

Trabajo de Fin de Grado  
Grado en Ingeniería Electrónica, Robótica y  
Mecatrónica

Diseño de una red neuronal en Matlab para análisis  
de señales de electroencefalograma

Autor: Nagore Peirotén López de Arbina

Tutor: Alfredo Pérez Vega-Leal

Alejandro del Real Torres

Dpto. de Ingeniería Electrónica  
Escuela Técnica Superior de Ingeniería  
Universidad de Sevilla

Sevilla, 2018





Trabajo de Fin de Grado  
Grado en Ingeniería Electrónica, Robótica y Mecatrónica

# **Diseño de una red neuronal en Matlab para análisis de señales de electroencefalograma**

Autor:

Nagore Peirotén López de Arbina

Tutor:

Alfredo Pérez Vega-Leal

Profesor titular

Alejandro del Real Torres

Profesor titular

Dpto. de Ingeniería Electrónica  
Escuela Técnica Superior de Ingeniería  
Universidad de Sevilla  
Sevilla, 2018



Trabajo de fin de grado: Diseño de una red neuronal en Matlab para análisis de señales de electroencefalograma

Autor: Nagore Peirotén López de Arbina

Tutor: Alfredo Pérez Vega-Leal  
Alejandro del Real Torres

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

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2018

El Secretario del Tribunal



*A mi familia*

*A mis amigos*

*A mis maestros*



# Agradecimientos

---

En primer lugar, me gustaría agradecer a mi familia todo el apoyo ofrecido a lo largo de estos años, ya que sin ellos no hubiese sido posible llegar hasta aquí. Gracias por todos los consejos ofrecidos y por la confianza depositada en mí.

En segundo lugar, agradecer a mis amigos de toda la vida haberme animado en todo momento a seguir adelante y haber estado ahí cuando lo necesitaba a pesar de la distancia, y a los conocidos a lo largo de esta etapa gracias por compartir conmigo cada momento vivido en estos cuatro años.

Por último, agradecer a mis profesores Alfredo y Alejandro la oportunidad de trabajar en este proyecto en el que tanto interés tenía, y por la ayuda proporcionada durante la realización de este.

*Nagore Peirotén López de Arbina  
Escuela Técnica Superior de Ingeniería  
Sevilla, 2018*



En el presente documento se muestra el procedimiento llevado a cabo para la clasificación de los movimientos realizados con la mano izquierda y derecha, a partir de las señales obtenidas de un EEG. A continuación, se muestran los pasos realizados para dicha distinción.

Para empezar, se hace una breve introducción de los sistemas BCI. Estos sistemas se encargan de analizar las ondas cerebrales en tiempo real con el fin de poder controlar dispositivos externos, es decir, se encargan de traducir las intenciones del usuario en comandos de control.

Por un lado, se presenta el funcionamiento del cerebro y las características de las distintas ondas cerebrales en el momento en el que se produce un movimiento por parte del individuo.

Por otro lado, se muestran los distintos procedimientos llevados a cabo para el análisis de las ondas cerebrales obtenidas mediante un EEG. Más concretamente, se analiza la transformada Wavelet y los filtros de paso banda.

A continuación, se realiza la extracción de características de las señales obtenidas del apartado anterior, con el fin de caracterizar cada una de las señales de manera que luego sea posible la identificación de los distintos movimientos.

Por último, se exponen las redes neuronales artificiales como herramientas para realizar la clasificación de los distintos movimientos. Tras esto, se analizan los resultados obtenidos para conocer el mejor clasificador.



# Abstract

---

The present document shows the procedure carried for the classification of the movements made with the left and right hand using the signals obtained with an EEG. The steps made for this distinction are shown below.

To begin, a brief introduction of the BCI systems is made. These systems are responsible for analyzing the brain waves in real time in order to be able to control external devices, in other words, they are responsible for translating the user's intention into control demands.

On the one hand, it is shown the behavior of the brain and the characteristics of the different brain waves that are presented when a person makes a movement.

On the other hand, the different procedures carried out for the analysis of brain waves obtained by an EEG are shown. More specifically, Wavelet transform and bandpass filters are analyzed.

Then, the extraction of characteristics of the signals obtained from previous section is performed, in order to characterize each of the signals so the identification of the different movements is possible.

Finally, ANNs are exposed as tools to perform the classification of the different movements. After that, the results obtained from then are analyzed to choose the best classifier.

# Índice

---

<b>Agradecimientos</b>	<b>ix</b>
<b>Resumen</b>	<b>xi</b>
<b>Abstract</b>	<b>xiii</b>
<b>Índice</b>	<b>xiv</b>
<b>Índice de Tablas</b>	<b>xvi</b>
<b>Índice de Figuras</b>	<b>xvii</b>
<b>Notación</b>	<b>xix</b>
<b>1 Introducción</b>	<b>11</b>
1.1. <i>Contexto y objetivos</i>	11
1.2. <i>Software</i>	11
1.2.1. <i>Matlab</i>	11
1.2.2. <i>BioSig</i>	13
<b>2 Ondas cerebrales</b>	<b>15</b>
<b>3 Movimiento del cuerpo y cerebro</b>	<b>18</b>
<b>4 Introducción a los sistemas bci</b>	<b>21</b>
4.1. <i>Adquisición de señales</i>	21
4.1.1. <i>Método invasivo</i>	21
4.1.2. <i>Método no invasivo</i>	22
4.2. <i>Preprocesamiento de señales</i>	23
4.3. <i>Extracción de características</i>	24
4.4. <i>Clasificación</i>	24
4.4.1. <i>Clasificador de Bayes</i>	24
4.4.2. <i>Redes neuronales artificiales</i>	24
<b>5 Experimento</b>	<b>26</b>
<b>6 Análisis de las señales</b>	<b>29</b>
6.1. <i>Transformada Wavelet</i>	29
6.2. <i>Filtros de paso banda</i>	33
<b>7 Extracción de características</b>	<b>37</b>
<b>8 Red neuronal</b>	<b>39</b>
8.1. <i>Neurona y red neuronal biológica</i>	39
8.2. <i>Neurona y red neuronal artificial</i>	39
8.3. <i>Historia de las redes neuronales</i>	40
8.4. <i>Clasificación redes neuronales artificiales</i>	41
8.4.1. <i>Perceptrón simple</i>	41
8.4.2. <i>Perceptrón multicapa</i>	41
8.5. <i>Red neuronal diseñada</i>	41
8.6. <i>Mejora de los resultados obtenidos</i>	47
8.6.1. <i>30 neuronas en la capa oculta</i>	48

8.6.2.	50 neuronas en la capa oculta	49
<b>9</b>	<b>Análisis de los resultados y conclusiones</b>	<b>53</b>
<b>10</b>	<b>Códigos</b>	<b>55</b>
9.1.	<i>Código Transformada Wavelet</i>	55
9.2.	<i>Código separación de canales y extracción de características</i>	56
<b>11</b>	<b>Referencias</b>	<b>65</b>

# ÍNDICE DE TABLAS

---

Tabla 4-1. <i>Nomenclatura para el posicionamiento de electrodos</i>	23
Tabla 6-1: <i>Asociación de colores a cada una de las ondas cerebrales</i>	32
Tabla 6-2: <i>Descomposición de las señales</i>	33
Tabla 9-1: <i>Resultados obtenidos de las distintas redes neuronales</i>	53

# ÍNDICE DE FIGURAS

---

Ilustración 1-1: <i>Logotipo Matlab</i>	11
Ilustración 1-2: <i>Interfaz gráfica del usuario del Wavelet Toolbox</i>	12
Ilustración 1-3: <i>Logotipo BioSig</i>	13
Ilustración 2-1: <i>Onda cerebral Delta</i>	15
Ilustración 2-2: <i>Onda cerebral Theta</i>	15
Ilustración 2-3: <i>Onda cerebral Alfa</i>	15
Ilustración 2-4: <i>Onda cerebral Beta</i>	16
Ilustración 2-5: <i>Onda cerebral Gamma</i>	16
Ilustración 3-1: <i>Áreas funcionales de la corteza cerebral</i>	18
Ilustración 3-2: <i>Anatomía del cerebro humano</i>	18
Ilustración 4-1: <i>Componentes sistemas BCI</i>	21
Ilustración 4-2: <i>Posición Nasión, Inión y punto preauricular</i>	22
Ilustración 4-3: <i>Posicionamiento de los electrodos en el cuero cabelludo siguiendo el International Estándar 10-20 System</i>	23
Ilustración 5-1: <i>Esquema temporal de las pruebas</i>	26
Ilustración 6-1: <i>Distintas representaciones de la señal</i>	29
Ilustración 6-2: <i>Representación tiempo-frecuencia de la señal</i>	30
Ilustración 6-3: <i>Adaptación del principio de incertidumbre de Heisenberg</i>	30
Ilustración 6-4: <i>Diferencias entre la DWT y la WPT</i>	31
Ilustración 6-5: <i>Descomposición de la señal utilizando la WPT</i>	32
Ilustración 6-6: <i>Filtro para la obtención de la onda Alfa</i>	34
Ilustración 6-7: <i>Filtro para la obtención de la onda Beta</i>	35
Ilustración 8-1: <i>Estructura de una neurona biológica</i>	39
Ilustración 8-2: <i>Función sigmoideal</i>	40
Ilustración 8-3: <i>Selección de los datos de entrada y salida de la red</i>	42
Ilustración 8-4: <i>Representación de las muestras de los datos de entrada y salida de la red</i>	42
Ilustración 8-5: <i>Selección del número de neuronas de la capa oculta</i>	42
Ilustración 8-6: <i>Arquitectura de la red neuronal con 10 neuronas en la capa oculta</i>	42
Ilustración 8-7: <i>División de los datos en los distintos sets</i>	43
Ilustración 8-8: <i>Porcentaje de errores cometidos en la identificación de los datos de cada uno de los sets</i>	44
Ilustración 8-9: <i>Estructura de la matriz de confusión</i>	45
Ilustración 8-10: <i>Matrices de confusión de la red neuronal con 10 neuronas en la capa oculta</i>	46
Ilustración 8-11: <i>Selección del número de neuronas en la capa oculta</i>	48
Ilustración 8-12: <i>Arquitectura de la red neuronal con 30 neuronas en la capa oculta</i>	48

Ilustración 8-13: <i>Porcentaje de errores de la red neuronal con 30 neuronas en la capa oculta</i>	48
Ilustración 8-14: <i>Matrices de confusión de la red neuronal con 30 neuronas en la capa oculta</i>	49
Ilustración 8-15: <i>Estructura de la red neuronal con 50 neuronas en la capa oculta</i>	50
Ilustración 8-16: <i>Porcentaje de errores de la red neuronal con 50 neuronas en la capa oculta</i>	50
Ilustración 8-17: <i>Matrices de confusión de la red neuronal con 50 neuronas en la capa oculta</i>	51

BCI	Brain-Computer Interface
ICO	Interfaz Cerebro-Ordenador
EEG	Electroencefalograma
MEG	Magnetoencefalografía
ECoG	Electrocorticografía
ANN	Artificial Neural Network
FFT	Fast Fourier Transform
WT	Wavelet Transform
WPT	Wavelet Packet Transform
IIR	Infinite Impulse Response
PSD	Power Spectral Density
Hz	Hercio
FIR	Finite Impulse Response
ADL	Análisis Discriminante Lineal



# 1 INTRODUCCIÓN

---

**E**n este primer capítulo de la memoria se expone una introducción del proyecto y los objetivos de este, además de la explicación del software utilizado.

## 1.1. Contexto y objetivos

En 1924, el EEG fue inventado por Hans Berger. Este descubrió la existencia de actividad eléctrica en el cerebro gracias al EEG. Realizó una investigación sobre los ritmos cerebrales humanos y sus modificaciones en función de las condiciones fisiológicas y patológicas. Dándoles nombre así a las distintas ondas cerebrales.

Más adelante, en 1970 la DARPA inició un programa para explorar las comunicaciones cerebrales haciendo uso del EEG. Se inició entonces la investigación de los sistemas BCI en la Universidad de California.

Inicialmente, el campo de investigación BCI se centró en fines médicos. Como claro ejemplo de esto, está la implantación de prótesis neuronales para poder recuperar la audición, vista o movilidad dañadas en un individuo. Sin embargo, en la primera década del siglo XXI la investigación no solo se centró en fines médicos, sino también en la industria de los videojuegos. Con esto llegó el siguiente cambio, los BCI no invasivos.

A pesar de existir distintos tipos de sistemas BCI, todos se basan en el mismo principio de funcionamiento: primero se realiza la adquisición de las señales, luego estas son preprocesadas, tras esto, se obtienen sus características más importantes, y por último, a través de la clasificación se consigue una interacción con el entorno.

El principal objetivo de este proyecto consiste en realizar la clasificación de los movimientos realizados con la mano izquierda y derecha, a partir de las señales obtenidas de un EEG con la mayor precisión posible. Las aplicaciones que esto podría tener son varias, tales como poder realizar el control de un coche en un videojuego y decidir si se quiere que este vaya hacia la izquierda (realizando el movimiento con la mano izquierda) o hacia la derecha (realizando el movimiento con la mano derecha).

## 1.2. Software

### 1.2.1. Matlab

Para el desarrollo de la parte software de este proyecto se ha usado la herramienta de software matemático Matlab. Esta herramienta se utiliza para investigación y resolución de problemas prácticos de ingeniería y matemáticas. Se centra en aplicaciones de control y procesamiento de señales.



Ilustración 1-1: *Logotipo Matlab*

Se ha hecho uso de varios *toolboxes* para poder ampliar sus prestaciones. Concretamente se han utilizado los siguientes:

- Neural Network Toolbox.
- Wavelet Toolbox.
- Signal Processing Toolbox.

El Neural Network Toolbox proporciona algoritmos, modelos y aplicaciones para poder crear, entrenar, visualizar y simular redes neuronales.

El Wavelet Toolbox a su vez, proporciona funciones y aplicaciones para el análisis y síntesis de señales e imágenes. Este toolbox incluye algoritmos para realizar el análisis continuo o discreto Wavelet, o análisis de paquetes Wavelet. Soporta varias familias Wavelet incluyendo las familias Daubechies, Morlet y Gausiana.

Esta toolbox además de contar con funciones que se pueden utilizar desde la línea de comandos, consta de una interfaz gráfica de usuario.

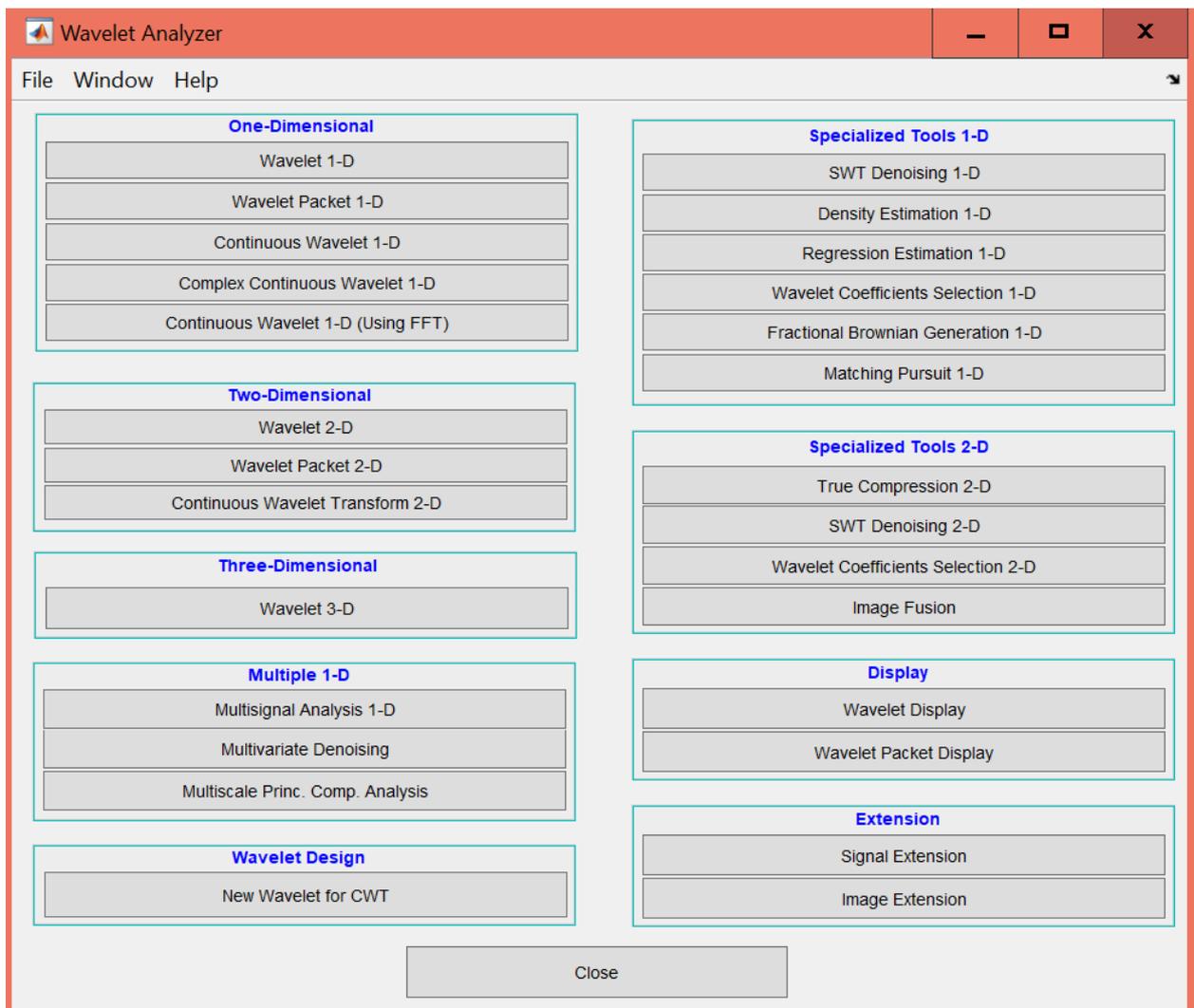


Ilustración 1-2: Interfaz gráfica del usuario del Wavelet Toolbox

Por último, se ha utilizado el Signal Processing Toolbox. Este proporciona funciones y aplicaciones para analizar, preprocesar y extraer características de señales. En este toolbox están incluidas herramientas para el análisis y diseño de filtros.

