BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA

INSTITUTO NACIONAL DE ASTROFÍSICA, ÓPTICA Y ELECTRÓNICA

FACULTAD DE CIENCIAS DE LA COMPUTACIÓN

Control de movimiento de objetos a través del uso de electro-encefalogramas y redes neuronales artificiales con equipo de bajo costo.

TESIS

PARA OBTENER EL TITULO DE:

LICENCIADO EN INGENIERÍA EN CIENCIAS DE LA COMPUTACIÓN

PRESENTA:

JESSICA NAYELI LÓPEZ ESPEJEL

ASESOR 1:

DR. ARTURO OLVERA LÓPEZ, BUAP

ASESOR 2:

DRA. MARÍA DEL PILAR GÓMEZ GIL, INAOE

TONANTZINTLA, PUE. **OCTUBRE 2015**

Para mis padres, hermanos, tías, amigos y asesores.

Índice general

Agradecimientos III

Índice de figuras 1

Índice de tablas 3

Capítulo I 5

Introducción 5

Capítulo II 9

Marco teórico y trabajos relacionados 9

2.1 Marco teórico 9

2.1.1 Redes neuronales 9

2.1. 2 Transformada Wavelet 13

2.1.3 Filtros Digitales 16

2.2 Trabajos relacionados 18

Capítulo III 27

Método Propuesto 27

3.1 Descripción general. 27

3.2 Adquisición de la señal. 27

3.3 Extracción de características 34

3.4 Clasificación 37

3.5 Movimiento del robot. 45

Capítulo IV 47

Resultados 47

4.1 Evaluación de resultados 48
4.2 Análisis de resultados 56
Capítulo V 57
Conclusiones y trabajo futuro 57
5.1 Conclusiones 57
5.2 Trabajo futuro 58
Bibliografía 61
Apéndices 65
Apéndice A 65
Matrices de confusión 65
Apéndice B 85
Manual de usuario 85

Agradecimientos

A tres mujeres que han llenado mi vida de alegrías, fortalezas, apoyo en cada una de las etapas de mi vida, y sobretodo han estado conmigo cuando más lo he necesitado: a mi mamá, mi tía Hote y mi tía Mari.

Agradezco a mis hermanos Jocelyn y Joshua por todos los momentos que hemos pasado juntos, y porque son de las personas que más me han enseñado a vivir, y a valorar cada instante.

Agradezco a toda mi familia por haberme acompañado en todos estos años, y en especial a tres personas que estarían felices de presenciar este momento, pero la vida no se los permitió: a mi hermano Iván, a mi abuelita Chucha, y a mi papá.

A Hazziel Pimentel por todos los buenos momentos que hemos pasado y por tu incondicional apoyo.

A mis asesores de tesis, Dra. Pilar Gómez Gil y Dr. Arturo Olvera por el aprendizaje obtenido durante la elaboración de la tesis, la paciencia que me tuvieron, y todo el apoyo que me dieron.

A mis sinodales, Dra. Josefina Guerrero y M.C Consuelo Molina por compartir conmigo sus conocimientos, apoyo en cada una de las actividades escolares, y por ser grandes académicas y personas.

A mis amigos por siempre estar conmigo en las buenas, en las malas, y en las peores, a todos ustedes con mucho cariño.

Índice de figuras

Figura 2.1 Diagrama general de una red neuronal artificial… ……..10

Figura 2.2 Típica red neuronal feed- forward compuesta por tres capas……11

Figura 2.3 Conexión entre la neurona i y la neurona j……..11

Figura 2.4 Ejemplo de una función wavelet……..14

Figura 2.5 Árbol de descomposición wavelet con tres niveles……..16

Figura 3.1 Etapas del sistema BLINK……27

Figura 3.2 Diadema Emotiv- EPOC……...28

Figura 3.3 Diagrama de actividades de la captura de señales……..28

Figura 3.4 Sistema de posicionamiento de electrodos de Emotiv- EPOC……..28

Figura 3.5. Ubicación de los electrodos y puntos de medición para su colocación en el Sistema 10- 20…. ……29

Figura 3.6. Posicionamiento correcto de electrodos en el usuario……..30

Figura 3.7 Diagrama de actividades desde el inicio de la grabación del EEG hasta la conversión del archivo .edf a .csv……31

Figura 3.8 Colores que verifican la recepción inalámbrica de la diadema Emotiv- EPOC……32

Figura 3.9 EmotivTestBench- Sujeto 1……33

Figura 3.10 Un ejemplo de EEG al realizar parpadeo de ojos……36

Figura 3.11 Procedimiento de *Three-folder cross-validation*……..37

Figura 3.12 Dos individuos a los que se les tomaron los EEG……39

Figura 3.13 Ventanas con corrimiento de 32 puntos……40

Figura 3.14 Aplicación del 3- *fold- validation* con n número de variables……43

Figura 3.15. Perro Lego Mindstorms NXT construido para visualizar los experimentos……45

Figura 3.16. Instrucciones que se le envían al robot……..46

Figura 4.1 Eficiencias promedio del “archivoDeEntrenamientoNumeros45.txt” para diferente número de neuronas……48

Figura 4.2 Matrices de confusión del mejor caso. Caso 1- experimento 1……48

Figura 4.3 Eficiencias promedio generadas a partir del archivo de entrenamiento "archivoDeEntrenamientoNumeros45.txt" para 10 iteraciones……..49

Figura 4.4 Matrices de confusión del mejor caso – caso 1- experimento 1.1……..49

Figura 4.5. Eficiencias promedio del "ArchivoComparacion_45x19_E2.txt" para diferente número de neuronas……50

Figura 4.6 Matrices de confusión del mejor caso- caso 1- experimento 2……50

Figura 4.7 Eficiencias a partir del archivo de entrenamiento "ArchivoComparacion_45x19_E2.txt" para 10 iteraciones……51

Figura 4.8 Matrices de confusión del mejor caso– caso 1- experimento 2.1…. ……52

Figura 4.9 Eficiencias a partir del archivo de entrenamiento "archivoDeEntrenamientoNumeros585.txt" para 10 iteraciones……53

Figura 4.10 Matrices de confusión del mejor caso – caso 2- experimento 3……54

Figura 4.11 Eficiencias a partir del archivo de entrenamiento "archivoEntrenamiento585x19_E2.txt" para 10 iteraciones……54

Figura 4.12 Matrices de confusión – caso 2- experimento 4……55

Figura A.1 Matrices de confusión para el número de neuronas en las capas internas- caso 1- experimento……..69

Figura A.2 Matrices de confusión – caso 1- experimento 1.1……...71

Figura A.3 Matrices de confusión para el número de neuronas en las capas internas- caso 1- experimento……...75

Figura A.4 Matrices de confusión – caso 1- experimento 2.1……...78

Figura A.5 Matrices de confusión – caso 2- experimento 3……..81

Figura A.6 Matrices de confusión – caso 2- experimento 4……..84

Figura B.1 Ventana inicial del software……..85

Figura B.2. Ventana de BLINK……..85

Figura B.3. Conectando la PC con Lego Mindstorms vía *bluetooth*……86

Figura B.4 Extracción de características……86

Figura B.5 Directorio de ojo derecho……..87

Figura B.6 Guardar nuevo archivo de la extracción de características……87

Figura B.7 Datos de la extracción de características……88

Figura B.8 Red neuronal……88

Figura B.9. Clasificador de comandos……...89

Figura B.10 Robot de Lego Mindstorms NXT……89

Índice de tablas

Tabla 3.1 Área cerebral y nodos que le corresponden (ver Figura 3.5 para el significado de los nodos) ……30

Tabla 3.2 Movimiento del robot de acuerdo al parpadeo…….33

Tabla 3.3. Archivos utilizados en los experimentos…….....41

Capítulo I

Introducción

En México, del total de personas discapacitadas, el 58% es referente a problemas para caminar o moverse. Dicho dato es alarmante, puesto que implica que este sector de la población carece de motricidad en el cuerpo, por lo que dependen directamente de la ayuda brindada por otras personas. La principal dificultad, sin embargo, radica en que su vida es más complicada, debido a que no pueden desplazarse de un lugar a otro, y en muchas ocasiones las expresiones faciales son su único medio de comunicación.

Pensando en las personas cuadripléjicas y con la motivación de ofrecer una alternativa para que se puedan desplazar fácilmente, en este trabajo de tesis se propone realizar un sistema basado en una red neuronal de retro-propagación, para el reconocimiento del movimiento de los párpados, mismos que a cada reacción en específico corresponde una dirección u orden de desplazamiento. Las señales EEG, sensadas por una diadema de bajo costo, como la Emotiv- EPOC (Emotiv I. , 2014), son caracterizadas por el sistema través de una técnica de procesamiento de señales conocida como análisis Wavelet (Burrus Sidney, Gopinath, & Guo, 1998). Las características del EEG sensado se utilizan como entrada a una red neuronal para el reconocimiento de comandos. El sistema se evalúa a través del envío de comandos a un pequeño robot mascota, construido con el sistema Mindstorms NXT (Lego Mindstorms, 2015). Los comandos que pueden reconocerse son tres: adelante, vuelta a la derecha y vuelta a la izquierda.

Debido a que se desea que el sistema sea accesible para todo el público, se ha diseñado para trabajar con dispositivos de sensado de bajo costo. Sin embargo, existen algunas limitaciones en el uso de diademas Emotiv- EPOC para la detección de señales EEG, con respecto a otros dispositivos. Entre las principales limitaciones se encuentra

el que no ofrecen señales de alta calidad, ya que están más orientadas a las aplicaciones de videojuegos. Por esta razón, en esta investigación se propone el mejoramiento de la señal a través del uso de técnicas para la eliminación de ruido. De esta manera, se consigue aprovechar la diadema Emotiv-EPOC, el cual es un dispositivo inalámbrico y cuyo costo es más económico que el de otros dispositivos.

