corteza somastésica primaria, homúnculo sensorial, que controlan dichas partes del cuerpo. Esta representación puede observarse en la Figura 4 [7].

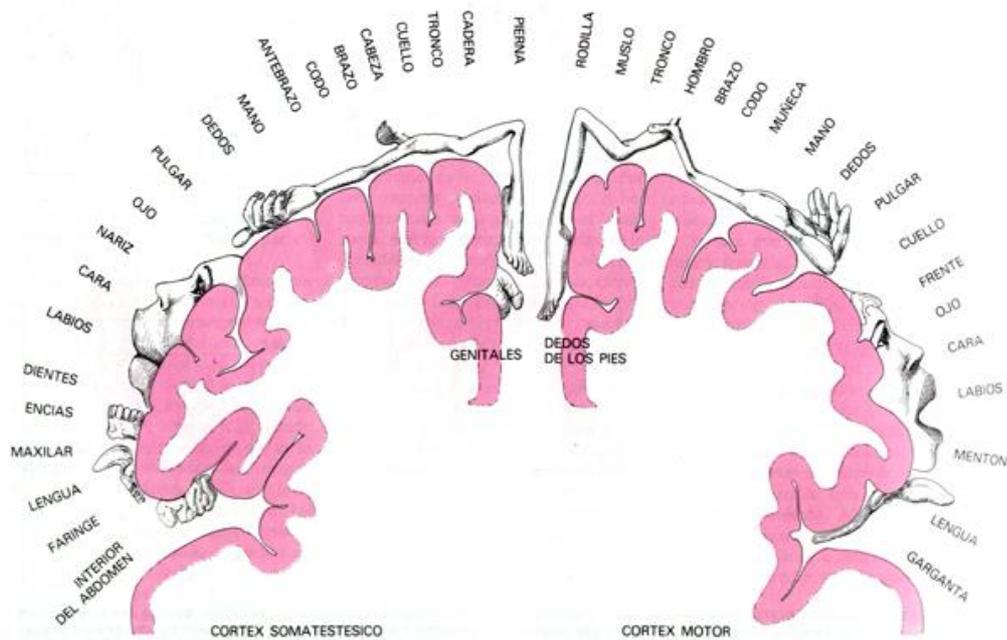


Figura 4 Homúnculos motor y sensorial

Para el electroencefalograma es necesario saber cómo y dónde se generan las señales bioeléctricas, es decir, la electrogénesis, para la cual, podemos distinguir entre electrogénesis cerebral y electrogénesis cortical.

2.2.2.1. Electrogénesis cerebral

Todo el sistema nervioso tiene capacidad electrogénica, esto es, la capacidad de generar potenciales eléctricos. Sin embargo, para el electroencefalograma tan solo nos va a interesar la corteza cerebral y las partes cercanas a la misma. La corteza cerebral está formada por distintas capas, que se pueden observar en la Figura 5. Estas son [7]:

- Capa molecular o plexiforme: es la parte más exterior de la corteza. Tiene una red densa de fibras nerviosas orientadas tangencialmente.
- Capa granular externa: se encuentra bajo la capa molecular y contiene una gran cantidad de células piramidales y estrelladas. Las células de esta capa son receptoras.
- Capa piramidal externa: formada por células piramidales y presenta una forma irregular, incrementándose su tamaño al profundizar en la capa. Las células de esta capa son efectoras.
- Capa granular interna: compuesta de células estrelladas de forma muy compacta y tiene una elevada concentración de fibras horizontales. Las células de esta capa son receptoras.
- Capa ganglionar o capa piramidal interna: contiene células piramidales grandes y medianas, y un elevado número de fibras horizontales. Las células de esta capa son efectoras.
- Capa multiforme: contiene células fusiformes.

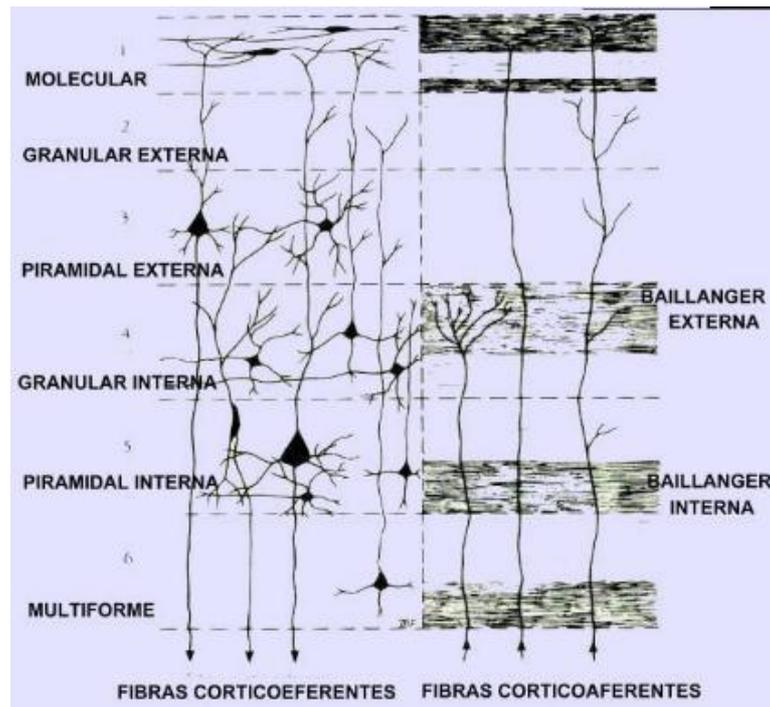


Figura 5 Capas de la corteza cerebral

2.2.2.2. Electrogénesis cortical

Una porción de tejido cortical aislado emite actividad eléctrica espontánea. Llamamos generador a cada una de estas porciones que es capaz de emitir actividad eléctrica espontánea. Se distinguen 3 tipos de generadores corticales [7]:

- Generador A: Situado a unas 500 micras de la superficie cortical está producido por la despolarización de las dendritas apicales de las células piramidales. Su actividad produce ondas negativas en la superficie de la corteza. No tiene relación con la descarga de potenciales de acción de las células.
- Generador B: Situado a 900 micras de profundidad está formado por las despolarizaciones de los somas de las células piramidales. Produce ondas positivas en la superficie cortical y su actividad coincide con la aparición de potenciales de acción en las células.
- Generador C: Está situado también a 900 micras, pero su actividad determina ondas negativas en la superficie cortical y es el resultado de la hiperpolarización de las células. Su actividad coincide con una interrupción de la descarga de potenciales de acción en las células piramidales.

De forma general, una tensión positiva en la superficie cortical traduce una despolarización en las capas más profundas de la corteza. En cambio, una tensión negativa puede ser resultado, bien de una despolarización superficial, o de una hiperpolarización profunda.

2.2.3. Señales EEG

Las señales EEG tienen una magnitud de 5 a 300 μV y un ancho de banda que va desde los 0.5 Hz hasta los 100 Hz, aunque normalmente se usa hasta los 70 Hz.

Las ondas emitidas por el cerebro, que pueden captarse mediante el EEG, se pueden clasificar en distintos tipos, dependiendo de la frecuencia. Los principales tipos de ondas cerebrales son los siguientes:

- Ondas Delta: son las ondas que provienen del cerebro y que ocupan el rango de frecuencias entre 0.5 y 4 Hz. Estas ondas delta suceden en situación de sueño profundo, cuando se está durmiendo sin sonar.
- Ondas Theta: son las que ocupan el rango de frecuencias entre 4 y 7 Hz. Relacionadas con las primeras etapas del sueño y se generan tras la interacción entre los lóbulos temporal y frontal.
- Ondas Alpha: son las que ocupan el rango de frecuencias entre 8 y 12 Hz. Se originan en el lóbulo occipital en periodos de relajación estando despierto.
- Ondas Beta: son las que ocupan el rango de frecuencias entre 13 y 30 Hz. Están asociadas a cuando se está despierto y consciente.
- Ondas Gamma: son las que ocupan el rango de frecuencias entre 30 y 100 Hz. Están asociadas a situaciones de gran estrés o de gran concentración.

Un ejemplo de estas ondas puede observarse en la Figura 6.

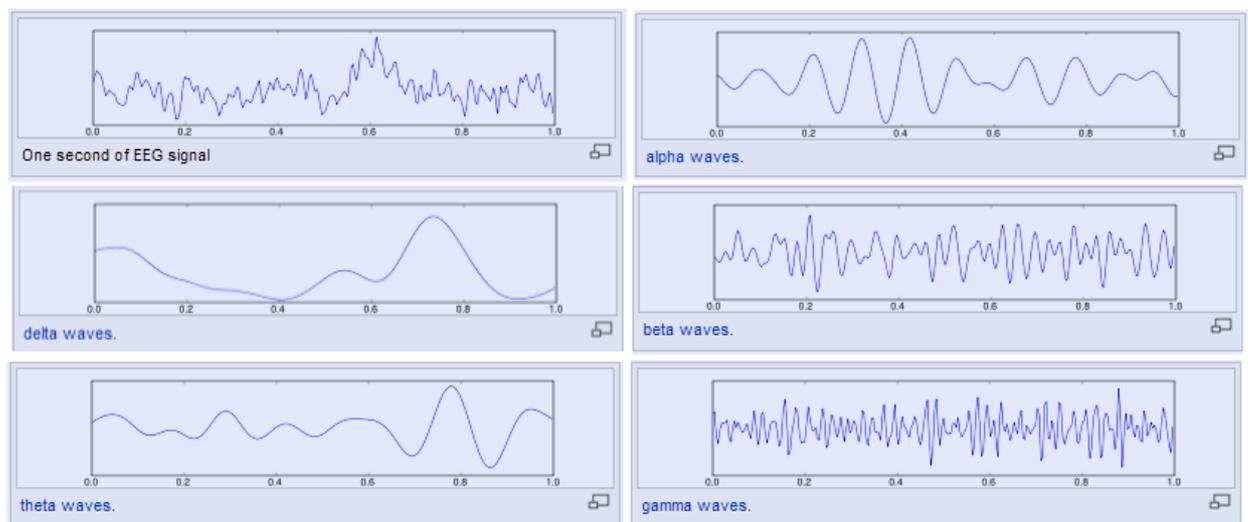


Figura 6 Principales tipos de ondas cerebrales

2.2.4. Captación del EEG

La actividad bioeléctrica puede captarse mediante diversos procedimientos:

- Sobre el cuero cabelludo.
- En la base del cráneo.
- En el cerebro expuesto.
- En localizaciones cerebrales profundas.

2.2.4.1. Tipos de electrodos

Existen distintos tipos de electrodos [7]:

- Electrodos superficiales: se aplican sobre el cuero cabelludo y se dividen en varios tipos a su vez:
 - 1) Adheridos: Son pequeños discos metálicos de 5 mm de diámetro. Se adhieren con pasta conductora y se fijan con colodión que es aislante. Aplicados correctamente dan resistencias de contacto muy bajas (1-2 kilo ohmios).
 - 2) De contacto: Consisten en pequeños tubos de plata clorurada roscados a soportes de plástico. En su extremo de contacto se colocan una almohadilla que se humedece con solución conductora. Se sujetan al cráneo con bandas elásticas y se conectan con pinzas de “cocodrilo”. Son de colocación muy fácil, pero incómodos para el paciente. Por esto no permiten registros de larga duración.
 - 3) En casco de malla: Los electrodos están incluidos en una especie de casco elástico. Existen cascos de diferentes tamaños, dependiendo de la talla del paciente. Se sujetan con cintas a una banda torácica. Como características más importantes presentan la comodidad de colocación, la comodidad para el paciente en registros de larga duración, su gran inmunidad a los artefactos y la precisión de su colocación, lo que los hace muy útiles en estudios comparativos, aunque para sacar provecho de esta característica es precisa una técnica muy depurada. Este tipo de electrodo se puede observar en la Figura 7.



Figura 7 Electrodo en casco de malla

- 4) De aguja. Su uso es muy limitado y solo se emplean en recién nacidos y en UCI. Pueden ser desechables o de uso múltiple, en los que, su esterilización y manipulación deben ser muy cuidadosos.
- Electrodo basales: los electrodo descritos hasta aquí registran solamente la convexidad superior de la corteza. Para el estudio de la cara basal del encéfalo se utilizan electrodo especiales como el faríngeo, el esfenoidal, y el timpánico. Estos electrodo son semi-invasivos, requieren de cirugía menor con anestesia local. Los esfenoidales recogen información de la actividad en la región media temporal y se pueden dejar fijos durante varias semanas sin tener efectos secundarios.
 - Electrodo quirúrgicos: Se utilizan durante el acto quirúrgico y son manipulados exclusivamente por el neurocirujano. Pueden ser duros, corticales o intracerebrales. Este tipo de electrodo se pueden observar en la Figura 8.

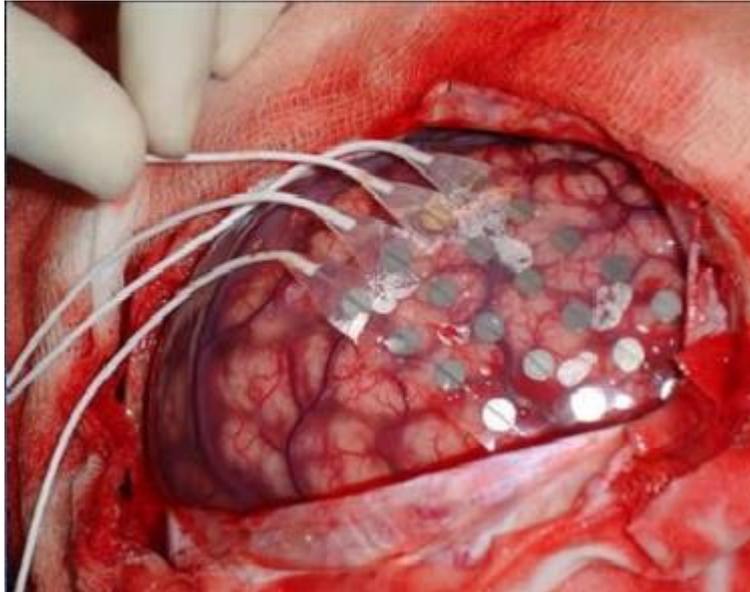


Figura 8 Electrodo quirúrgico

Si utilizamos electrodos de superficie o basales obtendremos el electroencefalograma (EEG). Sin embargo, si usamos electrodos quirúrgicos en la superficie de la corteza obtendremos el electrocorticograma (ECoG), y si usamos electrodos quirúrgicos de aplicación profunda obtendremos el estéreo electroencefalograma (E-EEG).

2.2.4.2. Sistema de posicionamiento de electrodos superficiales

La colocación de los electrodos se realiza normalmente siguiendo el sistema internacional 10-20, que emplea cuatro referencias anatómicas de la cabeza del paciente: nasión, inión y puntos pre-auriculares izquierdo y derecho. Es un método reconocido internacionalmente para describir y aplicar la ubicación de los electrodos en el cuero cabelludo en el contexto de una prueba EEG. Este método fue desarrollado para asegurar la reproducibilidad estandarizada para que los estudios de un sujeto pudieran compararse para distintas pruebas de EEG y los estudios de distintos sujetos pudieran compararse entre sí. Este sistema se basa en la relación entre la ubicación de un electrodo y el área subyacente de la corteza cerebral. El sistema 10/20 hace referencia a la manera de colocar los electrodos en la cabeza siguiendo una proporción (10% y 20%) respecto a la medida de la distancia nasion-inion.

Cada zona tiene una letra para identificar el lóbulo y un número para identificar la ubicación del hemisferio. Las letras F, T, C, P y O representan los lóbulos frontal, temporal, central, parietal y occipital, respectivamente. Los números pares (2,4,6,8) se refieren a las posiciones de los electrodos en el hemisferio derecho, mientras que los números impares (1,3,5,7) se refieren a las del hemisferio izquierdo. La "z" (zero) se refiere a un electrodo colocado en la línea media. Además de estas combinaciones, los códigos de letras A y Fp identifican los lóbulos de las orejas y del frontal, respectivamente.

Se utilizan dos puntos de referencia anatómicos para el posicionamiento de los electrodos EEG: en primer lugar, el nasion, que es el área entre los ojos, justo por encima del puente de la nariz; En segundo

lugar, el inion, que es el punto más bajo del cráneo de la parte posterior de la cabeza, normalmente marcado por un gran bulto. Este sistema puede observarse en la Figura 9.

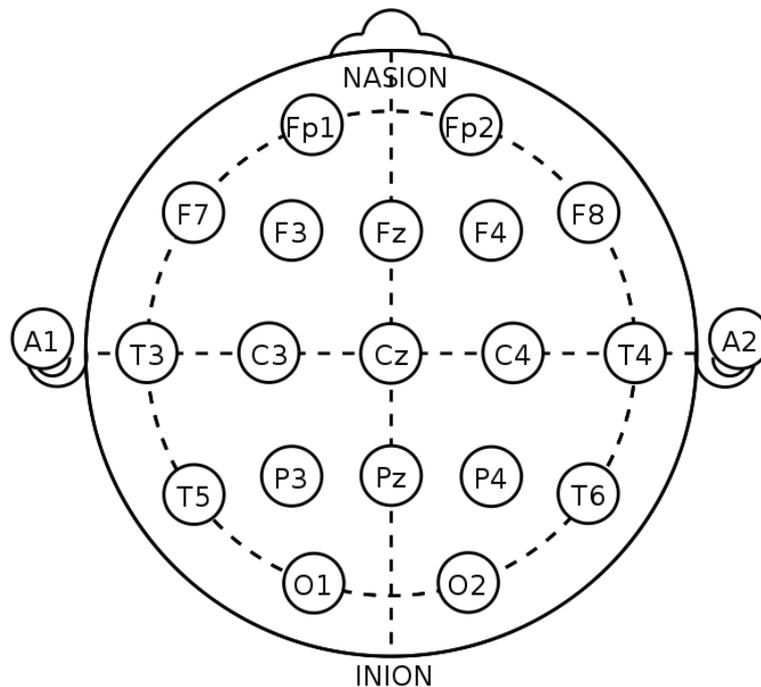


Figura 9 Sistema Internacional 10/20 de colocación de electrodos

2.2.4.3. Montaje de un EEG

Cada electrodo conectado en el cuero cabelludo constituye un punto de registro. Para realizar este registro es necesario disponer de dos terminales para establecer una diferencia de potencial. Según el propósito de la prueba a realizar y el número de canales disponibles habrá que seleccionar de que electrodos queremos realizar el registro. En primer lugar, hay que seleccionar entre registro monopolar o registro bipolar.

En los registros monopolares se toma el registro de cada electrodo de forma independiente a los demás, usando un electrodo activo, que es el que está conectado al registro, y un electrodo referencial, con el que establecer la diferencia de potencial. En este tipo de registros lo ideal es poner el electrodo a un potencial de cero voltios, pero en la realidad no hay seguridad de que ese sea el potencial, por ello, lo que se hace es conectar el electrodo de referencia en el lóbulo de la oreja o en el mentón, para obtener un valor aproximado (Esto se muestra en la Figura 10). Otra forma es usar como electrodo de referencia un electrodo conectado a la suma de los demás electrodos, suma que será cero (Esto se muestra en la Figura 11).