### 1.2.2. BioSig

También se ha hecho uso de la librería de software libre BioSig. Esta sirve para realizar el procesamiento de señales biomédicas. Ha sido necesario el uso de esta librería debido a que los ficheros de los que se disponían para realizar este proyecto estaban en formato gdf. Este es el formato general de datos para señales biomédicas, pero Matlab no puede trabajar directamente con el.



Ilustración 1-3: *Logotipo BioSig*



## 2 ONDAS CEREBRALES

---

Debido a la comunicación entre las neuronas, se genera actividad eléctrica en el cerebro, siendo la onda cerebral el resultado de esta.

Existen diferentes tipos de ondas cerebrales en función de su frecuencia, entre las que se encuentran las ondas Delta, Theta, Alfa, Beta y Gamma. A continuación, se muestra la clasificación, las características, los beneficios que aportan y la forma que tiene de cada una de ellas.

Para empezar, esta la onda Delta, siendo esta la de menor frecuencia (1 a 3 Hz) y por lo tanto la de mayor amplitud. Estas ondas son más habituales en niños pequeños, ya que a medida que se envejece se produce una cantidad menor. Se producen cuando el individuo duerme, si aparecen en el estado de vigilia, se relaciona con defectos físicos en el cerebro. Esta onda nos permite conseguir un buen descanso mientras soñamos, sintiéndonos así renovados una vez despiertos. Además, su aparición se relaciona con el aumento de la hormona del crecimiento.



Ilustración 2-1: *Onda cerebral Delta*

La siguiente onda que nos encontramos con una frecuencia entre 3.5 y 8 Hz es la onda Theta. Esta onda se asocia con las capacidades imaginativas, con la reflexión y el sueño debido a que representan la delgada línea entre estar dormido o despierto. Además, surge de la tensión emocional, por lo que en el momento que el individuo experimenta emociones profundas, muestra una elevada actividad en este rango de frecuencias. Su aparición se asocia con la reducción de la fatiga mental, la ansiedad y el estrés.



Ilustración 2-2: *Onda cerebral Theta*

La tercera clase de onda es la Alfa, esta se encuentra en el rango de frecuencias de 8 a 13 Hz. Está asociada a la relajación y a la desconexión, pero no al sueño. El pensamiento de algo pacífico con los ojos cerrados, por ejemplo, conlleva un aumento de la actividad de la onda Alfa. El nivel de actividad de esta onda, es mayor en los niños y adolescentes que en los adultos. Esta onda, permite una reducción de la presión arterial e incremento del flujo sanguíneo del cerebro.

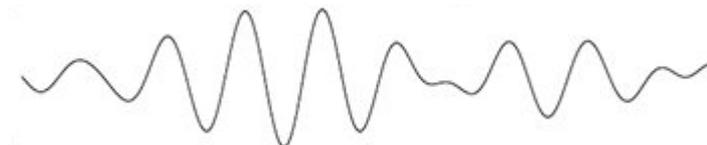


Ilustración 2-3: *Onda cerebral Alfa*

También está la onda Beta. Esta se produce en el rango de frecuencias de 13 a 33 Hz. Estas ondas son pequeñas y rápidas. Están asociadas a la concentración enfocada o al pánico, lo que implica una actividad neuronal intensa. Un exceso de este tipo de onda puede derivar en ansiedad. Esta onda conlleva la mejora de la concentración y el estado de alerta, permitiendo esto una mejora de la lógica, el razonamiento y el pensamiento crítico.



Ilustración 2-4: *Onda cerebral Beta*

Por último, está la onda Gamma, Esta es la onda de mayor frecuencia (mayor a 30 Hz) y es la que tiene una amplitud menor. Su aparición se relaciona con una gran actividad mental y con la consolidación de la información.



Ilustración 2-5: *Onda cerebral Gamma*

A pesar de que se diferencian los tipos de ondas que se han mencionado, hay que tener en cuenta que no son realmente ondas separadas, sino que la actividad cerebral es una mezcla de todas las frecuencias al mismo tiempo.

Debido a las características que se han mencionado, las ondas más interesantes a la hora de analizar el movimiento realizado por un individuo son las Alfa y Beta.



# 3 MOVIMIENTO DEL CUERPO Y CEREBRO

Existe una parte del cerebro, llamada corteza motora que se encarga de la planificación, control y ejecución de las funciones motoras voluntarias. La corteza motora se divide en dos partes: corteza motora primaria y corteza motora secundaria.

La corteza motora primaria o M1, se encuentra en la parte posterior del lóbulo frontal. Esta zona se encarga de generar y enviar la orden de movimiento a todos los músculos voluntarios del organismo.

La corteza motora secundaria, sin embargo, no realiza los movimientos, sino que se encarga de organizar el movimiento antes de que la primaria lo lleve a cabo. Esta zona incluye las siguientes regiones: corteza parietal posterior, corteza premotora y área motora suplementaria.

La corteza parietal posterior se encarga de transformar la información visual en instrucciones motoras. La corteza premotora, sin embargo, se encarga de la programación y guía del movimiento de los músculos proximales y del tronco corporal. Esta es la zona en la que se almacenan los programas motores que se han ido aprendiendo con la experiencia. Por último, está el área motora suplementaria o AMS que se encarga de la inicialización de los movimientos y de la planificación, programación y coordinación de los movimientos complejos.

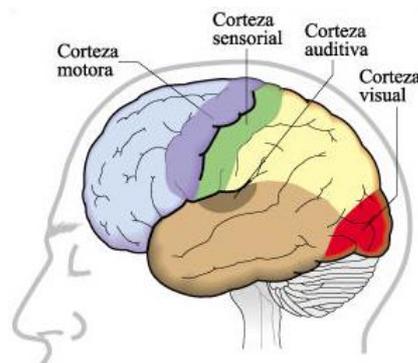


Ilustración 3-1: *Áreas funcionales de la corteza cerebral*

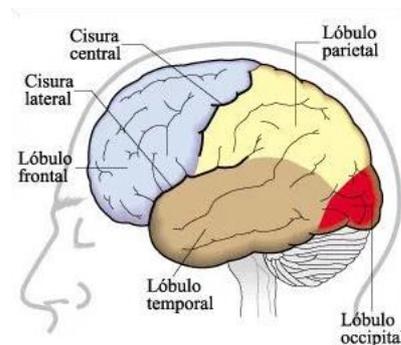


Ilustración 3-2: *Anatomía del cerebro humano*

La cantidad de materia que está destinada a cada parte del cuerpo representa la cantidad de control que la corteza motora primaria tiene sobre esa parte del cuerpo.

Se necesita mucho espacio cortical para controlar movimientos complejos como los movimientos de los dedos de la mano, por lo que estas partes del cuerpo tienen una mayor representación en M1 que las piernas o el tronco.

Para poder ver que parte de la corteza motora está asociada a cada parte del cuerpo podemos ver la representación del Homúnculo motor de Penfield, donde se representa esta área como si de una forma humana se tratase. El homúnculo motor está situado en el surco central de la corteza frontal.

Como se puede ver en la siguiente imagen hay zonas en nuestro cuerpo que ocupan una parte mayor de la corteza cerebral, como las manos, la boca y en líneas generales la cara. De todas formas, esta área se desarrolla de manera distinta en cada ser humano, dependiendo de las partes del cuerpo que se utilicen más.

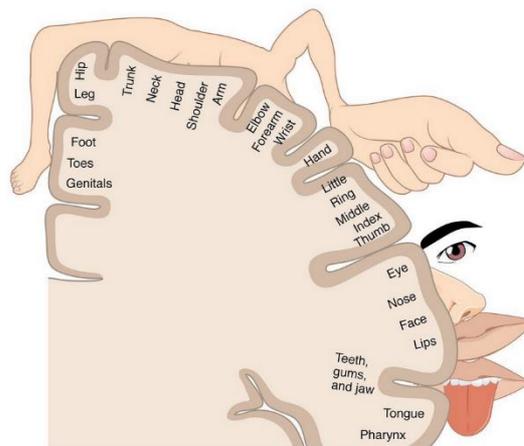


Ilustración 3-3: Representación Homúnculo motor de Penfield

A pesar de que previamente se ha comentado la importancia de las ondas Alfa y Beta a la hora de analizar el movimiento realizado por un sujeto, existe también el ritmo mu. Este se da en las mismas frecuencias que la onda cerebral alfa, con la diferencia de que la onda alfa aparece en el cortex visual en reposo en la parte posterior del cuero cabelludo y el ritmo u onda mu aparece en el cortex motor, en una banda que va a aproximadamente de oreja a oreja. Debido a esto a pesar de que se tienen las señales obtenidas de los 21 electrodos situados en las distintas partes del cuero cabelludo, solo se han seleccionado los electrodos C3 y C4.

Se sabe que la parte izquierda del cerebro se encarga de controlar la parte derecha del cuerpo, mientras que la parte derecha del cerebro se encarga de controlar la parte izquierda del cuerpo. Debido a esto, se ha descubierto que, si se imagina o se realiza un movimiento de la parte izquierda del cuerpo, las ondas mu solo deberían suprimirse en la parte izquierda del cerebro. Al tener las señales obtenidas de los electrodos C3 (situado en la parte izquierda del cerebro en la corteza motora) y C4 (situado en la parte derecha del cerebro en la corteza motora) es posible ver en cual de los dos lados del cerebro se suprime la onda mu.

En el caso de querer diferenciar no solo el movimiento de la parte izquierda del movimiento de la parte derecha del cuerpo, sino también diferenciar el movimiento de distintas partes del cuerpo, es necesario fijarse en la parte de la corteza sensorimotora es suprimida la onda mu.



# 4 INTRODUCCIÓN A LOS SISTEMAS BCI

Los sistemas BCI analizan las ondas cerebrales en tiempo real con el fin de controlar dispositivos externos. Estos sistemas se encargan de traducir las intenciones del usuario en comandos de control convirtiéndolos así en herramientas que permiten que personas que se encuentran en situación de dependencia mejoren su calidad de vida.

Las distintas funcionalidades que pueden tener son las siguientes: Control de una silla de ruedas, control de los movimientos de una extremidad protésica, traducción de nuestros pensamientos a texto, etc.

Estos sistemas contienen los siguientes componentes:

- Adquisición de señales.
- Pre-procesamiento.
- Extracción de características.
- Clasificación.



Ilustración 4-1: Componentes sistemas BCI

## 4.1. Adquisición de señales

Uno de los componentes principales de un sistema BCI es la adquisición de señales mediante las cuales se mide la oscilación de la actividad cerebral. Existen dos métodos distintos para llevar a cabo la adquisición de señales: método invasivo y método no invasivo.

### 4.1.1. Método invasivo

Por una parte, existe el método invasivo, en esta tecnología los electrodos son implantados quirúrgicamente bajo el cuero cabelludo. Existen ciertas ventajas en esta técnica. Para empezar, se mejora de la calidad de la señal obtenida, la relación señal-ruido que se obtiene es mucho mejor que en los sistemas no invasivos.

Sin embargo, este método puede dar varios problemas. El primer problema es el procedimiento quirúrgico que hay que llevar a cabo para la implantación de los electrodos, además de la adaptación del cuerpo a un objeto nuevo. Además de esto, la zona cerebral que se monitoriza es muy pequeña, y una vez que el electrodo ha sido

implantado no puede moverse para medir la actividad en otra área distinta del cerebro.

Un ejemplo de este método es la ECoG, esta es un tipo de EEG invasiva en la que los electrodos se ubican directamente sobre la superficie del cerebro y se conectan a un electroencefalógrafo.

#### 4.1.2. Método no invasivo

Por otra parte, está el método no invasivo. A diferencia del método anterior, esta técnica no requiere implantación ninguna de objetos externos en el cerebro del sujeto. Esto permite evitar los problemas que surgen del procedimiento quirúrgico en la técnica anterior. Como principal inconveniente de este método se encuentra el nivel de la señal conseguido, ya que esta es mucho más pobre que la obtenida en el método invasivo. Dentro de este método existen diversas técnicas como la MEG y la EEG.

##### 4.1.2.1. MEG

La magnetoencefalografía es una técnica que registra la actividad funcional cerebral, mediante la captación de campos magnéticos. Este método, utiliza un dispositivo superconductor de interferencia cuántica y una computadora, para así medir la actividad neuromagnética dentro del cerebro.

##### 4.1.2.2. EEG

La electroencefalografía consiste en el registro de la actividad eléctrica del cerebro mediante electrodos aplicados sobre el cuero cabelludo. Estos electrodos se suelen situar siempre en el mismo lugar, siguiendo el Sistema 10-20 explicado más adelante. Es importante que los electrodos sean colocados siempre en el mismo lugar, para que se puedan comparar los estudios realizados a un sujeto en el transcurso de tiempo, o sea posible la comparación de los resultados entre distintos sujetos.

Debido a la necesidad de un buen contacto entre los electrodos y el cuero cabelludo, a cada electrodo se le aplica un gel a través de una jeringuilla sin aguja.

El *International Standard 10-20 system* es un sistema de posicionamiento de electrodos en el que a cada sector del cuero cabelludo se le asigna una letra y un número. Los números 10 y 20 se deben a que la distancia entre los electrodos adyacentes va desde el 10% al 20% de la distancia total entre los puntos reconocibles del cráneo, estos puntos son:

- Nasió n indentación entre frente y nariz.
- Inión; protuberancia occipital.
- Punto preauricular; delante del trago de cada pabellón de la oreja.

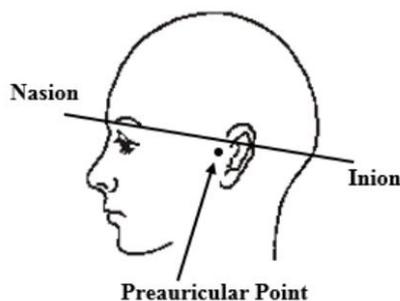


Ilustración 4-2: Posición Nasió n, Inión y punto preauricular

A continuación se muestra la imagen con la nomenclatura utilizada para los distintos electrodos posicionados en el cuero cabelludo:

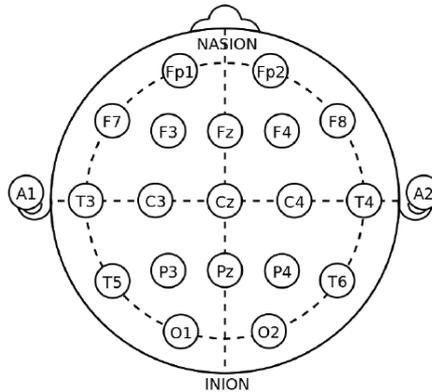


Ilustración 4-3: Posicionamiento de los electrodos en el cuero cabelludo siguiendo el *Insternacional Estándar 10-20 System*

Por una parte, la letra indica el lóbulo en el que está posicionado el electrodo. Las letras que aparecen son las siguientes:

Tabla 4-1. *Nomenclatura para el posicionamiento de electrodos*

Letra	Lóbulo
Fp	Prefrontal
F	Frontal
C	Central
P	Parietal
O	Occipital
T	Temporal

En la imagen anterior se puede apreciar que también aparece la letra A. Esta letra se corresponde con el proceso óseo prominente que generalmente se encuentra justo detrás del oído externo.

Por otra parte, el número de cada uno de los electrodos se asigna en función de la localización del hemisferio en el que está posicionado el electrodo. En el caso de que el electrodo esté situado en el hemisferio derecho, su número será par, mientras que, si está situado en el hemisferio izquierdo, a este se le asignará un número impar. Como se puede comprobar en la imagen, en alguno de los electrodos la letra Z es la que acompaña a la letra encargada de indicar el lóbulo en el que esta situado el electrodo en lugar de un número. Esta letra indica que el electrodo está situado en la línea media del plano sagital del cráneo.

## 4.2. Preprocesamiento de señales

La parte de preprocesamiento de las señales consiste en filtrar las señales y eliminar algunos posibles artefactos que se encuentren superpuestos a la señal de interés.

En el caso de haber obtenido las señales a partir de la EEG, interesa eliminar artefactos como el parpadeo, el movimiento de los ojos, movimientos musculares, etc.

### 4.3. Extracción de características

Una vez se han obtenido las ondas cerebrales, se procede con la extracción de características. En esta parte del sistema hay que seleccionar las características más significativas del tipo de señal que se está analizando para así poder realizar la clasificación de manera correcta.

Las razones para realizar la extracción de características son las siguientes:

- Proveer un conjunto relevante de características al clasificador.
- Reducir redundancia.
- Recuperar características latentes significativas.
- Visualización de los datos.

### 4.4. Clasificación

Ciertas veces antes de realizar la clasificación se realiza el ADL. El análisis discriminante lineal, se utiliza para el reconocimiento de patrones. Se basa en encontrar una combinación lineal de rasgos para poder separar dos o más clases de objetos o eventos. Se consigue una reducción de dimensiones muy útil a la hora de realizar la clasificación.

Existen diversos métodos de clasificación como el clasificador de Bayes, las redes neuronales artificiales, etc. Como algoritmo de clasificación de este proyecto se va a usar una red neuronal artificial.