Se realizaron dos casos, y en ambos, los resultados fueron favorables. En el primer caso se realizaron cuatro experimentos, en los que a la información del EEG se le aplicó la extracción de características a 10 segundos de sensado; en el segundo caso se realizaron dos experimentos, haciendo la extracción de características con ventanas entrecortadas de un segundo cada una y un corrimiento de 32 puntos. En ambos casos se aplicó la Transformada Discreta Wavelet (Burrus Sidney, Gopinath, & Guo, 1998). Para el primer caso el promedio de eficiencia en el primer experimento fue 0.624, en el experimento 2 fue de 0.678. En el segundo caso se realizaron dos experimentos, cuya mejor eficiencia resultó de 0.844 La principal aportación de este trabajo es la forma en que se diseñó la base de datos a partir de la toma de EEG y su extracción de características, para después someter los datos a clasificación con el uso de una red neuronal. (Galvão Barbosa, Zacarias Freitas, de Moura Guedes, & Meggiolaro, 2013) obtuvieron como mejor eficiencia de la clasificación de comandos por EEG 0.8875, y (Nazari Kousarrizi, Asadi Ghanbari, Teshnehlab, Aliyari, & Gharaviri, 2009) de igual manera, obtuvieron como mejor resultado 0.8875 de eficiencia en la precisión de clasificación. Comparados con los trabajos de experimentación realizados en este trabajo de investigación, no están muy separados de nuestros resultados hechos con un dispositivo de bajos costo.

Las tareas a futuro que se puede realizar son diversas, por ejemplo: la toma del EEG se puede llevar a cabo con un dispositivo más sofisticado que Emotiv- EPOC; de igual manera, para la extracción de características, sería interesante calcular el vector característico con otras medidas estadísticas, y por qué no, haciendo uso de una, dos, o todas las sub- bandas cerebrales. Aunado a lo anterior, otra actividad que resultaría de gran utilidad es hacer la clasificación en tiempo real, y si se tuviera el equipo necesario, implementarlo en una silla de ruedas real.

El presente escrito se encuentra dividido en 5 capítulos. El capítulo II describe los trabajos relacionados que se han elaborado, así como el marco teórico más importante para entender el resto de documento. En el capítulo III se presenta la descripción del modelo propuesto para la implementación del sistema informático; el capítulo IV muestra los resultados obtenidos en base a diversas pruebas. Finalmente, el capítulo V discute las conclusiones y el trabajo futuro.

Capítulo II

Marco teórico y trabajos relacionados

El área de BCI (*Brain Computer Interface*, en español Interfaz- Humano Computadora) ha sido un campo de investigación muy expandido en las últimas décadas (Galvão Barbosa, Zacarias Freitas, de Moura Guedes, & Meggiolaro, 2013). Este campo de estudio trata sobre las técnicas asociadas a la adquisición de señales cerebrales, para su posterior procesamiento e interpretación, a través de una computadora o un dispositivo inteligente. En seguida se explican algunos conceptos relacionados a esta investigación y posteriormente se describen algunos trabajos interesantes que se relacionan con el tema presentado.

2.1 Marco teórico

En este apartado se describe el marco teórico relacionado con la presente investigación, algunos de los aspectos a tratar son: redes neuronales, transformadas Wavelets, y filtros digitales.

2.1.1 Redes neuronales

Las redes neuronales artificiales (ANN, por sus siglas en inglés *Artificial Neural Networks*) están inspiradas biológicamente; son enfoques que simulan la forma en que el cerebro humano procesa la información. Las ANN reúnen conocimientos mediante la detección de patrones y relaciones entre los datos, y aprenden a través de la experiencia, no de la programación. Una ANN está formada por diversas unidades individuales, las neuronas artificiales o elementos de procesamiento (PE, por sus siglas en inglés), conectadas con coeficientes (pesos), que constituyen la estructura neural y se organizan en capas. El poder de los cálculos neuronales proviene de la conexión de las neuronas en una red. Cada PE tiene entradas ponderadas, una función de transferencia y una salida. El comportamiento de una red neuronal está determinado

por las funciones de transferencia de sus neuronas, por la regla de aprendizaje, y por la propia arquitectura. Los pesos son los parámetros ajustables y, en ese sentido, una red neuronal es un sistema parametrizado. La suma ponderada de las entradas constituye la activación de la neurona. La señal de activación se pasa a través de la señal de transferencia, para producir una única salida de la neurona. La función de transferencia introduce la no linealidad a la red. Durante el entrenamiento, las conexiones entre las unidades están optimizadas hasta que se minimiza el error en las predicciones y la red alcanza el nivel especificado de precisión.

Existen diversos tipos de redes neuronales, pero todas son descritas por las funciones de transferencia de sus neuronas, por la regla de aprendizaje, y por la fórmula de conexión (Agatonovic-Kustrin, 2000). La Figura 2.1 muestra el diagrama general de una red neuronal artificial con sus respectivos componentes.

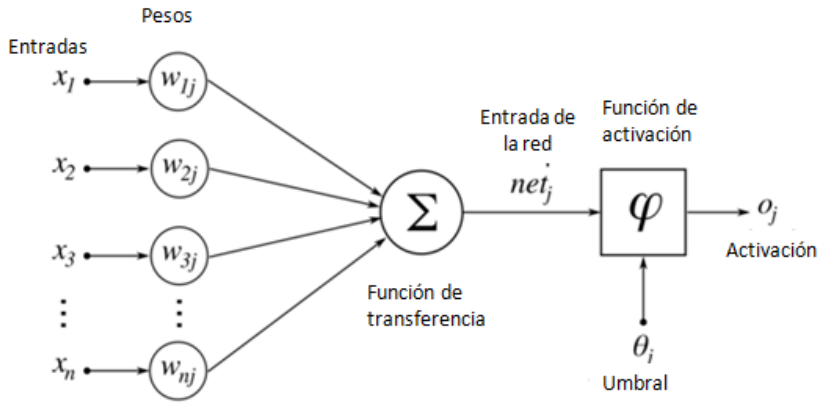


Figura 2.1 Diagrama general de una red neuronal artificial (Agatonovic-Kustrin, 2000).

2.1.1.1 Redes neuronales *feed- forward* multicapa

Las redes neuronales *feed- forward* multicapa (MLF, por sus siglas en inglés *Multi- Layer Feed-forward*) o también conocidas como perceptrones de varios niveles (MLP por sus siglas en inglés, *Multi- Layer perceptrons*), se caracterizan porque las salidas de unas neuronas sirven como entradas a otras, sin existir recurrencia en las conexiones. Existen varios algoritmos para entrenar estas redes, siendo el más popular el algoritmo de retro- propagación. Estás redes son aplicadas en una amplia variedad de problemas de todo tipo (Svozil, KvasniEka, & Pospichal, 1997).

Una red neuronal MLF consiste de neuronas, que están ordenadas en capas (Figura 2.2). La primera capa es llamada capa de entrada; la última capa es llamada la capa de salida, y las capas que se encuentre la primera y la última capa se llaman capas ocultas.

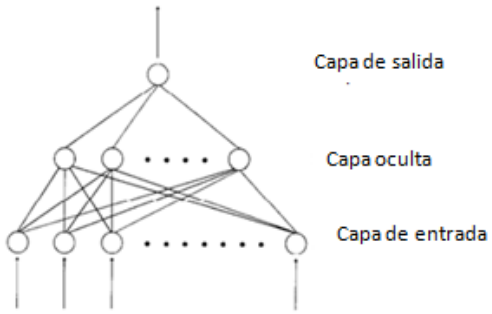


Figura 2.2 Típica red neuronal feed- forward compuesta por tres capas (Svozil, KvasniEka, & Pospichal, 1997).

Para una descripción formal de neuronas se puede usar la denominada función de mapeo Γ, que asigna a cada neurona i un subconjunto $\Gamma(i) \subseteq V$ que consta de todos los antecesores de la neurona dada (Svozil, KvasniEka, & Pospichal, 1997). Un subconjunto $\Gamma^{-1}(i) \subseteq V$ consta de todos los predecesores de la neurona i. Cada neurona en una capa particular está conectada con todas las neuronas en la capa siguiente. La conexión entre la neurona i y la neurona j está caracterizada por el coeficiente de ponderación ω_{ij} y la neurona i por el coeficiente umbral ϑ_i (Figura 2.3). El coeficiente de ponderación refleja el grado de importancia de las conexiones dadas en la red neuronal. El valor de salida (actividad) de la i- ésima neurona x_i está determinada por las ecuaciones (2.1) y (2.2).

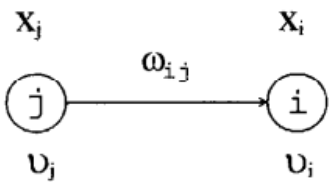


Figura 2.3 Conexión entre la neurona i y la neurona j.

$$x_i = f(\xi_i) \qquad \textbf{2.1}$$

$$\boldsymbol{\xi_i = \vartheta_i + \sum_{j\epsilon\Gamma_i^{-1}} \omega_{ij} x_j} \quad \textbf{2.2}$$

Donde ξ_i es la ponderación de la i-ésima neurona y la función $f(\xi_i)$ es llamada función de transferencia. El coeficiente umbral puede ser entendido como un coeficiente de peso de la conexión con una neurona j formalmente agregada, donde $x_j = 1$ (llamado sesgo).

En muchos casos, para la función de transferencia se utiliza la ecuación 2.3:

$$\boldsymbol{f(\xi) = \frac{1}{1+\exp(-\xi)}} \quad \textbf{2.3}$$

El proceso de adaptación supervisado varía el coeficiente umbral ϑ_i y los coeficientes de peso ω_{ij} para reducir al mínimo la suma de las diferencias al cuadrado entre los valores de salida calculados y requeridos. Esto se logra mediante la minimización de una función objetivo E, expresada como:

$$\boldsymbol{E = \sum_0 \frac{1}{2}(x_0 - \neg x_0)^2} \quad \textbf{2.4}$$

Donde x_0 y $\neg x_0$ son vectores compuestos por el cálculo y salidas requeridas de las neuronas de la capa de salida, y la suma se extiende sobre todas las neuronas de salida 0 (Svozil, KvasniEka, & Pospichal, 1997).

2.1.1.2 Algoritmos de entrenamiento de retro- propagación

El algoritmo de retro- propagación utiliza las ecuaciones 2.5 y 2.6 para modificar de manera incremental los coeficientes de peso y de umbral, respectivamente (Svozil, KvasniEka, & Pospichal, 1997).

$$\boldsymbol{\omega_{ij}^{(k+1)} = \omega_{ij}^{(k)} - \lambda \left(\frac{\partial E}{\partial \omega_{ij}}\right)^{(k)}} \quad \textbf{2.5}$$

$$\boldsymbol{\vartheta_i^{(k+1)} = \vartheta_i^{(k)} - \lambda \left(\frac{\delta E}{\delta \vartheta_i}\right)^{(k)}} \quad \textbf{2.6}$$

Donde $\boldsymbol{\lambda}$ es la tasa de aprendizaje ($\boldsymbol{\lambda > 0}$). El problema principal es el cálculo de las derivadas de $\boldsymbol{\partial E/\partial \omega_{ij}}$ a $\boldsymbol{\delta E/\delta \vartheta_i}$. Esto puede realizarse de varias maneras. Una de las derivaciones más populares para obtener esta derivadas, está basada en gradientes y fue propuesta originalmente por Werbos (Werbos, 1974), y de manera independiente posteriormente con el nombre de retro-propagación por Rumelhart y colaboradores (Rumelhart, Geoffrey, & R.J, 1986). Una descripción detallada y didáctica de la derivación y desglose del algoritmo de retro-propagación puede encontrarse en (Pérez Águila, 2012). Existen librerías que implementan el algoritmo de retropropagación; en este trabajo se utilizó la librería "Neural networks toolbox" de Matlab (MathWorks, 2015).