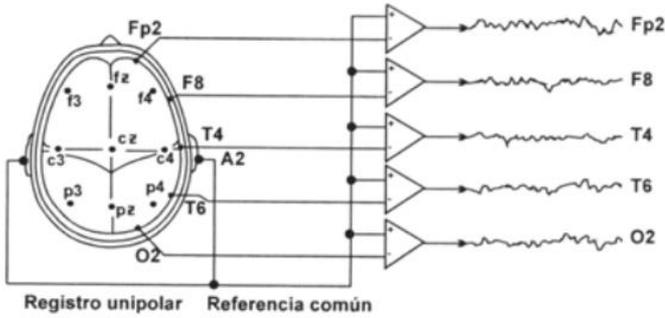


Figura 10 Esquema de montaje para un registro unipolar

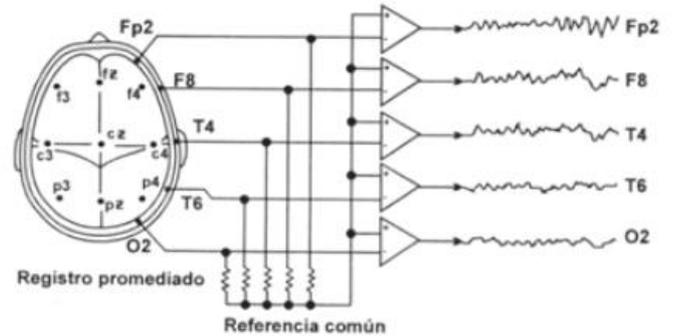


Figura 11 Esquema de montaje para un registro unipolar promediado

En el montaje de la Figura 11 se hace uso de sistema Wilson, al realizar la interconexión entre todos los electrodos usando resistencias de valor bajo, entre 1 y 1'5 Mohmios. Este sistema permite tomar tantos registros como se desee, tan solo dependiendo del número de canales disponibles.

También se pueden tomar otro tipo de referencias, con el objetivo de eliminar ciertas interferencias. Esto se aplica, por ejemplo, para el caso de atenuar los artefactos debidos a los parpadeos o, también, para los debidos a pulsaciones cardiacas. Para estos casos se sitúan dos o más electrodos en los párpados, para el caso de los parpadeos, o en zonas cercanas al corazón, para el caso de las pulsaciones cardiacas. Así se consigue que en estos electrodos se cortocircuite la señal de los parpadeos o la cardiaca, atenuándose la señal en gran medida.

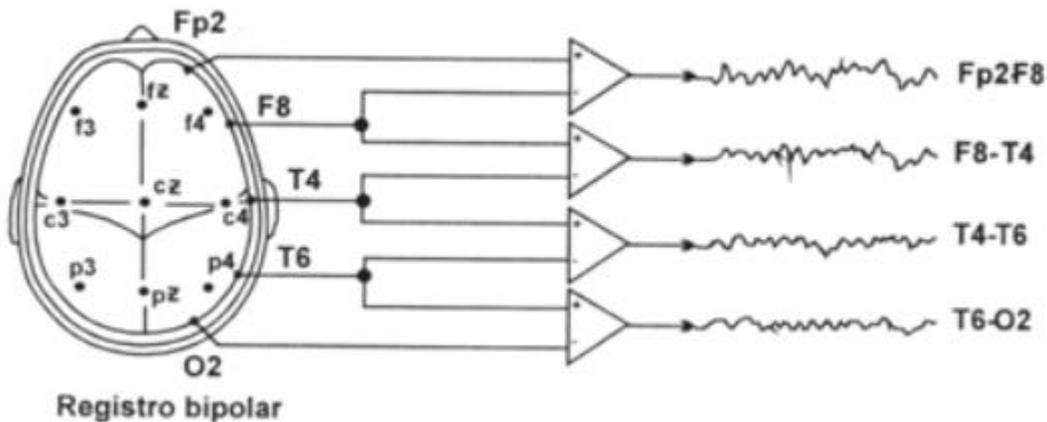


Figura 12 Esquema de montaje para un registro bipolar

En la Figura 12, se puede observar que se toman parejas de electrodos activos, midiéndose la diferencia de potencial entre cada par. Esto se conoce como registro bipolar y permite obtener un número enorme de combinaciones entre electrodos en grupos de 8, de 12, de 16 o más, según el número de canales. Al haber tantas combinaciones posibles, es necesario indicar que combinaciones tendrán mayor interés. A cada una de estas combinaciones se le llama montaje.

Se puede distinguir entre montajes de largas distancias o de cortas distancias, dependiendo si se usan parejas de electrodos alejados o de electrodos vecinos. Sin embargo, los montajes también han sido clasificados por la Federación Internacional de EEG y Neurofisiología en longitudinales y transversales, que se pueden observar en la Figura 13.

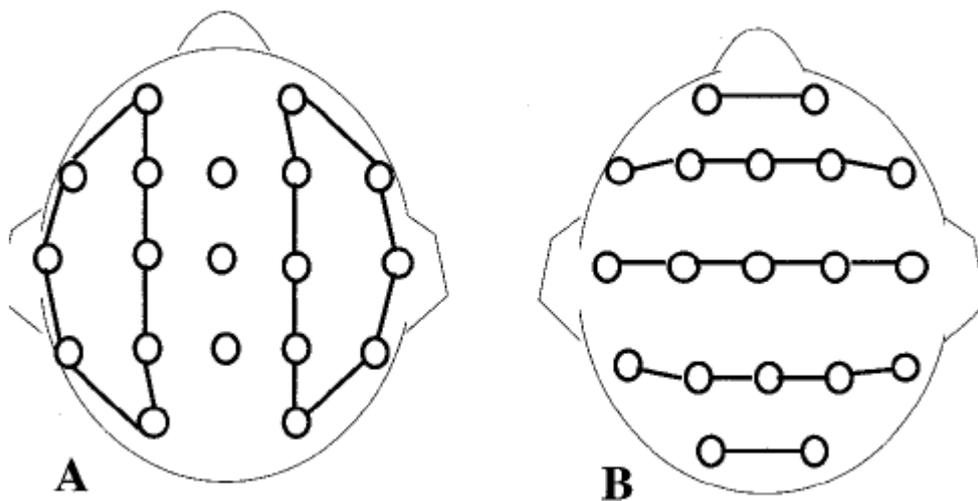


Figura 13 Montajes bipolares longitudinal, o de doble banana, (A) y transversal (B)

Además, hay otros montajes, como el temporal ring o las Zetas del Standard Cero o, también, las triangulaciones.

En el diseño de montajes se recomienda usar como mínimo 8 canales, utilizar el sistema 10-20 de posicionamiento de electrodos y se recomienda también que cada sesión rutinaria incluya uno de los montajes de los tres tipos principales: referencial, longitudinal bipolar y transversal bipolar.

2.3. Base de datos

La base de datos, que se usará en este trabajo, fue recogida en el Children's Hospital Boston y está formada por registros EEG de sujetos pediátricos con crisis intratable. Los sujetos fueron monitorizados hasta varios días después de la retirada de la medicación antiepiléptica, para caracterizar sus crisis y evaluar la posibilidad de intervención quirúrgica [8].

Se obtuvieron grabaciones, agrupadas en 24 casos, de 23 sujetos (5 varones, de edades entre 3 y 22 años, y 17 mujeres, de edades entre 1,5 y 19 años). El caso chb21 se obtuvo 1,5 años después del caso chb01 y pertenecen al mismo sujeto [8].

En cada sujeto se tienen entre 9 y 42 grabaciones en ficheros de extensión '. Edf' continuos. Casi todas estas grabaciones tienen duración de una hora, menos en el caso chb10 que duran dos horas y cuatro horas en los casos chb04, chb06, chb07, chb09 y chb23 [8].

Todas las señales se digitalizaron a 256 muestras por segundo con una resolución de 16 bits. La mayoría de los archivos contienen 23 señales EEG (24 o 26 en algún caso). Se utilizó el Sistema Internacional 10-20 para posicionar los electrodos [8].

Para cada sujeto se dispone de un fichero llamado chbnn.summary.txt, en el cual se indica el montaje utilizado, la hora en la que se realizaron las distintas grabaciones e, incluso, se indica en que segundo empiezan y en cual acaban las crisis epilépticas que se produzcan [8].

A continuación, en la Figura 14, se incluye el contenido del fichero SUBJECT-INFO, en el que se informa de la edad y el sexo de los sujetos, excepto del caso chb24.

Case	Gender	Age (years)
chb01	F	11
chb02	M	11
chb03	F	14
chb04	M	22
chb05	F	7
chb06	F	1.5
chb07	F	14.5
chb08	M	3.5
chb09	F	10
chb10	M	3
chb11	F	12
chb12	F	2
chb13	F	3
chb14	F	9
chb15	M	16
chb16	F	7
chb17	F	12
chb18	F	18
chb19	F	19
chb20	F	6
chb21	F	13
chb22	F	9
chb23	F	6

Figura 14 Fichero SUBJECT-INFO con la edad y el género de los sujetos

3 INTELIGENCIA ARTIFICIAL

Se trata de un programa o software diseñado para realizar operaciones propias de la inteligencia humana, como pueden ser tareas de autoaprendizaje o de clasificación.

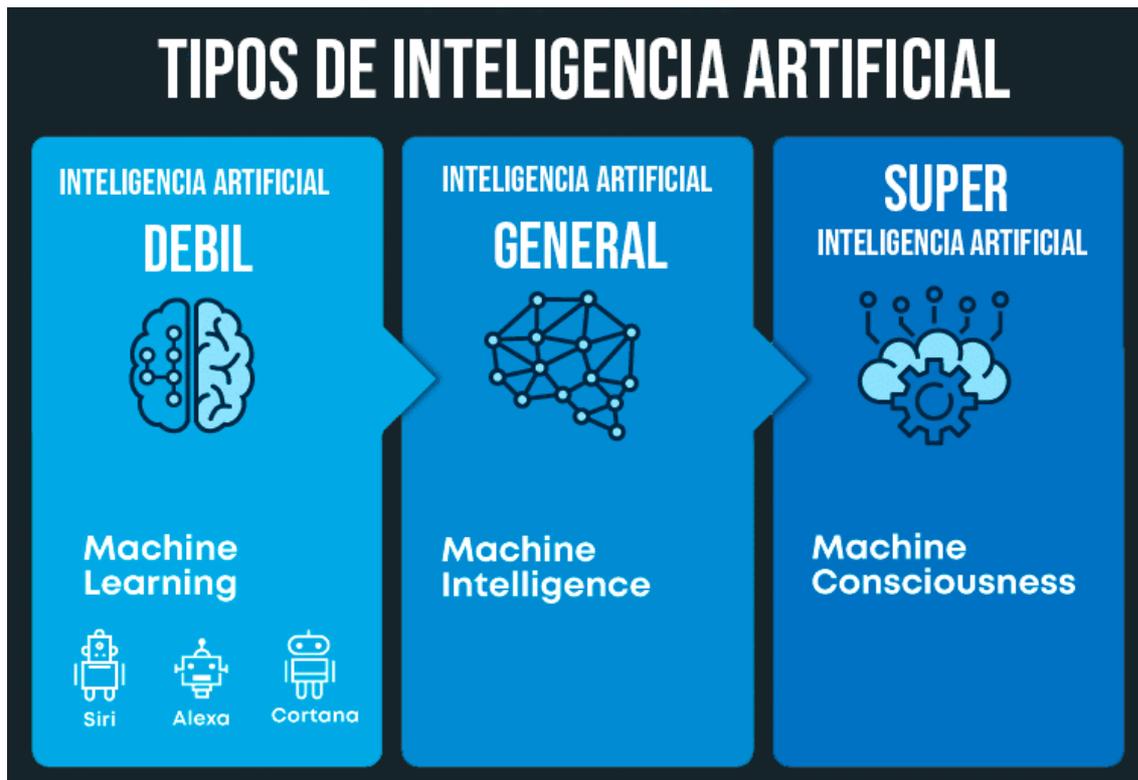


Figura 15 Tipos de inteligencia artificial [9]

La inteligencia artificial se ha dividido en tres grandes categorías. Estas se muestran en la Figura 15 y son:

- Inteligencia artificial débil

Es un tipo de inteligencia racional que se limita a un ámbito estrecho, es decir, que solo pretende ser aplicable a un tipo específico de problemas bien definidos y acotados. La inteligencia utilizada por la IA estrecha es limitada, ya que no son capaces de adaptarse a su entorno sino, tan solo, utilizar una serie de casos posibles ya definidos para resolver de la mejor

forma posible el problema [10].

Hay muchos ejemplos de este tipo de inteligencia, como los asistentes inteligentes (como Siri, Cortana, Alexa o Google Assistant), los chatbots (Eliza, Vergic, Ada...) o los asistentes de voz (Movistar, Orange...).

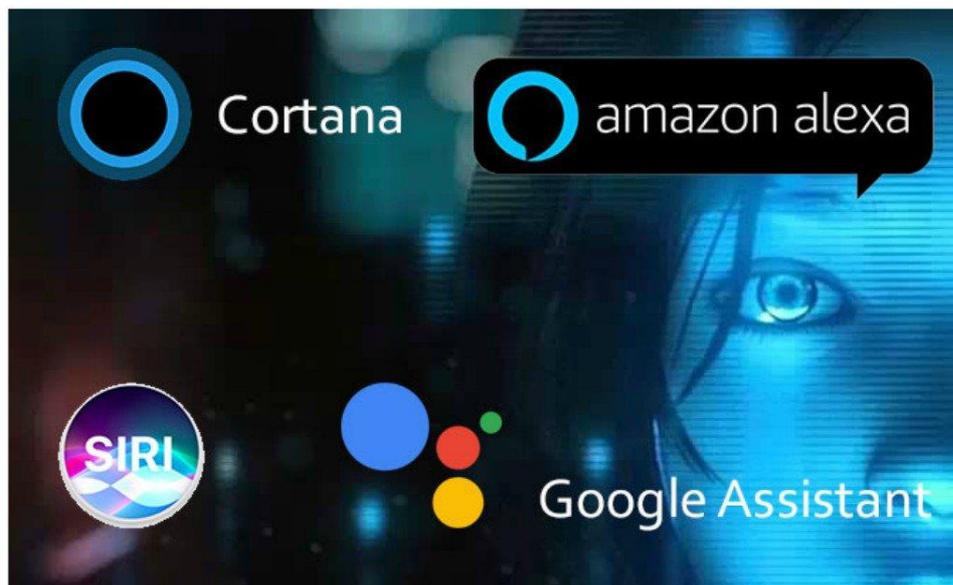


Figura 16 Asistentes inteligentes [11]

- Inteligencia artificial general

Es una inteligencia algo más compleja, similar a la humana, la que podría razonar y realizar juicios.

Esta inteligencia solo existe de momento a nivel teórico y no sería una unión de inteligencias débiles, sino que, al poder razonar, podría resolver miles de tareas distintas, tal como los seres humanos.

Este tipo de IA pasaría sin problemas el test de Turing que es una prueba que permite comprobar que una máquina es capaz de tener un comportamiento inteligente similar o indistinguible al de un ser humano. Lo que mide esta prueba viene reflejado en el diagrama de la Figura 17.

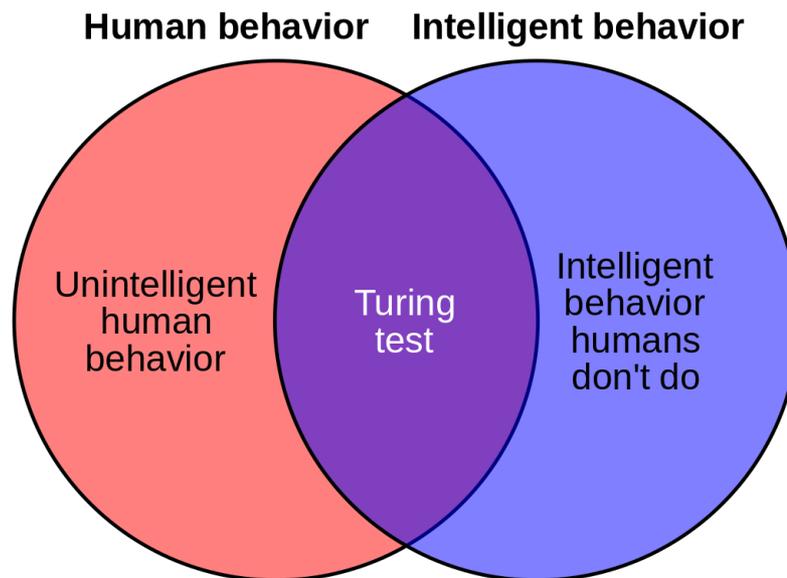


Figura 17 Esquema del test de Turing [12]

- Inteligencia artificial fuerte

Un último tipo de inteligencia artificial es la IA fuerte, inteligencia que dispondría de “estados mentales” y sería consciente de sí misma.

Esta inteligencia iría más allá de la humana al tener una gran potencia de cálculo y al ser capaz de adaptarse al entorno con rapidez. Sería capaz de resolver cualquier problema, sentir emociones o contar con una experiencia subjetiva propia.

Esta inteligencia, al igual que la anterior, existe solo teóricamente y plantearía en un principio grandes problemas éticos [10].

Otras fuentes clasifican la inteligencia artificial en cuatro tipos, que son: máquinas reactivas (decisiones sobre el momento presente sin tener en cuenta el pasado, sin evolución), memoria limitada (pueden almacenar información del pasado pero de forma limitada), teoría de la mente (capaces de comprender el entorno que les rodea y entender y aprender cómo funcionan sistemas, objetos o personas) y autoconciencia (una IA que tiene consciencia de sí misma que se reconoce como un ente independiente y es capaz de tomar decisiones) [13].

La inteligencia artificial actual está dividida en diversos campos, siendo los más importantes los que se muestran en la Figura 18. Consistiendo cada uno de estos campos en lo siguiente:

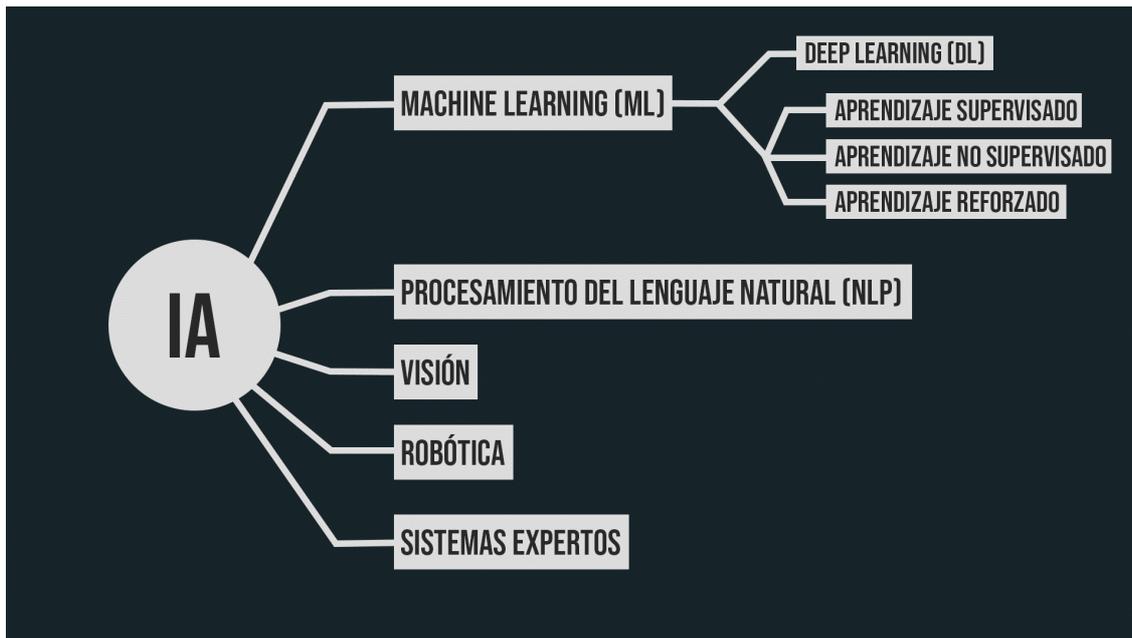


Figura 18 Categorías más importantes de inteligencia artificial [14]

➤ Machine Learning (ML – aprendizaje automático)

Desarrolla sistemas que realizan tareas específicas a través del reconocimiento de patrones. Dependiendo de cómo se presenten los datos se divide en:

- Aprendizaje supervisado: aprendizaje a partir de datos etiquetados. Este tipo de aprendizaje se suele utilizar en:
 - Problemas de clasificación.
 - Problemas de regresión.