#### 4.4.1. Clasificador de Bayes

Cuando se utiliza el clasificador de Bayes, los objetos a clasificar son elementos de una población. Cada objeto se clasifica en la clase que más probabilidades tiene de haber producido.

Para calcular dicha probabilidad se utiliza el teorema de Bayes:

$$P(c_i/x^o) = \frac{P(x^o/c_i)P(c_i)}{P(X)}$$

Se elige la clase que más probabilidades tiene de haberse producido de la siguiente manera:

$$k = \arg \max_i P(c_i/x^o)$$

Siendo:

- $c_i$  los distintos tipos de clases.
- $x^o$  el objeto, evento a clasificar.
- $P(A)$  es la probabilidad de que ocurra un evento de tipo A.
- $P(A/B)$  es la probabilidad condicional de A dado B

#### 4.4.2. Redes neuronales artificiales

Las redes neuronales, son una nueva forma de computación. Son un modelo matemático compuesto por un gran número de elementos simples de procesos muy interconectados que procesan información por medio de su estado dinámico como respuesta a entradas externas.

Las redes neuronales artificiales presentan muchas características semejantes a las del cerebro humano. Por ejemplo, son capaces de aprender de la experiencia y de realizar generalizaciones. Estas se explicarán con mas detalle en los capítulos posteriores.



# 5 EXPERIMENTO

Para este proyecto se han utilizado los datos proporcionados por “BCI Competition IV”. Más concretamente se ha utilizado el set de datos 2a.

Este set contiene los datos de 9 sujetos diferentes. El experimento realizado por cada uno de los sujetos consta de cuatro tipos de movimientos distintos, mano izquierda, mano derecha, ambos pies y lengua.

La adquisición de las señales en todos los sujetos se realizó a través de 22 canales para la obtención del EEG, en el que el posicionamiento de dichos electrodos se llevó a cabo siguiendo el sistema internacional 10-20 mencionado anteriormente. Las señales se obtuvieron con una frecuencia de muestreo de 250 Hz y filtradas entre 0.5 y 100 Hz. Además, se utilizó un filtro notch para eliminar la frecuencia de 50 Hz de la red eléctrica.

Cada uno de los experimentos realizado por cada sujeto contiene 288 pruebas, siguiendo cada una de las pruebas el siguiente esquema:

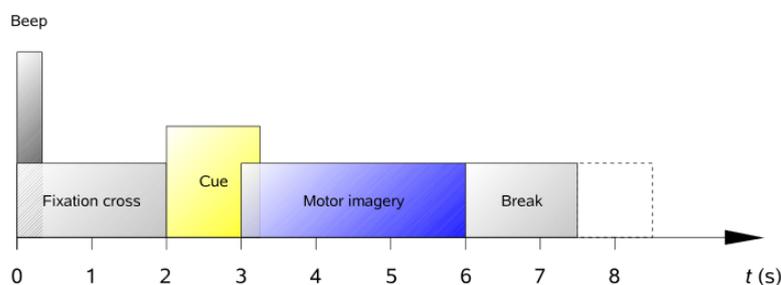


Ilustración 5-1: *Esquema temporal de las pruebas*

Como se puede apreciar en la imagen anterior, la duración total de cada una de las pruebas es de 7,5 segundos. En el instante 0, aparece una cruz en la pantalla y un corto tono acústico que sirve para notificar del inicio de la prueba. En  $t = 2s$  aparece por pantalla una flecha apuntando a la izquierda (mano izquierda), derecha (mano derecha), abajo (pies) o arriba (lengua) indicando el movimiento que tiene que realizar el sujeto. Esta indicación permanece en la pantalla 1.25s. Al sujeto se le pedía que continuase con la realización del movimiento hasta que la cruz de la pantalla desapareciese en  $t = 6s$ . Tras esto, la pantalla se vuelve de color negro durante un corto periodo de tiempo.

Además de las señales, se tiene una estructura que contiene información sobre el set de datos. Nos da información como la posición, muestra en la que comienza cada prueba, el tipo de movimiento que se realiza en dicha prueba y la duración de esta.

Como para este proyecto solo resultaban interesantes los movimientos realizados con las manos, solo se han utilizado las pruebas correspondientes a estas clases de movimientos. Por lo tanto, antes de comenzar el análisis de señales, se separó cada una de las pruebas en las distintas clases y se seleccionaron solo las clases que correspondían al movimiento de la mano izquierda o al movimiento de la mano derecha.

Cabe destacar, que existía cierto número de pruebas que tenían superpuestas algunos artefactos a la señal de interés. Estas muestras estaban identificadas y han sido descartadas.

Una vez separadas las pruebas, se separó cada una de ellas en dos muestras distintas, la primera de ellas correspondiente a la referencia, cuando todavía no se había iniciado el movimiento de la mano, y la segunda de ellas correspondiente a la realización del movimiento. Debido a la frecuencia de muestreo con el que se obtuvo cada una de las señales, se escogió un total de 512 muestras para la referencia de cada prueba y otras 512 muestras para la realización del movimiento de cada prueba.

El siguiente paso a realizar, fue seleccionar los canales interesantes para cada una de las señales. Como se ha mencionado antes, debido a la respuesta cerebral al movimiento corporal, solo se seleccionaron dos canales,

los correspondientes a los electrodos C3 y C4.

Por lo tanto para analizar cada una de las pruebas se usan 4 señales distintas:

- Referencia del canal C3.
- Movimiento del canal C3.
- Referencia del canal C4.
- Movimiento del canal C4.



# 6 ANÁLISIS DE LAS SEÑALES

Se han planteado dos herramientas diferentes para realizar el análisis de las ondas cerebrales. El objetivo de ambas, es obtener las señales Alfa y Beta por separado, ya que cada una tiene un rango de frecuencias distinto y son las ondas en las que más se ve reflejado el movimiento realizado por el individuo.

## 6.1. Transformada Wavelet

Una señal puede ser representada de varias maneras distintas, se puede tener una representación Tiempo-Amplitud, Tiempo-Frecuencia etc. En el caso de tener una señal obtenida a través de un EEG es interesante tener la información del tiempo y de la frecuencia al mismo tiempo.

Esto no sería posible en el caso de que se use la Transformada de Fourier (FT), ya que esta herramienta se utiliza para el análisis en frecuencia de señales, por lo que cuando se le aplica dicha transformada a la señal, se obtiene una representación Frecuencia-Amplitud de la señal, es decir, esta transformada nos dice las componentes de frecuencia de la señal, pero no nos dice en qué momento se da cada una de las componentes.

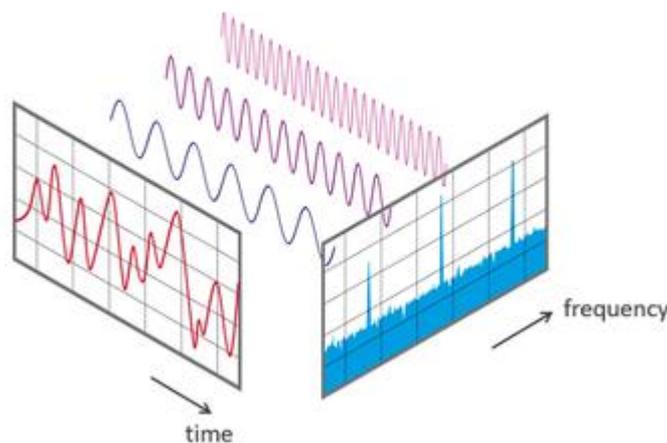


Ilustración 6-1: *Distintas representaciones de la señal*

Por lo tanto, el problema es que no hay información de la frecuencia de la señal en el dominio del tiempo, ni se tiene información del tiempo en el dominio de la frecuencia. Debido a esto, aparece la necesidad de buscar otra transformada que permita la representación Tiempo-Frecuencia como lo es la WT (Transformada Wavelet). Esta, nos permite tener información del tiempo y de la frecuencia al mismo tiempo.

La DWT (Transformada Wavelet Discreta) realiza un filtrado continuo de la señal de entrada usando dos filtros (filtro paso baja y filtro paso alta), y así se descompone la señal de entrada en diferentes escalas. Lo que se consigue con esto, es dividir la señal en distintas frecuencias conservando la información temporal de la señal. Es decir, se obtienen varias señales que representan la misma señal, pero cada una corresponde a una banda de frecuencia distinta. Como sabemos que señal corresponde a cada frecuencia, es posible juntar las distintas señales obtenidas al aplicar la DWT y representarlas en un gráfico 3D en los que los ejes son: frecuencia, tiempo y amplitud. El tiempo está directamente relacionado con la traslación, mientras que el parámetro de la escala es la inversa de la frecuencia.

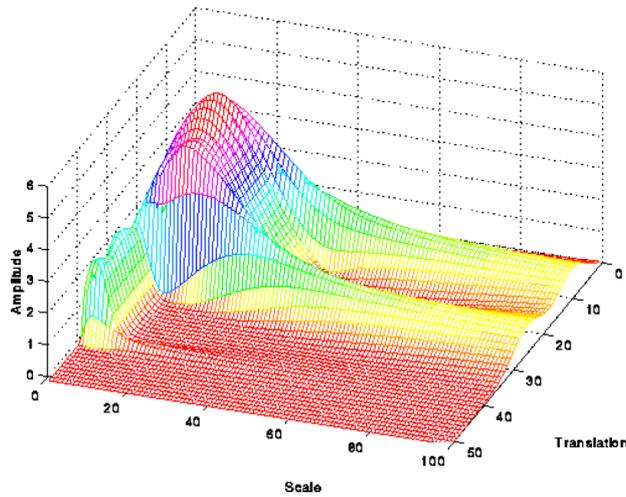


Ilustración 6-2: Representación tiempo-frecuencia de la señal

Antes de continuar, se tiene que conocer el principio de incertidumbre de Heisenberg. Este dice que es imposible medir simultáneamente, y con precisión absoluta el valor de la posición y la cantidad de movimiento de una partícula. Esto significa que la precisión con la que se pueden medir las cosas es limitada. Este principio también puede aplicarse a la transformada Wavelet de la siguiente manera: no podemos conocer el valor exacto del tiempo y de la frecuencia de un punto.

Mirando la siguiente imagen, se puede ver que, aunque el ancho y la altura de los rectángulos cambie, el área de los rectángulos sigue siendo el mismo. Cada uno de los rectángulos representa una porción equitativa del plano tiempo-frecuencia, por lo que todos los puntos que caen dentro de la misma caja son representados por el mismo valor de la transformada Wavelet.

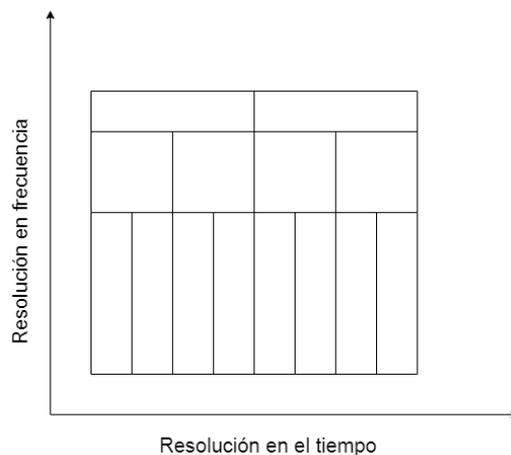


Ilustración 6-3: Adaptación del principio de incertidumbre de Heisenberg

Cuando estamos a bajas frecuencias, la altura de los rectángulos es menor, lo que corresponde con una mejor precisión en la frecuencia, y el ancho del rectángulo es mayor, lo que corresponde con una baja resolución en el tiempo. Para altas frecuencias, sin embargo, ocurre lo contrario el ancho de los rectángulos se hace menor, lo que equivale a una mejora en la precisión del tiempo, pero se incrementa la altura del rectángulo, haciendo que la precisión de la frecuencia disminuya.

Con esto se llega a la conclusión de que la transformada Wavelet tiene buena precisión de tiempo y mala precisión de frecuencia a altas frecuencias, y tiene buena resolución de frecuencia y baja de tiempo para bajas frecuencias.

La DWT primero divide la señal en detalle y coeficientes aproximados, donde el detalle y los coeficientes aproximados se corresponden a las componentes de alta y baja frecuencia de la señal respectivamente. En el siguiente nivel, solo las componentes de baja frecuencia (coeficientes aproximados) pueden descomponerse en dos partes.

También existe la WPT (Wavelet Packet Transform) que se basa en la transformada Wavelet, pero al contrario que esta, hace la descomposición de altas y bajas frecuencias al mismo tiempo, es decir, en cada nivel se descompone la señal en aproximación y detalle, lo que crea un árbol binario proporcionando así una información más detallada que la DWT. Además de esto, ahorra computación y es más adecuada para casos en los que la señal tiene discontinuidades y cambios abruptos. Por todas estas razones, se ha decidido utilizar la WPT para la descomposición de la señal.

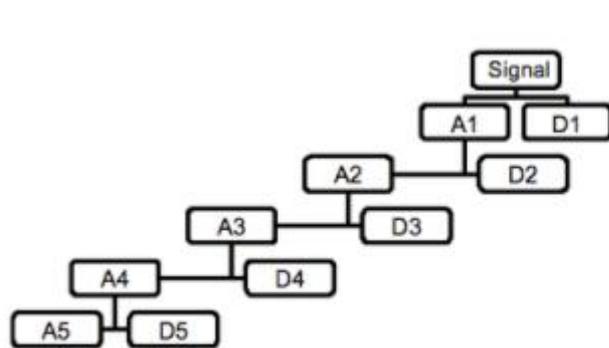


Fig. 1. wavelet tree decomposition

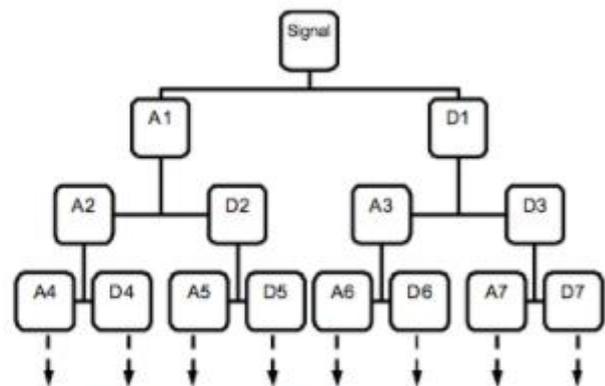


Fig. 2. Wavelet Packet decomposition tree

Ilustración 6-4: *Diferencias entre la DWT y la WPT*

Una vez decidido esto, queda escoger el tipo de Wavelet madre que se va a usar. Se ha decidido hacer uso de la familia Daubechies. Esta transformada se caracteriza por un número máximo de desvanecimientos (momentos nulos en el dominio del tiempo) de la señal.

Se ha decidido hacer uso de esta ya que, en todos los artículos relacionados con la aplicación de la transformada Wavelet a las ondas cerebrales, es esta la que se utiliza.

Además de esto, la Daubechie se denota por dbN, siendo N el orden de esta. Se ha escogido la de orden 4, o como se le conoce, db4. Cada señal va a poder tener la mitad de números de desvanecimientos que el orden que se haya escogido, en nuestro caso 2.

A continuación, se puede ver la descomposición en árbol resultante para nuestra señal.

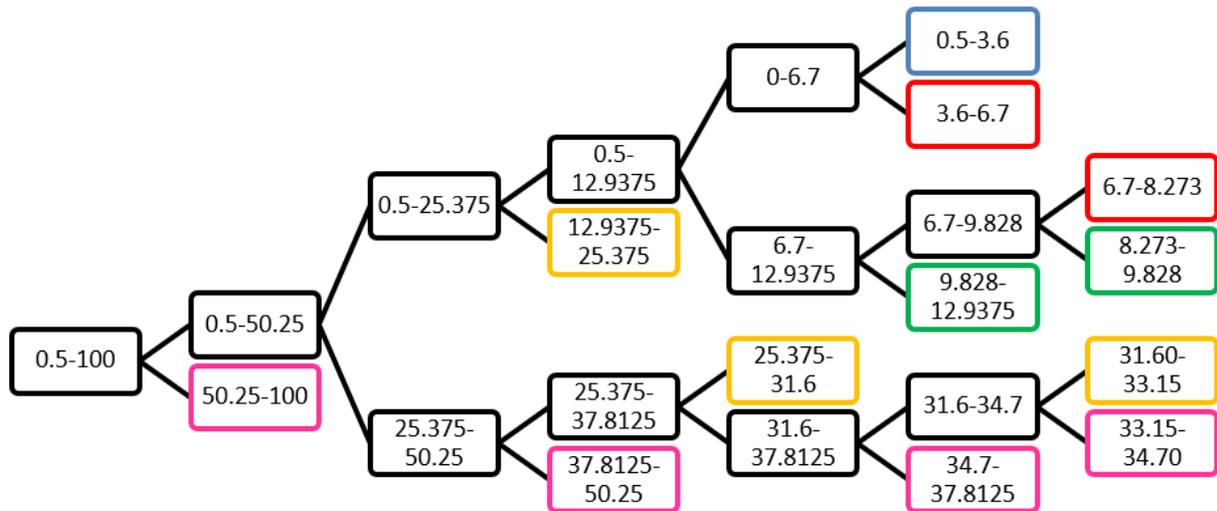


Ilustración 6-5: Descomposición de la señal utilizando la WPT

Cada uno de los colores indica una onda cerebral distinta. A continuación, se muestra la tabla de los colores asociados a cada una de ellas.