2.1.1.3 Entrenamiento y generalización

La red neuronal MLF trabaja de dos modos: entrenamiento y estimación. Para el entrenamiento de la red neuronal MLF y para la estimación usando la red neuronal MLF se necesitan dos conjuntos de datos, el conjunto de entrenamiento y el conjunto que queremos estimar (de prueba).

El modo de entrenamiento comienza con valores arbitrarios de los pesos, generalmente éstos son números arbitrarios, y se procesan iterativamente. Cada iteración del conjunto de entrenamiento completo se llama época. En cada época la red ajusta los pesos y reduce el error. A medida que continúa el proceso iterativo de ajuste incremental, los pesos convergen gradualmente al conjunto local óptimo de valores. Muchas épocas suelen ser necesarias antes de llegar a los valores esperados. (Svozil, KvasniEka, & Pospichal, 1997).

2.1. 2 Transformada Wavelet

El término 'wavelet' se refiere a una onda oscilatoria que viaja por el tiempo, y puede ser descrita en el plano tiempo-frecuencia (Adam, 2013). En general, las wavelets se elaboraron a propósito para tener propiedades específicas que las hacen útiles para el procesamiento de señales. Estas funciones representan una herramienta adecuada para el análisis de los fenómenos no estacionarios o transitorios. En la figura 2.4 se muestra un ejemplo de señal wavelet (Adam, 2013).

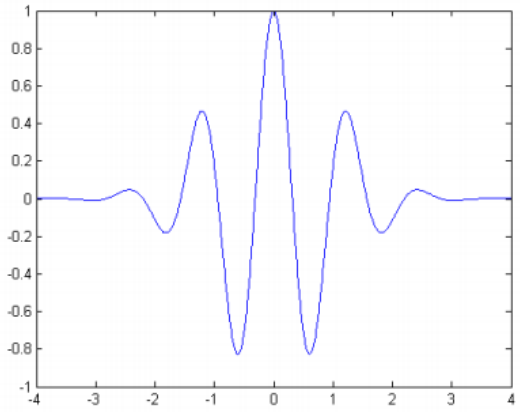


Figura 2.4 Ejemplo de una función wavelet (Adam, 2013).

Matemáticamente, la wavelet ψ, es una función de promedio cero, que tiene la energía concentrada en el tiempo (ecuación 2.7):

$$\int_{-\infty}^{\infty} \boldsymbol{\psi(t)dt = 0} \qquad \textbf{2.7}$$

Con el fin de ser más flexibles en la extracción de la información tiempo-frecuencia, una familia de ondas puede ser construida a partir de una función $\psi(t)$, también conocidas como 'Wavelet Madre', que están confinadas en un intervalo finito. La 'Wavelet Hija', $\psi_{u,s}(t)$ está formada por el factor de traslación u y un parámetro de escala de dilatación representado con s. La ecuación 2.8 describe a la wavelet hija (Adam, 2013).

$$\boldsymbol{\psi_{u,s}(t) = \frac{1}{\sqrt{s}} . \psi\left(\frac{t-u}{s}\right)} \qquad \textbf{2.8}$$

2.1.2.1 ¿Por qué usar wavelets?

La transformada de Fourier, basada en el análisis espectral, es la herramienta analítica dominante para el análisis de dominio de frecuencia. Sin embargo, la transformada de Fourier no puede dar ninguna información sobre los cambios espectrales con respecto al tiempo. La transformada de Fourier asume que la señal es estacionaria, pero una señal no siempre lo es. Para superar esta deficiencia, existe un método que permite representar la señal en tiempo y en frecuencia a través de una

función de ventanas de tiempo (Adam, 2013). La longitud de la ventana determina una constante de tiempo así como la resolución de la frecuencia. Por lo tanto, una ventana de tiempo más pequeña se utiliza para capturar el comportamiento transitorio de una señal, sacrificando la resolución de frecuencia. La naturaleza de algunas señales no es periódica y transitoria, por lo que las señales no pueden ser fácilmente analizadas por transformadas convencionales. Así, las transformadas wavelets son una herramienta matemática alternativa que debe ser seleccionada para extraer información relevante tiempo- amplitud de una señal.

2.1.2.2 Transformada Discreta Wavelet

Se sabe que cualquier función se puede expresar como una descomposición lineal (ecuación 2.9):

$$\boldsymbol{f(t) = \sum_l a_l \varphi_l(t)} \qquad \textbf{2.9}$$

Donde a_l son los coeficientes de expansión (valores reales) y $\varphi_l(t)$ es un conjunto de funciones reales, conocidas como el conjunto de expansión (Burrus Sidney, Gopinath, & Guo, 1998).

Usando wavelets como el conjunto de expansión, se puede definir la expansión de una función cualquiera $f(t)$ como se específica en la ecuación 2.10:

$$\boldsymbol{f(t) = \sum_k \sum_j a_{j,k} \varphi_{j,k}(t)} \qquad \textbf{2.10}$$

Donde j y k son índices (valores enteros) y las funciones wavelets $\varphi_{j,k}(t)$ forman un conjunto ortogonal de funciones (base).

Los coeficientes $a_{j,k}$ se conocen como **Transformada Discreta Wavelet** (DTW, por sus siglas en inglés *Discrete Wavelet Transform*).

La ecuación 2.11 corresponde a la transformación inversa discreta wavelet:

$$\boldsymbol{f(t) = \sum_k \sum_j a_{j,k} \varphi_{j,k}(t)} \qquad \textbf{2.11}$$

Esta ecuación se expresa en términos de dos índices: el tiempo de traslación k y el índice de escalamiento j. (Burrus Sidney, Gopinath, & Guo, 1998).

Existen varias maneras de estimar coeficientes de la Transformada Discreta Wavelet, una de ellas consiste en utilizar banco de filtros, en este método, la señal pasa por un filtro de pasa altas y un filtro de pasa bajas. (Burrus Sidney, Gopinath, & Guo, 1998). La salida de cada filtro contiene la mitad de las frecuencias de la señal original, pero el mismo número de elementos; las salidas de ambos filtros se concatenan, lo que genera una nueva señal con el doble de elementos que la original. Para evitar esto, la salida de cada filtro se pasa por un proceso de sub- muestreo. Las salidas sub-muestreadas de los filtros pasa bajas se conocen como *coeficientes de aproximación* y las salidas sub- muestreadas del filtro pasa altas se conocen como *coeficientes de detalle.* Los coeficientes de aproximación se introducen nuevamente a un banco de filtros, y el proceso se repite tantas veces como se desee. La figura 2.5 muestra un árbol de descomposición wavelet con tres niveles. (Gómez Gil, 2014)

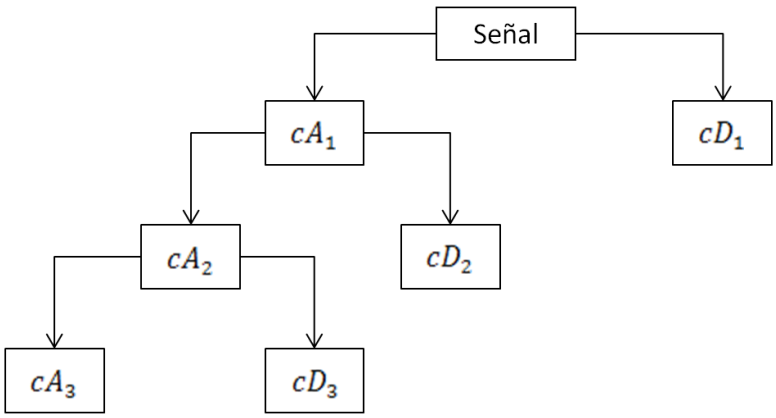


Figura 2.5 Árbol de descomposición wavelet con tres niveles (Burrus Sidney, Gopinath, & Guo, 1998).

2.1.3 Filtros Digitales

Los filtros digitales son utilizados para dos propósitos generales (Tan, 2008):

1. Separación de señales que han sido combinadas.

La separación de señales es necesaria cuando una señal ha sido contaminada con ruido, interferencia u otras señales. Por ejemplo, en la medición de la actividad eléctrica del corazón de un bebé cuando aún está en el vientre materno es necesaria la separación de señales, debido a que la señal será corrompida por la respiración del producto y de la madre.

2. Restauración de señales que se han distorsionado.
   Por ejemplo, una grabación con equipo deficiente puede ser filtrada para representar mejor el sonido.

Los filtros analógicos pueden ser usados para estas mismas tareas, sin embargo, los filtros digitales generalmente logran resultados muy superiores. (Tan, 2008)

2.1.3.1 Filtros IIR

Los filtros de respuesta infinita al impulso o IIR (por sus siglas en inglés, *Infinite Impulse Response*), son la primera opción cuando la velocidad es extrema, y la fase de no linealidad es aceptable (Tan, 2008). Los filtros IIR son computacionalmente más eficientes que los filtros FIR, ya que requieren un menor número de coeficientes debido al hecho de que utilizan retroalimentación o polos. Sin embargo, la retroalimentación puede resultar ser inestable en el filtro, si los coeficientes se desvían de sus valores verdaderos. La ecuación general para los filtros IIR puede ser expresada como se describe en la ecuación 2.12. (Tan, 2008).

$$\boldsymbol{H(z) = \frac{b_0 + b_1 z^{-1} + \cdots + b_N Z^{-N}}{1 + a_1 z^{-1} + \cdots + a_M Z^{-M}} = \frac{\sum_{k=0}^{N} b_k z^{-k}}{1 + \sum_{k=1}^{M} a_k Z^{-k}}} \qquad \mathbf{2.12}$$

Donde a_k y b_k son coeficientes del filtro.