Los algoritmos de aprendizaje supervisado más habituales son:

- Árboles de decisión
 - Clasificación de Naïve Bayes – clasificador bayesiano.
 - Regresión por mínimos cuadrados.
 - Regresión logística.
 - Support Vector Machine (SVM – máquina de soporte vectorial): se utilizará en este proyecto.
 - Métodos “Ensemble”
- Aprendizaje no supervisado: aprendizaje a partir de datos no etiquetados, teniendo un carácter exploratorio. Este tipo de aprendizaje se suele utilizar en:

- Problemas de clustering – problema de agrupación,
- Agrupamientos de co-ocurrencias.
- Perfilado o profiling

Los algoritmos de aprendizaje no supervisado son:

- Algoritmos de clustering – algoritmos de agrupación.
 - Análisis de componentes principales (ICA).
 - Descomposición en valores singulares.
- Aprendizaje reforzado: este aprendizaje se caracteriza por aprender y mejorar la respuesta del modelo utilizando un proceso de retroalimentación [15].
 - Deep Learning (DL – aprendizaje profundo): los algoritmos de aprendizaje profundo fueron constituidos para trabajar como una red neuronal real, esto es, con distintas capas superpuestas no lineales dispuestas de forma que el procesamiento de datos va aumentando su complejidad con cada capa, al igual que funciona el cerebro [16].

➤ Procesamiento del lenguaje natural (NLP)

Es un campo dentro de la inteligencia artificial que se encarga de estudiar como interaccionan mediante el lenguaje natural humanos y máquinas, permitiendo a las máquinas descifrar, comprender, traducir o, incluso, generar lenguaje natural hablado automáticamente [17].

➤ Visión

Es la rama de la inteligencia artificial que se encarga de procesar y analizar imágenes y videos. Las utilidades de esta IA son amplias e incluyen, entre otras, el reconocimiento facial, la clasificación de imágenes y videos o el reconocimiento de objetos.

➤ Robótica

Se encarga de producir robots que aporten soluciones a problemas mecánicos, como brazos robóticos en una cadena de montaje o robots de atención al público.

➤ Sistemas expertos

Tienen como objetivo realizar la toma de decisiones tal y como lo haría un humano, pudiendo así solucionar problemas complejos razonando.

4 INTRODUCCIÓN A TÉCNICAS DE IA

En este proyecto se van a utilizar varias técnicas de inteligencia artificial. El propósito de esta sección es introducir dichas técnicas, a definir sus características y a compararlas entre sí.

4.1. Perceptrón multicapa (MLP)

Es una red neuronal artificial (RNA), es decir, un modelo computacional que pretende emular el comportamiento de una red neuronal biológica, por lo que dispone de neuronas artificiales conectadas y comunicadas entre sí. Estas neuronas artificiales están dispuestas en capas de neuronas, con lo que se consigue una red neuronal multicapa de alimentación directa en la que las entradas se procesan secuencialmente, capa por capa, es decir, una capa solo recibe señales de la capa inmediatamente anterior y las envía solo a la siguiente capa.

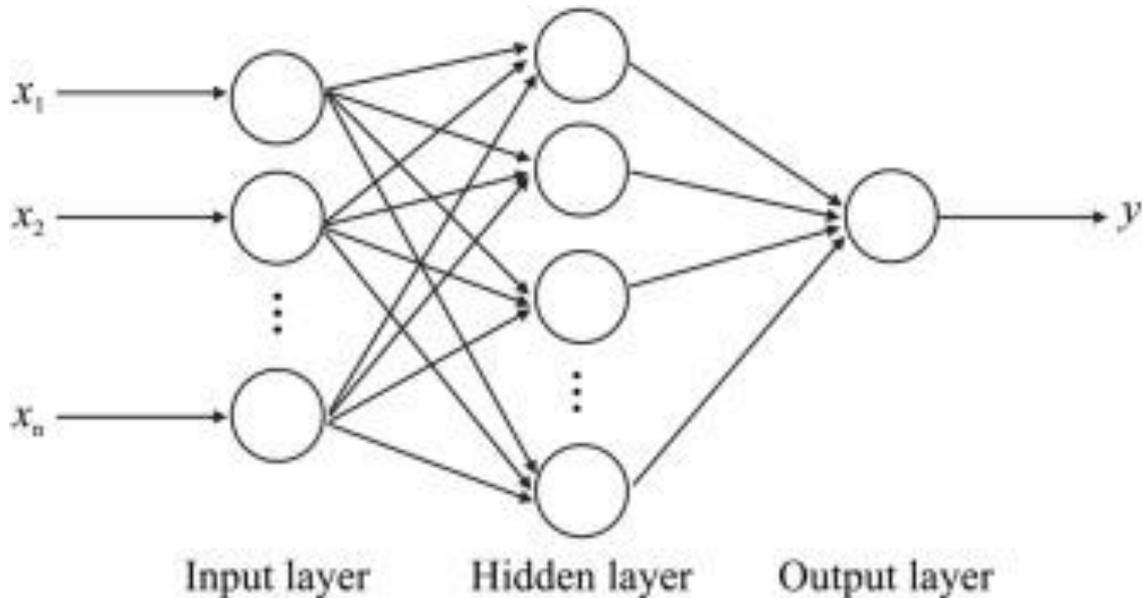


Figura 19 MLP [18]

Como se observa en la Figura 19, se tiene de una capa de entrada, una capa oculta y una de salida, estando todas las neuronas de cada capa completamente conectadas tanto a las neuronas de la capa anterior, como a las neuronas de la siguiente capa. Si se tuviera más de una capa oculta se estaría en el caso de una red neuronal artificial profunda [19].

Se puede agregar una cantidad arbitraria de capas de neuronas, así como una cantidad arbitraria de

neuronas en cada capa, por lo que al final se tiene un sistema en el que se pueden tratar el número de capas y el número de neuronas por capa como hiperparámetros. Por tanto, el objetivo será optimizar dichos hiperparámetros para realizar la tarea de la forma más eficiente posible [19].

Sin embargo, hay un límite al número de capas que se pueden añadir, puesto que cuantas más capas se añadan más probable es la convergencia hacia un mínimo local de la función objetivo.

Estas redes son en esencia una combinación de las neuronas, los sesgos, las interconexiones y de los pesos de estas interconexiones. Cada neurona de la red se puede definir con las ecuaciones (1) y (2) [18].

$$y_k = f(u_k + b_k) \quad (1)$$

$$u_k = \sum_{i=1}^N w_{ki} x_i \quad (2)$$

Donde la k es el número de neurona, las x_i son las señales de entrada, las w_{ki} son los pesos de las conexiones de cada neurona, u_k es la combinación lineal de los pesos de entrada, N es el número de interconexiones, b_k es el sesgo, f es la función de activación e y_k es la salida de la neurona.

La función de activación, ‘ f ’, es aquella que se aplica en todas las neuronas tras calcular la combinación lineal de pesos y entradas. Esta función de activación es una forma de transmitir la información por las conexiones de salida. En el caso en el que se quisiera que dicha información se transmita sin modificar se utilizaría la función identidad como función de activación. Sin embargo, el principal cometido de la función de activación es la de aportar una “no linealidad” al modelo, pudiendo así resolver problemas más complejos. Una red sin funciones de activación no sería más que una simple regresión lineal [20].

Las funciones de activación más utilizadas son las siguientes:

- Función escalón

Esta función transmite un ‘0’ para valores negativos y un ‘1’ para valores positivos. Se utiliza para realizar una clasificación estricta que no permita casos intermedios. Se representa en la Figura 20 y sigue la ecuación (3).

$$\phi = \begin{cases} 0 & \text{si } x < 0 \\ 1 & \text{si } x > 0 \end{cases} \quad (3)$$

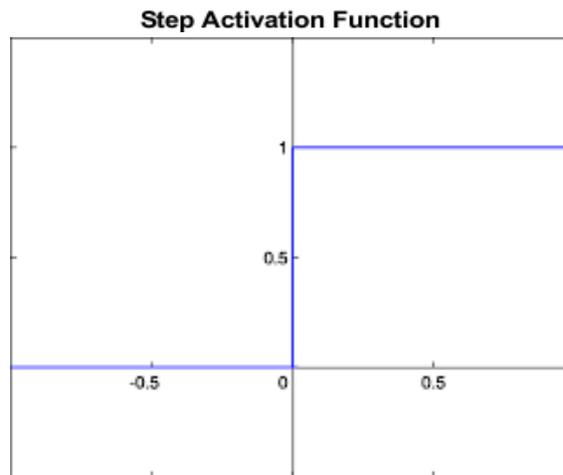


Figura 20 Función escalón

- Función softmax

Se mantiene en el rango entre '0' y '1' y distingue entre los valores positivos y negativos, pero, en este caso, la clasificación no es tan estricta como en la función escalón, siendo más suave y permitiendo casos intermedios. Es utilizada normalmente en tareas de clasificación multiclase. Se representa en la Figura 21 y sigue la ecuación (5).

$$\text{softmax}(x_i) = \frac{\exp(x_i)}{\sum_j \exp(x_j)} \quad (4)$$

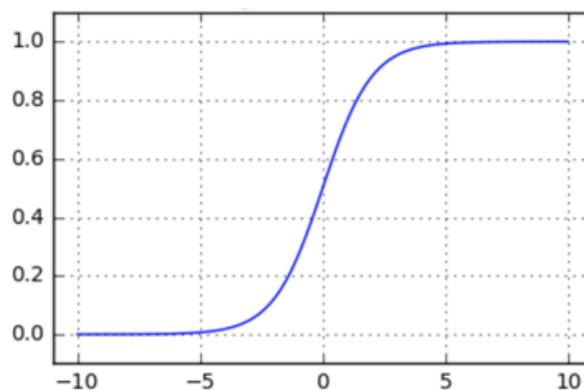


Figura 21 Softmax

- Función sigmoide

Es un caso particular de la función Softmax, donde el número de clases es dos. Se representa en la Figura 22 y sigue la ecuación (5).

$$\phi = \frac{1}{1 + e^{-x}} \quad (5)$$

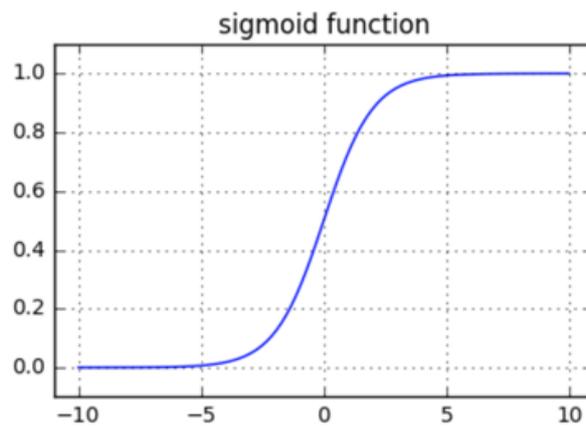


Figura 22 Sigmoide

- Función rectificadora (ReLU)

Esta función descarta los valores negativos poniéndolos a '0' y deja pasar los valores positivos, manteniendo su valor. Se representa en la Figura 23 y sigue la ecuación (6).

$$\phi = \max(x, 0) \quad (6)$$

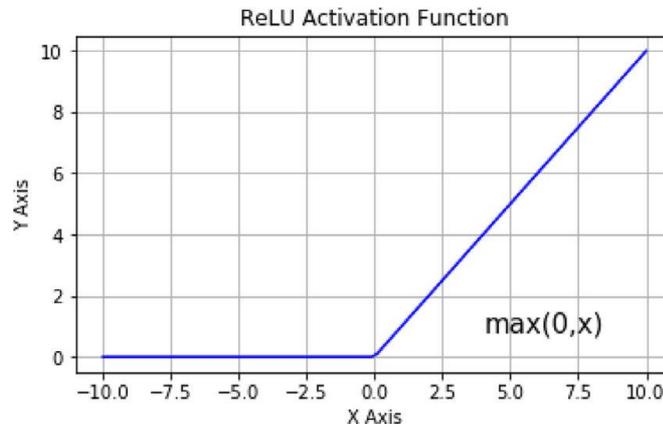


Figura 23 ReLU

- Función tangente hiperbólica

Esta es similar a la función sigmoide, con la salvedad de que esta se mueve en el rango entre '-1' y '1'. Se representa en la Figura 24 y sigue la siguiente ecuación (7).

$$\phi = \frac{1 - e^{-2x}}{1 + e^{-2x}} \quad (7)$$

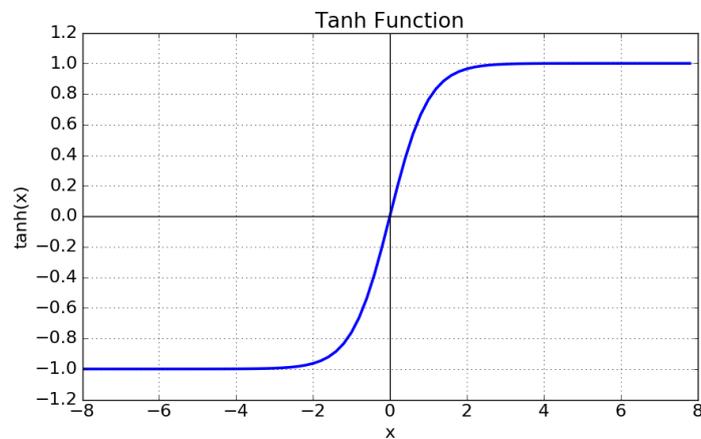


Figura 24 Tangente hiperbólica

Algunas peculiaridades a tener en cuenta son:

- **Inicialización de pesos y sesgos**

Al comienzo de la construcción de la red se deben asignar un valor inicial a todos los pesos y sesgos de la red. Generalmente los sesgos siempre se inician a 0 y, por otra parte, los pesos se suelen iniciar con valores aleatorios, aunque hay técnicas con las que se obtienen mejores resultados.

Si los pesos se inicializan a 0 la derivada con respecto a la función de pérdidas es la misma para todos los pesos, por lo que todos los pesos tendrían los mismos valores en las siguientes iteraciones. De esta forma, lo que se acaba consiguiendo es que las capas ocultas sean simétricas y continuas en las distintas iteraciones, por lo que, establecer inicialmente los pesos a 0 no hace a este modelo mejor que un modelo lineal.

Es más adecuado si se inician los pesos en valores aleatorios, sin embargo, no todos los valores son adecuados, presentándose varias posibilidades:

- Si los valores establecidos fueran muy altos, se conseguiría que la pendiente del gradiente cambie lentamente, por lo que se tendrían problemas de convergencia, haciendo que el modelo no consiga alcanzar la solución o tarde más tiempo.
- Si estos fueran demasiado bajos, serían muy cercanos a 0 y se tendrían un resultado similar que cuando se asignan a 0.
- Asignar valores más adecuados, con los que se consiguen mejores resultados y se puede realizar mediante varias técnicas que se utilizan habitualmente, como, por ejemplo:
 - La inicialización de He: la cual consiste en utilizar la función de activación ReLU e inicializar aleatoriamente los pesos, pero multiplicándolos por una distribución normal de media 0 y desviación estándar $\sqrt{\frac{2}{N_{in}}}$, siendo N_{in} el número de entradas de la capa [21].
 - La inicialización de Xavier Glorot: similar a la inicialización de He, pero utilizando $\tanh()$ como función de activación y multiplicando los pesos aleatoriamente escogidos por una distribución normal de media 0 y desviación estándar $\sqrt{\frac{2}{N_{in}+N_{out}}}$, siendo N_{in} y N_{out} el número de entradas y salidas, respectivamente [22].
 - Otra técnica utilizada es el esquema LSUV donde los pesos se preinician con matrices ortonormales y posteriormente se obliga a que la varianza de cada capa sea igual a 1. Esta técnica funciona correctamente con diversas funciones de activación [23].

- **Función de costes**

La función de costes mide el error a la salida, que es la diferencia entre el valor de salida calculado por la red y el valor real que debería obtenerse. Esta información dada por la función de costes se utiliza en las sucesivas iteraciones para realizar el ajuste de los pesos de la red propagando hacia atrás (backpropagation) la función de costes obtenida en las distintas iteraciones, teniendo como objetivo que dicha diferencia entre el valor de salida y el valor real sea lo más pequeña posible, esto es, reduciendo la función de costes.

Hay distintos estimadores de error de la función de coste, entre los que destacan:

- Error cuadrático medio (RMSE), que se obtiene mediante la ecuación (8).

$$RMSE = \sqrt{\sum_{i=1}^n \frac{(\hat{y}_i - y_i)^2}{n}} \quad (8)$$

Siendo \hat{y}_i es el valor estimado a la salida, y_i la salida real y n el número de salidas.

- Error absoluto medio (MAE), que se obtiene mediante la ecuación (9).

$$MAE = \frac{\sum_{i=1}^n |\hat{y}_i - y_i|^2}{\frac{n}{n-1} \sum_{i=2}^n |\hat{y}_n - y_n|^2} \quad (9)$$

- Error de escala absoluta media (MASE), que se obtiene mediante la ecuación (10).

$$MASE = \sum_{i=1}^n \frac{|\hat{y}_i - y_i|^2}{n} \quad (10)$$

- Entropía cruzada categórica (Categorical Cross-Entropy), que se obtiene mediante la ecuación (11).

$$\mathcal{L}(\theta) = \sum_{i=1}^n y_i \log(\hat{y}_i) \quad (11)$$

- Entropía cruzada binaria (Binary Cross-Entropy), que se obtiene mediante la ecuación (12).

$$\mathcal{L}(\theta) = \sum_{i=1}^n y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i) \quad (12)$$

Este último estimador del error es el que se aplicará en los distintos modelos de este trabajo.

- **Descenso del gradiente**

En cada iteración, una vez calculada la función de costes, se calcula el gradiente como una derivada de varias variables respecto a todos los parámetros de la red, que gráficamente es la pendiente de la tangente de la función de costes con los pesos y sesgos actuales. Este gradiente es un vector que indica la dirección y el sentido de máximo aumento de la función de costes, esto es, del error. Como se quiere que el error se reduzca, se utiliza el sentido contrario al indicado por el gradiente.

Esto último es lo que se conoce como descenso del gradiente, el cual, en una red neuronal profunda, podría ser complicado de calcular si no fuera por el algoritmo de propagación hacia atrás que se detalla a continuación.

- **Propagación hacia atrás – backpropagation**

Este algoritmo consiste en calcular las derivadas parciales de la función de coste con respecto únicamente a las variables de la última capa. Tras esto, se pasa a la capa anterior y se calculan de nuevo las derivadas parciales de la función de costes con respecto a los parámetros de dicha capa anterior. Estos cálculos se facilitan en gran medida por la regla de la cadena y se realizan hasta recorrer todas las capas de la red y llegar al inicio de la misma.

Una vez obtenido el vector gradiente se actualizan todos los parámetros de la red restándoles el valor de gradiente correspondiente y se comienza una nueva iteración. Esto se realiza hasta que termine el proceso con el objetivo de obtener el mínimo error posible.

- **Algoritmos de optimización**

Estos se aplican sobre el descenso del gradiente y hay diferentes técnicas, algunas de las cuales son:

— Momentum:

Esta técnica se basa en actualizar los parámetros sumándole un término adicional que tenga en cuenta la iteración anterior, y, por tanto, teniendo en cuenta los gradientes de las iteraciones anteriores. Con esto se consigue que se acelere el descenso del gradiente cuando el modelo vaya encaminado hacia la dirección correcta y que se atenúen las oscilaciones.