Onda cerebral	Color
Delta	Azul
Theta	Rojo
Alfa	Verde
Beta	Naranja
Gamma	Rosa

Tabla 6-1: Asociación de colores a cada una de las ondas cerebrales

En algunos casos, ha sido necesaria la descomposición de una onda en más de una para poder recoger todo su rango de frecuencias. A continuación, se muestra la clasificación de las ondas obtenidas a través de la WPT y los filtros necesarios para obtener cada una de ellas:

Onda cerebral	Banda de frecuencia (Hz)	Filtros
Delta 1	0-3.125	LLLLL
Theta 1	3.125-6.25	LLLLH
Theta 2	6.25-7.8125	LLHLL
Alfa 1	7.8125-9.375	LLLHLH
Alfa 2	9.375-12.5	LLLHH

Beta 1	12.5-25	LLH
Beta 2	25-31.25	LHLL
Beta 3	31.25-32.8125	LHLHLL
Gamma 1	50-100	H
Gamma 2	37.5-50	LHH
Gamma 3	34.375-37.5	LHLHH
Gamma 4	32.8125-34.375	LHLHLH

Tabla 6-2: *Descomposición de las señales*

A pesar de haber realizado un programa en Matlab que permitiese realizar la descomposición mencionada, como se ha comentado antes, cada uno de los filtros de la transformada va seguido por una reducción del factor de muestreo de 2. Como consecuencia, en los casos en los que hay que aplicar una gran cantidad de filtros hasta obtener la onda deseada, el número de muestras se reduce de tal manera que es insuficiente para realizar la extracción de características correctamente.

Debido a la frecuencia de muestreo con la que se obtuvo la señal, esta herramienta no va a ser útil para la señal que se quiere analizar.

## 6.2. Filtros de paso banda

La otra opción para realizar el análisis de las ondas cerebrales y obtener las ondas Alfa y Beta es diseñar unos filtros de paso banda para los rangos de frecuencia de cada una de las ondas.

El primer filtro que se ha diseñado, ha sido con el fin de obtener la onda alfa, como se ha mencionado antes en la clasificación y descripción de las distintas ondas cerebrales, la alfa tiene un rango de frecuencia de 8 a 13 Hz. Como se puede apreciar en la siguiente foto, se ha seleccionado un filtro de paso banda FIR, de orden 250, cuyas frecuencias de corte son 7 y 13.5 Hz. El tipo de ventana que se ha utilizado ha sido la Kaiser.

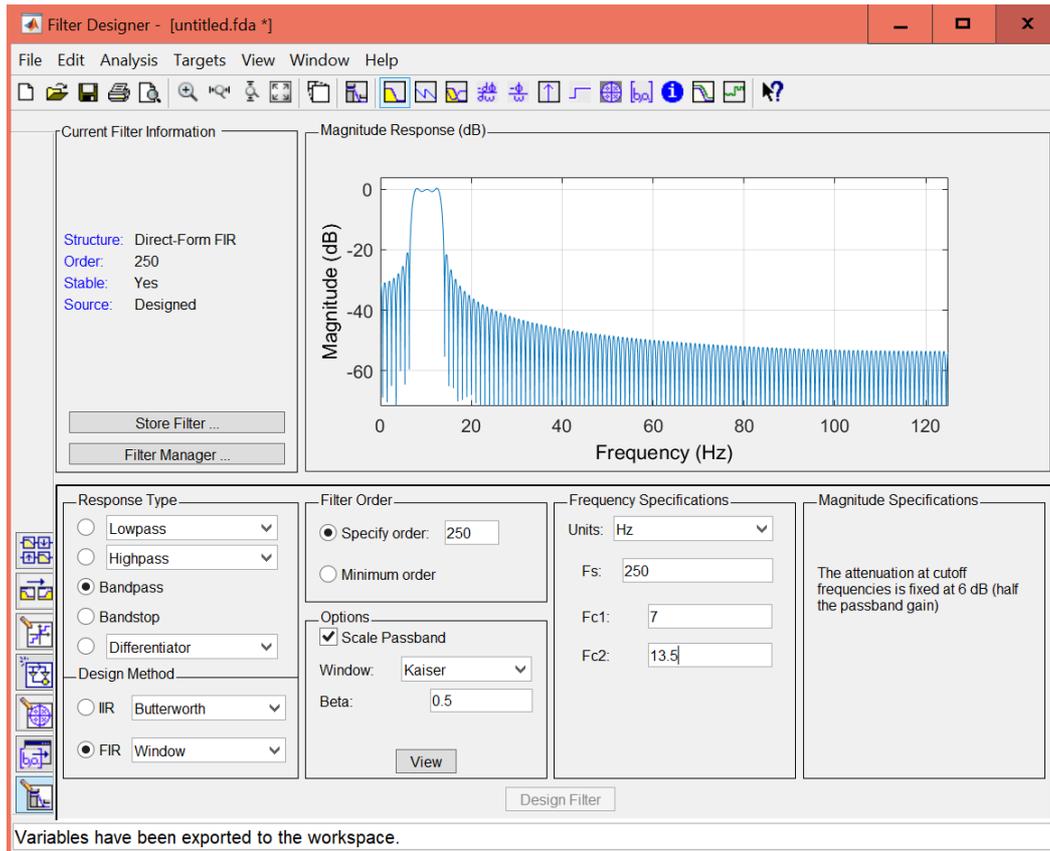


Ilustración 6-6: Filtro para la obtención de la onda Alfa

El segundo filtro que se ha diseñado, es el que se ha usado para obtener la onda beta. Como se ha mencionado antes en la clasificación y descripción de las distintas ondas cerebrales, la onda beta tiene un rango de frecuencia de 13 a 13 Hz. Como se puede apreciar en la siguiente foto, se ha seleccionado un filtro de paso banda FIR, de orden 250, cuyas frecuencias de corte son 12 y 33 Hz. El tipo de ventana que se ha utilizado ha sido la Kaiser.

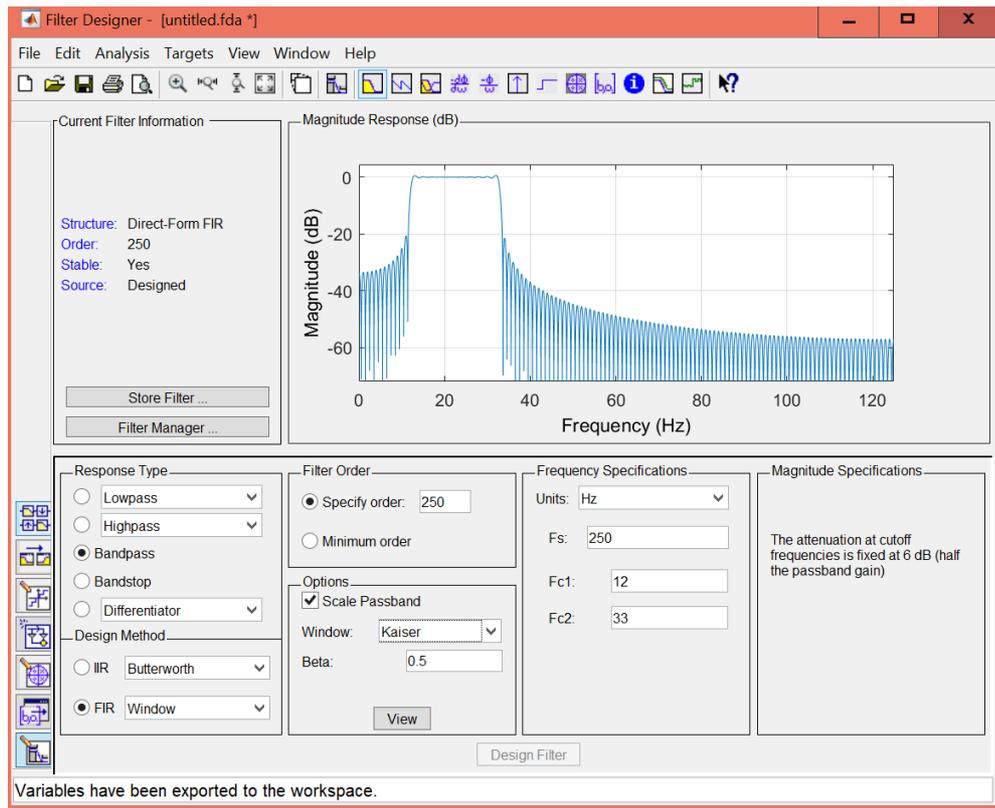


Ilustración 6-7: Filtro para la obtención de la onda Beta



# 7 EXTRACCIÓN DE CARACTERÍSTICAS

---

La extracción de características de una señal se usa para poder describir y representar dicha señal. En el caso de estar hablando de una señal de un EEG, la extracción de características es una parte muy importante para la clasificación de patrones relevantes correspondientes a una tarea concreta.

Lo que se busca en esta parte del proceso es ser capaces de encontrar características en las ondas cerebrales Alfa y Beta, previamente obtenidas para que en función de estas sea posible diferenciar el movimiento de la mano izquierda del movimiento de la mano derecha.

Basándose en la información obtenida de como afecta el movimiento corporal al cerebro, se ha decidido calcular las siguientes características para las ondas Alfa y Beta obtenidas de los electrodos C3 y C4 durante el instante de referencia y durante el tiempo en el que se ha realizado el movimiento:

- Valor medio.
- Desviación típica.
- Varianza.
- Valor mínimo.
- Valor máximo.
- Curtosis.
- Sesgo.
- PSD.

La primera característica que se ha calculado ha sido el valor medio de la señal, es decir, la media de todos los valores que definen la señal. Esto se calcula de la siguiente manera:

$$\mu_x = \frac{1}{N} \sum_{i=1}^N X_i$$

Siendo N el número de muestras y X la amplitud instantánea de la señal.

El siguiente valor que se va a calcular es la desviación típica o desviación estándar. Esta es una medida de dispersión que indica lo dispersos que están los datos con respecto a la media. Se calcularía de la siguiente manera:

$$Std = \sqrt{\frac{\sum_{i=1}^N (X_i - \mu_x)^2}{N - 1}}$$

Una vez que tenemos la desviación típica, es muy sencillo realizar el cálculo de la varianza de X, ya que la desviación típica es la raíz cuadrada de la varianza, es decir:

$$\sigma_x = \sqrt{Std}$$

Como se ha mencionado antes, los valores máximos y mínimos que adquiere la señal en un periodo de tiempo también pueden ser de interés.

Lo siguiente que se ha calculado es el sesgo de la señal. Esto es una medida de asimetría de la distribución de probabilidad y de su media. Se calcula de la siguiente manera:

$$Skew = \frac{\sum_{i=1}^N (X_i - \mu_x)^3}{(N - 1) \cdot \sigma_x^3}$$

También se ha analizado la Curtosis. Este coeficiente analiza el grado de concentración que presentan los valores alrededor de la zona central de la distribución. Una mayor Curtosis implica una mayor concentración de valores de la variable muy cerca de la media de la distribución (pico) y muy lejos de la misma (colas), existiendo una menor frecuencia de valores intermedios (hombros).

Esta se calcularía de la siguiente manera:

$$Kurt = \frac{\sum_{i=1}^N (X_i - \mu_x)^4}{(N - 1) \cdot \sigma_x^4}$$

También puede ser útil el cálculo de la Densidad Espectral de Potencia, PSD. Este valor muestra como está distribuida la potencia de la señal sobre las distintas frecuencias de las que está formada. La densidad espectral de potencia va a ser la siguiente:

$$S_{xx}(f) = TF\{R_{xx}(\tau)\}$$

Siendo FT la Transformada de Fourier y  $R_{xx}$  la función de autocorrelación de  $x(t)$ . La función de autocorrelación es útil para encontrar patrones repetitivos dentro de la señal. No se ha utilizado la densidad espectral de potencia como tal, sino su media.

Una vez calculadas estas características de la señal en todos los casos mencionados previamente, se ha llegado a la conclusión de que no todos ellos son útiles para la descripción de las señales de un EEG. Más concretamente, se ha descartado el uso de la Curtosis y del Sesgo, debido a que su valor era prácticamente 0 en todos los casos y no aportaba información ninguna de la señal analizada.

Por lo tanto, se han seleccionado 6 características distintas para cada una de las señales. Como se ha mencionado anteriormente, se han seleccionado las ondas Alfa y Beta de dos canales distintos en dos instantes de tiempo distintos (referencia y movimiento), lo que hacen un total de 48 datos para la clasificación de un movimiento. Estos 48 datos son los que se utilizan como entrada para la red neuronal artificial explicada con detalle más adelante. Como salida para la red, se utilizará un 0 en el caso de que el movimiento se corresponda con el de la mano izquierda y un 1 en el caso de que el movimiento se corresponda con el de la mano derecha.

# 8 RED NEURONAL

## 8.1. Neurona y red neuronal biológica

La neurona es la célula responsable de transferir la información y los impulsos dieléctricos alrededor del cuerpo. Todos los datos se transfieren de neurona a neurona a través del impulso eléctrico.

Las neuronas tienen tres partes principales, el cuerpo celular, el axón y la dendrita. El cuerpo celular o soma, es la parte principal de la neurona en la que se encuentran todos los componentes necesarios de la célula. El axón es la parte más larga de la neurona y la que se encarga de enviar señales eléctricas a otras neuronas. Por último, está la dendrita, que es la parte que crea las conexiones con otras células. Esta parte puede estar en uno o ambos lados de la célula. La dendrita sería la parte de la neurona que actúa como entrada, mientras que el axón sería la parte que actúa como salida.

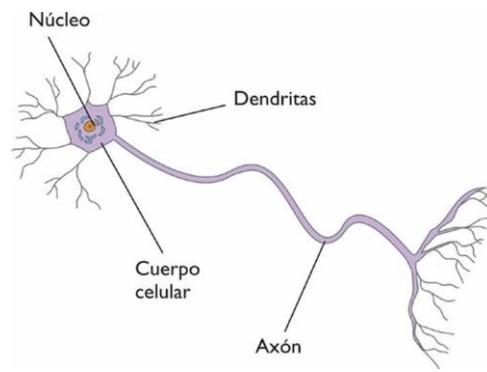


Ilustración 8-1: *Estructura de una neurona biológica*

Las redes neuronales biológicas están compuestas por un gran número de neuronas dispuestas en capas. Las neuronas de una capa suelen tener como entrada las neuronas de otra capa, y como salida las neuronas de una tercera capa distinta de las anteriores. Las conexiones entre estas neuronas tienen pesos asociados. Estos pesos representan la influencia que tiene una sobre la otra, es decir, un peso pequeño implica que el valor de entrada tenga una pequeña influencia independientemente del valor de dicha entrada, y un peso grande por el contrario implica una gran influencia del valor de entrada.

Además de esto, una neurona biológica puede estar activa o inactiva, o lo que es lo mismo, excitada o no excitada.

## 8.2. Neurona y red neuronal artificial

Una red neuronal artificial, al igual que una biológica tiene entradas, utiliza pesos y genera salidas. Existen tres funciones distintas: la función de entrada, la función de activación y la función de salida.

La función de entrada más usada es la sumatoria de las entradas pesadas, esta función, es la suma de cada uno de los valores de entrada a la neurona, multiplicados por sus pesos asociados.

Por otra parte, está la función de activación, esta se encarga de calcular el estado de actividad de una neurona. La función de activación más conocida es la función sigmoideal. Esta es una función no lineal que tiene límites finitos en  $-\infty$  y  $\infty$ . Normalmente es una función que va desde 0 a 1, pudiendo ir también de -1 a 1. Suele tener un punto de inflexión en  $x = 0$ . Es una función diferenciable. Es una función monótonica pero su derivada no lo es. Tiene la siguiente forma:

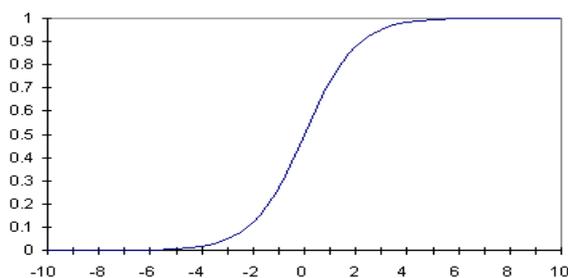


Ilustración 8-2: *Función sigmoideal*

La ecuación que describe la función sigmoideal es la siguiente:

$$A = \frac{1}{1 + e^{-x}}$$

Por último, está la función de salida, cuyo resultado es la salida de la neurona. La función más común es la función identidad, en este caso la salida será la misma que la entrada.

A pesar de su gran similitud, existen ciertas diferencias entre las neuronas biológicas y artificiales. Para empezar, las neuronas artificiales se basan en funciones lógicas, no admitiéndose raciocinios que no se encuentren entre sus “bases”, mientras que las biológicas son el resultado de procesos complejos en varias partes del cuerpo. Además, las neuronas artificiales se basan en el análisis, comienzan por una idea fundamental y la descomponen hasta conocer todos sus componentes, mientras que las biológicas se basan en la síntesis.

### 8.3. Historia de las redes neuronales

Alan Turing fue el primero en estudiar el cerebro como una forma de ver el mundo de la computación en 1936, pero fueron el neurofisiólogo Warren McCulloch y el matemático Walter Pitts quienes, en 1943, lanzaron una teoría acerca de la forma de trabajar de las neuronas. Estos realizaron el modelo de una red neuronal simple mediante el uso de circuitos eléctricos.

El proceso de aprendizaje sin embargo, no fue desarrollado hasta 1949 por Donald Hebb. Este proceso es el fundamento de la mayoría de las funciones de aprendizaje que pueden darse en una red neuronal hoy en día. Los trabajos de Hebb, formaron las bases de la Teoría de las Redes Neuronales.