Existe una gran cantidad de filtros. Un tipo de filtro IIR que a continuación se describe es el filtro Chevyshev Tipo II, el cual presenta buenos resultados en aplicaciones asociadas al procesamiento de electro encefalogramas (Juárez Guerra, Alarcon Aquino, & Gómez Gil, Epilepsy Seizure Detection in EEG Signals Using Wavelet Transforms and Neural Networks, 2014).

2.2 Trabajos relacionados

Enseguida se presentan algunos trabajos reportados recientemente, que son similares al presentado en este proyecto. Los trabajos descritos aquí se han organizado de acuerdo al área de conocimiento.

2.2. 1 Reconocimiento basado en Interfaces Cerebro-Computadora

Galvão y otros (Galvão Barbosa, Zacarias Freitas, de Moura Guedes, & Meggiolaro, 2013), desarrollaron técnicas para implementar operaciones de cuatro actividades mentales de BCI, para la activación de una silla de ruedas. Estas actividades están basadas en el uso de BCI con actividades mentales intuitivas, tales como el movimiento imaginario del brazo izquierdo, para girar una silla de ruedas a la izquierda, sin la necesidad de imaginar operaciones matemáticas o imaginar movimientos de objetos, como hacen otras aplicaciones. Los autores encontraron que las características relacionadas con la potencia de la señal en las bandas Alfa y Beta, que aparecen al imaginar funciones motoras, representan adecuadamente el comportamiento de las señales de EEG en el dominio de frecuencia- tiempo para estos casos. Los métodos que utilizaron están basados en algoritmos de redes neuronales perceptrón multinivel. Los dos métodos de clasificación que emplearon los autores fueron: el sistema implementado con una multi-red y el sistema de modelo jerárquico, cuyo éxito de clasificación en comandos fue de 88.75% y 91.0%, respectivamente.

En Morales y otros (Morales Flores, Ramírez Cortés, Gómez Gil, & Alarcón, 2012) se aplica una arquitectura basada en sistemas de inferencia difusa (ANFIS), a problemas de salud para tareas de clasificación ejecutadas sobre datos temporales. La información sobre potencia de la señal se obtuvo de las bandas conocidas como "Alfa" y "Beta", que constituyen un descriptor bueno para este tipo de problemas y con adecuada separabilidad, proporcionando un buen equilibrio entre la complejidad del problema y la tasa de clasificación. Los vectores característicos que representan a cada tarea mental después de aplicar un paradigma difuso, proporcionan una buena descripción de la evolución temporal de la señal de potencia. Este trabajo se realizó

utilizando dos redes neuronales artificiales, una tipo Elman y una conocida como FCRNN (*Fully Connected Recurrent Neural Networks,* en español: red neuronal recurrente totalmente conectada). Los autores encontraron que la red FCRNN obtiene un mejor desempeño que la red Elman durante la etapa de entrenamiento (94.61% y 91.75%, respectivamente); sin embargo, la red Elman logra un mejor rendimiento que la red FCRNN en la generalización de la clasificación (90.16% y 88.12%, respectivamente). En ambos casos, se ocupó una red con 10 nodos en la capa oculta.

Nazari K., y otros (Nazari Kousarrizi, Asadi Ghanbari, Teshnehlab, Aliyari, & Gharaviri, 2009) muestran el diseño de una BCI, que convierte EEG en el movimiento de un cursor. El método automatizado propuesto para la clasificación de la actividad EEG, consiste en tres etapas: pre-procesamiento de la señal, extracción de características y clasificación. Para el pre-procesamiento de la señal los autores utilizaron cuatro métodos, con el fin de quitar la perturbación que causan los artefactos: valores extremos, tendencias lineales, improbabilidad de datos y curtosis. En la extracción de características ocuparon el espectrograma de potencia, varianza y media de las wavelet madre Haar. Finalmente, los autores implementaron un MLP de retro- propagación (*MultiLayer Perceptron,* en Español perceptrón multicapa) con una capa oculta de cinco neuronas, una red neuronal probabilística (PNN, *Probabilistic Neural Network*), y un clasificador SVM (*Support Vector Machines*) con núcleo RBF (*Radial Base Function,* función de base radial) Gaussiana. En comparación con la red neuronal, SVM tiene mayor precisión de clasificación en el entrenamiento (99.56% y 100%, respectivamente); sin embargo, la red neuronal es mejor que SVM durante las pruebas (precisión de 88.75% en la red y 82.2% en SVM). Esto se debe a que SVM es un clasificador más general.

2.2. 2 Diadema Emotiv- Epoc

La diadema EPOC es un sistema EEG inalámbrico de 16 electrodos: 14 canales para extracción de señales y 2 electrodos de referencia. Este sistema fue desarrollado por Emotiv Systems (Emotiv I. , 2014), una empresa electrónica australiana dedicada a la elaboración de tecnología de electroencefalografía (EEG) y sistemas BCI. Se han realizado investigaciones recientes que denotan el interés de muchas personas por

alcanzar resultados significativos en BCI usando Emotiv- EPOC; a continuación se mencionan algunas.

Liu y colaboradores (Liu, y otros, 2012) desarrollaron una plataforma basada en SSVEP (*steady-state visual evoked potential,* en Español "potencial evocado visual de estado estable") usando Matlab y Emotiv- EPOC. En comparación con g.tec[1] (g.tec medical engineering, 2015), el cual es un sistema sofisticado de captura y procesamiento de señales EEG, Emotiv EPOC es un dispositivo inalámbrico de bajo costo que requiere de menor tiempo de preparación para su uso. Sin embargo, el software de apoyo para Emotiv-EPOC no es de código abierto y tiene algunas limitaciones en la programación con Matlab. Con la finalidad de superar estas dificultades, para el procesamiento de la señal sin el uso de internet, los autores utilizaron Emotiv TestBench (Emotiv I. , 2014), que es un conjunto de rutinas que permiten guardar y convertir los datos del EEG a un formato que Matlab puede leer. Posteriormente, los autores emplearon el análisis de correlación canónica (CCA, *canonical correlation analysis*) para la detección de características en los datos del EEG y compararon el rendimiento de g.tec y Emotiv. El ITR[2] (*Information Transfer Rate,* tasa de transferencia de información) y la precisión obtenidos usando g.tec fueron de 94.79±1.94 % y 35.66±5.71 bits/min, respectivamente. Para Emotiv, el promedio de precisión fue de 82.99±4.98% y el ITR de 28.06±6.45 bits/min.

En segundo lugar, debido a la dificultad de conexión, estos autores ocuparon el sistema de software BCI2000 para hacer comunicación entre Emotiv y Matlab, de esta manera implementaron el sistema en tiempo real. Este sistema alcanzó un promedio de precisión de 95.83±3.59 %, un ITR de 18.99±1.68 bits/min y detección de tiempo de 5.25±2.14 segundos. Finalmente, los autores utilizaron el sistema en línea y su prueba de control del entorno ayudó a los usuarios, especialmente a las personas discapacitadas, al manejo de los aparatos electrodomésticos sin necesidad de utilizar el cuerpo.

[1] g.tec medical engineering es una empresa austriaca, fundada en 1999 y dedicada a la investigación y desarrollo de sistemas BCI.
[2] La velocidad de transferencia de información (ITR) es un índice muy importante para evaluar sistemas BCI. ITR está representado por la información transmitida dada en bits/ minuto.

Moreno y otros (Moreno, Peña, & Gualdrón, 2014) presentan los avances obtenidos en la realización de un sistema de neuro- mercadotecnia. El sistema permite evaluar las emociones que presentan los televidentes al observar comerciales publicitarios. Utilizaron la interfaz Emotiv- EPOC para la adquisición de las señales, misma que es capaz de identificar los siguientes estados de ánimo: aburrimiento, excitación, calma, frustración y meditación. El algoritmo que propusieron tiene siete etapas: preparación de los equipos y zonas de trabajo, tiempo de relajación para preparar a los usuarios, inicio de la ejecución de una pauta publicitaria, captura del evento de atención, fin de la ejecución de la pauta publicitaria, análisis estadístico de la señal correspondiente a la atención del usuario durante la pauta y la realización de una entrevista (método tradicional de evaluación). Los indicadores estadísticos de las pruebas realizadas (media, mediana, desviación estándar y varianza) demostraron que el análisis de neuro-señales a través de Emotiv- EPOC es una herramienta útil para la captura del nivel de atención en los usuarios, y de esta manera mejorar la realización de comerciales.

Los autores (Vourvopoulos & Liarokapis, 2011) desarrollaron dos prototipos basados en la interacción humano-robot a través de tele-operación con BCI. El primer prototipo experimental permite a un robot Mindstorms NXT avanzar o detenerse, dependiendo del nivel de atención o meditación que el usuario presente. El segundo prototipo lo implementaron con el uso de la diadema Emotiv-EPOC, debido a las ventajas que presenta, como su extensibilidad y adaptabilidad en su lenguaje de programación LEJOS de Java, logrando obtener un mejor control del usuario hacia el robot Mindstorms NXT. El robot controlado por la diadema Emotiv-Epoc fue capaz de realizar cuatro movimientos: derecha, izquierda, adelante y atrás, por lo que el estudio de los niveles de atención y meditación en el cerebro humano pueden ser de gran utilidad en trabajos futuros. En este trabajo de investigación el porcentaje máximo de eficiencia para el nivel de meditación en calma fue de 90%, en estado normal: 75%, y en estado de concentración fue de 83%.

Campbell y colaboradores (Campbell, y otros, 2010) presentan el diseño, implementación y evaluación del sistema *Neurophone.* El sistema detecta señales neuronales con el uso de la diadema Emotiv-EPOC para impulsar aplicaciones de telefonía móvil en el iPhone. La aplicación consiste en marcar un número específico de contacto de la libreta de direcciones de un teléfono móvil únicamente controlado por el cerebro, mediante dos formas: con el uso de la señal P300 y a través de guiños emitidos por el usuario. En la pantalla del teléfono parpadea una secuencia de fotos que hacen alusión a la persona que el usuario desea marcar, las señales EEG captadas se transmiten de forma inalámbrica al iPhone donde funciona de forma nativa un clasificador ligero para discriminar señales con ruido. En la evaluación del sistema los autores obtuvieron que la precisión de la diadema Emotiv- EPOC, en los resultados de clasificación por guiño para individuos sentados, es de 95.58%; para personas caminando es de 92.58%. Además, en la evaluación de la clasificación a través de pensamiento con P300, obtuvieron que el porcentaje de precisión para un sujeto sentado sin escuchar música y para un sujeto sentado escuchando música es el mismo, es decir, 88.89%, finalmente parado es de 66.67%.