— AdaGrad (Adaptive Gradient Algorithm):

Esta técnica se basa en ir ajustando el learning rate – tasa de aprendizaje - a los parámetros de la red, realizando actualizaciones grandes para los parámetros que apenas cambian y actualizaciones pequeñas para los parámetros que cambian frecuentemente.

— Adadelta:

Es una mejora de AdaGrad donde, en vez de tener en cuenta el gradiente acumulado desde el principio de la red, este se restringe a una ventana de tamaño fijo n y, por tanto, a los últimos n gradientes.

— RMSProp (Root Mean Square Propagation):

Permite solucionar el problema de los minibatches (grupo de muestras a partir de las que se calcula el gradiente), en los que se divide entre un gradiente diferente cada vez. RMSProp obliga a que minibatches adyacentes tengan gradientes similares. Se obtienen medias móviles de los gradientes al cuadrado para cada peso y se dividen los gradientes entre la raíz del cuadrado de la media [24].

— Adam (Adaptive Moment Estimation):

Adam es el algoritmo resultante de la combinación entre los algoritmos RMSProp y momentum. Realiza una combinación lineal entre el gradiente y el incremento anterior y tiene en cuenta los gradientes recientemente aparecidos en las actualizaciones para mantener diferentes tasas de aprendizaje para cada parámetro.

- **Modelo propuesto**

En el modelo que se propone, se aplican, una vez segmentadas, las señales EEG crudas a un perceptrón multicapa que consta de cuatro capas ocultas. El número de neuronas en cada capa oculta será de 300, 100, 50 y 20, respectivamente, aunque se realizarán pruebas cambiando el número de capas y de neuronas. Al ser, el perceptrón multicapa, un modelo totalmente conectado, se tendrá un número total de parámetros entrenables elevado. El modelo se entrena con backpropagation, o propagación hacia atrás, y se optimiza mediante el algoritmo RMSprop. Como función de pérdidas se utiliza la entropía cruzada binaria que viene definida en la ecuación (13).

Como funciones de activación se utiliza la función ReLU (Rectifier Linear Unit) para las capas ocultas, con el fin de añadir no linealidad y robustez contra el ruido en los datos de entrada. La ReLU se ha definido anteriormente en la ecuación (6). En la capa de salida se utiliza la función sigmoide, con la cual decidir a qué clase pertenecen los datos de entrada. La sigmoide viene dada por la ecuación (5).

$$l(y, \hat{y}) = -\frac{1}{n} \sum_{i=1}^n [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)] \quad (13)$$

Donde y es la salida calculada, \hat{y} es la salida deseada y $l(y, \hat{y})$ es la función de pérdidas. Esta arquitectura de clasificador MLP se representa en la Figura 25.

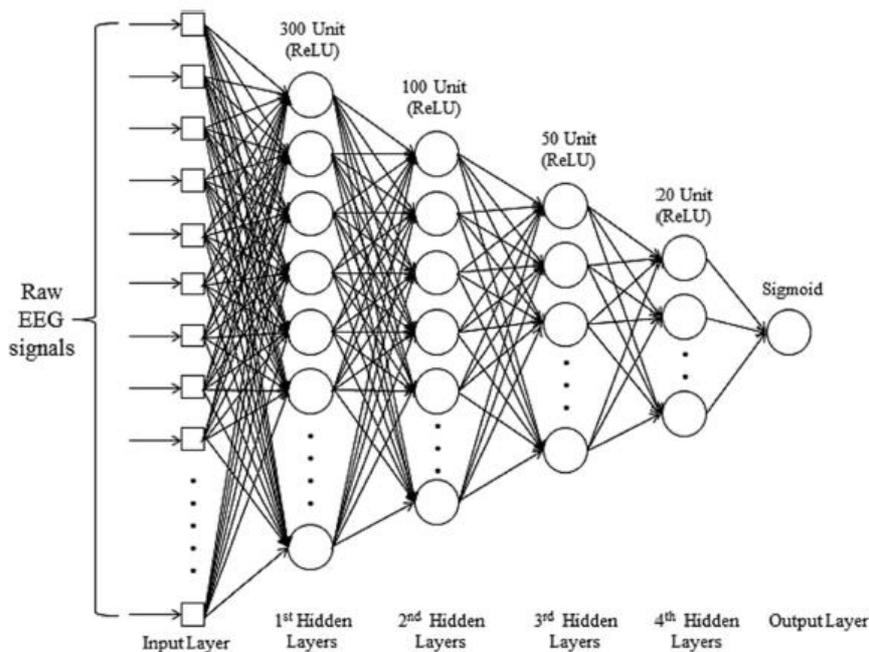


Figura 25 Arquitectura propuesta para el clasificador basado en MLP [4]

4.2. Red neuronal convolucional (CNN)

Las redes neuronales convolucionales están constituidas por un conjunto de modelos que se basan en cómo funciona la corteza visual primaria del cerebro humano cuando reconoce objetos, y al utilizar matrices bidimensionales son muy efectivas para tareas de visión artificial [19].

Estas redes se utilizan mayoritariamente en reconocimiento de imágenes y vídeo, en sistemas de recomendación, en clasificación de imágenes, en análisis de imágenes médicas, en el procesamiento del lenguaje natural y en series temporales.

Debido al buen funcionamiento de las redes CNN, cada vez se han utilizado más y se han obtenido grandes mejoras en las aplicaciones de visión artificial y de aprendizaje automático.

Las redes neuronales convolucionales aprovechan el patrón jerárquico en los datos y combinan patrones o características más pequeñas y simples (en imágenes serían bordes, manchas, etc) para obtener así patrones más complejos (en imágenes serían formas de objeto como, por ejemplo, una silla, una casa o un gato). Esto hace que las CNN sean como versiones regularizadas del perceptrón multicapa, teniendo una escala de conectividad y complejidad menor. La conectividad total de las neuronas en los perceptrones multicapa los hace propensos al sobreajuste de los datos, problema que se corrige con las CNN al tener menor número de conexiones [19].

Las CNN están formadas por una capa de entrada, una capa de salida y de múltiples capas ocultas, que pueden ser capas convolucionales de una o más dimensiones, capas de submuestreo (pooling), capas totalmente conectadas (Fully Connected), que son en esencia perceptrones multicapa, o capas de normalización.

A diferencia de las capas convolucionales o de las capas totalmente conectadas, las capas de submuestreo no añaden a la red ningún parámetro que se pueda aprender, como pesos o sesgos. [19]

En primer lugar, se encuentra la fase de extracción de características, en la cual se incluyen alternativamente capas convolucionales y de submuestreo que hacen que se reduzca la dimensionalidad de los datos. Al final se incluyen capas de perceptrón para realizar la clasificación final sobre las características extraídas. Esto se muestra en la Figura 26.

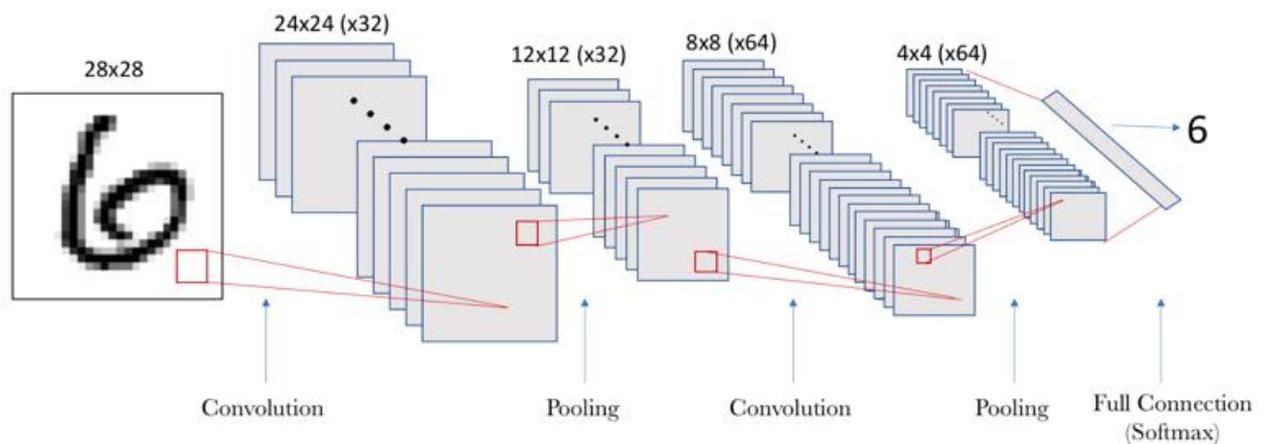


Figura 26 CNN [25]

A continuación, se van a detallar cada uno de los componentes por separado:

- Neuronas convolucionales:

En la parte de extracción de características, las neuronas de las capas de convolución llevan a cabo operaciones sobre los datos en 2D que pasan por estas. La salida de cada neurona convolucional se calcula con la ecuación dada por (14) [26].

$$X_{out} = f \left(b_j + \sum_i K_{ij} \otimes X_{int} \right) \quad (14)$$

Siendo X_{out} la matriz de salida de la neurona j , que se calcula mediante la combinación lineal de las salidas X_{int} de las neuronas de la capa anterior aplicándole a cada una de ellas el núcleo convolucional K_{ij} correspondiente a cada conexión, denotando el operando de convolución con el símbolo \otimes . A lo anterior se le suma el sesgo b_j y se le aplica la función de activación f [26].

- Neuronas de submuestreo:

Estas neuronas utilizan la operación de max-pooling que funciona cogiendo el valor máximo de una ventana de muestra de tamaño predefinido y asigna este valor a toda la zona de datos o píxeles del tamaño de la ventana de muestra. La salida de esta operación son los valores máximos de todas las zonas del tamaño de la ventana de muestra en los que se pueden dividir las matrices de datos. Por tanto, a la salida, las matrices de datos se habrán reducido por un factor igual a la ventana de muestra [19]

- Neuronas de clasificación

Los datos de entrada de estas neuronas ya han pasado la parte de extracción de características, por lo que ya han sido reducidos y disponen de un número mucho menor de características que contienen más información.

En esta fase de clasificación se clasifican estas características asignándole una etiqueta u otra.

Estas neuronas funcionan igual que las de un perceptrón multicapa, por lo que, las salidas se obtendrán, como ya se vio anteriormente, con las ecuaciones dadas en (1) y (2).

- **Modelo propuesto**

Estas redes tienen un buen rendimiento, obteniéndose en general buenos resultados, lo que se debe a que son capaces de extraer automáticamente las características espaciales significativas que mejor representan los datos sin disponer de ninguna decisión humana a la hora de elegir dichas características más representativas y habiéndole pasado los datos sin ningún procesamiento previo. La conectividad escasa les aporta una gran ventaja frente a otras redes, ya que el número total de parámetros entrenables es mucho más reducido, siendo más eficiente.

Para este modelo los segmentos de señal EEG se convertirán en matrices 2D para que sea adecuado para una red neuronal convolucional. Se tienen cuatro capas convolucionales y tres capas de agrupación máxima (maximum pooling). En cada convolución se elige un número de 32 filtros (kernels), con un tamaño de 3x2, para cubrir las matrices de los datos EEG que no son cuadradas. Las capas de agrupación máxima tienen un tamaño de agrupación de 2x2. Se utiliza la función ReLU como función de activación de las capas convolucionales, así como, una técnica de normalización por lotes (Batch Normalization technique) para mejorar la velocidad de entrenamiento y reducir el sobreajuste añadiendo algo de ruido a la activación de cada capa. Esta red neuronal convolucional descrita, encargada de la extracción de características, se muestra en la Figura 27.

La transformada de normalización por bloques se define en la ecuación (15).

$$BN_{\gamma, \beta}(x_i) = \gamma \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} + \beta \quad (15)$$

Donde x_i es el vector que se va a normalizar en un mini-batch $B = \{x_1, x_2, x_3, \dots, x_m\}$, μ_B y σ_B^2 son la media y la varianza del actual mini-batch de x_i , respectivamente, ϵ es una constante añadida a la varianza para aportar estabilidad numérica, γ y β son los parámetros de aprendizaje que se utilizan para escalar y cambiar el valor normalizado, respectivamente.

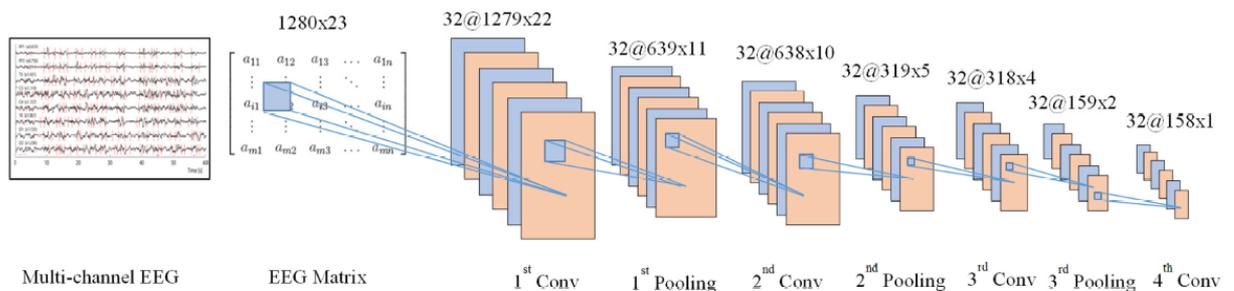


Figura 27 Arquitectura CNN propuesta [4]

Como se puede observar en la figura anterior, las señales EEG se segmentan, obteniéndose matrices 2D de tamaño 1280x23 adecuadas para el modelo CNN y, como se ha indicado anteriormente, se asignan 32 filtros para convolucionarlos con los segmentos, lo que viene indicado en la figura como “32@” seguido por el tamaño de la matriz resultante en cada capa.

Este modelo de CNN se utiliza para realizar una extracción de las características espaciales más representativas. Tras aplicar este modelo y extraer las características más representativas, se utilizan capas completamente conectadas, tales como un perceptrón multicapa, para realizar la tarea de clasificación. Este modelo se muestra en la Figura 28.

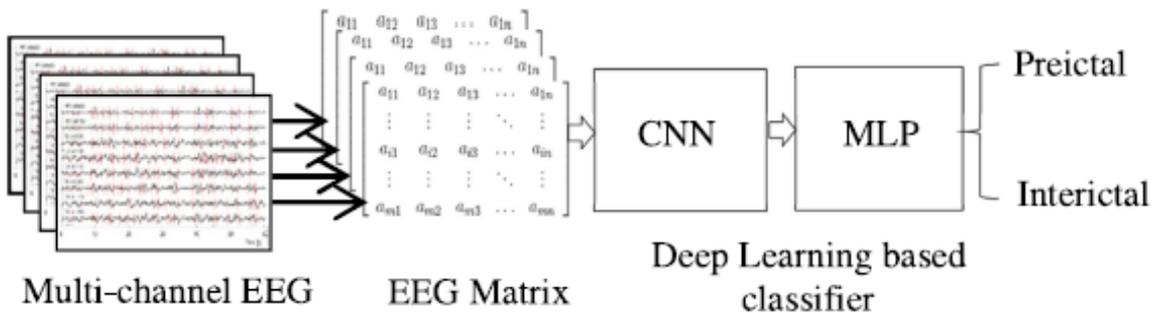


Figura 28 Diagrama de bloques de CNN+MLP [4]

Como se mencionó anteriormente, se está diferenciando entre muestras “preictales” e “interictales”, siendo las “preictales” aquellas muestras que se encuentran hasta una hora antes de la primera convulsión, y, las “interictales”, aquellas que se encuentran a más de cuatro horas antes y después de cualquier convulsión. Esta diferenciación se realiza porque lo que interesa es detectar los ataques epilépticos antes de que sucedan, con la finalidad de poder evitar el ataque o, al menos, reducir todo lo posible los efectos del este.

4.3. Redes neuronales recurrentes (RNN)

Las redes neuronales recurrentes son un tipo de redes que analizan datos de series temporales teniendo un comportamiento dinámico temporal en el que las conexiones entre nodos forman un gráfico dirigido a lo largo de una secuencia temporal [27]. Integran bucles de realimentación, lo que permite que estas redes tengan memoria y la información persista en pasos de tiempo posteriores [28].

En cada instante de tiempo, cada neurona recibe la salida de la capa anterior, así como, la propia salida que esta neurona ha obtenido en el instante de tiempo anterior a modo de realimentación recurrente.

El flujo de información en pasos de tiempos adyacentes hace que la red recuerde eventos pasados y se resuelvan las salidas de las neuronas considerando estos eventos pasados [19]. Esto hace que se tenga en cuenta todo el contexto a la hora de hacer predicciones, imitando la forma de proceder de los humanos al procesar secuencias, con lo que se consigue una mejor resolución que si no se tuvieran en cuenta dichos eventos pasados.

En la Figura 29 se muestra un ejemplo de neurona recurrente.

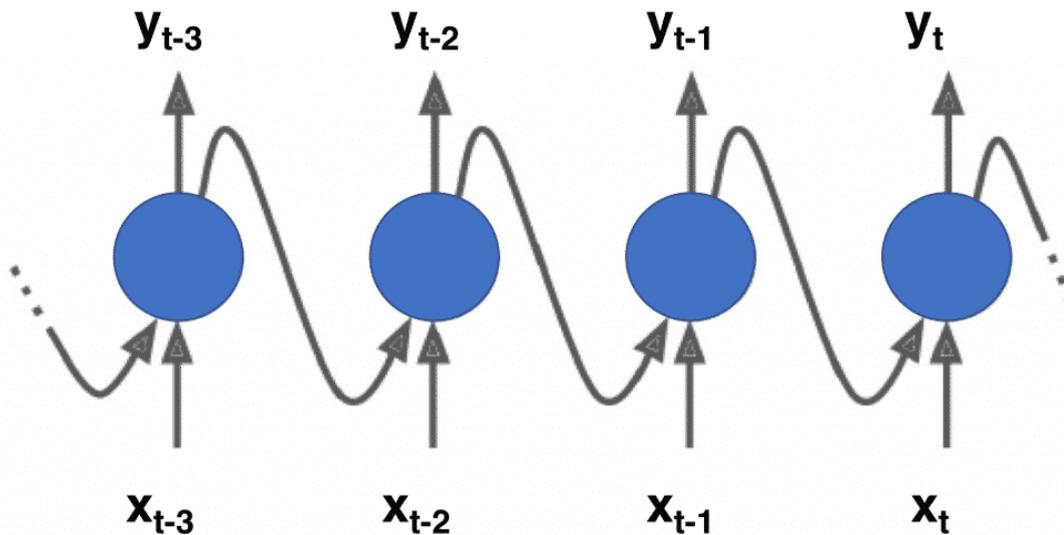


Figura 29 Funcionamiento de una neurona recurrente en una red RNN [28]

Los aspectos a destacar de este tipo de red neuronal son los siguientes:

- Celda de memoria

Tener modelos con memoria es un factor clave, sobre todo a la hora de tratar secuencias temporales. Esta característica hace que las redes neuronales recursivas sean idóneas para tareas como el reconocimiento de voz, la traducción automática o el modelado del lenguaje, entre otras.

Que estas redes tengan esta memoria y recuerden los eventos pasados es importante, sobre todo en secuencias temporales, porque la información ‘del pasado’, esto es, de las muestras anteriores o ya procesadas, puede aportar mucha información sobre las muestras futuras y, por tanto, facilitar en gran medida la labor de predicción, lo cual es el aspecto más importante a la hora de tratar casos de clasificación como, por ejemplo, los mencionados en el párrafo anterior.