No fue hasta 1950 cuando Karl Lashley descubrió que la información no es almacenada de forma centralizada en el cerebro, sino que es distribuida encima de él. Seis años más tarde, en el Congreso de Dartmouth de 1956 se produjo el Nacimiento de la inteligencia artificial. En esta conferencia se discutió sobre la capacidad de las computadoras para simular el aprendizaje.

En 1957 Frank Rosenblatt inició el desarrollo del Perceptrón. Una vez aprendido una serie de patrones, este modelo podía reconocer otros patrones similares aunque estos no se hubiesen usado en su entrenamiento. A pesar del gran avance que esto supuso, este modelo no tenía la capacidad suficiente como para clasificar clases no separables linealmente. Dos años más tarde, publicó el libro *Principios de Neurodinámica*, en el cual confirmó que en ciertas condiciones, el aprendizaje del Perceptrón convergía hacia un estado finito.

Un año después, en 1960 el profesor Bernard Widrow junto con su alumno Ted Hoff desarrollaron el modelo Adaline, *Adaptive Linear Elements*. Esta fue la primera red neuronal que se utilizó para solucionar un problema real, concretamente se utilizó para eliminar ecos en las líneas telefónicas.

En 1969 se produjo una caída de las investigaciones de las redes neuronales debido a la crítica que recibió el Perceptrón por parte de Marvin Minsky y Seymour Papert. Estos probaron que el perceptrón no era capaz de resolver problemas relativamente fáciles como el aprendizaje de una función no lineal.

Años más tarde, en 1974, Paul Werbos describió por primera vez el algoritmo de aprendizaje *backpropagation* (de propagación hacia atrás).

El renacimiento de la redes neuronales no ocurrió hasta 1985, año en el que John Hopfield publicó el libro *Computación neuronal de decisiones en problemas de optimización*.

Actualmente, se realizan numerosos trabajos en relación a las redes neuronales. Cada año nuevas aplicaciones surgen y nuevos productos son lanzados por las empresas.

## **8.4. Clasificación redes neuronales artificiales**

Existen distintos tipos de redes neuronales, entre los que se encuentran el perceptrón simple y el perceptrón multicapa.

### **8.4.1. Perceptrón simple**

El perceptrón simple es una red neuronal que está formada por dos capas de neuronas. En este tipo de red todas las neuronas de la primera capa (capa de entrada) están unidas con todas las neuronas de la segunda capa (capa de salida). El inconveniente de este modelo es que solo permite la resolución de problemas linealmente separables.

### **8.4.2. Perceptrón multicapa**

El perceptrón multicapa es una generalización del perceptrón simple, pero al contrario que este, permite la resolución de problemas que no son linealmente separables. Esto se debe a que además de las capas de entrada y salida tiene una o más capas ocultas. La unión de las capas se realiza de la siguiente manera: las neuronas de la capa de entrada se unen con las neuronas que forman la primera capa oculta, estas a su vez se unen con las neuronas de la siguiente capa oculta, y así sucesivamente hasta la última capa oculta cuyas neuronas se unen con las de la capa de salida.

Además de esto existe otra clasificación dentro del perceptrón multicapa, ya que este puede ser totalmente o parcialmente conectado.

Se dice que un perceptrón multicapa está totalmente conectado cuando todas las neuronas de una capa están conectadas con todas las neuronas de la siguiente capa.

Por el contrario, se dice que un perceptrón multicapa está parcialmente conectado cuando cada una de las neuronas de una capa está conectada con una serie de neuronas de la siguiente capa.

## **8.5. Red neuronal diseñada**

Una vez realizada la extracción de características del EEG previamente explicada, siguiendo el esquema de un sistema BCI, se ha realizado la clasificación. Como algoritmo de clasificación se ha usado una red neuronal artificial diseñada a través del toolbox de Matlab Neural Network Toolbox previamente mencionado.

Lo primero que hay que tener en cuenta a la hora de realizar el diseño de una red neuronal es que hay que seguir ciertos pasos para realizarla correctamente.

El primer paso es la recogida de datos. Los datos usados para la red diseñada son los obtenidos tras realizar la extracción de características. Concretamente se han utilizado 1034 pruebas distintas, de las cuales 510 son de clase 0 (correspondientes al movimiento de la mano izquierda) y 524 de clase 1 (correspondientes al

movimiento de la mano derecha). Cada una de las pruebas cuenta con 48 entradas distintas y una única salida.

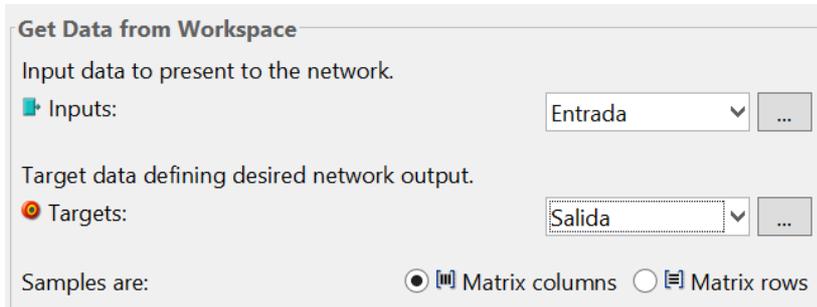


Ilustración 8-3: Selección de los datos de entrada y salida de la red

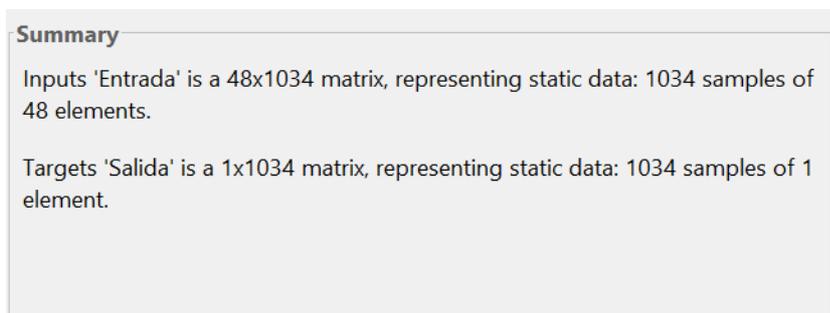


Ilustración 8-4: Representación de las muestras de los datos de entrada y salida de la red

El siguiente paso a realizar es la creación de la red, en este paso hay que especificar el número de neuronas que va a tener en la capa de entrada (será igual al número de entradas), en la capa oculta y en la capa de salida (igual que el número de salidas). En este caso, tendrá 48 neuronas en la capa de entrada, 10 en la capa oculta y 1 en la capa de salida.

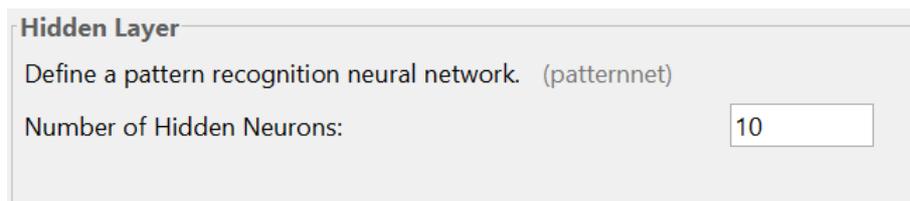


Ilustración 8-5: Selección del número de neuronas de la capa oculta

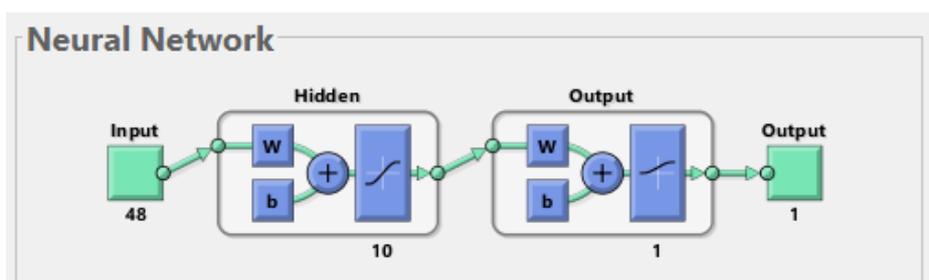


Ilustración 8-6: Arquitectura de la red neuronal con 10 neuronas en la capa oculta

Una vez creada, hay que configurar la red. En este caso al hacer uso del toolbox la inicialización de los pesos y bías se hace de manera automática.

El próximo paso es el entrenamiento de la red. Para ello es necesario dividir los datos obtenidos en los siguientes 3 subsets:

- Set de Entrenamiento: Es el que se usa para calcular el gradiente y actualizar los pesos y bías de la red.
- Set de Validación: Durante la primera fase de la validación disminuye el error de validación, hasta que en un punto el error empieza a aumentar otra vez, en ese momento se queda con el valor de los pesos y bías que menor error hayan dado.
- Set de Test: No se usa para el entrenamiento, sino para la comparación de modelos, ver el porcentaje de error etc.

Se realiza dicha separación debido a que, a la hora de realizar el entrenamiento, lo que se hace primero es utilizar los datos del set de entrenamiento. Para estos datos se obtiene un error de entrenamiento muy grande y una función inicial. Se prueba esta función con los datos del set de validación, obteniéndose un error muy alto en la clasificación.

Una vez hecho esto, se sigue entrenando la red con los datos de entrenamiento. Con esto lo que se va consiguiendo es una reducción del error de la clasificación de los datos del set de entrenamiento y se van calculando nuevas funciones. Cuando se prueban estas funciones con los datos del set de validación se va apreciando una disminución en el error de clasificación de los datos del set de validación. Este error irá disminuyendo a medida que se entrena la red con los datos del set de entrenamiento hasta que llegue un punto en el cual el error comience a aumentar de nuevo. El aumento del error al usar la función obtenida mediante los datos del set de entrenamiento se debe a que llega un momento en el que el error de entrenamiento es nulo, y por consecuencia la función que se obtiene se ajusta perfectamente a los datos del set de entrenamiento, pero no a los de validación.

Debido a esto, se seleccionan los pesos y bías de la red que producen menor error con los datos del set de validación. Cuando esto ocurre el error de los datos del set de validación es igual al error de los datos del set de entrenamiento.

Finalmente se utiliza el set de test para comprobar los resultados que se obtienen al utilizar la red para la clasificación de unos datos que no se han utilizado durante el entrenamiento.

En este caso se tienen datos de 1034 pruebas, los datos de 724 pruebas se han utilizado para el set de entrenamiento, los de 155 pruebas para el set de validación y los datos de 155 pruebas para el set de test. Estas pruebas han sido escogidas de manera aleatoria. La división de las muestras se puede ver en la siguiente imagen.

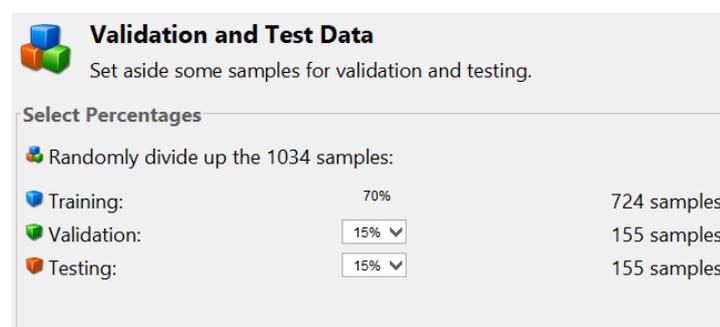


Ilustración 8-7: División de los datos en los distintos sets

Una vez realizado el entrenamiento se obtiene como resultado el porcentaje de errores cometidos por la red al clasificar cada uno de los sets. Para conseguir un resultado que refleje de manera correcta la capacidad de clasificación de la red, se ha seleccionado una red neuronal artificial cuyo porcentaje de errores/aciertos ha sido aproximadamente el obtenido al realizar la media entre los resultados obtenidos al entrenar la misma red

15 veces.

Los porcentajes de error de la red seleccionada son los siguientes:

Results			
	 Samples	 CE	 %E
 Training:	724	6.65483e-1	33.56353e-0
 Validation:	155	7.66754e-1	36.12903e-0
 Testing:	155	7.78922e-1	39.35483e-0

Ilustración 8-8: *Porcentaje de errores cometidos en la identificación de los datos de cada uno de los sets*

Como se puede ver en la imagen anterior, el porcentaje de error en las muestras del set de entrenamiento ha sido del 33,5%, en las muestras del set de validación ha sido aproximadamente del 36,1 % y finalmente en las muestras del set de test ha sido aproximadamente de 39,3%.

No solo se obtienen estos datos para la red neuronal diseñada, también se obtienen por ejemplo varias matrices de confusión. Antes de analizar las matrices de confusión obtenidas se van a comentar algunos conceptos relacionados con estas.

La matriz de confusión es una herramienta que se utiliza como método de evaluación de un modelo de clasificación. A partir de esta se pueden obtener los siguientes parámetros:

- Sensibilidad o Tasa de Verdaderos Positivos: Proporción de casos positivos que fueron correctamente identificados.
- Especificidad o Tasa de Verdaderos Negativos: Proporción de casos negativos que fueron correctamente identificados.
- Tasa de Falsos Positivos: Proporción de casos negativos que fueron identificados como casos positivos.
- Tasa de Falsos Negativos: Proporción de casos positivos que fueron identificados como casos negativos.
- Precisión: Proporción de casos que fueron identificados correctamente.
- Exactitud: Proporción entre los positivos reales identificados y todos los casos positivos.

Las matrices de transformación que se han obtenido tienen la siguiente forma:

<b>VALOR PREDICHO</b>	0	1	2	3
	1	4	5	6
		7	8	9
		0	1	
		<b>VALOR REAL</b>		

Ilustración 8-9: Estructura de la matriz de confusión

Cada uno de los cuadrantes mostrados en la imagen es el siguiente parámetro:

1. Número de casos negativos que fueron correctamente clasificados.
2. Número de casos positivos que fueron clasificados como casos negativos.
3. Proporción de casos negativos que lo son realmente de entre todos los identificados como negativos.
4. Número de casos negativos que fueron clasificados como casos positivos.
5. Número de casos positivos que fueron correctamente clasificados.
6. Proporción de casos positivos que lo son realmente de entre todos los identificados como positivos.
7. Tasa de Verdaderos Negativos.
8. Tasa de Verdaderos Positivos.
9. Precisión.

Los parámetros definidos antes se calculan de la siguiente manera:

$$\text{Precisión} = \frac{(1) + (5)}{(1) + (2) + (4) + (5)}$$

$$\text{Tasa de Verdaderos Positivos} = \frac{(5)}{(2) + (5)}$$

$$\text{Tasa de Verdaderos Negativos} = \frac{(1)}{(1) + (4)}$$

$$\text{Tasa de Falsos Positivos} = \frac{(4)}{(1) + (4)}$$

$$\text{Tasa de Falsos Negativos} = \frac{(2)}{(2) + (5)}$$

$$\text{Exactitud} = \frac{(5)}{(4) + (5)}$$

Una vez conocidos los conceptos relacionados con las matrices de confusión, se van a analizar las matrices de confusión obtenidas del clasificador.

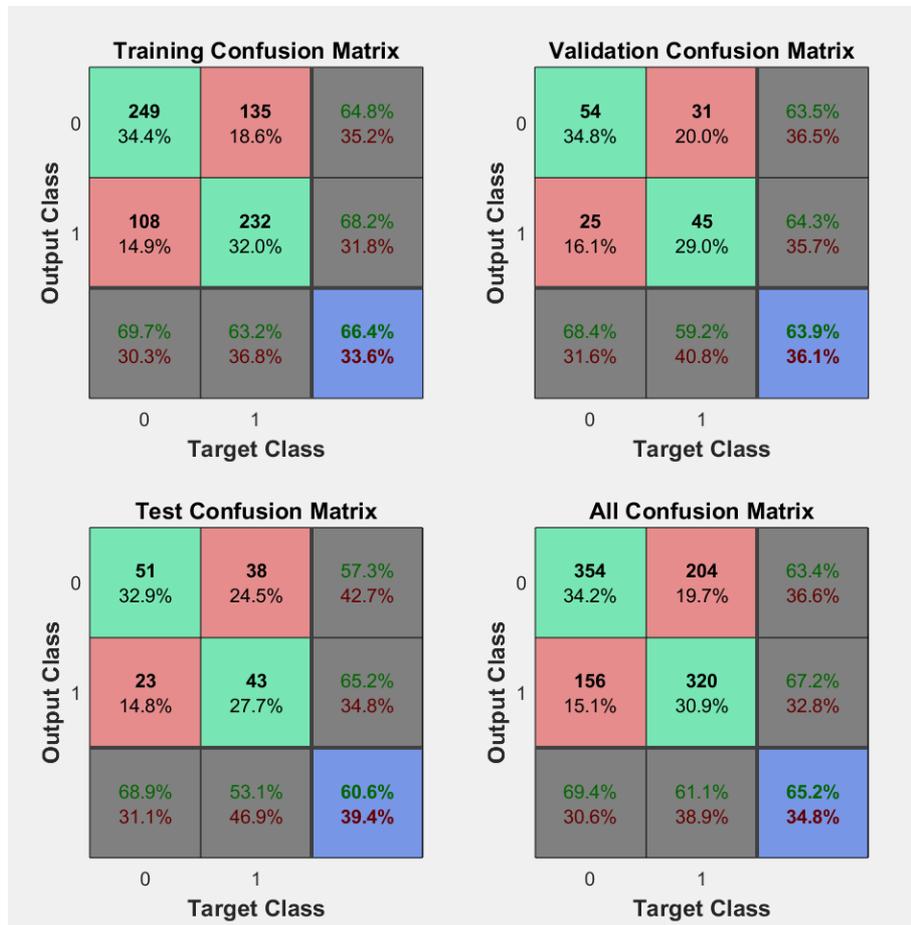


Ilustración 8-10: Matrices de confusión de la red neuronal con 10 neuronas en la capa oculta

La matriz de confusión más interesante va a ser la situada abajo a la izquierda en la imagen, ya que esta es la matriz de confusión general y el resto son las matrices de confusión para cada uno de los sets.