2.2.3 Robot Mindstorms NXT

El robot Mindstroms NXT está formado por un conjunto de elementos básicos de las teóricas robóticas, como la unión de piezas y la programación de acciones orientada a objetos o en forma iterativa. Este robot es fabricado por la empresa LEGO y se usa en este proyecto de investigación para ejecutar las instrucciones de movimiento interpretadas por el sistema BCI implementado. Los robots Mindstorms se usan continuamente en proyectos de investigación y para una gran variedad de aplicaciones, por ejemplo: la simulación de problemas de inteligencia tipo enjambre (Pásztor, 2014) , la enseñanza en la disciplina de Gestión de Proyectos de Ingeniería de Software (Silveira Duarte, y otros, 2014), la validación de estrategias para fusión de sensores (Marín, Vallés, Soriano, Valera, & Albertos, 2014) o para el aprendizaje de la mecatrónica (MechLab, 2013).

2.2.4 Wavelets

Se han realizado algunas investigaciones de la transformada Wavelets relacionadas con el análisis de EEG, lo que demuestra la utilidad de estas transformadas matemáticas en el campo de la ciencia, a continuación se mencionan algunas.

Los autores (Sheoran, Kumar, & Kumar, 2014) propusieron un algoritmo para eliminar el ruido que provocan los artefactos[3], cuando se toman EEG's empleando un solo canal, basado en la utilización de análisis wavelet y la técnica FastICA[4]. El análisis wavelet lo utilizaron para descomponer el vector de señal en una matriz de entrada usando la técnica ICA. El ruido blanco que utilizaron en los EEG's analizados fue de 50 Hz, y los resultados obtenidos del método propuesto, en comparación con los filtros pasa bajas, mostraron mejores resultados con insignificante pérdida de información. El método propuesto con wavelets mostró una mejora en la eliminación de ruido de 4.7%.

Los autores (Mahajan & I. Morshed, 2015) presentaron un nuevo algoritmo estadístico sin supervisión, que utiliza entropía multiescala modificada y curstosis, para identificar automáticamente qué ojo parpadea independientemente de los artefactos. Posteriormente, se elimina el ruido de dichos artefactos mediante la descomposición wavelet biortogonal. El algoritmo preserva la actividad neuronal en los componentes independientes y elimina solo la actividad de los artefactos. Los resultados mostraron un mejor desempeño en las señales EEG reconstruidas usando un algoritmo no supervisado propuesto en términos de coeficiente de correlación y coherencia espectral, en comparación con la reducción a cero-ICA convencional y técnicas de eliminación wavelet de artefactos. El nuevo algoritmo logra una sensibilidad media de 90%, con un tiempo promedio de ejecución para los conjuntos de datos (N= 7) de 0.06 s. El algoritmo propuesto no requiere de la identificación manual de artefactos, ni de

[3] Los artefactos son señales indeseadas que distorsionan el EEG y que surgen de acciones cotidianas, como parpadear o mover un músculo.
[4] FastICA es un algoritmo que se aplica a las señales descompuestas para averiguar la mezcla y desmezcla de la matriz (A y W respectivamente) a lo largo de la matriz de componentes independientes.

un canal electro-encefalográfico adicional. El algoritmo fue probado para 12 canales, pero podría ser útil para sistemas de EEG más grandes.

Los autores (Nasreddine Belkacem, Shin, Kambara, Yoshimura, & Koike, 2015) propusieron un algoritmo para la clasificación de seis tipos de movimientos de ojos (arriba, abajo, izquierda, derecha, centro y parpadeo no voluntario). Para captar las señales cerebrales ocuparon dos sensores, posicionados sobre el lóbulo temporal, realizando la clasificación en tiempo real. En este trabajo, los autores usaron la transformada continua wavelet para la detección en línea y extracción de características, pudiendo discriminar entre cuatro direcciones cardinales para el movimiento de ojos y dos clases adicionales (parpadeo y mirar al centro). Los autores aplicaron la transformada wavelet porque se ha demostrado en otros trabajos que es una técnica muy efectiva para el reconocimiento de patrones en EEG, además de que detecta ocurrencias muy precisas en el movimiento de ojos en señales no estacionarias. A través de usar una clasificación lineal, los resultados mostraron un promedio de precisión del 85.2% y una susceptibilidad de 77.6% en todas las clases. Por lo tanto, el algoritmo propuesto fue eficiente en la detección y clasificación del movimiento de ojos, logrando mayor precisión y baja latencia para pocas pruebas.

Juárez Guerra y otros (Juárez Guerra, Alarcon Aquino, & Gómez Gil, Epilepsy Seizure Detection in EEG Signals Using Wavelet Transforms and Neural Networks, 2014) presentaron los resultados de un modelo basado en análisis wavelet y redes neuronales, para la identificación de eventos de convulsiones en la epilepsia. La base de datos que utilizaron es la proporcionada por la Universidad de Bonn, y el tratamiento de la señal pasó por tres etapas: la primera, que consistió en el pre-procesamiento de la señal, utilizó dos filtros pasa bajas IIR y FIR, cada uno diseñado para frecuencias de 64 Hz y al menos 60 dB de atenuación en lo más alto de la banda. Posteriormente, la extracción de características se llevó a cabo con la Transformada Discreta Wavelet (DWT, *Discrete Wavelet Transform*) y con la Transformada Discreta Wavelet de Superposición Máxima (MODWT, *Maximal Overlap Discrete Wavelet Transform*), aplicando en ambas las siguientes wavelets madre: Haar wavelet, Daubechies de segundo orden (Db2) y Daubechies de cuarto orden (Db4). Por último,

los autores realizaron la clasificación a través de una red neuronal de retro-propagación con seis nodos de entrada (uno por característica) y dos nodos de salida (uno por cada clase). Para realizar los experimentos utilizaron los dos tipos de transformadas wavelet mencionados anteriormente. En el primer experimento se descompuso la señal del EEG en segmentos y con ellos entrenaron a la red, logrando una precisión de clasificación de 93.23%. En el segundo experimento la señal del EEG se descompuso en sub- segmentos para el entrenamiento y obtuvieron el 99.26% de precisión.

Capítulo III

Método Propuesto

En el presente capítulo se describen cada una de las etapas del método propuesto para lograr el control del movimiento de objetos a través del uso de electro-encefalogramas y redes neuronales artificiales con la diadema Emotiv-EPOC.

3.1 Descripción general.

La descripción general del sistema, llamado BLINK, se basa en cuatro etapas mostradas en la Figura 3.1; en las siguientes secciones se describen cada una de éstas.

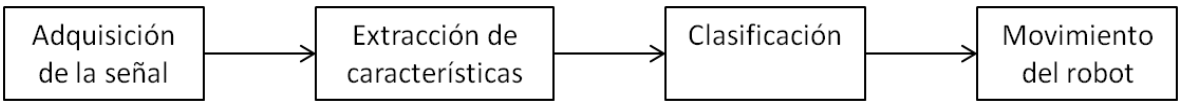


Figura 3.1 Etapas del sistema BLINK.

3.2 Adquisición de la señal.

Existen diferentes técnicas para la captura de las señales neuro-eléctricas, sin embargo, el EEG y el electrocorticograma (ECoG) son los dispositivos más utilizados. En nuestro trabajo se utilizó el EEG para obtener las señales cerebrales, a través del uso de la diadema Emotiv- EPOC mostrada en la Figura 3.2. Los primeros pasos para capturar las señales cerebrales se describen en el diagrama de actividades de la Figura 3.3.

Figura 3.2 Diadema Emotiv- EPOC (Emotiv I. , 2014).

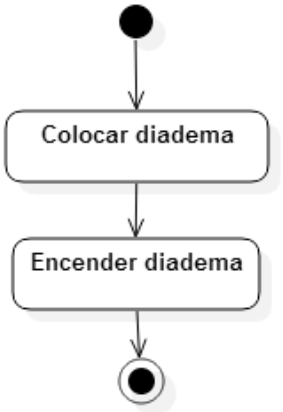


Figura 3.3 Diagrama de actividades de la captura de señales.

3.2.1 Colocar diadema

La colocación correcta de la diadema Emotiv- EPOC es muy importante para obtener resultados óptimos y confiables. El sistema de posicionamiento de electrodos Emotiv- EPOC se basa en el sistema internacional 10-20 (Figura 3.4).

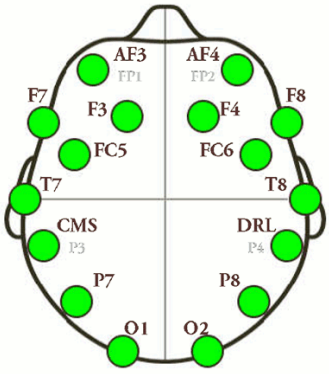


Figura 3.4 Sistema de posicionamiento de electrodos de Emotiv- EPOC (Reyes Rueda, 2013).

El sistema 10-20 (Figura 3.5) se basa en la relación de distancia entre electrodos que están espaciados entre el 10% y el 20% de distancia entre puntos identificables del cráneo. Estos puntos clave, a partir de los cuales se realizan las medidas, son (Talamillo García, 2011):

- Nasión: identación entre frente y nariz.
- Inión: protuberancia occipital.
- Punto pre-auricular: delante del trago de cada pabellón de la oreja.

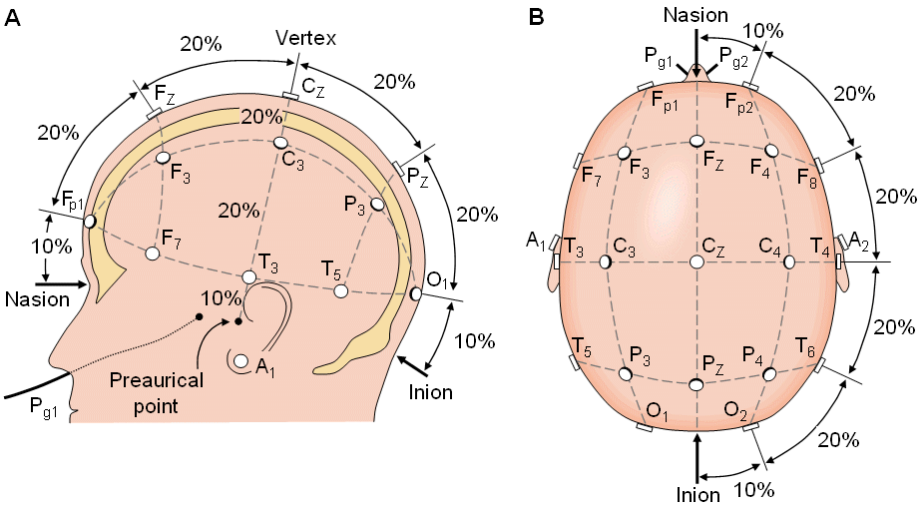


Figura 3.5. Ubicación de los electrodos y puntos de medición para su colocación en el Sistema 10- 20 (Talamillo García, 2011).