- Backpropagation - propagación hacia atrás

En el caso de una red neuronal recurrente, el proceso de backpropagation – propagación hacia atrás - es similar al de las redes anteriores, con la salvedad de que en este caso este proceso se realiza a través del tiempo y se lo conoce como Backpropagation Through Time (BPTT – propagación hacia atrás a través del tiempo).

El proceso de propagación hacia atrás se realiza siempre comprobando el valor de pérdidas obtenido en la red, con el objetivo de minimizarlo.

En el caso de una RNN, es algo más complejo, al depender de instantes de tiempo anteriores, el valor de pérdidas está calculado teniendo en cuenta esos instantes anteriores y, por tanto, a la hora de realizar la propagación hacia atrás, hay que tenerlos en cuenta y se tiene que realizar

lo que se conoce como un proceso de ‘desenrollado’, en el que se desenrolla la red en tantas capas como pasos temporales o datos tenga la secuencia temporal, resultando un grafo enorme que hace que el proceso de BPTT sea costoso computacionalmente. En cada capa desenrollada se mantienen los mismos pesos para acelerar el proceso.

- Problemas de convergencia

Estos problemas que afectan a cualquier red que tenga un alto número de parámetros, sea o no una red neuronal recurrente, y pueden ser originados tanto por tener valores de gradientes altos como por tener valores de gradientes bajos.

Ambos problemas son opuestos, aunque tienen un efecto similar, que es que se tengan problemas de entrenamiento al ser imposible conseguir que el resultado converja.

En el caso en el que se tienen valores altos de gradiente, se le está dando una gran importancia a los pesos, viéndose el resultado muy afectado al variar estos y teniendo así, por tanto, un problema de entrenamiento al resultar imposible que el proceso converja debido a estas grandes variaciones del resultado.

Por otra parte, en el caso de que se tengan valores bajos de gradientes se dirá que los gradientes se han desvanecido, dando lugar a que la variación de los pesos apenas afecte al resultado y, por tanto, no produciendo una variación suficiente con la que poder alcanzar la solución.

- Long-Short Term Memory (LSTM – memoria a largo-corto plazo)

Son una extensión de RNN donde se tiene una memoria más amplia que puede recordar eventos que han pasado hace mucho tiempo.

La memoria de un LSTM permite leer, escribir y borrar información. Esto último se decide según la importancia que se le asigne a la información que se está recibiendo, que se asigna según los pesos, que, a su vez, también van siendo actualizados.

La red está dividida en celdas y cada celda tiene un estado. La función de cada una de estas celdas se decide por los parámetros que se aprenden durante el entrenamiento.

La función de cada celda se decide en última instancia por los parámetros (pesos) que se aprenden durante el entrenamiento.

Las celdas de información de una red LSTM tienen tres puertas: una puerta de entrada para manipular la entrada actual, una puerta de olvido para descartar información irrelevante, y una puerta de salida para hacer predicciones en cada paso de tiempo.

Las LSTM mantienen los gradientes lo suficientemente altos, con una pendiente suficiente para resolver los problemas de desvanecimiento de gradiente consiguiendo que el peso no se diluya entre las capas y consiguiendo, a su vez, que los entrenamientos sean relativamente cortos y que se obtengan precisiones altas.

- **Modelo propuesto**

Como se ha comentado, las RNN son un tipo de red neuronal que cuentan con memoria y recuerdan los eventos pasados, lo cual les aporta una gran ventaja frente a otras redes, ya que permiten hacer predicciones teniendo en cuenta todo el contexto. Esta propiedad hace que estas redes sean adecuadas para la predicción de datos utilizando series temporales, como es el caso de las señales EEG que se emplearán.

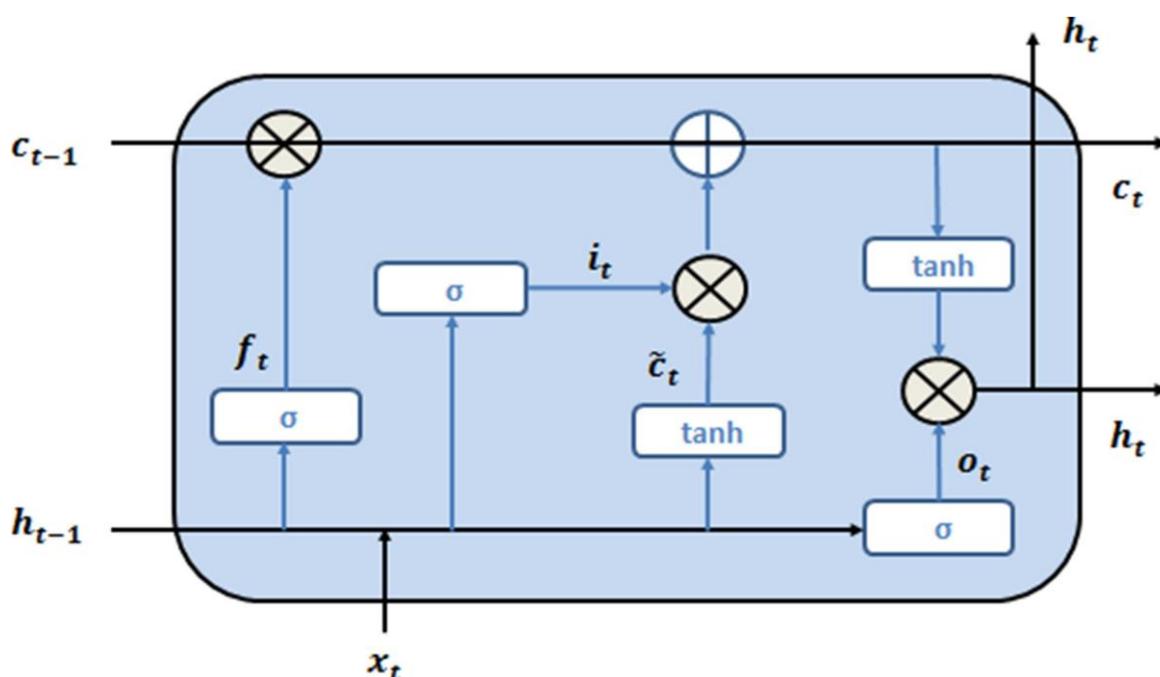


Figura 30 Celda básica LSTM [4]

Para superar los problemas de este tipo de red, que son el desvanecimiento del gradiente y los gradientes explosivos, se van a utilizar redes LSTM (Long Short Term Memory network). Las celdas LSTM, como se ha mencionado anteriormente, constan de tres puertas de control con las que son capaces de almacenar u olvidar el estado anterior y de utilizar o descartar el estado actual. Estas celdas calculan en todo momento dos estados, que son: un estado de celda (c) que podría mantenerse durante muchos pasos temporales y un estado oculto (h), que es la nueva salida de la celda en cada paso de tiempo. Una celda básica se muestra con detalle en la Figura 30 y las ecuaciones que gobiernan a las puertas de control son las dadas por (16), (17), (18), (19), (20) y (21).

$$f_t = \sigma(W_{fh}h_{t-1} + W_{fx}x_t + b_f) \quad (16)$$

$$i_t = \sigma(W_{ih}h_{t-1} + W_{ix}x_t + b_i) \quad (17)$$

$$o_t = \sigma(W_{oh}h_{t-1} + W_{ox}x_t + b_o) \quad (18)$$

$$\tilde{c}_t = \tanh(W_{ch}h_{t-1} + W_{cx}x_t + b_c) \quad (19)$$

$$c_t = f_t \circ c_{t-1} + i_t \circ \tilde{c}_t \quad (20)$$

$$h_t = o_t \circ \tanh(c_t) \quad (21)$$

Donde x_t es la entrada en el tiempo t , c_t y h_t son el estado de la célula y el estado oculto en el tiempo t , respectivamente, W y b son el peso y el sesgo, respectivamente, σ es la función sigmoide, \circ es el operador del producto de Hadamard y \tilde{c}_t es un candidato para actualizar c_t a través de la puerta de entrada.

La puerta de entrada i_t decide la actualización de la célula con un nuevo estado de célula \tilde{c}_t , mientras la puerta de olvido f_t decide qué mantener y qué olvidar del estado anterior de la célula y finalmente la puerta de salida o_t decide cuanta información será pasada a la siguiente célula.

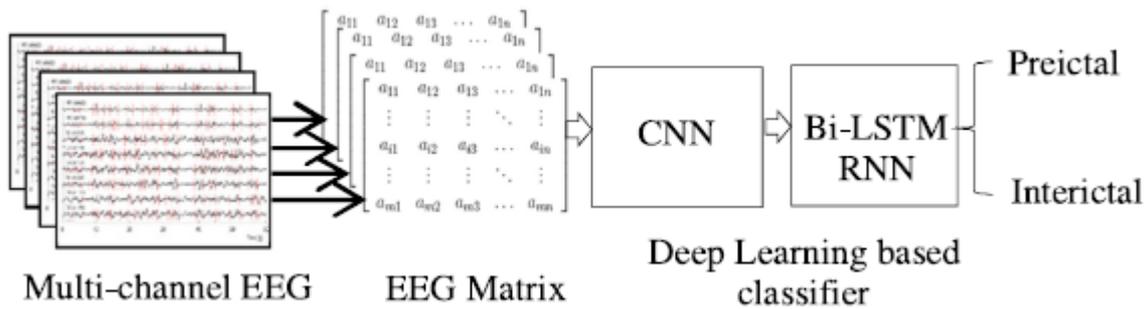


Figura 31 Diagrama de bloques CNN+Bi-LSTM [4]

En este modelo, como se muestra en la Figura 31, se va a utilizar la red CNN para realizar la extracción de características y para la clasificación en lugar de utilizar una red LSTM se va a utilizar la red Bi-LSTM o LSTM bidireccional, en la que cada bloque LSTM es sustituido por dos bloques LSTM que procesan la secuencia temporal al mismo tiempo pero en sentidos opuestos, por lo que, la salida de la red en cada paso de tiempo son las salidas de ambos bloques combinadas en ese paso de tiempo. La utilización de la red Bi-LSTM mejora la precisión ya que con esta se extraen las características temporales representativas, a lo que se suma la extracción de las características espaciales más representativas realizada por la CNN inicial. Un ejemplo de red LSTM bidireccional desenrollada se representa en la Figura 32.

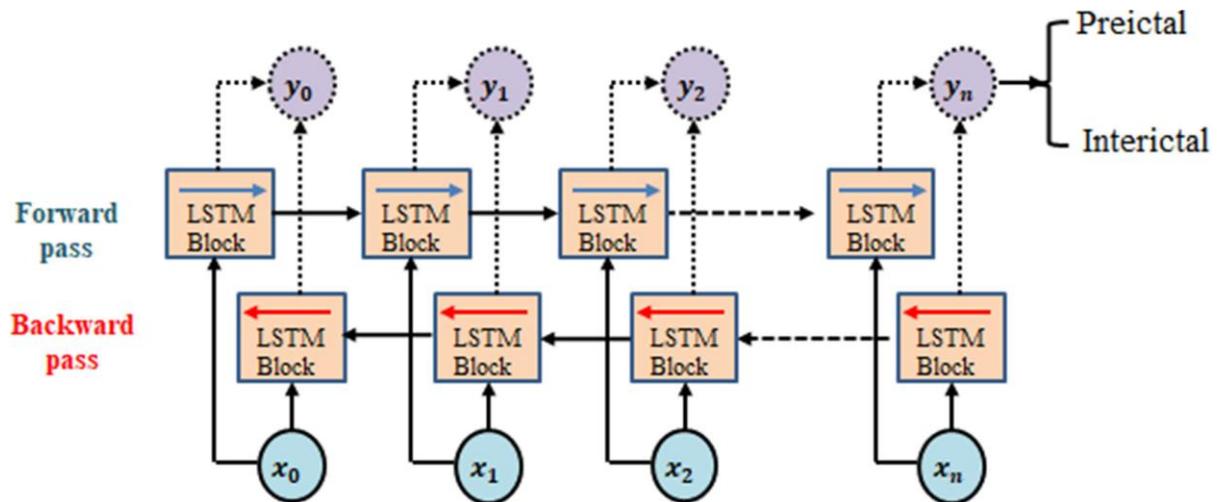


Figura 32 La red LSTM bidireccional desenrollada [4]

La parte de clasificación de la red propuesta consiste en una única capa bidireccional que realiza la tarea de clasificación en la última instancia de tiempo después de procesar todos los segmentos de EEG. Se elige que el número de unidades, es decir, la dimensionalidad del espacio de salida sea 20, aunque se variará este valor para comprobar cómo afecta en la predicción. Se utiliza la técnica de regularización “dropout” para evitar el sobreajuste. El “dropout” es aplicado en la entrada y en el estado recurrente (recurrent state) al 10% y al 50%, respectivamente. Se utiliza la función de activación sigmoide para la predicción y RMSprop para la optimización.

La finalidad de la técnica de regularización “dropout” es reducir el sobreajuste (overfitting). Las neuronas cercanas aprenden patrones que están muy relacionados entre sí y estas relaciones pueden llegar a producir un patrón muy específico con los datos de entrenamiento y, por tanto, sobreajuste. Para evitar esto último, mediante la técnica “dropout” se desactivan diferentes neuronas de forma aleatoria en cada iteración de la red neuronal, con lo que estas no se tienen en cuenta durante todo el proceso en la iteración, consiguiendo así que las neuronas cercanas no dependan tanto de las que se han desactivado y funcionen mejor en solitario, reduciendo así el sobreajuste.

5 RESULTADOS

Este trabajo está basado en el artículo “Efficient Epileptic Seizure Prediction Based on Deep Learning” de Hisham Daoud y Magdy A. Bayoumi [4], del que se pretende replicar algunas de sus técnicas, añadirle variaciones y analizar cómo afectan a los distintos modelos. Las especificaciones indicadas en esta sección se han obtenido de este artículo científico.

El objetivo es detectar lo antes posible y con precisión el estado “preictal” (periodo de tiempo justo antes de la convulsión) diferenciándolo del estado “interictal” (periodo entre convulsiones separado por más de cuatro horas antes y después de cualquier convulsión), y hacerlo de forma que se pueda aplicar en tiempo real. Para ello se introduce al sistema las señales EEG crudas, sin procesar, y, tras esto, se realizará la extracción de características y la clasificación en un único sistema automatizado.

Se van a utilizar varias técnicas, que son: un perceptrón multicapa, una red neuronal convolucional y una red neuronal recurrente. A estas técnicas se les introducirá algunos cambios y se analizarán los resultados obtenidos.

La base de datos de grabaciones EEG de pacientes con epilepsia que se va a utilizar se ha detallado en la sección 1.3 de este trabajo.

Las distintas técnicas se han implementado en Python, haciendo uso de la librería “Keras”. En el apartado de *Anexos* se incluyen todos los códigos.

Todo se ha realizado mediante Google Colaboratory, que es un entorno de Google Research que permite que cualquier persona escriba y ejecute código en Python a través del navegador. Se ha elegido este entorno porque, además de ser muy útil al tener ya la mayoría de las librerías instaladas, permite utilizar las GPUs de Google, con lo que se consigue una gran capacidad de procesamiento.

5.1. Metodología

En este trabajo se utilizan tres modelos de inteligencia artificial y se tiene el objetivo de una predicción lo más precisa y temprana posible de los ataques epilépticos buscando que se pueda realizar en tiempo real.

Como se ha mencionado, las predicciones se acometen como una tarea de clasificación entre estados cerebrales “preictales” e “interictales”, alertando cuando se produce un estado “preictal”. En este trabajo se consideran como “preictales” todas aquellas muestras de las grabaciones de los pacientes con epilepsia que hay hasta una hora antes de que el ataque epiléptico ocurra, y como “interictales”, aquellas que se encuentran más de cuatro horas antes y después de cualquier ataque epiléptico.

Como se ha indicado, se introducirán al sistema señales EEG en crudo, esto es, sin ningún preprocesado previo.

Las características discriminatorias se aprenderán automáticamente mediante técnicas de inteligencia artificial, para así reducir la carga de la tarea de clasificación.

Al disponer de un número de muestras “interictales” muy superior al de muestras “preictales” se realiza una tarea de balanceo, mediante la cual se les pasan a los modelos el mismo número de muestras

“preictales” que “interictales”, ya que, en caso contrario, si se le pasaran muchas más muestras de una categoría, el clasificador sería mucho más preciso con esta categoría que con la de menos muestras.

Las señales EEG se dividen en segmentos de cinco segundos no superpuestos, a los que se considera como “batches” de entrenamiento.

En el primer modelo se realiza un perceptrón multicapa (MLP), esto es, una red neuronal profunda simple, que se entrena para que aprenda los parámetros más discriminativos, para con ellos realizar la tarea de clasificación.

En el segundo, se realiza una red neuronal convolucional (CNN) con la cual se mejora la extracción de características, por lo que, al realizar la clasificación, que se realiza con un MLP, se obtendrán mejores resultados.

En el tercer modelo se vuelve a utilizar la red CNN del segundo modelo para realizar la extracción de características, y esta se concatena a continuación con una red Bidireccional Long Short-Term Memory (Bi-LSTM) con la que se realiza la clasificación. Las redes LSTM son eficaces en el aprendizaje de características temporales ya que mantienen las dependencias existentes en secuencias a largo plazo haciendo uso de su memoria. Las LSTM bidireccionales mejoran aún más la predicción al utilizar la información de las instancias temporales anterior y siguiente.

5.2. Red MLP

En el primer modelo, que se ha detallado en la sección “Modelo propuesto” del apartado 4.1, se obtienen los resultados dados por la Tabla 1. En esta tabla y las siguientes se analizan varios pacientes (pacientes 1, 3, 5, 9, 16, 20, 22 y 23), de los que se obtienen las siguientes características:

— Sensitivity - sensibilidad:

Que mide el porcentaje los verdaderos positivos y se puede calcular como se indica la ecuación (22).

$$Sensitivity = tp / (tp + fn) \quad (22)$$

Siendo tp (true positive) los verdaderos positivos y fn (false negative) los falsos negativos.

— Specificity - especificidad:

Mide el porcentaje de falsos positivos, esto es, de que no se detecten muestras “preictales” y sean clasificadas correctamente. Este parámetro se calcula como indica la ecuación (23).

$$\textit{Specificity} = tn/(tn + fp) \quad (23)$$

Siendo tn (true negative) los verdaderos negativos y fp (false positive) los falsos positivos.

— Accuracy – precisión:

Dada automáticamente por el sistema, aunque también puede ser calculada como se indica la ecuación (24).

$$\textit{Accuracy} = (tp + tn)/(tp + tn + fp + fn) \quad (24)$$

Siendo tn los verdaderos negativos (true negative) y fp los falsos positivos (false positive).

— False alarm h^{-1} :

Calculada como se indica la ecuación (25).

$$h^{-1} = fp/(fp + fn + tn + tp) \quad (25)$$

Esto es, dividiendo el número de falsos positivos (falsas alarmas) entre el número de casos total.

— Tiempo de entrenamiento (min):

Es el tiempo que tarda cada modelo en hacer el entrenamiento y el test. Este tiempo depende de la GPU asignada por Google Colab que varía con el tiempo, asignando unas veces GPUs más potentes y otras veces GPUs menos potentes. Por este motivo, cada técnica se han realizado las pruebas durante una misma sesión para los distintos pacientes utilizados, asegurando así que se están utilizando los mismos recursos y se puedan comparar tiempos entre los distintos pacientes utilizados en cada técnica.

— Número de parámetros:

Indica el número de parámetros que se han utilizado en el modelo para cada paciente.