Para empezar, mirando el primer y el cuarto cuadrante, se puede ver que de las 500 muestras de clase 0 que había 354 fueron clasificadas de manera correcta, mientras que 156 fueron clasificadas de manera incorrecta. Con estos datos se obtiene la Tasa de Verdaderos Negativos y la Tasa de Falsos Positivos que se muestran en el séptimo cuadrante, 69,4% y 30,6% consecutivamente. Estos valores podrían calcularse también con las formulas mostradas anteriormente:

$$Tasa\ de\ Verdaderos\ Negativos = \frac{(1)}{(1) + (4)} = \frac{354}{354 + 156} = 0,694$$

$$Tasa\ de\ Falsos\ Positivos = \frac{(4)}{(1) + (4)} = \frac{156}{354 + 156} = 0,3058$$

Para continuar, mirando el segundo y quinto cuadrante, se pudo ver que de las 524 muestras de clase 1 que había 320 fueron clasificadas de manera correcta, mientras que 204 fueron clasificadas de manera incorrecta.

Con estos datos se obtiene la Tasa de Verdaderos Positivos y la Tasa de Falsos Negativos que se se muestra en el octavo cuadrante, 61,1% y 38,9% consecutivamente. Estos valores podrían calcularse también con las formulas mostradas anteriormente:

$$Tasa\ de\ Verdaderos\ Positivos = \frac{(5)}{(2) + (5)} = \frac{320}{204 + 320} = 0,6106$$

$$Tasa\ de\ Falsos\ Negativos = \frac{(2)}{(2) + (5)} = \frac{204}{204 + 320} = 0,389$$

Una vez conocidas estas tasas, si uno se centra en la primera fila de la matriz, se puede ver que de 558 muestras identificadas como clase 0, solo 354 lo eran realmente. Como resultado, se obtiene la proporción de casos negativos que lo son realmente entre todos los casos identificados como negativos en el tercer cuadrante. Como se puede ver, el 63,4% de las muestras identificadas como clase 0 realmente eran de dicha clase, mientras que el 36,6% de las muestras identificadas como clase 0 eran de clase 1.

Si en lugar de en la primera uno se fija en la segunda fila, se puede ver que de 476 muestras identificadas como clase 1, solo 320 lo eran realmente. Como resultado, se obtiene la proporción de casos positivos que lo son realmente entre todos los casos identificados como positivos en el sexto cuadrante. Como se puede ver, el 67,2% de las muestras identificadas como clase 1 realmente eran de dicha clase, mientras que el 32,86% de las muestras identificadas como clase 1 eran de clase 0. La proporción entre los positivos reales identificados y todos los casos positivos, se conoce también como exactitud como se ha mencionado previamente, por lo que podría calcularse de la siguiente manera:

$$Exactitud = \frac{(5)}{(4) + (5)} = \frac{320}{320 + 156} = 0.672$$

Finalmente, quedaría por analizar el noveno cuadrante, en el que se muestra la precisión de nuestro clasificador. En este caso se obtiene una precisión del 65.2%. Este resultado también se podría obtener de la siguiente manera:

$$Precisión = \frac{(1) + (5)}{(1) + (2) + (4) + (5)} = \frac{354 + 320}{354 + 204 + 156 + 320} = 0.6518$$

## 8.6. Mejora de los resultados obtenidos

Existen distintas posibilidades para mejorar los resultados obtenidos por la red neuronal artificial diseñada. En el caso de que se quiera mejorar su precisión puede realizarse cualquiera de los siguientes procedimientos:

- Inicializar la red nuevamente y volverla a entrenar. Esto cambiará los resultados obtenidos por la red debido a que cada vez que esta se inicializa, los pesos y bías iniciales son distintos.
- Incrementar el número de neuronas de la capa oculta. Se utiliza un número mayor de neuronas en la capa oculta ya que esto implica una mayor flexibilidad de la red, es decir, existe un mayor número de parametros que optimizar.
- Implementar la red neuronal con distintas funciones de entrenamiento.

En este caso se ha decidido mejorar los resultados de la red siguiendo el Segundo procedimiento mencionado,

es decir, incrementando el número de neuronas de la capa oculta. Para ver como varían los resultados se han realizado dos redes neuronales artificiales más, siguiendo el mismo procedimiento usado para el diseño, creación y entrenamiento. La primera de las redes se ha diseñado con 30 neuronas en la capa oculta, mientras que la segunda se ha diseñado con 50 neuronas en la capa oculta.

### 8.6.1. 30 neuronas en la capa oculta

Los datos utilizados para realizar el entrenamiento de la red han sido los mismos que en el caso anterior y se ha realizado la división de las muestras de la misma manera, es decir, se han utilizado nuevamente el 70% de las muestras para el set de entrenamiento, el 15% para el set de validación y el 15% para el set de entrenamiento.

A la hora de crear la red, el número de neuronas de la capa de entrada y de la capa de salida va a ser el mismo, lo único que va a cambiar es que, en lugar de 10 neuronas, la capa oculta va a estar formada por 30 neuronas.

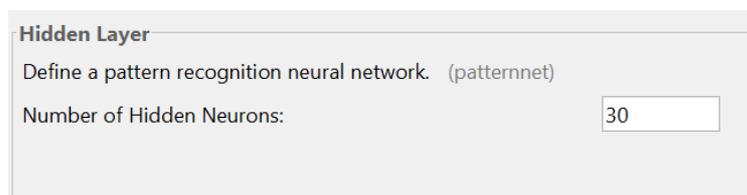


Ilustración 8-11: Selección del número de neuronas en la capa oculta

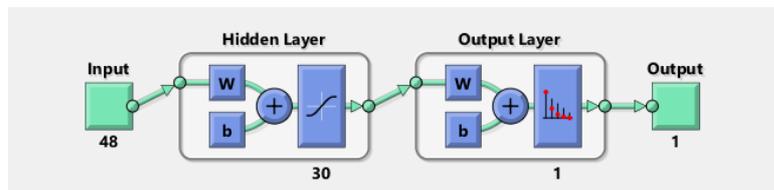


Ilustración 8-12: Arquitectura de la red neuronal con 30 neuronas en la capa oculta

En este caso, una vez entrenada la red se han obtenido los siguientes errores de clasificación:

Results			
	Samples	CE	%E icon"/> %E
Training:	724	6.39306e-1	32.04419e-0
Validation:	155	6.71298e-1	34.19354e-0
Testing:	155	6.88031e-1	38.70967e-0

Ilustración 8-13: Porcentaje de errores de la red neuronal con 30 neuronas en la capa oculta

Como se puede ver, los errores obtenidos en los distintos sets de datos son ligeramente menores que los obtenidos en la red neuronal anterior con 10 neuronas en la capa oculta. Aún y todo, es necesario ver como ha cambiado la matriz de confusión para ver como han variado los resultados.

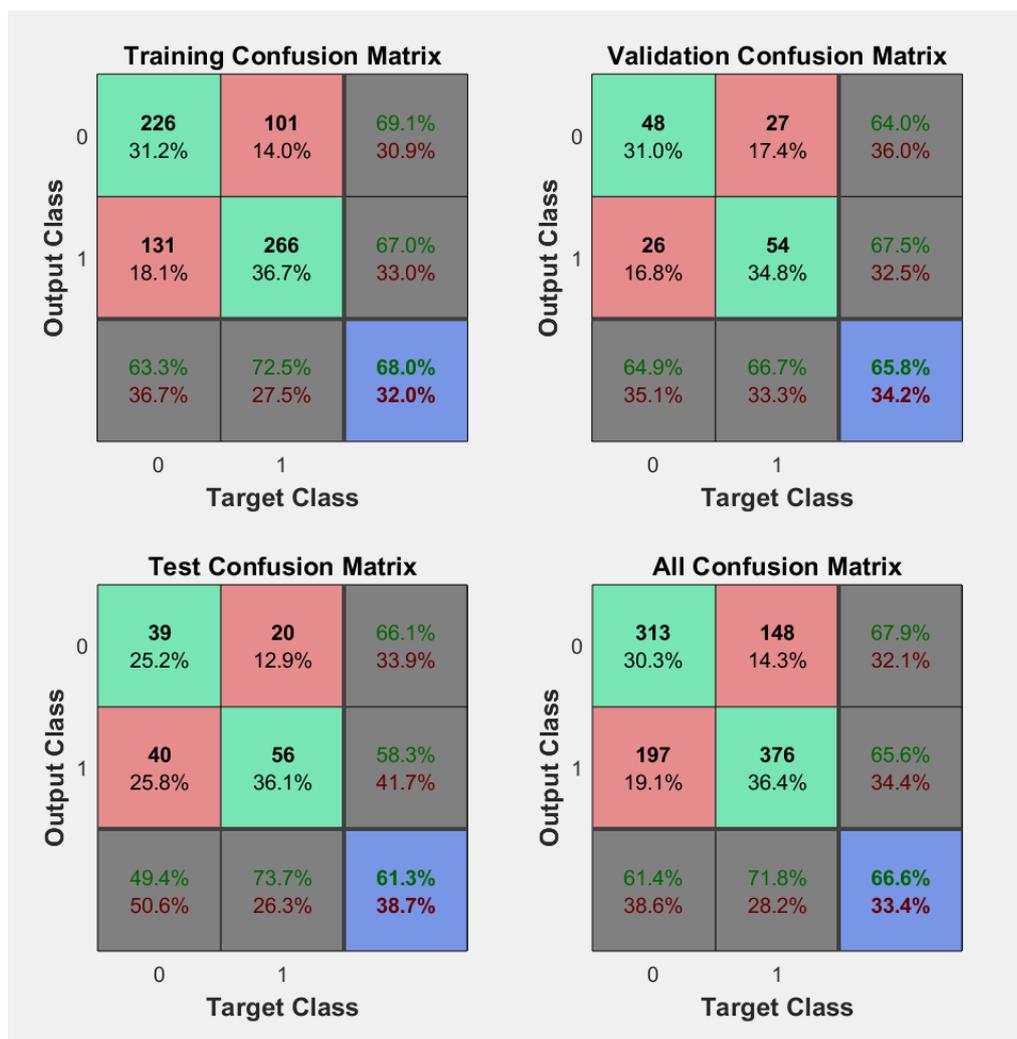


Ilustración 8-14: *Matrices de confusión de la red neuronal con 30 neuronas en la capa oculta*

Nuevamente se comentará la matriz de confusión general, debido a que refleja mejor los resultados de la red neuronal artificial.

En este caso, de las 500 muestras de clase 0 que había 313 fueron clasificadas de manera correcta y 197 de manera incorrecta, debido a esto la Tasa de Verdaderos Negativos con esta nueva red ha disminuido de un 69,4% a un 61,4%, y a su vez la Tasa de Falsos positivos a aumentado de un 30,6% a un 38,6%. Como se puede ver la nueva red neuronal ha identificado de peor manera las muestras de clase 0.

Sin embargo, de las 524 muestras de clase 1 que había, 376 han sido clasificadas de manera correcta y 148 de manera incorrecta. Con estos datos se puede observar que la Tasa de Verdaderos Positivos ha aumentado respecto a la red anterior de un 61,1% a un 71,8% y por lo tanto la Tasa de Falsos Negativos ha disminuido de un 38,9% a un 28,2%. Como se puede ver, esta red neuronal ha identificado mejor las muestras de clase 1.

A pesar de los cambios que ha habido en las predicciones de cada uno de los casos, la precisión de la red con 30 neuronas en la capa intermedia ha mejora ligeramente. Esto se puede apreciar en el novena cuadrante, ya que esta vez la precisión es del 66,6%, es decir, 1,4% mejor que en el caso anterior.

### 8.6.2. 50 neuronas en la capa oculta

Por ultimo se ha diseñado una red neuronal con 50 neuronas en la capa oculta. Nuevamente los datos que se han utilizado para su entrenamiento y la división de estos datos en los distintos subsets han sido los mismos que en los dos casos anteriores. Además el número de neuronas en la capa de entrada y de salida se ha

mantenido constante. La forma de esta red neuronal es la siguiente:

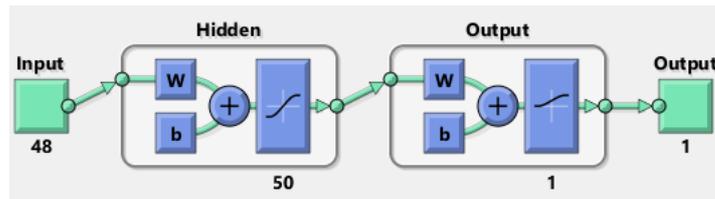


Ilustración 8-15: Estructura de la red neuronal con 50 neuronas en la capa oculta

Los errores que se han obtenido en la clasificación de las muestras en los distintos subsets han sido los siguientes:

Results			
	Samples	CE	%E
Training:	724	6.59014e-1	32.73480e-0
Validation:	155	7.51692e-1	33.54838e-0
Testing:	155	7.46868e-1	33.54838e-0

Ilustración 8-16: Porcentaje de errores de la red neuronal con 50 neuronas en la capa oculta

Se puede comprobar una vez más, que el hecho de incrementar el número de neuronas en la capa oculta mejora los resultados de la red neuronal diseñada, ya que como se ve en la imagen anterior, el porcentaje de errores en la clasificación de las muestras de los distintos sets ha decrementado.

A continuación, se muestran las matrices de confusión que se han obtenido para la red neuronal artificial con 50 neuronas en la capa oculta:

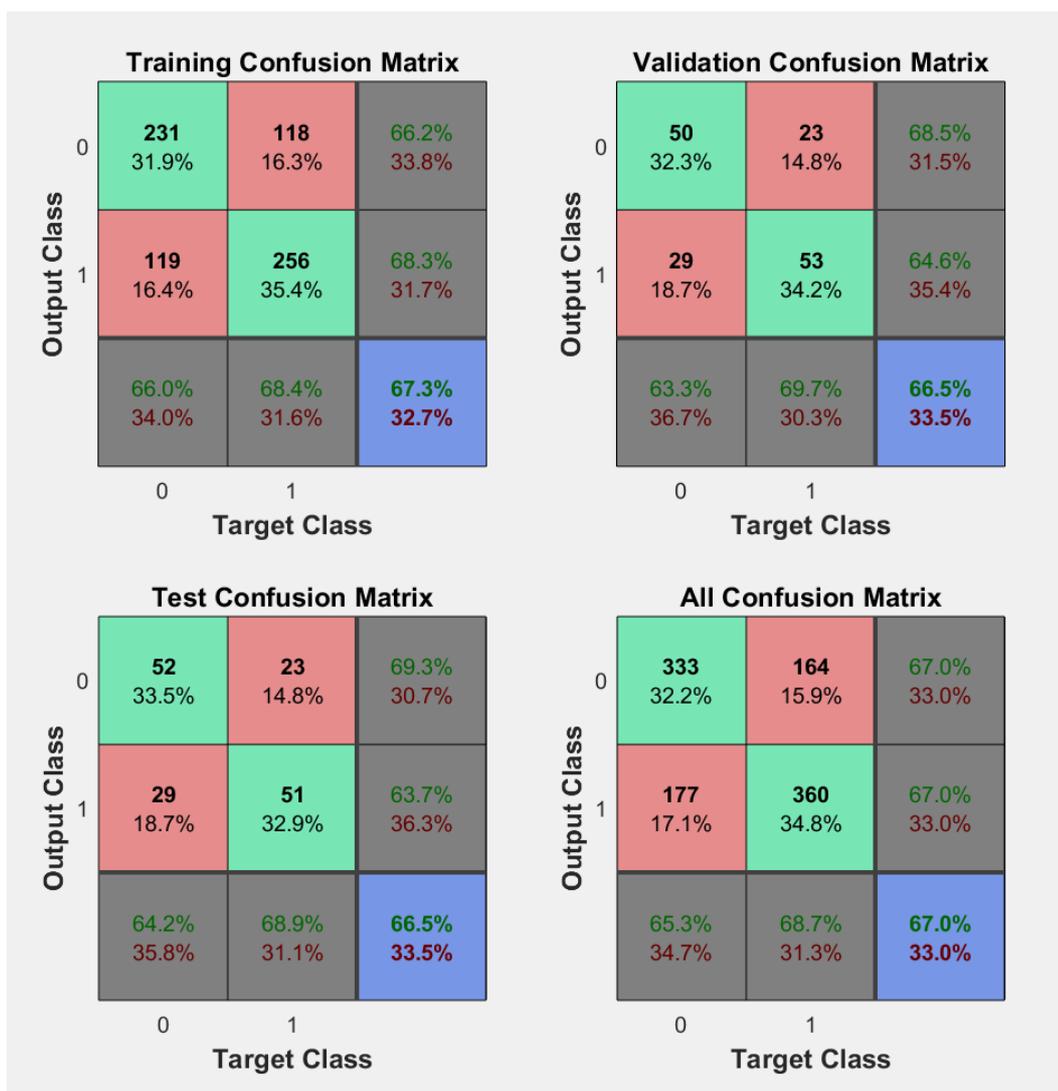


Ilustración 8-17: Matrices de confusión de la red neuronal con 50 neuronas en la capa oculta

Al igual que en los dos casos anteriores se comentará la matriz de confusión general, debido a su relevancia en el funcionamiento general de la red.

En este caso, de las 500 muestras de clase 0 que había 333 fueron clasificadas de manera correcta y 177 de manera incorrecta, debido a esto la Tasa de Verdaderos Negativos con esta nueva red ha disminuido respecto de la red original de un 69,4% a un 65,3%, y a su vez la Tasa de Falsos Positivos a aumentado de un 30,6% a un 34,7%. Como se puede ver la nueva red neuronal ha identificado de peor manera las muestras de clase 0.