Existe una nomenclatura de los electrodos, que obedece a la región cerebral sobre la que yacen y a una numeración que va de menor a mayor, empezando desde áreas anteriores hacia posteriores; los números impares corresponden al lado izquierdo y los pares al derecho (Talamillo García, 2011), (Ver tabla 3.1). En la Figura 3.6 se ilustra el sistema de posicionamiento correcto en la cabeza del usuario con el sistema Emotiv- EPOC.

3.2.2 Encender diadema

El encendido de la diadema comprende la conexión de la USB al computador para establecer comunicación inalámbrica, así como verificar que el *led* de la diadema esté en color azul para asegurar que tiene suficiente batería y funcione correctamente.

Tabla 3.1 Área cerebral y nodos que le corresponden (ver Figura 3.5 para el significado de los nodos). (Talamillo García, 2011)

Área cerebral	**Hemisferio Izquierdo**	**Línea Media**	**Hemisferio Derecho**
Frotopolar	FP1		FP2
Frontal	F3	Fz	F2
Fronto Temporal	F7 C3	Cz	F8 C4
Temporal Medio y Parietal	T3 P3	Pz	T4 P4
Temporal Posterior y Occipital	T5 O1		T6 O2

Figura 3.6. Posicionamiento correcto de electrodos en el usuario (Emotiv I. , 2014).

Continuando con el proceso de la adquisición de la señal, la Figura 3.7 muestra el diagrama de actividades correspondiente al momento en el que se inicia la grabación de señales cerebrales con el usuario, hasta el instante en el que se obtiene el archivo *.edf que contiene los datos crudos del EEG. En los siguientes apartados se describen cada una de las actividades de manera explícita.

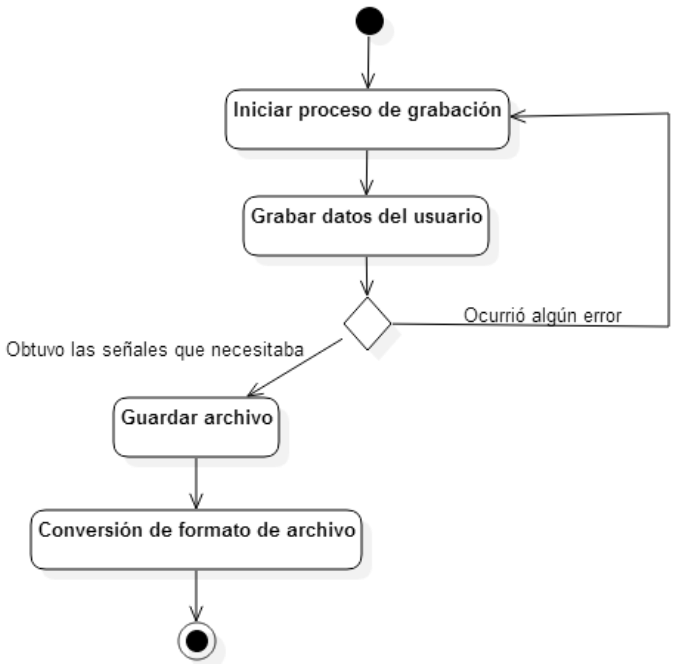


Figura 3.7 Diagrama de actividades desde el inicio de la grabación del EEG hasta la conversión del archivo .edf a .csv

3.2.3 Iniciar proceso de grabación

El software utilizado para grabar las señales electro-encefalográficas de los usuarios en nuestra investigación fue Emotiv TestBench v1.5.0.3 (Informer Technologies, 2015). Antes de comenzar a grabar el EEG, es necesario verificar la recepción de la señal inalámbrica, a través de los colores mostrados en la ilustración representativa de la diadema (Figura 3.8). El color negro indica que no hay señal, el rojo revela que la señal es casi nula, el anaranjado representa que la señal es baja, el amarillo muestra un estado de aceptación y el verde corresponde al color de una excelente señal. Para nuestro sistema es muy importante tener todos los canales funcionando de manera óptima, por lo que el color de exigencia es verde.

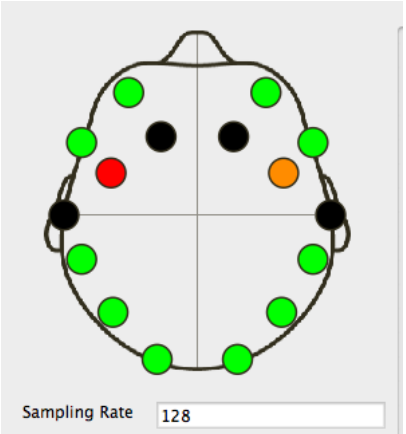


Figura 3.8 Colores que verifican la recepción inalámbrica de la diadema Emotiv- EPOC (Emotiv).

Antes de comenzar a grabar el EEG de los sujetos que se utilizaron para realizar los experimentos detallados aquí, se practicaron diversas actividades mentales y musculares con los usuarios durante cinco minutos. Estos ejercicios mentales se realizaron utilizando los programas ya disponibles en Emotiv. Una actividad consistió en tratar de mover un cubo en 4 cuatro posiciones diferentes (arriba, abajo, derecha, izquierda) con solo pensarlo, y la segunda actividad fue parpadear primero un ojo, después otro y por último ambos ojos por dos minutos, respectivamente. Las actividades anteriormente descritas se realizaron con la finalidad de que las personas se familiarizarán con el equipo y sus acciones se efectuarán de manera natural.

3.2.3 Grabar datos del usuario

Para realizar los experimentos presentados en esta investigación, se creó una pequeña base de datos de EEG, que se describe en la sección 3.4.2. Durante la grabación del EEG de cada sujeto, éste(a) se encontraba sentado(a) cómodamente en una silla posicionada a 30 cm de la computadora, y el lugar contaba con buena iluminación. Cuando el usuario estaba relajado(a) y preparado(a) para comenzar la captura de señales electro-encefalográficas, se comenzó el parpadeo de ojos, de acuerdo a las indicaciones que el individuo le quería mandar al robot y de esta manera efectuar el movimiento del lego. La tabla 3.2 específica el movimiento del robot que corresponde al parpadeo provocado de cada uno de los ojos. La Figura 3.9 muestra como ejemplo, el parpadeo provocado del ojo derecho con el usuario 1.

Tabla 3.2 Movimiento del robot de acuerdo al parpadeo.

Parpadeo de ojo	Movimiento del robot
Ojo derecho	Vuelta a la derecha
Ojo Izquierdo	Vuelta a la izquierda
Ambos ojos	Avanza hacia adelante.

.

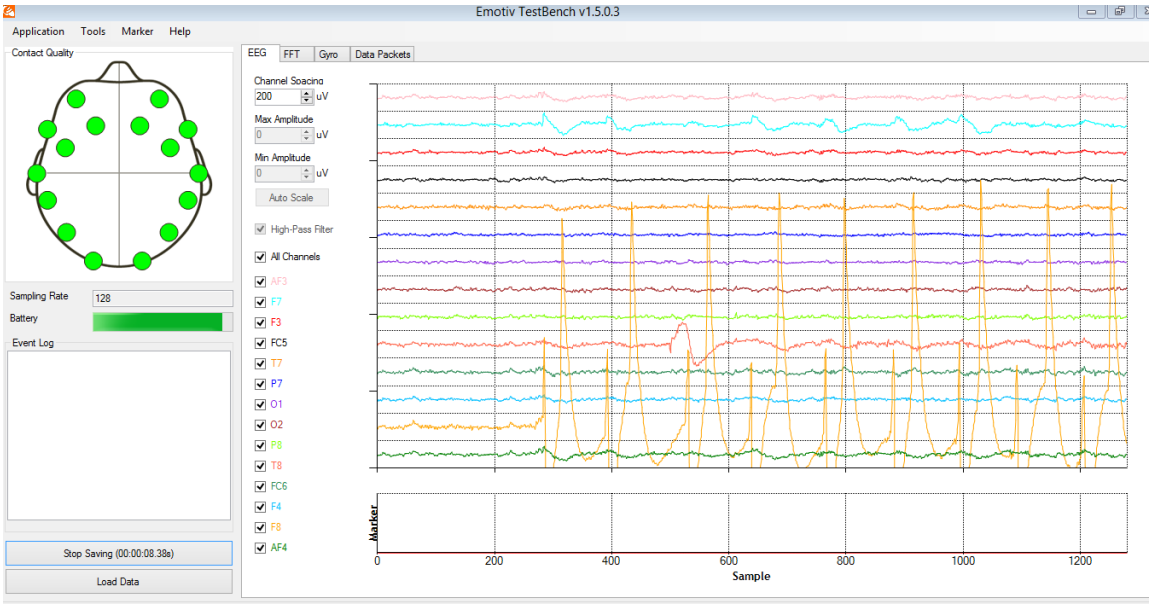


Figura 3.9 EmotivTestBench- Sujeto 1

En el momento en el que el usuario está satisfecho con las indicaciones que desea mandar al robot a través de las señales electro- encefalográficas, la grabación se debe detener y se procede a guardar el archivo en el formato que ofrece el programa TestBench, es decir, un archivo con extensión .edf.

3.2.3 Guardar archivo y conversión de formato

Como se nombró anteriormente, la extensión por defecto que ofrece TestBench v1.5.0.3 para guardar archivos de señales cerebrales es .edf. Sin embargo, en este formato no es posible visualizar la información de manera inmediata. Por lo anterior, en nuestra investigación se realizó una conversión de formato de *.edf a .csv*. El formato

.csv muestra toda la información que nos interesa del EEG y además el archivo se puede abrir desde Excel.