○ PRIMERA PRUEBA CON LA RED MLP

En primer lugar, se va a probar el modelo MLP con las especificaciones dadas en el subapartado “Modelo propuesto” dentro del apartado 4.1. El esquema del modelo viene representado en la Figura 25, la tabla con los resultados se muestra en la Tabla 1 y sus especificaciones resumidas son:

- Cuatro capas ocultas con un número de unidades de 300, 100, 50 y 20, respectivamente.
- 10 epochs.
- 1280 muestras de batch_size.
- **Todas las muestras se utilizan para entrenamiento y test.**
- RMSProp como optimizador.
- Entropía cruzada binaria como función de costes o pérdidas (Loss).

Tabla 1 Resultados de la primera prueba del modelo MLP (R=Mejor. A=Peor)

PATIENT	SENSITIVITY	SPECIFICITY	ACCURACY	FALSE ALARM h ⁻¹	TRAINING TIME (min)	No. OF PARAMETERS
P 1	0.70195	0.70251	0.70223	0.148	13.28	9756512
P 3	0.91122	0.91071	0.91099	0.044	15.17	9756512
P 5	0.61721	0.61648	0.61681	0.191	12.01	9756512
P 9	0.71108	0.71143	0.71125	0.144	14.72	9756512
P 16	0.53811	0.54071	0.53941	0.22	10.21	9906512
P 20	0.74421	0.74417	0.74419	0.127	11.36	9906512
P 22	0.82145	0.81392	0.81771	0.093	13.51	9906512
P 23	0.92666	0.92692	0.92687	0.036	11.75	9756512
MEAN	0.73543	0.73501	0.73523	0.106	12.65	9812493

Se puede observar que, en general, se obtienen buenos resultados, por encima del 70%, salvo en los pacientes 5 y 16, y que la media de todos también se mantiene por encima del 70%.

Se observa también que, al ser una red MLP y estar, por tanto, sus neuronas completamente conectadas, se tiene un alto número de parámetros, lo cual, hace a esta red ineficiente si se compara con el siguiente modelo que utiliza una red convolucional que reduce ampliamente el número de parámetros utilizados.

También destaca que hay una gran variabilidad entre los distintos pacientes, teniendo algunos que superan incluso el 90% de precisión y otros que ni llegan al 70%. Esto puede ser debido a una gran variedad de motivos, como puede ser, por ejemplo, el hecho de que para algunos pacientes se disponga de un mayor número de muestras “preictales” e “interictales” con las que realizar el entrenamiento que con otros, lo que se debe a su vez a la gran casuística de los pacientes con epilepsia, teniendo algunos

muy pocos ataques epilépticos durante la sesión de grabación, lo que implica pocas muestras “preictales” e “interictales” con las que entrenar, y teniendo otros pacientes muchos ataques epilépticos durante el mismo periodo de tiempo, lo que implica que tienen muchas muestras “preictales” e “interictales” con las que realizar el entrenamiento. Particularmente, los pacientes 1, 3, 9 y 23, tienen muchas más muestras que los pacientes 5, 16, 20 y 22, dándose que para los pacientes 1, 3, 9 y 23 se dispone de más de diez mil muestras tanto “preictales” como “interictales”, mientras que de otros casos como en paciente 16, se disponen menos de siete mil muestras tanto “preictales” como “interictales”, o del paciente 20, por debajo de las ocho mil.

La disposición o no de muestras “preictales” en un paciente depende directamente del número de ataques epilépticos que tenga el paciente durante la sesión de grabación.

Aparentemente, siempre se va a disponer de más muestras “interictales” que “preictales”, sin embargo, se tiene la consideración de que para que una muestra sea “interictal” tiene que estar alejada más de cuatro horas antes y después de cualquier ataque epiléptico y, como las grabaciones de los pacientes están realizadas durante un mismo día en el que el paciente presenta varios ataques epilépticos, el número de muestras “interictales” posibles se reduce considerablemente.

Otro motivo de la gran variabilidad en los resultados entre los diferentes pacientes se debe también a la gran casuística de la epilepsia en los distintos pacientes, ya que la epilepsia afecta a cada paciente de forma distinta, produciendo convulsiones en distintas zonas del cerebro, por lo que, se pueden tener desde pacientes que tengan convulsiones en una zona del cerebro muy reducida y que solo es detectada por pocos sensores, a pacientes a los que dichas convulsiones afectan a zonas más amplias del cerebro o incluso a varias zonas, siendo detectadas en estos casos por un mayor número de sensores. En definitiva, que es posible que en algunos pacientes tan solo unos pocos sensores capten las convulsiones, y que el resto de los sensores no solo no aporte información, sino que, además, hacen que los resultados empeoren.

Debido también a esta gran casuística de la epilepsia, otro posible motivo de los diferentes resultados en los distintos pacientes es la diferente intensidad de las señales y las convulsiones en cada paciente. Si la intensidad de las señales en algún paciente es más pequeña, sus muestras “preictales” pueden ser más difíciles de detectar y diferenciar de las muestras “interictales”.

En lo que se ha comentado en estos últimos párrafos se puede ver la gran complejidad que presenta un trastorno como la epilepsia y justifica los esfuerzos que se realizan por entender más esta enfermedad y por mejorar la vida de los pacientes epilépticos.

○ SEGUNDA PRUEBA CON LA RED MLP

En esta segunda prueba se va a aumentar el número de unidades de las capas ocultas para ver cómo afectan a los resultados. La tabla con los resultados se muestra en la Tabla 2 y las especificaciones resumidas quedarían como sigue:

- Cuatro capas ocultas con un número de unidades de 3000, 1000, 50 y 20, respectivamente.
- 10 epochs.
- 1280 muestras de batch_size.

- **Todas las muestras se utilizan para entrenamiento y test.**
- RMSProp como optimizador.
- Entropía cruzada binaria como función de costes o pérdidas (Loss).

Tabla 2 Resultados de la segunda prueba del modelo MLP (R=Mejor. A=Peor)

PATIENT	SENSITIVITY	SPECIFICITY	ACCURACY	FALSE ALARM h ⁻¹	TRAINING TIME (min)	No. OF PARAMETERS
P 1	0.69228	0.69228	0.69231	0.153	51.08	93775112
P 3	0.91517	0.91536	0.91526	0.042	49.18	93775112
P 5	0.60727	0.60686	0.60709	0.196	56.43	93775112
P 9	0.70218	0.70912	0.70570	0.145	54.72	93775112
P 16	0.55199	0.55520	0.55376	0.222	44.03	93925112
P 20	0.72616	0.72573	0.72601	0.137	48.72	93925112
P 22	0.81714	0.81595	0.81650	0.092	52.45	93925112
P 23	0.92261	0.92269	0.92265	0.038	46.15	93775112
MEAN	0.73111	0.73232	0.73176	0.108	50.19	93831334

Al aumentar el número de unidades de las capas ocultas, se observan unos notables incrementos tanto del tiempo de entrenamiento como del número de parámetros y, sin embargo, los resultados obtenidos son prácticamente iguales al caso anterior, **por lo que, no es eficiente en este caso aumentar el número de unidades de las capas de la red.**

El gran incremento tanto del tiempo de entrenamiento como del número de parámetros se debe a que, como se ha comentado anteriormente, las neuronas de una red perceptrón multicapa están completamente conectadas, lo que hace que, al aumentar el número de unidades, aumenten exponencialmente el número de conexiones de la red, lo que conlleva a un gran aumento de la complejidad, aumentando en gran medida el número de parámetros de la red y el tiempo de ejecución.

○ TERCERA PRUEBA CON LA RED MLP

En esta segunda prueba se van a añadir ciertos cambios para ver cómo afectan a los resultados. **Se le va a añadir un “validation_split” de 0.1, esto es, en el entrenamiento, un 10% de las pruebas se van a utilizar como validación.** Por otra parte, se va a aumentar el número de epochs a 20. Los resultados se indican en la Tabla 3. Las especificaciones resumidas, quedarían como sigue:

- Cuatro capas ocultas con un número de unidades de 300, 100, 50 y 20, respectivamente.
- **Un “validation_split” del 10%.**
- 20 epochs.
- 1280 muestras de batch_size.
- RMSProp como optimizador.
- Entropía cruzada binaria como función de costes o pérdidas (Loss).

Tabla 3 Resultados de la tercera prueba del modelo MLP (R=Mejor. A=Peor)

PATIENT	SENSITIVITY	SPECIFICITY	ACCURACY	FALSE ALARM h ⁻¹	TRAINING TIME (min)	No. OF PARAMETERS
P 1	0.70032	0.69493	0.69767	0.152	35.32	9756512
P 3	0.91033	0.91057	0.91046	0.045	34.85	9756512
P 5	0.57837	0.58145	0.57994	0.216	39.16	9756512
P 9	0.69369	0.69369	0.69369	0.153	38.25	9786512
P 16	0.55099	0.54833	0.54965	0.225	30.54	9906512
P 20	0.72387	0.72452	0.72421	0.142	33.54	9906512
P 22	0.82890	0.82875	0.82881	0.094	36.55	9906512
P 23	0.90117	0.90122	0.90121	0.051	32.06	9756512
MEAN	0.72484	0.72428	0.72458	0.116	34.92	9816260

Añadirle validación en el entrenamiento y aumentar a 20 epochs tampoco hace que mejoren los resultados, obteniéndose valores similares. En este caso, tampoco es efectivo aumentar el número de epochs, puesto que lo único que se consigue con esto es aumentar el tiempo de entrenamiento sin mejorar los resultados.

○ CUARTA PRUEBA CON LA RED MLP

En esta tercera prueba se van a añadir, además de las modificaciones de la segunda prueba (‘validation_split y 20 epochs’), más capas al modelo, teniendo ahora como unidades de las capas ocultas 300, 100, 70, 50, 30 y 20, respectivamente. Los resultados se muestran en la Tabla 4. Las especificaciones resumidas quedarían como sigue:

- Cuatro capas ocultas con un número de unidades de 3000, 1000, 70, 50, 30 y 20, respectivamente.
- Un “validation_split” del 10%.
- 20 epochs.
- 1280 muestras de batch_size.
- RMSProp como optimizador.
- Entropía cruzada binaria como función de costes o pérdidas (Loss).

Tabla 4 Resultados de la cuarta prueba del modelo MLP (R=Mejor. A=Peor)

PATIENT	SENSITIVITY	SPECIFICITY	ACCURACY	FALSE ALARM h ⁻¹	TRAINING TIME (min)	No. OF PARAMETERS
P 1	0.68685	0.68617	0.68651	0.156	54.68	9763212
P 3	0.90893	0.90901	0.90895	0.045	53.82	9763212
P 5	0.58981	0.59006	0.58991	0.204	59.29	9763212
P 9	0.69461	0.69488	0.69475	0.152	58.72	9793212
P 16	0.54929	0.55104	0.55021	0.224	46.83	9913212
P 20	0.70101	0.70135	0.70117	0.149	51.96	9913212
P 22	0.82528	0.82532	0.82529	0.087	56.40	9913212
P 23	0.89517	0.89341	0.89455	0.053	49.52	9913212
MEAN	0.72066	0.72081	0.72076	0.116	53.74	9841699

Nuevamente, en esta última prueba con la red MLP, se vuelven a obtener resultados similares, por lo que, aumentar el número de capas ocultas tampoco funciona en este caso.

Según se ha observado, cambiar los parámetros de esta red MLP no hace que se obtengan mejores resultados, por lo que, los resultados obtenidos en la primera prueba son los máximos valores que se pueden obtener con esta red y, si se quiere mejorarlos, habría que cambiar la red y/o añadirle capas más complejas.

En el siguiente modelo, se van a añadir capas convolucionales, más complejas que las de MLP y permitirán obtener mejores resultados.

5.3. Red CNN+MLP

Las tablas incluidas en este apartado constan de los mismos campos que las tablas del apartado anterior, esto permitirá observar de forma clara las diferencias entre los distintos resultados obtenidos por los modelos.

○ PRIMERA PRUEBA CON LA RED CNN+MLP

En primer lugar, se va a probar el modelo CNN+MLP con las especificaciones dadas en el subapartado “Modelo propuesto” dentro del apartado 4.2. La red CNN viene representada en la Figura 27, el esquema del modelo completo en la Figura 28 y los resultados de esta prueba se indican en la Tabla 5. Las especificaciones resumidas son:

- Se le pasan matrices de 1280x23 de las señales EEG recortadas.
- La CNN intercala capas convolucionales con capas de agrupación (pooling), tal y como se muestra en la Figura 27. Se utilizan cuatro capas convolucionales, de 32 unidades y de 3x2 para hacerla adecuada para matrices rectangulares, y tres de pooling de 2x2.
- Una capa para aplicar la técnica de normalización de Batch, explicada en el apartado 4.2.
- Una red MLP cuyas cuatro capas ocultas tienen 300, 100, 50 y 20 unidades, respectivamente.
- 50 epochs.
- 32 muestras de batch_size.
- **Todas las muestras se utilizan para entrenamiento y test.**
- RMSProp como optimizador.
- Entropía cruzada binaria como función de costes o pérdidas (Loss).

Tabla 5 Resultados de la primera prueba del modelo CNN+MLP (R=Mejor. A=Peor)

PATIENT	SENSITIVITY	SPECIFICITY	ACCURACY	FALSE ALARM h ⁻¹	TRAINING TIME (min)	No. OF PARAMETERS
P 1	0.99908	0.99908	0.99908	0.0004	3.45	1552992
P 3	0.97161	0.97508	0.97508	0.012	3.13	1552992
P 5	0.85333	0.86511	0.86111	0.067	3.87	1552992
P 9	0.73317	0.73708	0.73474	0.131	3.74	1552992
P 16	0.73608	0.71958	0.72577	0.141	3.54	1552992
P 20	0.98437	0.98437	0.98437	0.007	3.92	1552992
P 22	0.91281	0.90966	0.91071	0.045	4.13	1552992
P 23	0.99420	0.99516	0.99420	0.0024	2.93	1552992
MEAN	0.89159	0.89131	0.891416	0.016	3.56	1552992

Para este primer caso de la red CNN se obtienen muy buenos resultados, la mayoría por encima del 90%, obteniendo una media del 89%.

Se puede observar que el número de parámetros es mucho inferior al que se tenía con la red MLP, lo que se debe a que la red CNN hace agrupación y consigue reducir el número de características.

Otro aspecto importante es el tiempo de entrenamiento, el cual es mucho inferior al de la red MLP. Esto se debe también al uso de GPU, muy adecuada a la hora de entrenar con redes convolucionales.

○ SEGUNDA PRUEBA CON LA RED CNN+MLP

Se va a probar el modelo CNN+MLP con las especificaciones utilizadas en la primera prueba, pero añadiéndole más capas de normalización de batch y entrenando con 200 epochs. Los resultados vienen indicados en la Tabla 6. Las especificaciones resumidas son:

- Se le pasan matrices de 1280x23 de las señales EEG recortadas.
- La CNN intercala capas convolucionales con capas de agrupación (pooling), tal y como se muestra en la Figura 27. Se utilizan cuatro capas convolucionales, de 32 unidades y de 3x2 para hacerla adecuada para matrices rectangulares, y tres de pooling de 2x2.
- Varias capas intercaladas para aplicar la técnica de normalización de Batch, explicada en el apartado 4.2.
- Una red MLP cuyas cuatro capas ocultas tienen 300, 100, 50 y 20 unidades, respectivamente.
- 200 epochs.

- 32 muestras de batch_size.
- **Todas las muestras se utilizan para entrenamiento y test.**
- RMSProp como optimizador.
- Entropía cruzada binaria como función de costes o pérdidas (Loss).

Tabla 6 Resultados de la segunda prueba del modelo CNN+MLP (R=Mejor. A=Peor)

PATIENT	SENSITIVITY	SPECIFICITY	ACCURACY	FALSE ALARM h ⁻¹	TRAINING TIME (min)	No. OF PARAMETERS
P 1	0.99817	0.99817	0.99817	0.0009	5.61	1552992
P 3	0.90793	0.90801	0.90796	0.045	5.36	1552992
P 5	0.82166	0.82666	0.82666	0.086	6.01	1552992
P 9	0.74569	0.74961	0.74726	0.125	5.61	1552992
P 16	0.65154	0.65361	0.65154	0.173	5.21	1552992
P 20	0.98437	0.98437	0.98437	0.007	5.73	1552992
P 22	0.97373	0.97268	0.97268	0.013	6.11	1552992
P 23	0.99516	0.99516	0.99516	0.002	4.23	1552992
MEAN	0.87547	0.87695	0.87625	0.018	5.57	1552992

En este segundo caso, los resultados son algo inferiores pero similares, bajando la media del 89 al 87%. Por tanto, aumentar el número de capas de la técnica de normalización de batch y cambiar el número de epochs a 200 no es muy efectivo en este caso. El sistema es ahora menos eficiente al tardar casi el doble de tiempo en ejecutarse mientras que se obtienen los mismos resultados.

○ TERCERA PRUEBA CON LA RED CNN+MLP

Se va a probar el modelo CNN+MLP con las especificaciones utilizadas en la segunda prueba, pero añadiéndole más capas ocultas a la red MLP, teniendo 300, 100, 70, 50, 30 y 20 unidades, respectivamente, y añadiendo un validation_split del 10%. Los resultados vienen indicados en la Tabla 7 y las especificaciones resumidas son:

- Se le pasan matrices de 1280x23 de las señales EEG recortadas.
- La CNN intercala capas convolucionales con capas de agrupación (pooling), tal y como se muestra en la Figura 27. Se utilizan cuatro capas convolucionales, de 32 unidades y de 3x2 para hacerla adecuada para matrices rectangulares, y tres capas de agrupación o pooling de 2x2.
- Varias capas intercaladas para aplicar la técnica de normalización de Batch, explicada en el apartado 4.2.
- Una red MLP cuyas cuatro capas ocultas tienen 300, 100, 70, 50, 30 y 20 unidades, respectivamente.
- 200 epochs.
- 32 muestras de batch_size.
- **Un “validation_split” del 10%.**
- RMSProp como optimizador.
- Entropía cruzada binaria como función de costes o pérdidas (Loss).

Tabla 7 Resultados de la tercera prueba del modelo CNN+MLP (R=Mejor. A=Peor)

PATIENT	SENSITIVITY	SPECIFICITY	ACCURACY	FALSE ALARM h ⁻¹	TRAINING TIME (min)	No. OF PARAMETERS
P 1	0.99908	0.99908	0.99908	0.0004	7.51	1560076
P 3	0.98551	0.98551	0.98551	0.007	6.86	1560076
P 5	0.87333	0.87333	0.87333	0.063	8.57	1560076
P 9	0.74256	0.74178	0.74256	0.129	8.09	1560076
P 16	0.74432	0.74432	0.74432	0.127	7.64	1560076
P 20	0.96701	0.96701	0.96701	0.016	8.51	1560076
P 22	0.95378	0.95378	0.95378	0.023	8.91	1560076
P 23	0.99421	0.99421	0.99421	0.002	6.41	1560076
MEAN	0.90138	0.90126	0.90138	0.014	7.76	1560076

En este último caso del modelo CNN+MLP, sí que están mejorando algo los resultados alcanzando el 90% en media, sin embargo, esto se consigue a fuerza de aumentar el tiempo de ejecución de los entrenamientos a más del doble de tiempo, por lo que, habría que analizar hasta qué punto interesa mejorar los resultados. Teniéndose como objetivo en este trabajo que el sistema sea aplicable a tiempo real pienso que la mejor opción es la primera prueba que se ha realizado con este modelo, ya que los resultados son muy buenos y el tiempo de entrenamiento es el menor de los tres casos probados.

5.4. Red CNN+Bi-LSTM

Al igual que en los modelos anteriores, las tablas en este caso tienen los mismos campos.