Sin embargo, de las 524 muestras de clase 1 que había, 360 han sido clasificadas de manera correcta y 164 de manera incorrecta. Con estos datos se puede observar que la Tasa de Verdaderos Positivos ha aumentado respecto a la red original de un 61,1% a un 68,7% y por lo tanto la Tasa de Falsos Negativos ha disminuido de un 38,9% a un 31,3%. Como se puede ver, esta red neuronal ha identificado mejor las muestras de clase 1.

Como se puede ver en el novena cuadrante, la precisión obtenida con esta última red es la mayor obtenida hasta ahora. La precisión conseguida es de un 67%, es decir, un 1,8% mayor que la de la red original y un 0,4% mayor que la de la red con 30 neuronas en la capa oculta.



# 9 ANÁLISIS DE LOS RESULTADOS Y CONCLUSIONES

En este trabajo se han mostrado distintas herramientas para realizar el análisis de las señales obtenidas mediante un EEG.

Una de las herramientas mostradas es la WPT. Como se ha mencionado anteriormente al utilizar esta herramienta para el análisis, cada vez que se aplica un filtro a la señal se produce una reducción del factor de muestreo de 2, lo que conlleva a reducir el número de muestras de la señal.

A pesar de haber obtenido las distintas ondas Alfa, Beta, Delta, Gamma y Theta por separado utilizando la WPT, no se han podido calcular las características deseadas debido a la insuficiencia de muestras de cada una de las ondas. Convirtiendo así la WPT en una herramienta inadecuada para el análisis de las señales disponibles.

Por otra parte, al utilizar distintos filtros de paso banda, se han podido obtener las ondas Alfa y Beta a partir de las señales originales sin disminuir el número de muestras, siendo así estos filtros una herramienta adecuada para las señales disponibles.

Tras esto, se ha llegado a la conclusión de que, aun existiendo distintas herramientas para realizar el análisis de las señales obtenidas de un EEG, no siempre es adecuado o posible el uso de todas ellas para cada una de las señales.

Debido a que solo ha sido posible realizar la extracción de características de manera satisfactoria haciendo uso de los filtros de paso banda, se ha llevado a cabo la creación de las distintas redes neuronales a partir de las características de las ondas obtenidas de estos filtros y no a partir de las ondas obtenidas a través de la WPT.

Finalmente, se han realizado tres redes neuronales distintas, cada una con un número distinto de neuronas en la capa oculta. Como se puede observar tanto en los porcentajes de los errores cometidos en cada uno de los sets como en cada una de las matrices de confusión de las redes neuronales diseñadas, a medida que se incrementa el número de neuronas de la capa oculta, se obtiene una mayor precisión en la clasificación.

Cabe destacar, que, a la hora de realizar la media de los resultados de las distintas redes neuronales, se han obtenido menores variaciones en los porcentajes de los errores de las redes con mayor número de neuronas en la capa oculta, es decir, no había tanta diferencia entre los resultados de las distintas redes sobre los que se ha hecho la media.

A continuación se muestra una tabla con los resultados obtenidos en función del número de neuronas en la capa oculta.

Tabla 9-1: *Resultados obtenidos de las distintas redes neuronales*

Resultados	10 neuronas	30 neuronas	50 neuronas
Error en el set de entrenamiento (%)	33.56	32.04	32.73
Error en el set de validación (%)	36.12	34.19	33.54
Error en el set de test (%)	39.35	38.7	33.54
Precisión (%)	65.2	66.6	67
Exactitud (%)	67.2	65.6	67
Tasa de Verdaderos Positivos (%)	61.1	71.8	68.7

Tasa de Verdaderos Negativos (%)	69.4	61.4	65.3
Tasa de Falsos Negativos (%)	38.9	28.2	31.3
Tasa de Falsos Positiivos (%)	30.6	38.6	34.7

---

## 9.1. Código Transformada Wavelet

El código mostrado a continuación, es el creado para realizar la WPT de señales obtenidas de un EEG. Se introduce por teclado la onda cerebral que se quiere visualizar y el programa la muestra por pantalla.

```
% TRANSFORMADA WPT
clc; clear all;
load('Prueba1.mat') % Se carga el archivo en el que está la señal.
original=Prueba; % Se guarda en original la señal a la que se le quiere
aplicar la transformada

% Muestra por pantalla las siguientes frases para poder seleccionar la
% señal a visualizar
fprintf('Introduzca el número de la señal que quieres ver \n');
fprintf('1. Onda Delta (0.5-3.6 Hz)\n');
fprintf('2. Onda Theta 1 (3.6-6.718 Hz)\n');
fprintf('3. Onda Theta 2 (6.718-8.273 Hz)\n');
fprintf('4. Onda Alfa 1 (8.273-9.828 Hz)\n');
fprintf('5. Onda Alfa 2 (9.828-12.938 Hz)\n');
fprintf('6. Onda Beta 1 (12.938-25.375 Hz)\n');
fprintf('7. Onda Beta 2 (25.375-31.60 Hz)\n');
fprintf('8. Onda Beta 3 (31.60-33.15 Hz)\n');
fprintf('9. Onda Gamma 1 (33.15-34.70 Hz)\n');
fprintf('10. Onda Gamma 2 (34.70-37.8125 Hz)\n');
fprintf('11. Onda Gamma 3 (37.8125-50.25 Hz)\n');
fprintf('12. Onda Gamma 4 (50.25-100 Hz)\n');
dato = input(''); %Se pide al usuario que introduzca el dato

% Los siguientes datos de nivel y coeficiente se obtienen del arbol de
% decomposición sabiendo los filtros que se aplica a la señal original

if (dato==1) % Si se quiere ver Delta 1 LLLLL
    nivel=5; coef=0;
end

if (dato==2) % Si se quiere ver Theta 1 LLLLLH
    nivel=5; coef=1;
end

if (dato==3) % Si se quiere ver Theta 2 LLLHLL
    nivel=6; coef=4;
end

if (dato==4) % Si se quiere ver Alfa 1 LLLHLH
    nivel=6; coef=5;
end

if (dato==5) %Si se quiere ver Alfa 2 LLLHH
    nivel=5; coef=3;
end

if (dato==6) % Si se quiere ver Beta 1 LLH
    nivel=3; coef=1;
end

if (dato==7) % Si se quiere ver Beta 2 LHLL
    nivel=4; coef=4;
```

```

end

if (dato==8) % Si se quiere ver Beta 3 LHLHLL
    nivel=6; coef=20;
end

if (dato==9) % Si se quiere ver Gamma 1 LHLHLH
    nivel=6; coef=21;
end

if (dato==10) % Si se quiere ver Gamma 2 LHLHH
    nivel=5; coef=11;
end

if (dato==11) % Si se quiere ver Gamma 3 LHH
    nivel=3; coef=3;
end

if (dato==12) % Si se quiere ver Gamma 4 H
    nivel=1; coef=1;
end

% La siguiente función devuelve el árbol T de la wavelet packet.
% wpt = wpdec(X,N,'wname');
% X es el vector, archivo, N es el nivel de descomposición y wname es el tipo
de wavelet.
wpt = wpdec(Prueba,6,'db4');

% Se representa el árbol del paquete wavelet
plot (wpt)

% Para leer el paquete
wpc = wpccoef(wpt,[nivel,coef]);

% Se representa la señal original arriba y la del nivel escogido abajo
figure(1)
subplot(2,1,1); plot(original); title('Señal original');
subplot(2,1,2); plot(wpc); title('Señal elegida');

```

## 9.2. Código separación de canales y extracción de características

A continuación, se muestra el código creado para realizar la extracción de características a las señales obtenidas del EEG.

```

%% SEPARACIÓN DE CANALES Y EXTRACCIÓN DE CARACTERÍSTICAS
clc; clear all;

% evento -> Variable para contar el numero de eventos de MD e MI
% eventos_totales -> Variable para contar el numero de eventos totales
% Salida -> Aquí es donde se va a guardar si son mov MD o MI
% Si es MD Salida=1 si es MI Salida=0
evento=1;
malos=0;
eventos_totales=1;
[s,h]=sload('A09T.gdf'); % Se carga el archivo

for i=1:length(h.EVENT.TYP) % Se recorren los distintos eventos

```

```

    if (h.EVENT.TYP(i)==770 || h.EVENT.TYP(i)==771 || h.EVENT.TYP(i)==772 ||
h.EVENT.TYP(i)==769) % Si el evento es del tipo movimiento manos, pies o
lengua

        if h.ArtifactSelection(eventos_totales)==0 % Si es un evento libre de
artefactos

            muestra=h.EVENT.POS(i-1);

            if h.EVENT.TYP(i)==770 % Movimiento mano derecha
                Salida(evento)=1; % Se le asigna un 1 a la salida
            end
            if h.EVENT.TYP(i)==769 % Movimiento mano izquierda
                Salida(evento)=0; % Se le asigna un 0 a la salida
            end
            if (h.EVENT.TYP(i)==770 || h.EVENT.TYP(i)==769) % Si es el
movimiento de cualquiera de las dos manos se realiza la separación de las
señales y la extracción de características

cont=0;

% Se separa cada uno de los eventos en dos señales, una correspondiente ala
% referencia y otra correspondiente a la realización del movimiento

% Primero se obtiene la señal mientras se produce el movimiento y un poco
% antes de que este comience
for i=(muestra+700):(muestra+700+511)
    cont=cont+1;
    senal_mov(cont,:)=s(i,:);
end

cont=0;

% Luego se obtiene la señal en el periodo de referencia
for i=(muestra):(muestra+511)
    cont=cont+1;
    senal_ref(cont,:)=s(i,:);
end

% SEPARACIÓN DE CANALES
% Se separa cada una de las señales obtenidas previamente en dos señales,
% cada una de un canal distinto
canal8_mov = senal_mov(:,8); %C3
canal12_mov = senal_mov(:,12); %C4
canal8_ref = senal_ref(:,8); %C3
canal12_ref = senal_ref(:,12); %C4

C3_mov = canal8_mov; %C3
C4_mov = canal12_mov; %C4
C3_ref = canal8_ref; %C3
C4_ref = canal12_ref; %C4

% Se cargan los filtros de paso banda previamente diseñados
load('filtro_alfa_10.mat')
load('filtro_beta_4.mat')

% SE REALIZA EL FILTRADO DE LAS SEÑALES
% Ondas alfa
filtrada_C3_alfa_mov=filter(filtro_alfa_10,C3_mov);
filtrada_C4_alfa_mov=filter(filtro_alfa_10,C4_mov);

```

```

filtrada_C3_alfa_ref=filter(filtro_alfa_10,C3_ref);
filtrada_C4_alfa_ref=filter(filtro_alfa_10,C4_ref);

% Ondas beta
filtrada_C3_beta_mov=filter(filtro_beta_4,C3_mov);
filtrada_C4_beta_mov=filter(filtro_beta_4,C4_mov);
filtrada_C3_beta_ref=filter(filtro_beta_4,C3_ref);
filtrada_C4_beta_ref=filter(filtro_beta_4,C4_ref);

% PRIMERA CARACTERÍSTICA: VALOR MEDIO
% Se inicializan a 0 las sumas de cada una de las señales
suma_beta_C3_ref=0; suma_beta_C3_mov=0;
suma_beta_C4_ref=0; suma_beta_C4_mov=0;
suma_alfa_C3_ref=0; suma_alfa_C3_mov=0;
suma_alfa_C4_ref=0; suma_alfa_C4_mov=0;

L=length(filtrada_C3_beta_ref);

for i=1:L

    % Se realiza la suma de cada una de las componentes de cada señal
    suma_beta_C3_ref = suma_beta_C3_ref + filtrada_C3_beta_ref(i);
    suma_beta_C4_ref = suma_beta_C4_ref + filtrada_C4_beta_ref(i);
    suma_beta_C3_mov = suma_beta_C3_mov + filtrada_C3_beta_mov(i);
    suma_beta_C4_mov = suma_beta_C4_mov + filtrada_C4_beta_mov(i);

    suma_alfa_C3_ref = suma_alfa_C3_ref + filtrada_C3_alfa_ref(i);
    suma_alfa_C4_ref = suma_alfa_C4_ref + filtrada_C4_alfa_ref(i);
    suma_alfa_C3_mov = suma_alfa_C3_mov + filtrada_C3_alfa_mov(i);
    suma_alfa_C4_mov = suma_alfa_C4_mov + filtrada_C4_alfa_mov(i);

end

% Se realiza la media de cada una de las señales
media_beta_C3_ref = (1/L)*suma_beta_C3_ref;
media_beta_C3_mov = (1/L)*suma_beta_C3_mov;
media_beta_C4_ref = (1/L)*suma_beta_C4_ref;
media_beta_C4_mov = (1/L)*suma_beta_C4_mov;

media_alfa_C3_ref = (1/L)*suma_alfa_C3_ref;
media_alfa_C3_mov = (1/L)*suma_alfa_C3_mov;
media_alfa_C4_ref = (1/L)*suma_alfa_C4_ref;
media_alfa_C4_mov = (1/L)*suma_alfa_C4_mov;

% CARACTERÍSTICA 2: DESVIACIÓN TÍPICA
suma_2_beta_C3_ref=0; suma_2_beta_C3_mov=0;
suma_2_beta_C4_ref=0; suma_2_beta_C4_mov=0;
suma_2_alfa_C3_ref=0; suma_2_alfa_C3_mov=0;
suma_2_alfa_C4_ref=0; suma_2_alfa_C4_mov=0;

for i=1:L

    suma_2_beta_C3_ref = suma_2_beta_C3_ref + (filtrada_C3_beta_ref(i)-
media_beta_C3_ref)^2;
    suma_2_beta_C4_ref = suma_2_beta_C4_ref + (filtrada_C4_beta_ref(i)-
media_beta_C4_ref)^2;
    suma_2_beta_C3_mov = suma_2_beta_C3_mov + (filtrada_C3_beta_mov(i)-
media_beta_C3_mov)^2;

```

```

    suma_2_beta_C4_mov = suma_2_beta_C4_mov + (filtrada_C4_beta_mov(i)-
media_beta_C4_mov)^2;

    suma_2_alfa_C3_ref = suma_2_alfa_C3_ref + (filtrada_C3_alfa_ref(i)-
media_alfa_C3_ref)^2;
    suma_2_alfa_C4_ref = suma_2_alfa_C4_ref + (filtrada_C4_alfa_ref(i)-
media_alfa_C4_ref)^2;
    suma_2_alfa_C3_mov = suma_2_alfa_C3_mov + (filtrada_C3_alfa_mov(i)-
media_alfa_C3_mov)^2;
    suma_2_alfa_C4_mov = suma_2_alfa_C4_mov + (filtrada_C4_alfa_mov(i)-
media_alfa_C4_mov)^2;

end

desviacion_beta_C3_ref = sqrt(double((suma_2_beta_C3_ref/(L-1))));
desviacion_beta_C4_ref = sqrt(double((suma_2_beta_C4_ref/(L-1))));
desviacion_beta_C3_mov = sqrt(double((suma_2_beta_C3_mov/(L-1))));
desviacion_beta_C4_mov = sqrt(double((suma_2_beta_C4_mov/(L-1))));

desviacion_alfa_C3_ref = sqrt(double((suma_2_alfa_C3_ref/(L-1))));
desviacion_alfa_C4_ref = sqrt(double((suma_2_alfa_C4_ref/(L-1))));
desviacion_alfa_C3_mov = sqrt(double((suma_2_alfa_C3_mov/(L-1))));
desviacion_alfa_C4_mov = sqrt(double((suma_2_alfa_C4_mov/(L-1))));

% A partir de la desviación típica se puede calcular la varianza
varianza_beta_C3_ref=(desviacion_beta_C3_ref)^2;
varianza_beta_C4_ref=(desviacion_beta_C4_ref)^2;
varianza_beta_C3_mov=(desviacion_beta_C3_mov)^2;
varianza_beta_C4_mov=(desviacion_beta_C4_mov)^2;

varianza_alfa_C3_ref=(desviacion_beta_C3_ref)^2;
varianza_alfa_C4_ref=(desviacion_beta_C4_ref)^2;
varianza_alfa_C3_mov=(desviacion_beta_C3_mov)^2;
varianza_alfa_C4_mov=(desviacion_beta_C4_mov)^2;

% CARACTERÍSTICA 3: VALOR MÁXIMO Y MÍNIMO
max_beta_C3_ref=0; max_beta_C3_mov=0;
max_beta_C4_ref=0; max_beta_C4_mov=0;
max_alfa_C3_ref=0; max_alfa_C3_mov=0;
max_alfa_C4_ref=0; max_alfa_C4_mov=0;

min_beta_C3_ref=inf; min_beta_C3_mov=inf;
min_beta_C4_ref=inf; min_beta_C4_mov=inf;
min_alfa_C3_ref=inf; min_alfa_C3_mov=inf;
min_alfa_C4_ref=inf; min_alfa_C4_mov=inf;

for i=1:L

    % Primero se calculan los mínimos de las ondas beta
    if (filtrada_C3_beta_ref(i)<min_beta_C3_ref)
        min_beta_C3_ref=filtrada_C3_beta_ref(i);
    end
    if (filtrada_C4_beta_ref(i)<min_beta_C4_ref)
        min_beta_C4_ref=filtrada_C4_beta_ref(i);
    end
    if (filtrada_C3_beta_mov(i)<min_beta_C3_mov)
        min_beta_C3_mov=filtrada_C3_beta_mov(i);
    end
    if (filtrada_C4_beta_mov(i)<min_beta_C4_mov)
        min_beta_C4_mov=filtrada_C4_beta_mov(i);
    end
end