3.3 Extracción de características

La extracción de características se realizó en base a la investigación realizada por el M.C. Juárez Guerra y otros, descrita en (Juárez Guerra, Alarcon Aquino, & Gómez Gil, Epilepsy Seizure Detection in EEG Signals Using Wavelet Transforms and Neural Networks, 2014) siguiendo el procedimiento a continuación descrito. Parte del software utilizado en dicha investigación se adecuó para su uso en este trabajo. Como se establece en dicha investigación, el análisis wavelet se utiliza para descomponer las señales EEG en sus componentes de frecuencia *delta* (δ), *theta* (θ) y *alpha* (α). La banda *delta* se encuentra entre 0- 4 Hz, se sabe que está presente durante el sueño profundo y también en una enfermedad cerebral grave (Galvão Barbosa, Zacarias Freitas, de Moura Guedes, & Meggiolaro, 2013). La banda *theta* está entre las frecuencias de 4- 8 Hz, y se muestra principalmente en las regiones cerebrales pariental y temporal en los niños y durante estrés emocional en adultos. La banda *alpha* se localiza entre 8-12 Hz y se encuentra en todos los individuos cuando se despiertan y permanecen tranquilos, están descansando o en estado de relajación.

Después, se lleva a cabo la media de los coeficientes wavelet, utilizando la Transformada Discreta Wavelet (por sus siglas en inglés DWT, *Discrete Wavelet Transform*), y el filtro Chevyshev IIR (*Infinite Impulse Response,* en español: Respuesta Infinita al Impulso) tipo II. De acuerdo a Juarez-Guerra y otros (Juárez Guerra, Alarcon Aquino, & Gómez Gil, Epilepsy Seizure Detection in EEG Signals Using Wavelet Transforms and Neural Networks, 2014), la selección de wavelet debe estar relacionada con características comunes de eventos presentes en las señales EEG, es decir, las wavelet deben estar bien adaptadas a lo que se analizará. Las diferentes familias de wavelet están relacionadas con su grado de simetría, el cual es importante para reducir la fase de desplazamiento de características durante la descomposición wavelet. Si el desplazamiento de la fase es largo, puede llevar a distorsiones en la ubicación de características en los coeficientes de transformación.

Siguiendo lo establecido en Juárez Guerra y otros, el vector de características se construye de la siguiente manera: se calcula la DWT de cada segmento, de donde se obtienen las sub-bandas cerebrales (*delta*, *theta* y *alpha*) y sobre estos datos se extraen 4 medidas estadísticas: media, mediana, varianza y entropía. Es importante mencionar que la extracción de características se llevó a cabo usando la información de los electrodos F8 y F7, puesto que las señales EEG mostraron que, cuando se parpadea con el ojo derecho, el canal que recibe directamente el movimiento muscular es el canal F8. De manera similar, se observó que cuando se parpadea con el ojo izquierdo la señal del canal afectado es F7, y cuando se parpadea con ambos ojos se observa una modificación en las señales de los canales F7 y F8. La Figura 3.10 muestra como las señales a cada respectivo parpadeo son diferentes, por ejemplo, la imagen del inciso a) corresponde al ojo derecho, y el canal F8 es el que presenta mayor movimiento; en el inciso b) el canal F7 tiene mayor variación en la señal y representa al ojo izquierdo. Finalmente, en el inciso c) podemos observar que las señales que mayor variación tienen son F7 y F8, y corresponde al parpadeo de ambos ojos.

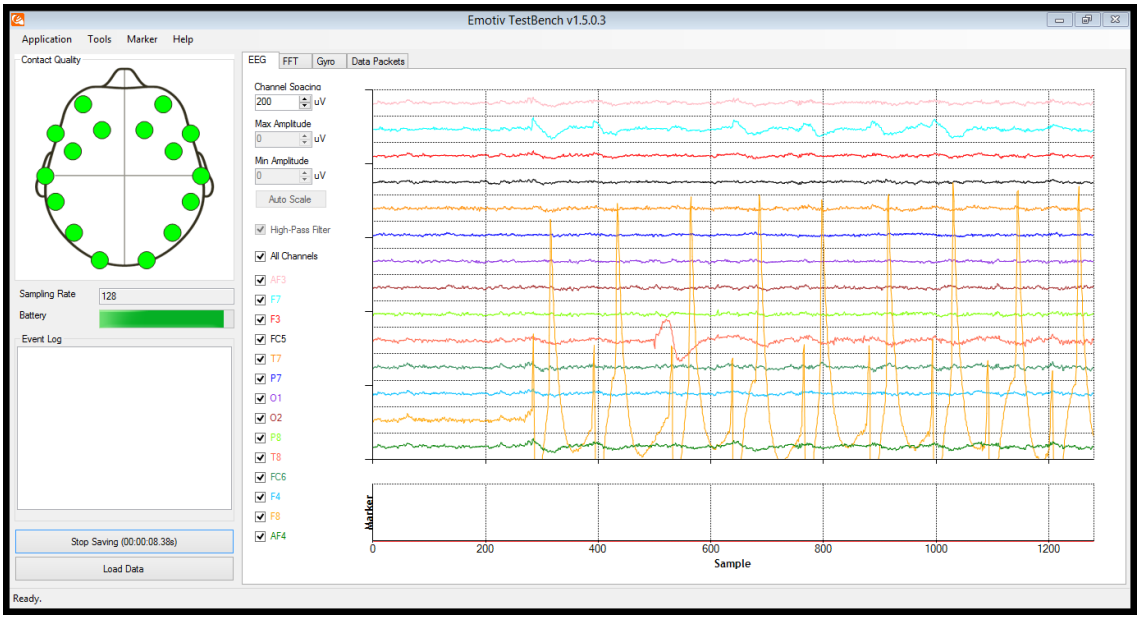


Figura 3.10 Un ejemplo de EEG al realizar parpadeo de ojos. a) EEG obtenido cuando se parpadea el ojo derecho.

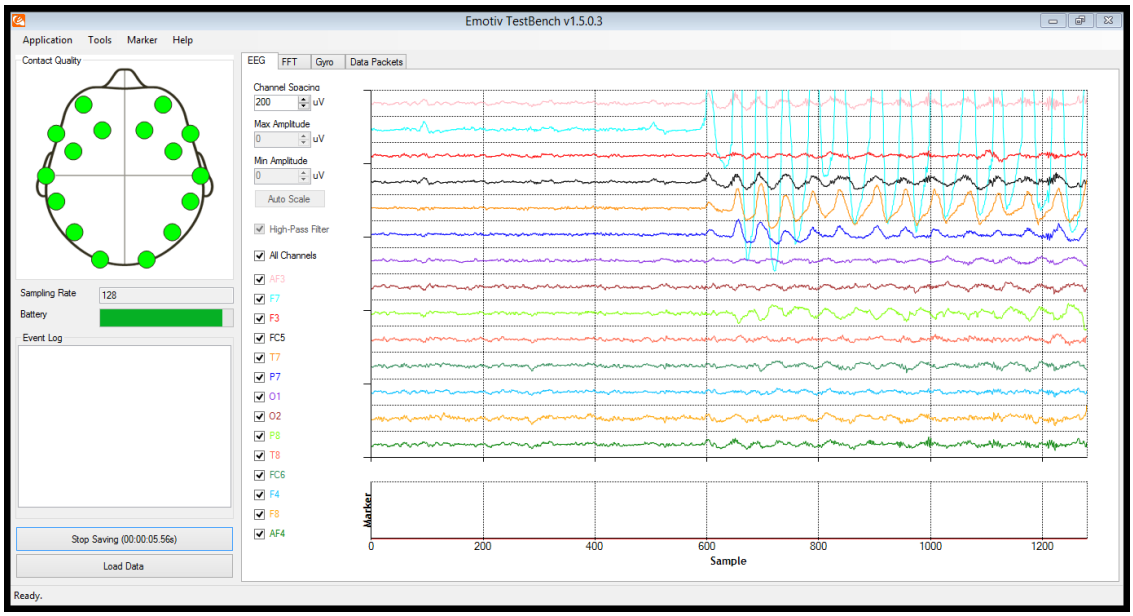


Figura 3.10 Un ejemplo de EEG al realizar parpadeo de ojos. b) EEG obtenido cuando se parpadea el ojo izquierdo.

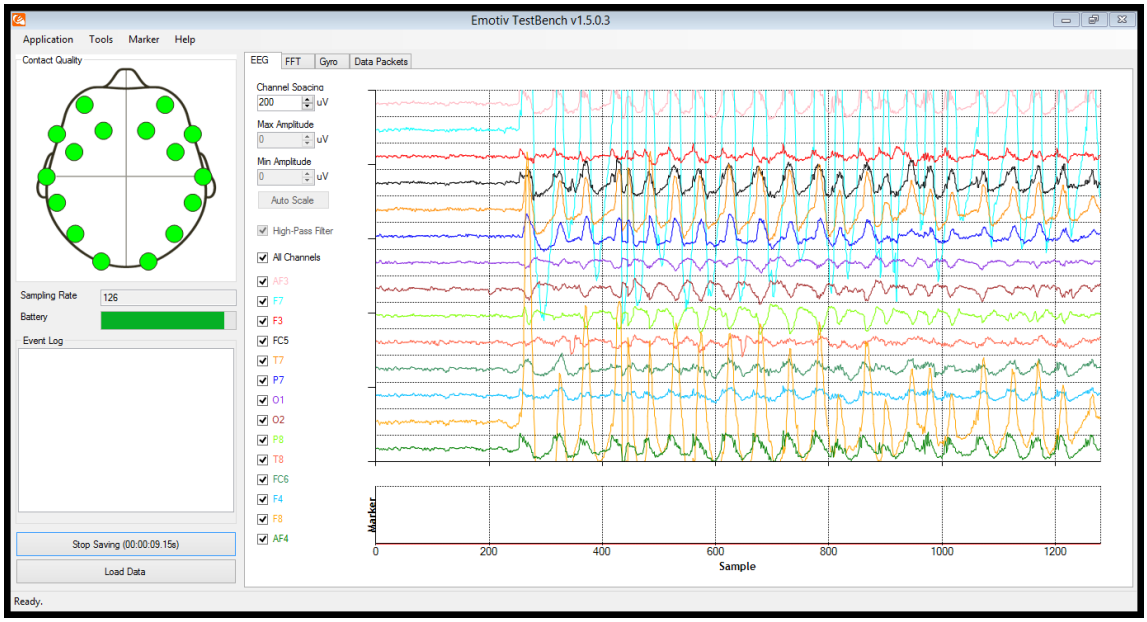


Figura 3.10 Un ejemplo de EEG al realizar parpadeo de ojos. c) EEG obtenido cuando se parpadean ambos ojos.