○ PRIMERA PRUEBA CON LA RED CNN+BI_LSTM

En primer lugar, se va a probar el modelo CNN+Bi+LSTM con las especificaciones dadas en el subapartado “Modelo propuesto” dentro del apartado 4.3. La red CNN viene representada en la Figura 27 y el esquema del modelo completo en la Figura 31. Los resultados se indican en la Tabla 8 y las especificaciones resumidas son:

- Se le pasan matrices de 1280x23 de las señales EEG recortadas.
- La CNN intercala capas convolucionales con capas de agrupación (pooling), tal y como se muestra en la Figura 27. Se utilizan tres capas convolucionales, de 32 unidades y de 3x2 para hacerla adecuada para matrices rectangulares, y dos capas de agrupación o pooling de 2x2.
- Varias capas intercaladas para aplicar la técnica de normalización de Batch, explicada en el apartado 4.2.
- Una capa LSTM bidireccional de 20 unidades.
- 50 epochs.
- 128 muestras de batch_size.
- **Todas las muestras se utilizan para entrenamiento y test.**
- RMSProp como optimizador.
- Entropía cruzada binaria como función de costes o pérdidas (Loss).

Tabla 8 Resultados de la primera prueba del modelo CNN+Bi-LSTM (R=Mejor. A=Peor)

PATIENT	SENSITIVITY	SPECIFICITY	ACCURACY	FALSE ALARM h ⁻¹	TRAINING TIME (min)	No. OF PARAMETERS
P 1	0.99908	0.99908	0.99908	0.0004	1.82	64002
P 3	0.97972	0.98030	0.97972	0.0098	1.52	40098
P 5	0.83166	0.83333	0.83333	0.083	1.88	40098
P 9	0.78482	0.78403	0.78403	0.107	1.77	40098
P 16	0.77938	0.77525	0.77525	0.112	1.71	40098
P 20	0.98611	0.98611	0.98611	0.006	1.89	40098
P 22	0.95063	0.94852	0.95063	0.025	1.98	40098
P 23	0.99710	0.99710	0.99710	0.0014	1.51	40098
MEDIA	0.90882	0.90815	0.90833	0.013	1.75	42511

Con este modelo se han obtenido buenos resultados, mejores incluso que los obtenidos en el modelo anterior.

Mejora en todos los aspectos comparándolo con el modelo CNN+MLP. En primer lugar, se observa que se están obteniendo precisiones algo mayores. En segundo lugar, estas mejores precisiones se consiguen con un tiempo de entrenamiento y test mucho menor, incluso menos de la mitad que en el modelo anterior en algunos casos, y esto, a su vez, implica que este modelo sea mucho más adecuado para aplicarlo en tiempo real. Por último, y relacionado con lo anterior, el número de parámetros utilizados es muy inferior que en los modelos anteriores.

Este sería un buen modelo para aplicarlo en tiempo real, no obstante, se van a realizar algunas modificaciones para analizar si estos resultados pueden mejorarse.

○ SEGUNDA PRUEBA CON LA RED CNN+BI_LSTM

En esta prueba se le va a aumentar el número de unidades de la capa LSTM a 200 unidades. Los resultados se indican en la Tabla 9 y las especificaciones resumidas son:

- Se le pasan matrices de 1280x23 de las señales EEG recortadas.
- La CNN intercala capas convolucionales con capas de agrupación (pooling), tal y como se muestra en la Figura 27. Se utilizan cuatro capas convolucionales, de 32 unidades y de 3x2 para hacerla adecuada para matrices rectangulares, y tres capas de agrupación o pooling de 2x2.
- Varias capas intercaladas para aplicar la técnica de normalización de Batch.
- Una capa LSTM bidireccional de 200 unidades.
- 50 epochs.

- 128 muestras de batch_size.
- **Todas las muestras se utilizan para entrenamiento y test.**
- RMSProp como optimizador.
- Entropía cruzada binaria como función de costes o pérdidas (Loss).

Tabla 9 Resultados de la segunda prueba del modelo CNN+Bi-LSTM (R=Mejor. A=Peor)

PATIENT	SENSITIVITY	SPECIFICITY	ACCURACY	FALSE ALARM h ⁻¹	TRAINING TIME (min)	No. OF PARAMETERS
P 1	0.99908	0.99817	0.99908	0.0009	2.75	840162
P 3	0.98493	0.98551	0.98551	0.007	1.86	516738
P 5	0.84333	0.845	0.84333	0.0775	2.29	516738
P 9	0.77699	0.77699	0.77699	0.111	2.16	516738
P 16	0.77113	0.77113	0.77113	0.114	2.02	516738
P 20	0.98958	0.98958	0.98958	0.005	2.22	516738
P 22	0.95693	0.95693	0.95588	0.02	2.44	516738
P 23	0.98937	0.98937	0.98937	0.005	1.82	516738
MEDIA	0.90892	0.90911	0.90887	0.015	2.17	549107

En este segundo caso, se observa que los cambios han hecho que se obtengan precisiones un poco mejores, sin embargo, también han hecho que tanto el número de parámetros como el tiempo de ejecución aumenten. Como se ha mencionado, se tiene como objetivo la aplicación en tiempo real, por lo que, aunque hayan aumentado algo las precisiones, el cambio realizado no conviene al aumentar el tiempo de entrenamiento y test.

○ TERCERA PRUEBA CON LA RED CNN+BI_LSTM

Por último, se va a realizar la primera prueba, pero **añadiéndole un “validation_split” del 10%**. Los resultados se presentan en la Tabla 10 y las especificaciones resumidas son:

- Se le pasan matrices de 1280x23 de las señales EEG recortadas.

- La CNN intercala capas convolucionales con capas de agrupación (pooling), tal y como se muestra en la Figura 27. Se utilizan cuatro capas convolucionales, de 32 unidades y de 3x2 para hacerla adecuada para matrices rectangulares, y tres capas de agrupación o pooling de 2x2.
- Varias capas intercaladas para aplicar la técnica de normalización de Batch.
- Una capa LSTM bidireccional de 20 unidades.
- 50 epochs.
- 128 muestras de batch_size.
- **Un ‘validation_split’ del 10%.**
- RMSProp como optimizador.
- Entropía cruzada binaria como función de costes o pérdidas (Loss).

Tabla 10 Resultados de la tercera prueba del modelo CNN+Bi-LSTM (R=Mejor. A=Peor)

PATIENT	SENSITIVITY	SPECIFICITY	ACCURACY	FALSE ALARM h ⁻¹	TRAINING TIME (min)	No. OF PARAMETERS
P 1	0.99908	0.99908	0.99908	0.0004	1.75	64002
P 3	0.97856	0.97856	0.97798	0.0107	1.46	40098
P 5	0.85833	0.86166	0.86001	0.0691	1.87	40098
P 9	0.72535	0.72378	0.72456	0.138	1.79	40098
P 16	0.72164	0.72371	0.72371	0.138	1.71	40098
P 20	0.93576	0.93402	0.93402	0.032	1.88	40098
P 22	0.92226	0.92016	0.92226	0.039	2.05	40098
P 23	0.99710	0.99516	0.99613	0.0024	1.51	40098
MEDIA	0.88548	0.88532	0.88552	0.0189	1.74	42511

Como se observa en la tabla anterior, los resultados obtenidos son algo peores a los obtenidos con la primera prueba de este modelo.

5.5. Selección de canal (Channel select)

Por último, se va a implementar un algoritmo de selección de canal, mediante el cual se va a buscar igualar o incluso mejorar los resultados que se obtienen utilizando todos los sensores.

El esquema de este algoritmo se representa en la Figura 33 y, como se puede observar, es iterativo. En cada una de las iteraciones el algoritmo va añadiendo un sensor más al grupo de sensores utilizados. El algoritmo seguirá iterando hasta que el resultado obtenido iguale o mejore al resultado que se obtiene utilizando todos los sensores.

Como resultado se tendrá la precisión obtenida, la cual será igual o mayor que la obtenida sin seleccionar canales, y el número de canales seleccionados con el cual se ha obtenido esta precisión.

Este algoritmo resulta interesante ya que, debido a la naturaleza de la epilepsia que dependiendo del paciente puede afectar a una zona del cerebro u otra, se verá como utilizando un número de canales inferior se pueden obtener mejores resultados. Esto se debe a que, al afectar la epilepsia normalmente solo a alguna zona concreta del cerebro del paciente, muchos de los canales no aportan ninguna información, llegando incluso a hacer que los resultados empeoren.

Algorithm 1: EEG Channel Selection Algorithm.

Input: Eight patients EEG preictal segments, the seizure prediction accuracy for each patient using all channels $Acc[1 : 8]$

Output: $Ch_{red} [1 : 8]$ (array of reduced channels for each patient that give the same accuracy Acc)

Initialization: $m = 8$ (initial number of channels), $done = 0$

```

for patient  $\leftarrow$  1 to 8 do
   $m = 8, done = 0;$ 
  for  $ch \leftarrow$  1 to 23 do
    compute variance[ $ch$ ];
    compute entropy[ $ch$ ];
    compute variance[ $ch$ ]*entropy[ $ch$ ];
  end
  sort the 23 channels with highest variance entropy product
  first in  $Temp$ ;
  while  $done \neq 1$  do
    select first  $m$  channels from  $Temp$ ;
    train and test the model with  $m$  channels;
    compute the prediction accuracy  $Acc_{new}$ ;
    if  $Acc_{new} \geq Acc[patient]$  then
       $done \leftarrow 1;$ 
       $Ch_{red} [patient] \leftarrow Temp[1: m];$ 
    else
       $m \leftarrow m+1$ 
    end
  end
end

```

Figura 33 Algoritmo de selección de canal (Channel select) [4]

Este algoritmo de selección se va a aplicar con el modelo CNN+Bi-LSTM, cuyos resultados se han indicado en el apartado anterior.

Como se puede observar en la Figura 33, en este algoritmo es necesario calcular la entropía y esta se calcula mediante la ecuación (26).

$$H = - \sum_{i=1}^k p_i \ln p_i \quad (26)$$

Dónde H es la entropía, p_i es la probabilidad de la clase i en el canal y \ln es el logaritmo neperiano. En este caso se está utilizando un logaritmo neperiano, por lo que la entropía se mide en nats, sin embargo, podría utilizarse igualmente un logaritmo en base dos, con lo que la entropía se mediría en bits y como esta se va a utilizar para comparar los resultados serían los mismos.

Calcular la entropía es necesario en el algoritmo para medir la cantidad de información que aporta cada canal y poder así ordenar dichos canales de mayor a menor entropía. Esto permite utilizar, en primer lugar, tan solo los canales que más información aportan e ir añadiendo un canal más en cada iteración del algoritmo. Por tanto, el algoritmo, en cada iteración, lo que hace es realizar el entrenamiento y el test del modelo con los canales que más información aportan y compara el resultado con el obtenido aplicando el modelo con todos los canales. El algoritmo continúa iterando y añadiendo canales hasta que se obtenga un resultado igual o mejor que el obtenido con todos los canales o hasta que se añadan todos los canales, resultando que no se ha podido conseguir obtener mejor o igual resultado con menos canales. En el caso en el que en alguna de las iteraciones se obtenga mejor o igual resultado, se tiene que se pueden eliminar el resto de los canales sobrantes, con lo que se consigue que el sistema tenga una mayor eficiencia, ya que se están quitando canales que no aportan apenas información o, incluso, que hacen que los resultados empeoren, reduciéndose la carga computacional.

Conviene recordar que este proceso es específico para cada paciente, por lo que, los canales que más información aportan para un paciente, en otro, podrían ser incluso los que menos información aporten, lo que obliga a realizar todo el proceso para cada uno de los pacientes.

La entropía se calcula calculando el número de clases, la probabilidad de cada clase y restando la probabilidad de cada clase por el logaritmo neperiano de esta probabilidad.

Los resultados obtenidos al aplicar este algoritmo en cada paciente son:

— Paciente 1:

Precisión con todos los canales: 0.99725

Precisión resultante: 0.99817. Número de canales: 10

El número de canales utilizado para obtener la misma o más precisión es de 10. Por lo que se dejan de utilizar 13 canales, lo que implica un gran ahorro computacional.

— Paciente 3:

Precisión con todos los canales: 0.9275782108306885

Precisión: 0.9663962721824646. Número de canales: 9

El número de canales utilizado para obtener la misma o más precisión es de 8. Dejando de utilizar 15 canales, lo que implica un gran ahorro computacional.

— Paciente 5:

Precisión con todos los canales: 0.8299999833106995

Precisión: 0.8299999833106995. Número de canales: 8

El número de canales utilizado para obtener la misma o más precisión es de 8. Dejando de utilizar 15 canales, lo que implica un gran ahorro computacional

— Paciente 9:

Precisión con todos los canales: 0.6071987748146057

Precisión: 0.7190923094749451. Número de canales: 8

El número de canales utilizado para obtener la misma o más precisión es de 8. Dejando de utilizar 15 canales, lo que implica un gran ahorro computacional

— Paciente 16:

Precisión con todos los canales: 0.6309278607368469

Precisión: 0.6721649765968323. Número de canales: 12

El número de canales utilizado para obtener la misma o más precisión es de 12. Dejando de utilizar 11 canales, lo que implica un gran ahorro computacional

— Paciente 20:

Precisión con todos los canales: 0.8888888955116272

Precisión: 0.96875. Número de canales: 10

El número de canales utilizado para obtener la misma o más precisión es de 10. Dejando de utilizar 13 canales, lo que implica un gran ahorro computacional

— Paciente 22:

Precisión con todos los canales: 0.9327731132507324

Precisión: 0.9337227005958557. Número de canales: 17

El número de canales utilizado para obtener la misma o más precisión es de 17.

Dejando de utilizar 6 canales, lo que implica un gran ahorro computacional

— Paciente 23:

Precisión con todos los canales: 0.9835748672485352

Precisión: 0.9942029118537903. Número de canales: 11

El número de canales utilizado para obtener la misma o más precisión es de 11.

Dejando de utilizar 12 canales, lo que implica un gran ahorro computacional

6 CONCLUSIONES

Después de realizar este estudio, podemos concluir que se han obtenido buenos resultados de los que se pueden extraer varios aspectos interesantes.

En la Tabla 11 se han incluido la media de los resultados obtenidos en las primeras pruebas de las tres técnicas utilizados en este trabajo.

Como puede observarse, de los tres modelos, el que mejor resultados ha obtenido es el modelo CNN+Bi-LSTM, con el que se ha obtenido una probabilidad media del 90%, además de utilizar un número bajo de parámetros y con tiempos de entrenamiento y test también bajos, entre uno y dos minutos.

Con el modelo MLP, que es el más básico, se han obtenido resultados por encima del 70%, que no están nada mal, pero, sin embargo, los altos tiempos de ejecución hacen que no sea una buena opción.

Por último, el segundo de los tres modelos, el de CNN+MLP, es preferible frente al básico MLP, ya que se obtienen resultados similares al tercer modelo, aunque sigue siendo mejor el tercero, ya que tiene tiempos de entrenamiento y test inferiores.

Por otra parte, como se ha mencionado anteriormente, llama la atención la gran variación de resultados en un mismo modelo de unos pacientes frente a otros, ya que hay pacientes cuyos resultados están por encima del 90% y otros incluso por debajo del 70%. Se puede concluir que esto se debe a la gran casuística de la que es característica la epilepsia. Este trastorno afecta a cada paciente de una forma diferente y muy particular, por lo que, para conseguir resultados óptimos, con los que mejorar la calidad de vida de los pacientes, se tienen que implementar soluciones que son específicas para cada paciente.

Esta gran casuística de la epilepsia es producida por el amplio abanico de posibilidades a las que se pueden enfrentar los pacientes con este trastorno. Algunas de las características en las que este trastorno puede ser diferente de un paciente a otro son, por ejemplo, la localización de las convulsiones en el cerebro, el número de convulsiones, la intensidad y duración de estas, etc. Sería interesante, para un estudio posterior, a partir de este trabajo, analizar qué diferencias hay entre las señales de un paciente en el cual se están obteniendo probabilidades superiores al 90% y las señales de un paciente en el que se obtienen probabilidades del 70%. Esto último permitiría, si se consiguiesen detectar las diferencias entre las señales, saber cómo modificar las técnicas y conseguir así mejores resultados en los pacientes en los que se obtienen probabilidades más bajas.

Tabla 11 Resultados medios obtenidos en la primera prueba de las tres técnicas

PATIENT	SENSITIVITY	SPECIFICITY	ACCURACY	FALSE ALARM h ⁻¹	TRAINING TIME (min)	No. OF PARAMETERS
MLP	0.73543	0.73501	0.73523	0.106	12.65	9812493
CNN+MLP	0.89159	0.89131	0.891416	0.016	3.56	1552992
CNN+Bi-LSTM	0.90882	0.90815	0.90833	0.013	1.75	42511

Tras los tres modelos, se ha realizado una última prueba en la que se ha utilizado el algoritmo de selección de canal, algoritmo que aporta resultados muy interesantes, ya que permite que se puedan obtener iguales o mejores resultados utilizando muchos menos canales, lo que es interesante, puesto que con esto se consigue un gran ahorro computacional que al final se traduce en un menor tiempo de ejecución, algo muy conveniente para la aplicación del sistema a tiempo real.

Una vez aplicado este algoritmo de selección de canal, puede observarse que este es muy útil, ya que, en todos los casos analizados, se consiguen mejores resultados utilizando menos canales, lo que resulta en una mayor eficiencia al reducir en gran medida la carga computacional. Además, permite saber, para cada paciente, qué canales son los que más información aportan, permitiendo que análisis posteriores se puedan realizar de forma más eficiente, limitándose tan solo a estos canales discriminativos específicos para cada caso.

REFERENCIAS

- [1] F. S. Vidoy, «Detección de crisis epilépticas a partir del electroencefalograma,» Dep. Teoría de la Señal y Comunicaciones de la Etsi (US), Sevilla, 2017.
- [2] Á. (. A. d. Epilepsia). [En línea]. Available: <http://www.apicepilepsia.org/Descripcion-de-los-diferentes-tipos-de-crisis-epilepticas>.
- [3] NINDS, «NINDS (National Institute of Neurological Disorders and Strokes),» [En línea]. Available: https://espanol.ninds.nih.gov/trastornos/crisis_epilepticas.htm.
- [4] H. D. y. M. A. Bayoumi, «Efficient Epileptic Seizure Prediction Based on Deep Learning,» 2019.
- [5] S. Giménez, «medicina21,» Farmavet, 2018. [En línea]. Available: <https://www.medicina21.com/articulos/ver/1178>.
- [6] M. Gratacós, «Tronco del encéfalo: funciones, partes y anatomía,» [En línea]. Available: <https://www.lifeder.com/tronco-del-encefalo/>.
- [7] R. B. Navarro, «Tema 5. Electroencefalografía,» *Instrumentación Biomédica. Departamento Electrónica. Universidad Alcalá*.
- [8] A. L., Goldberger; y otros, «PhysioBank, PhysioToolkit, and PhysioNet}: Components of a New Research Resource for Complex Physiologic Signals,» *Circulation*, vol. 101, n° 23, pp. e215--e220, 2000 (June 13).
- [9] J. Pastor, «Qué es la inteligencia artificial,» 2018. [En línea]. Available: <https://www.xataka.com/robotica-e-ia/que-inteligencia-artificial>.
- [10] R. L. d. M. Badía y P. M. González, *Inteligencia artificial (¿Qué sabemos de?)*, ES: La catarata, 2017.
- [11] Domotizados.co, «Los asistentes virtuales mas usados en los hogares,» Domotizados, Febrero 2018. [En línea]. Available: <https://domotizados.co/asistentes-digitales-tomando-el-control/>.
- [12] C. Gillingham, «A Venn diagram illustrating one of the weaknesses of the Turing test.,» 2007.
- [13] Iat.es, «INTELIGENCIA ARTIFICIAL: QUÉ ES, TIPOS, TÉCNICAS, VENTAJAS,» IAT, Febrero 2020. [En línea]. Available: <https://iat.es/tecnologias/inteligencia-artificial/>.
- [14] «Inteligenciayartificial,» 2019. [En línea]. Available: <https://inteligenciayartificial.com/>.
- [15] P. R. de los Santos, «Tipos de aprendizaje en Machine Learning: supervisado y no supervisado,» Telefónica, Noviembre 2017. [En línea]. Available: <https://empresas.blogthinkbig.com/que->

algoritmo-elegir-en-ml-aprendizaje/.