```

```

% Se calculan los mínimos de las ondas alfa
if (filtrada_C3_alfa_ref(i)<min_alfa_C3_ref)
    min_alfa_C3_ref=filtrada_C3_alfa_ref(i);
end
if (filtrada_C4_alfa_ref(i)<min_alfa_C4_ref)
    min_alfa_C4_ref=filtrada_C4_alfa_ref(i);
end
if (filtrada_C3_alfa_mov(i)<min_alfa_C3_mov)
    min_alfa_C3_mov=filtrada_C3_alfa_mov(i);
end
if (filtrada_C4_alfa_mov(i)<min_alfa_C4_mov)
    min_alfa_C4_mov=filtrada_C4_alfa_mov(i);
end

% Se calculan los máximos de las ondas beta
if (filtrada_C3_beta_ref(i)>max_beta_C3_ref)
    max_beta_C3_ref=filtrada_C3_beta_ref(i);
end
if (filtrada_C4_beta_ref(i)>max_beta_C4_ref)
    max_beta_C4_ref=filtrada_C4_beta_ref(i);
end
if (filtrada_C3_beta_mov(i)>max_beta_C3_mov)
    max_beta_C3_mov=filtrada_C3_beta_mov(i);
end
if (filtrada_C4_beta_mov(i)>max_beta_C4_mov)
    max_beta_C4_mov=filtrada_C4_beta_mov(i);
end

% Se calculan los máximos de las ondas alfa
if (filtrada_C3_alfa_ref(i)>max_alfa_C3_ref)
    max_alfa_C3_ref=filtrada_C3_alfa_ref(i);
end
if (filtrada_C4_alfa_ref(i)>max_alfa_C4_ref)
    max_alfa_C4_ref=filtrada_C4_alfa_ref(i);
end
if (filtrada_C3_alfa_mov(i)>max_alfa_C3_mov)
    max_alfa_C3_mov=filtrada_C3_alfa_mov(i);
end
if (filtrada_C4_alfa_mov(i)>max_alfa_C4_mov)
    max_alfa_C4_mov=filtrada_C4_alfa_mov(i);
end

end

% CARACTERÍSTICA 4: PSD

Fs = 250; % Frecuencia de muestreo (en Hz)
T = 1/Fs; % Periodo de muestreo
t = (0:511)*T; % Vector de tiempo
nfft=1024; % El numero de puntos de la fft

x_filtrada_C3_alfa_mov=filtrada_C3_alfa_mov;
Y_filtrada_C3_alfa_mov=fft(x_filtrada_C3_alfa_mov,nfft); % Tomar la FFT, y
llenando con ceros, de manera que el largo de la FFT sea nfft
Y_filtrada_C3_alfa_mov = Y_filtrada_C3_alfa_mov(1:nfft/2); % La FFT es
simétrica, así que se tira la mitad
my_filtrada_C3_alfa_mov = abs(Y_filtrada_C3_alfa_mov).^2; % Tomar la potencia
espectral,módulo al cuadrado de la FFT

x_filtrada_C4_alfa_mov=filtrada_C4_alfa_mov;

```

```

Y_filtrada_C4_alfa_mov=fft(x_filtrada_C4_alfa_mov,nfft);
Y_filtrada_C4_alfa_mov = Y_filtrada_C4_alfa_mov(1:nfft/2);
my_filtrada_C4_alfa_mov = abs(Y_filtrada_C4_alfa_mov).^2;

x_filtrada_C3_alfa_ref=filtrada_C3_alfa_ref;
Y_filtrada_C3_alfa_ref=fft(x_filtrada_C3_alfa_ref,nfft);
Y_filtrada_C3_alfa_ref = Y_filtrada_C3_alfa_ref(1:nfft/2);
my_filtrada_C3_alfa_ref = abs(Y_filtrada_C3_alfa_ref).^2;

x_filtrada_C4_alfa_ref=filtrada_C4_alfa_ref;
Y_filtrada_C4_alfa_ref=fft(x_filtrada_C4_alfa_ref,nfft);
Y_filtrada_C4_alfa_ref = Y_filtrada_C4_alfa_ref(1:nfft/2);
my_filtrada_C4_alfa_ref = abs(Y_filtrada_C4_alfa_ref).^2;

x_filtrada_C3_beta_mov=filtrada_C3_beta_mov;
Y_filtrada_C3_beta_mov=fft(x_filtrada_C3_beta_mov,nfft);
Y_filtrada_C3_beta_mov = Y_filtrada_C3_beta_mov(1:nfft/2);
my_filtrada_C3_beta_mov = abs(Y_filtrada_C3_beta_mov).^2;

x_filtrada_C4_beta_mov=filtrada_C4_beta_mov;
Y_filtrada_C4_beta_mov=fft(x_filtrada_C4_beta_mov,nfft);
Y_filtrada_C4_beta_mov = Y_filtrada_C4_beta_mov(1:nfft/2);
my_filtrada_C4_beta_mov = abs(Y_filtrada_C4_beta_mov).^2;

x_filtrada_C3_beta_ref=filtrada_C3_beta_ref;
Y_filtrada_C3_beta_ref=fft(x_filtrada_C3_beta_ref,nfft);
Y_filtrada_C3_beta_ref = Y_filtrada_C3_beta_ref(1:nfft/2);
my_filtrada_C3_beta_ref = abs(Y_filtrada_C3_beta_ref).^2;

x_filtrada_C4_beta_ref=filtrada_C4_beta_ref;
Y_filtrada_C4_beta_ref=fft(x_filtrada_C4_beta_ref,nfft);
Y_filtrada_C4_beta_ref = Y_filtrada_C4_beta_ref(1:nfft/2);
my_filtrada_C4_beta_ref = abs(Y_filtrada_C4_beta_ref).^2;

f= (0:nfft/2-1)*Fs/nfft; % Construccion del vector de frecuencias

% Interesa la media de la PSD
sumaPSD_alfa_C3_ref=0;
sumaPSD_alfa_C3_mov=0;
sumaPSD_alfa_C4_ref=0;
sumaPSD_alfa_C4_mov=0;

sumaPSD_beta_C3_ref=0;
sumaPSD_beta_C3_mov=0;
sumaPSD_beta_C4_ref=0;
sumaPSD_beta_C4_mov=0;

for i=1:512

sumaPSD_alfa_C3_ref=sumaPSD_alfa_C3_ref+my_filtrada_C3_alfa_mov(i);
sumaPSD_alfa_C3_mov=sumaPSD_alfa_C3_mov+my_filtrada_C3_alfa_ref(i);
sumaPSD_alfa_C4_ref=sumaPSD_alfa_C4_ref+my_filtrada_C4_alfa_mov(i);
sumaPSD_alfa_C4_mov=sumaPSD_alfa_C4_mov+my_filtrada_C4_alfa_ref(i);

sumaPSD_beta_C3_ref=sumaPSD_beta_C3_ref+my_filtrada_C3_beta_mov(i);
sumaPSD_beta_C3_mov=sumaPSD_beta_C3_mov+my_filtrada_C3_beta_ref(i);
sumaPSD_beta_C4_ref=sumaPSD_beta_C4_ref+my_filtrada_C4_beta_mov(i);
sumaPSD_beta_C4_mov=sumaPSD_beta_C4_mov+my_filtrada_C4_beta_ref(i);

end

Media_PDS_alfa_C3_ref=(1/512)*sumaPSD_alfa_C3_ref;

```

```

Media_PDS_alfa_C3_mov=(1/512)*sumaPSD_alfa_C3_mov;
Media_PDS_alfa_C4_ref=(1/512)*sumaPSD_alfa_C4_ref;
Media_PDS_alfa_C4_mov=(1/512)*sumaPSD_alfa_C4_mov;

Media_PDS_beta_C3_ref=(1/512)*sumaPSD_beta_C3_ref;
Media_PDS_beta_C3_mov=(1/512)*sumaPSD_beta_C3_mov;
Media_PDS_beta_C4_ref=(1/512)*sumaPSD_beta_C4_ref;
Media_PDS_beta_C4_mov=(1/512)*sumaPSD_beta_C4_mov;

% SE GENERA UNA MATRIZ CON LAS CARACTERÍSTICAS OBTENIDAS
% La columna va a depender de la muestra que sea
% Las filas van a ser las siguientes características:
% 1 -> Valor medio beta C3 ref
% 2 -> Valor medio beta C3 mov
% 3 -> Valor medio beta C4 ref
% 4 -> Valor medio beta C4 mov
% 5 -> Valor medio alfa C3 ref
% 6 -> Valor medio alfa C3 mov
% 7 -> Valor medio alfa C4 ref
% 8 -> Valor medio alfa C4 mov
% 9 -> Desviación típica beta C3 ref
% 10 -> Desviación típica beta C3 mov
% 11 -> Desviación típica beta C4 ref
% 12 -> Desviación típica beta C4 mov
% 13 -> Desviación típica alfa C3 ref
% 14 -> Desviación típica alfa C3 mov
% 15 -> Desviación típica alfa C4 ref
% 16 -> Desviación típica alfa C4 mov
% 17 -> Varianza beta C3 ref
% 18 -> Varianza beta C3 mov
% 19 -> Varianza beta C4 ref
% 20 -> Varianza beta C4 mov
% 21 -> Varianza alfa C3 ref
% 22 -> Varianza alfa C3 mov
% 23 -> Varianza alfa C4 ref
% 24 -> Varianza alfa C4 mov
% 25 -> Valor maximo beta C3 ref
% 26 -> Valor maximo beta C3 mov
% 27 -> Valor maximo beta C4 ref
% 28 -> Valor maximo beta C4 mov
% 29 -> Valor maximo alfa C3 ref
% 30 -> Valor maximo alfa C3 mov
% 31 -> Valor maximo alfa C4 ref
% 32 -> Valor maximo alfa C4 mov
% 33 -> Valor minimo beta C3 ref
% 34 -> Valor minimo beta C3 mov
% 35 -> Valor minimo beta C4 ref
% 36 -> Valor minimo beta C4 mov
% 37 -> Valor minimo alfa C3 ref
% 38 -> Valor minimo alfa C3 mov
% 39 -> Valor minimo alfa C4 ref
% 40 -> Valor minimo alfa C4 mov
% 41 -> Media PSD beta C3 ref
% 42 -> Media PSD beta C3 mov
% 43 -> Media PSD beta C4 ref
% 44 -> Media PSD beta C4 mov
% 45 -> Media PSD alfa C3 ref
% 46 -> Media PSD alfa C3 mov
% 47 -> Media PSD alfa C4 ref
% 48 -> Media PSD alfa C4 mov

```

```

Entrada(1,evento) = media_beta_C3_ref;
Entrada(2,evento) = media_beta_C3_mov;
Entrada(3,evento) = media_beta_C4_ref;
Entrada(4,evento) = media_beta_C4_mov;
Entrada(5,evento) = media_alfa_C3_ref;
Entrada(6,evento) = media_alfa_C3_mov;
Entrada(7,evento) = media_alfa_C4_ref;
Entrada(8,evento) = media_alfa_C4_mov;
Entrada(9,evento) = desviacion_beta_C3_ref;
Entrada(10,evento) = desviacion_beta_C3_mov;
Entrada(11,evento) = desviacion_beta_C4_ref;
Entrada(12,evento) = desviacion_beta_C4_mov;
Entrada(13,evento) = desviacion_alfa_C3_ref;
Entrada(14,evento) = desviacion_alfa_C3_mov;
Entrada(15,evento) = desviacion_alfa_C4_ref;
Entrada(16,evento) = desviacion_alfa_C4_mov;
Entrada(17,evento) = varianza_beta_C3_ref;
Entrada(18,evento) = varianza_beta_C3_mov;
Entrada(19,evento) = varianza_beta_C4_ref;
Entrada(20,evento) = varianza_beta_C4_mov;
Entrada(21,evento) = varianza_alfa_C3_ref;
Entrada(22,evento) = varianza_alfa_C3_mov;
Entrada(23,evento) = varianza_alfa_C4_ref;
Entrada(24,evento) = varianza_alfa_C4_mov;
Entrada(25,evento) = max_beta_C3_ref;
Entrada(26,evento) = max_beta_C3_mov;
Entrada(27,evento) = max_beta_C4_ref;
Entrada(28,evento) = max_beta_C4_mov;
Entrada(29,evento) = max_alfa_C3_ref;
Entrada(30,evento) = max_alfa_C3_mov;
Entrada(31,evento) = max_alfa_C4_ref;
Entrada(32,evento) = max_alfa_C4_mov;
Entrada(33,evento) = min_beta_C3_ref;
Entrada(34,evento) = min_beta_C3_mov;
Entrada(35,evento) = min_beta_C4_ref;
Entrada(36,evento) = min_beta_C4_mov;
Entrada(37,evento) = min_alfa_C3_ref;
Entrada(38,evento) = min_alfa_C3_mov;
Entrada(39,evento) = min_alfa_C4_ref;
Entrada(40,evento) = min_alfa_C4_mov;
Entrada(41,evento) = Media_PDS_alfa_C3_ref;
Entrada(42,evento) = Media_PDS_alfa_C3_mov;
Entrada(43,evento) = Media_PDS_alfa_C4_ref;
Entrada(44,evento) = Media_PDS_alfa_C4_mov;
Entrada(45,evento) = Media_PDS_beta_C3_ref;
Entrada(46,evento) = Media_PDS_beta_C3_mov;
Entrada(47,evento) = Media_PDS_beta_C4_ref;
Entrada(48,evento) = Media_PDS_beta_C4_mov;

evento=evento+1;
    end
end
    if h.ArtifactSelection(eventos_totales)==1
        if (h.EVENT.TYP(i)==770 || h.EVENT.TYP(i)==769)
            malos=malos+1;
        end
    end
    eventos_totales=eventos_totales+1;
end
end

```



# 11 REFERENCIAS

---

- [1] Alma Alanís, Edgar Sánchez, *Redes neuronales, conceptos fundamentales y aplicaciones a control automático.*
- [2] Mark Hudson Beale, Martin T. Hagan, Howard B. Demuth, *Neural Network Toolbox User's Guide.*
- [3] Robi Polikar, *The wavelet tutorial, second edition, part I.*
- [4] Nicholas Chimitt, William Misch, Li Tan, Alain Togbe, Jean Jiang, *Comparative Study of Simple Feature Extraction for Single-Channel EEG Based Classification.*
- [5] Mangalagowri S G, Cyril Prasanna Raj P, *EEG Feature Extraction and Classification using Feed Forward Backpropagation Algorithm for Emotion Detection.*
- [6] Saweta Pattnaik, S. K. Sabut, M. Dash, *DWT-based Feature Extraction and Classification for Motor Imaginary EEG Signals.*
- [7] Weijie Ren, Min Han, Jun Wang, Dan wang, Tieshan Li, *Efficient Feature Extraction Framework for EEG Signals Classification.*
- [8] Sheikh Md. Rabiul Islam, Ahsanullah Sajol, Xu Huang, Keng Liang Ou, *Feature Extraction and Classification of EEG signal for Different Brain Control machine.*
- [9] Syed Khairul Bashar, Mohammed Imamul Hassan Bhuiyan, *Identification of Arm Movements Using Statistical Features from EEG Signals in Wavelet Packet Domain.*
- [10] Marquez L. Alejandro P, Muñoz G. Roberto, *Analysis and Classification of Electroencephalographic Signals (EEG) to Identify Arm Movements.*
- [11] Germán Rodríguez Bermúdez, Pedro J. García Laencina, Domitien Brizion, Joaquín Roca Dorda, *Adquisición, procesamiento y clasificación de señales EEG para el diseño de sistemas BCI basados en imaginación de movimiento.*
- [12] E. Mattar, K. Al-Mutib, *EEG Complex Grasping Patterns Recognition for BMI Robotic and Prothesis Motion Control.*
- [13] Kayvan Najarian, Robert Splinter, *Biomedicas Signal and Image Processing.*
- [14] Teresa Talamillo García, *Manual básico para enfermeros en electroencefalografía.*
- [15] Andrea Noelia Bermúdez Cicchino, *Técnicas de procesamiento de EEG para detección de eventos.* Tesis, Dep. de Electrónica, Universidad Nacional de la Plata, Argentina.
- [16] Efrén L. Lema-Condo, Freddy L. Bueno-Palomeque, Susana E. Castro-Villalobos, Esteban F. Ordoñez-Morales, Luis J. Serpa-Andrade, *Comparison of Eavelet Transform Symlets (2-10) and Daubechies (2-10) for an Electroencephalographic Signal Analysys.*
- [17] L. Ramya Stephygraph, N. Arunkumar, V. Venkatraman, *Wireless Mobile Robot Control through Human Machine Interface using Brain Signals.*
- [18] Cabrera Maria E, Novak Keisha, Foti Daniel, Voyles Richard, Wachs Juan P, *What makes a gesture a gesture? Neural signatures involved in gesture recognition.*

- [19] Sunae Park, Jong-Ho Choi, Hyun-Kyo Jung, *Evaluation of features for electrode location robustness in brain-computer interface (BCI)*.
- [20] Stefan Haufe, Ryota Tomioka, Thorsten Dickhaus, Claudia Sanelli, Benjamin Blankertz, Guido Nolte and Klaus-Robert Müller, *Localization of class-related mu-rhythm desynchronization in motor imagery based Brain-Computer Interface sessions*.
- [21] Payam Aghaei Pour, Tauseef Gulrez, Omar AlZoubi, Gaetano Gargiulo, Rafael A. Calvo, *Brain-Computer Interface: Next Generation Thought Controlled Distributed Video Game Development Platform*.