- [16] Salesforce, «Machine Learning y Deep Learning: aprende las diferencias,» Julio 2018. [En línea]. Available: <https://www.salesforce.com/mx/blog/2018/7/Machine-Learning-y-Deep-Learning-aprende-las-diferencias.html>.
- [17] Decidesoluciones, «Procesamiento del Lenguaje Natural (PLN o NLP): qué es y para qué se utiliza,» 2019. [En línea]. Available: <https://decidesoluciones.es/procesamiento-del-lenguaje-natural-pln-o-nlp-que-es-y-para-que-se-utiliza/>.
- [18] A. H. Fath, F. Madanifar y M. Abbasi, «Implementation of multilayer perceptron (MLP) and radial basis function (RBF) neural networks to predict solution gas-oil ratio of crude oil systems,» de *Petroleum*, 2020, pp. Volume 6, Issue 1, 80-91.
- [19] V. Mirjalili y S. Raschka, Python machine learning, Barcelona: Marcombo, 2019.
- [20] J. D. G. Villanueva, «Redes neuronales desde cero,» 2019. [En línea]. Available: <https://www.iartificial.net/redes-neuronales-desde-cero-i-introduccion/>.
- [21] K. He, X. Zhang, S. Ren y J. Sun, «Delving deep into rectifiers: Surpassing human.level performance on imagenet classification,» de *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026-1034.
- [22] Y. Bengio y X. Glorot, «Understanding the difficulty of training deep feedforward neural networks,» de *Proceedings of the Thirteen International Conference on Artificial Intelligence and Statistics*, 2010, pp. 249-256.
- [23] D. Mishkin y J. Matas, «All you need is a good init,» de *Tetrahedron*, 2015, pp. 3013-3018.
- [24] J. M. V. Arias, «Modelo de aprendizaje profundo/red neuronal convolucional (CNN) para clasificación de calidad de ácidos grasos por imágenes de semillas de *Helianthus annuus*,» Universitat Oberta de Catalunya, 2019.
- [25] J. Torres, «Deep learning: Introducción práctica con Keras,» 2019. [En línea]. Available: <https://torres.ai/deep-learning-inteligencia-artificial-keras/>.
- [26] F. Huang y Y. LeCun, «Large-scale Learning with SVM and Convolutional for Generic Object Categorization,» de *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2006, pp. vol. 1, pp. 284-291.
- [27] S. Dupond, «A through review on the current advance of neural network structures,» de *Annual Reviews in Control*, 2019, pp. 14: 200-230.
- [28] J. Torres, «Redes Neuronales Recurrentes,» Septiembre 2019. [En línea]. Available: <https://torres.ai/redes-neuronales-recurrentes/>.
- [29] Y. Bengio y Y. Lecun, «Convolutional Networks for Images, Speech, and Time-Series,» 1997.

GLOSARIO

ECoG: Electrocorticograma	17
E-EEG: Estéreo-Electroencefalograma	17
EEG: Electroencefalograma	1, 10
ERFC: Exponencial Radial Base Function	31
ICA: Independent Component Analysis	22
KKT: Karush-Kuhn-Tucker	28
NINDS: National Institute of Neurological Disorders and Stroke	7
PhysioBank ATM: PhysioBank's Automated Teller Machine	22
RBF: Radial Base Function	31
SVM: Máquina de Soporte Vectorial	1
TAC: Tomografía Axial Computarizada	6
UCI: Unidad de Cuidados Intensivos	16

ANEXO. FUNCIONES DE PYTHON UTILIZADAS

En esta sección se añaden los códigos de todos los modelos utilizados en este trabajo. Todos han sido realizados en Python y, entre otras, con la librería Keras, utilizando Google Colaboratory como entorno de desarrollo.

➤ Código modelo MLP

```
import keras
from keras.models import Sequential
from keras.layers import Dense
import numpy as np
import lee_grabaciones_mlp # Lee_grabaciones_mlp es el fichero .py que contiene
una función por cada paciente y se encarga de leer de la base de datos las señales EEG
y seleccionar las muestras "preictales" e "interictales" de cada grabación.
from time import time

# CAMBIAR NÚMERO DE PACIENTE SEGÚN INTERESE. SE TIENEN LOS PACIENTES
1,3,5,9,16,20,22,23
train_signal, test_signal, train_labels, test_labels =
lee_grabaciones_mlp.lee_paciente_1() # De la función leer_paciente_*() se obtienen
directamente las señales de train, test y los vectores de etiquetas

network = Sequential()

network.add(Dense(30000, activation='relu'))
network.add(Dense(300, activation='relu'))
network.add(Dense(100, activation='relu'))
network.add(Dense(50, activation='relu'))
network.add(Dense(20, activation='relu'))
network.add(Dense(2, activation='sigmoid'))

network.compile(optimizer='rmsprop',
                loss='binary_crossentropy',

metrics=['accuracy',keras.metrics.SpecificityAtSensitivity(0.5),keras.metrics.Sensiti
vityAtSpecificity(0.5),keras.metrics.FalsePositives(),keras.metrics.TruePositives(),ke
ras.metrics.FalseNegatives(),keras.metrics.TrueNegatives()])

start = time() # Para calcular el tiempo de entrenamiento y test. Empieza a contar
desde aquí
network.fit(train_signal, train_labels, epochs=10, batch_size=1280) # Entrenamiento.
1280 muestras equivalen a 5 segundos de grabación

test_loss, test_acc, specificity, sensitivity, fp, tp, fn, tn =network.evaluate(test_signal,
test_labels) # Test

print("sensitivity calculada: ",tp/(tp+fn))# Imprime el valor de la sensibilidad
print("specificity calculada: ",tn/(tn+fp))# Imprime el valor de la especificidad
print("Acuraccy: ",test_acc)# Imprime el valor de la precisión
print("fp: ",fp) # Imprime el valor de los falsos positivos
```

```
print("False alarm h^-1:", fp/(fp+tp+tn+fn)) #Imprime el valor de las falsas alarmas
print("Train time: ", (time()-start)/60)# Imprime el valor del tiempo que ha tardado
en ejecutarse el entrenamiento y el test
print("No de parámetros: ", network.count_params())# imprime el número de parámetros
de la red
```

➤ Código modelo CNN+MLP

```
import keras
from keras.models import Sequential
from keras.layers import LSTM
from keras.layers import Dense
from keras.layers import TimeDistributed
from keras.layers import Bidirectional
from keras.layers.normalization import BatchNormalization
from keras.layers import Convolution2D, MaxPooling2D, UpSampling2D, Flatten,
Dropout, Reshape, Conv2D, ConvLSTM2D, MaxPooling2D
from keras.layers import Dense, Embedding, Activation
from sklearn.model_selection import train_test_split
import math
import os
import mne
import numpy as np
import lee_grabaciones_cnn_mlp # lee_grabaciones_cnn_mlp es el fichero .py que
contiene una función por cada paciente y se encarga de leer de la base de datos las
señales EEG y seleccionar las muestras "preictales" e "interictales" de cada
grabación.
from time import time

# CAMBIAR NÚMERO DE PACIENTE SEGÚN INTERESE. PACIENTES 1,3,5,9,16,20,22,23
train_X,test_X,train_labels,test_labels = lee_grabaciones_dcnn_mlp.leer_paciente_1()#
De la función leer_paciente_*() se obtienen directamente las señales de train, test y
los vectores de etiquetas

model = Sequential()
model.add(Conv2D(32, (3, 2), activation='relu', input_shape=(1280,
len(train_X[1,1,:]), 1)))
model.add(BatchNormalization())# Esta capa aplica la técnica de normalización de
Batch, que mejora la velocidad de entrenamiento y reduce el overfitting
model.add(MaxPooling2D((2, 2)))
model.add(Conv2D(32, (3, 2), activation='relu'))
model.add(MaxPooling2D((2, 2)))
model.add(Conv2D(32, (3, 2), activation='relu'))
model.add(MaxPooling2D((2, 2)))
model.add(Conv2D(32, (3, 2), activation='relu'))

model.add(Flatten())
model.add(Dense(300, activation='relu'))
model.add(Dense(100, activation='relu'))
model.add(Dense(50, activation='relu'))
model.add(Dense(20, activation='relu'))
```

```

model.add(Dense(2, activation='sigmoid'))

model.summary()

model.compile(optimizer='rmsprop',
              loss='binary_crossentropy',

metrics=['accuracy',keras.metrics.SpecificityAtSensitivity(0.5),keras.metrics.SensitivityAtSpecificity(0.5),keras.metrics.FalsePositives(),keras.metrics.TruePositives(),keras.metrics.FalseNegatives(),keras.metrics.TrueNegatives()])

start = time() # Para calcular el tiempo de entrenamiento y test. Empieza a contar desde aquí

model.fit(train_X, train_labels, epochs=200, batch_size=32, validation_split=0.1)

test_loss, test_acc,specificity,sensitivity,fp,tp,fn,tn = model.evaluate(test_X, test_labels)

print("sensitivity calculada: ",tp/(tp+fn))# Imprime el valor de la sensibilidad
print("specificity calculada: ",tn/(tn+fp))# Imprime el valor de la especificidad
print("Acuraccy: ",test_acc)# Imprime el valor de la precisión
print("fp: ",fp)# Imprime el valor de los falsos positivos
print("h-1:", fp/(fp+fn+tn+tp))# Imprime el valor de las falsas alarmas
print("Train time: ", (time()-start)/60)# Imprime el valor del tiempo que ha tardado en ejecutarse el entrenamiento y el test
print("No de parámetros: ", model.count_params())# Imprime el número de parámetros de la red

```

➤ Código modelo CNN+Bi-LSTM

```

import keras
import mne
import numpy as np
from keras.models import Sequential
from keras.layers import LSTM
from keras.layers import Dense
from keras.layers import TimeDistributed
from keras.layers import Bidirectional
from keras.layers.normalization import BatchNormalization
from keras.layers import Lambda, Convolution2D, MaxPooling2D,MaxPool1D, Flatten, Dropout, Reshape, Conv2D, ConvLSTM2D, MaxPooling2D,MaxPooling3D, LSTM
from keras.layers.convolutional import Conv1D
from keras.layers import Dense, Embedding, Activation
import lee_grabaciones_dcnm_mlp # lee_grabaciones_mlp es el fichero .py que contiene una función por cada paciente y se encarga de leer de la base de datos las señales EEG y seleccionar las muestras "preictales" e "interictales" de cada grabación.
from time import time

# CAMBIAR NÚMERO DE PACIENTE SEGÚN INTERESE. PACIENTES 1,3,5,9,16,20,22,23

```

```

train_X, test_X, train_labels, test_labels = lee_grabaciones_dcnn_mlp.lee_paciente_1()#
De la función leer_paciente_*() se obtienen directamente las señales de train, test y
los vectores de etiquetas

model = Sequential()

model.add(Conv2D(32, (3, 2), activation='relu',
input_shape=(1280, len(train_X[1,1,:]),1)))
model.add(MaxPooling2D((2, 2)))
model.add(BatchNormalization())# Esta capa aplica la técnica de normalización de
Batch, que mejora la velocidad de entrenamiento y reduce el overfitting
model.add(Dropout(0.1))
model.add(Conv2D(32, (3, 2), activation='relu'))
model.add(MaxPooling2D((2, 2)))
model.add(BatchNormalization())
model.add(Conv2D(32, (3, 2), activation='relu'))

model.add(TimeDistributed(Flatten()))
model.add(Bidirectional(LSTM(20, return_sequences=True,
recurrent_dropout=0, activation="tanh", recurrent_activation="sigmoid")))
model.add(Flatten())
model.add(Dense(2, activation="sigmoid"))
model.summary()

model.compile(optimizer='rmsprop',
              loss='binary_crossentropy',

metrics=['accuracy',keras.metrics.SpecificityAtSensitivity(0.5),keras.metrics.Sensiti
vityAtSpecificity(0.5),keras.metrics.FalsePositives(),keras.metrics.TruePositives(),ke
ras.metrics.FalseNegatives(),keras.metrics.TrueNegatives()])

start = time() # Para calcular el tiempo de entrenamiento y test. Empieza a contar
desde aquí
model.fit(train_X, train_labels, epochs=20, batch_size=128, validation_split=0.1) #
Entrenamiento. 1280 muestras equivalen a 5 segundos de grabación
test_loss, test_acc, specificity, sensitivity, fp, tp, fn, tn = model.evaluate(test_X,
test_labels)# Test

print("sensitivity calculada: ",tp/(tp+fn))# Imprime el valor de la sensibilidad
print("specificity calculada: ",tn/(tn+fp))# Imprime el valor de la especificidad
print("Acuraccy: ",test_acc)# Imprime el valor de la precisión
print("fp: ",fp)# Imprime el valor de los falsos positivos
print("h-1:", fp/(fp+fn+tn+tp))# Imprime el valor de las falsas alarmas
print("Train time: ", (time()-start)/60)# Imprime el valor del tiempo que ha tardado
en ejecutarse el entrenamiento y el test
print("No de parámetros: ", model.count_params())# Imprime el número de parámetros
de la red

```

➤ Código modelo ChannelSelect-CNN+Bi-LSTM

```

import keras
from keras.models import Sequential
from keras.layers import LSTM
from keras.layers import Dense

```

```

from keras.layers import TimeDistributed
from keras.layers import Bidirectional
from keras.layers.normalization import BatchNormalization
from keras.layers import Convolution2D, MaxPooling2D,
UpSampling2D, Flatten, Dropout, Reshape, Conv2D, ConvLSTM2D, MaxPooling2D, LSTM
from keras.layers import Dense, Embedding, Activation
from sklearn.model_selection import train_test_split
import math
import os
import mne
import numpy as np

import lee_grabaciones_funciones_CS # lee_grabaciones_funciones_CS es el fichero
.py que contiene una función por cada paciente y se encarga de leer de la base de
datos las señales EEG y seleccionar las muestras "preictales" e "interictales" de cada
grabación.
from time import time

# CAMBIAR NÚMERO DE PACIENTE SEGÚN INTERESE. PACIENTES 1,3,5,9,16,20,22,23
train_X, test_X, train_labels, test_labels, train_signal, test_signal, data_relleno =
lee_grabaciones_funciones_CS.leer_paciente_1() # De la función leer_paciente_*( ) se
obtienen directamente las señales de train, test y los vectores de etiquetas

# Esta función es para darle el formato adecuado tanto a la matriz de train como de
test. Normalmente esto se realiza con la función 'reshape', pero me da error, creo que
debido al alto número de muestras
def mireshape(vector, relleno):

    partes = math.ceil(len(vector)/1280) # La variable "partes" indica el número de
matrices de tamaño 1280x23 en el que se dividen los datos de train
    aux = np.empty((partes,1280, len(vector[1,:])))
    for i in range(0, len(vector),1280):
        if (i==0):
            ind=0
        else:
            ind=int(i/1280)

        if (ind != partes-1):
            aux[ind,:,:] = vector[i:i+1280,:]
        else:
            tam_falta = 1280-len(vector[i:-1,:])
            ult_matriz=np.concatenate((vector[i:-1,:],relleno[0:tam_falta,:]))
            aux[ind,:,:] = ult_matriz

    vector_res = aux[:,:,:,:None]

    return vector_res,partes

# Función para definir el modelo
def modelo(num_canales):

    model = Sequential()

    model.add(Conv2D(32, (3, 2), activation='relu', input_shape=(1280, num_canales,1)))
    model.add(MaxPooling2D((2, 2)))
    model.add(BatchNormalization()) # Esta capa aplica la técnica de normalización de
Batch, que mejora la velocidad de entrenamiento y reduce el overfitting
    model.add(Dropout(0.1))
    model.add(Conv2D(32, (3, 2), activation='relu'))
    model.add(MaxPooling2D((2, 2)))
    model.add(BatchNormalization())
    model.add(Conv2D(32, (3, 1), activation='relu'))

```

```

model.add(TimeDistributed(Flatten()))
model.add(Bidirectional(LSTM(20, return_sequences=True,
recurrent_dropout=0,activation="tanh",recurrent_activation="sigmoid")))
model.add(Flatten())
model.add(Dense(2,activation="sigmoid"))

return model

model = modelo(len(train_signal[1,:])) # Se crea el modelo llamando a la función
model.summary()

model.compile(optimizer='rmsprop',
              loss='binary_crossentropy',
              metrics=['accuracy'])
model.fit(train_X, train_labels, epochs=20, batch_size=128) # Entrenamiento
test_loss, acc = model.evaluate(test_X, test_labels) # Test

from math import log, e
# Creación de la función para calcular la entropía
def entropy(signal, base=None): #Función para calcular la entropía

    n_signal = len(signal)

    if n_signal <= 1:
        return 0

    valor,counts = np.unique(signal, return_counts=True)
    probs = counts / n_signal
    n_classes = np.count_nonzero(probs)

    if n_classes <= 1:
        return 0

    ent = 0.

    # Compute entropy
    base = e if base is None else base
    for i in probs:
        ent -= i * log(i, base)

    return ent

## EMPIEZA EL ALGORITMO DE SELECCIÓN DE CANAL
m=8
done=0
varxent = np.zeros([1,len(train_signal[1,:])])
for i in range(1,len(train_signal[1,:]),1):
    varianza = train_signal[:,i].var()
    entropia = entropy(train_signal[:,i])
    varxent[0,i] = varianza*entropia

Temp = (-varxent).argsort() # Para obtener un vector de los números de canal
ordenados de mayor a menor producto de varianza por entropía

m=8
while done != 1 and m < 23: # Mientras done no se ponga a 1 y no se llegue a los 23
canales

```

```

Temp_rec = Temp[0,0:m]      # Me quedo con los m canales con mayor valor del producto
varianza por entropía, que son los más representativos.

train_signal_rec = train_signal[:,Temp_rec]      # Me quedo solo con los m canales de
mayor producto varianza por entropía
test_signal_rec = test_signal[:,Temp_rec]
data_relleno_rec = data_relleno[:,Temp_rec]

# Creación de nuevo modelo utilizando solo m canales

model = modelo(len(train_signal_rec[1,:]))

# Reshape

[train_X_rec,partes] = mireshape(train_signal_rec,data_relleno_rec)
[test_X_rec,partes] = mireshape(test_signal_rec,data_relleno_rec)

# Train y test

model.compile(optimizer='rmsprop',
              loss='binary_crossentropy',
              metrics=['accuracy'])
model.fit(train_X_rec, train_labels, epochs=20, batch_size=128)
test_loss, acc_new = model.evaluate(test_X_rec, test_labels)
print("Precisión:",acc_new,". Número de capas:",m)

if acc_new >= np.trunc(100*acc)/100:
    done=1
    Ch = Temp[1:m]
else:
    m=m+1
if m >= 23:
    print("Para este caso, no se puede reducir el número de canales sin perder
precisión")
else:
    print("El número de canales utilizado para obtener la misma o más precisión
es de",m,". Dejando de utilizar", 23-m,"canales, lo que implica un gran ahorro
computacional")

```