3.4 Clasificación

El objetivo del clasificador del sistema BCI en esta investigación es identificar las acciones que un individuo le manda a un robot, basándose únicamente en la extracción de características del EEG. Para la clasificación del movimiento de parpados utilizamos una red neuronal de retro- propagación. Para validar si el sistema es adecuado para el sistema descrito, se utilizó la técnica estadística de validación cruzada conocida como *k- fold cross- validation*, que se describe a continuación.

3.4.1 *k- fold cross- validation*

La técnica k- *fold cross- validation* es utilizada para garantizar que todas las clases estén representadas en todos los archivos en aproximadamente las mismas proporciones, con la finalidad de asegurar que un sistema de clasificación entrene y pruebe adecuadamente todos los datos de entrenamiento y la clase a la que estos datos pertenecen (Refaeilzadeh, Tang, & Liu, 2009).

Para este trabajo se utilizó la técnica *Three-fold cross- validation* cuyo objetivo es dividir un archivo de entrenamiento en tres archivos diferentes, que contengan la misma cantidad de datos, elegidos de manera aleatoria y asegurando que cada uno de los archivos tenga la misma cantidad de cada clase. La figura 3.11 ejemplifica el procedimiento que sigue la técnica *threefold cros-validation.*

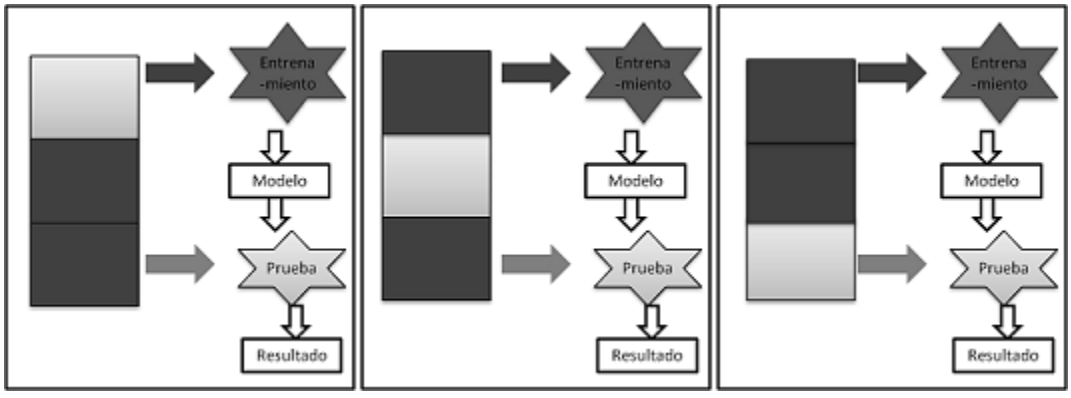


Figura 3.11 Procedimiento de *Three-folder cross-validation.*

El pseudo-código del algoritmo utilizado se describe a continuación. "*archivo1*", "*archivo 2*" y "*archivo3*" son los nombres de los archivos que se generaron aleatoriamente con el mismo número de datos, a partir del archivo de entrenamiento correspondiente. La función "Entrenar" recibe dos parámetros (los nombres de los archivos) y con los datos que tiene cada archivo se entrena la red neuronal. La función denominada "Validar" es la encargada de obtener los valores de clasificación de la red y de esta forma verificar cuantas clasificaciones se realizaron de manera correcta de un número total de clasificaciones, y obtener la eficiencia de la red neuronal.

```
Para  i=1 to 3
        Entrenar (archivo1, archivo 2)
        Validar (archivo 3)= eficiencia (1)
        Entrenar (archivo1, archivo3)
        Validar (archivo 2)= eficiencia (2)
        Entrenar (archivo2, archivo3)
        Validar (archivo 1)= eficiencia (3)
        PromedioEficiencia= eficiencia (1)+ eficiencia (2)+
´       eficiencia (3) / 3;
End
```

3.4.2 Diseño de los archivos de entrenamiento

Los archivos de entrenamiento utilizados para esta investigación son datos propios, recolectados en el laboratorio de Procesamiento de Señales e Inteligencia Computacional (PSIC) del Instituto de Astrofísica, Óptica y Electrónica (INAOE) en Tonantzintla, Puebla. Estos datos, recolectados durante el cuatrimestre de primavera de 2015, fueron generados a partir del procedimiento que a continuación se describe.

Primero, a 3 individuos de sexo masculino, en edades comprendidas entre 22 y 25 años se les tomaron 15 series de pruebas electro- encefalográficas con la diadema EMOTIV- Epoc. Para cada individuo, 5 pruebas correspondieron a parpadeo de ojo derecho, 5 de ojo izquierdo y 5 de ambos ojos. Cada uno de los ejercicios de la serie

originalmente tuvo una duración de 14 segundos; entre cada ejercicio el individuo tuvo un descanso de 60 segundos para que se relajara. El lugar en el que se realizó la toma de los EEG es un lugar limpio, fuera de ruido y los individuos estaban a 30 cm aproximadamente del ordenador, en una posición cómoda. La figura 3.12 muestra a dos de los tres sujetos a los que les tomaron los EEG.

a) Sujeto 2 b) Sujeto 3

Figura 3.12 Dos individuos a los que se les tomaron los EEG.

De los 14 segundos grabados en cada uno de los ejercicios, se recortó la señal a únicamente 10 segundos guardados en el archivo final, eliminando dos segundos de grabación del inicio del archivo y dos segundos de grabación del final del archivo. La razón por la cual se eliminaron dos segundos al principio del EEG es que se observó que el usuario tarda en empezar a parpadear aproximadamente un segundo y el sistema tiene un retraso de otro segundo. Por otra parte, los otros dos segundos que se eliminaron al final se deben a que, entre más tiempo pasa en la toma del EEG, el usuario realiza el parpadeo con menos fuerza, y deja de realizar la actividad deseada con uno o dos segundos de desincronización con el sistema de *TestBench*. Se construyeron 45 archivos en total: 15 de parpadeo con ojo derecho, 15 de parpadeo con ojo izquierdo y 15 de ambos ojos.

Los resultados presentados en este trabajo se realizaron utilizando y comparando cuatro estrategias. La primera estrategia implica la extracción de características sobre segmentos de 10 segundos; la segunda estrategia implica extraer características usando segmentos de un segundo; la tercera y cuarta estrategia implican el uso de un conjunto de características ligeramente diferente al de las otras dos

estrategias. Enseguida se detalla la construcción de los archivos de entrenamiento usados para cada estrategia.

Para obtener el primer archivo de entrenamiento, al que llamamos "archivoDeEntrenamientoNumeros45.txt", a los 45 archivos .csv que tenían los 10 segundos acotados de toma de EEG, se le aplicó la extracción de características (descrita en la sección 3.3). Además, estos datos se etiquetaron con la clase a la que correspondían, por lo que el archivo de entrenamiento consta de 45 filas y 27 columnas. Las primeras 24 columnas corresponden a las características extraídas, las cuales son 4 por cada banda de frecuencia analizada para los canales F7 y F8, recordando que hay 3 bandas de frecuencia involucradas. Las tres columnas restantes corresponden a la codificación asociada a la clase a la que pertenece. El etiquetado de clase se codificó de la siguiente manera: la clase ambos ojos corresponde al identificador "0 0 1", la clase ojo derecho "0 1 0", y la clase ojo izquierdo "1 0 0").

El segundo archivo de entrenamiento, denominado "archivoDeEntrenamientoNumeros585.txt", se obtuvo también de los 45 archivos que se acotaron a 10 segundos de los tres sujetos. Sin embargo, para generar este archivo la extracción de características se realizó con ventanas superpuestas, entrecortadas segundo a segundo. Cada segundo corresponde a 128 puntos, y el retraso de la ventana corresponde a 32 segundos (Figura 3.11), generándose de un solo archivo 13 vectores característicos. En total, de los 45 archivos analizados se obtienen 585 filas (45x13), por lo que el archivo final quedó de 585 filas por 27 columnas.

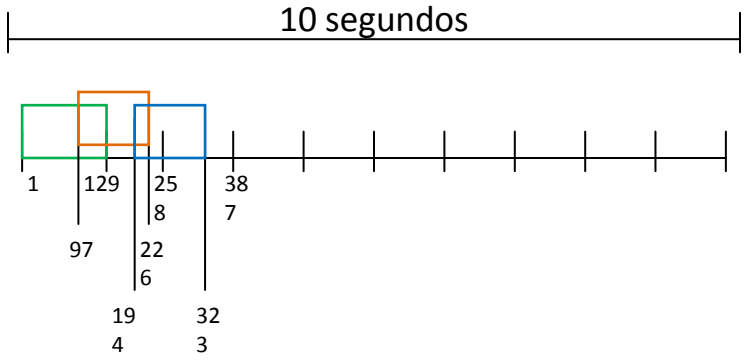


Figura 3.13 Ventanas con corrimiento de 32 puntos.

Para obtener el tercer archivo de entrenamiento, al que llamamos "ArchivoComparacion_45x19_E2.txt", a los 45 archivos .csv que tenían los 10 segundos acotados de toma de EEG, se les aplicó la misma extracción de características descrita en la sección 3.3, pero eliminando del vector a la media y mediana de las bandas *theta* y *alpha*. Esto se realizó debido a que se observó que estos valores son similares para todas las clases. A estos datos, como en los casos anteriores, se les agregó la clase a la que correspondían, por lo que el archivo de entrenamiento quedo de 45 filas y 19 columnas.

El cuarto archivo de entrenamiento, denominado "ArchivoComparacion_585x19_E2.txt", se obtuvo también de los 45 archivos que se acotaron a 10 segundos de los tres sujetos, sin embargo, la extracción de características se realizó con ventanas entrecortadas (sin las mismas 4 características que el tercer archivo) quedando el archivo final de 585 filas y 19 columnas.

La tabla 3.3 resume las características de cada archivo descrito anteriormente.

Tabla 3.3. Archivos utilizados en los experimentos

NOMBRE DEL ARCHIVO	TAMAÑO	DESCRIPCIÓN
archivoDeEntrenamientoNumeros45.txt	45 x 27	Contiene 45 instancias con 24 características cada una, calculadas sobre el segmento completo de 10 segundos
archivoDeEntrenamientoNumeros585.tx	585 x 27	Contiene 585 instancias con 24 características, calculadas sobre ventanas superpuestas
ArchivoComparacion_45x19_E2.txt	45 x 19	Contiene 45 patrones con 16 características, calculadas sobre el segmento completo de 10 segundos
ArchivoComparacion_585x19_E2.txt	585 x 19	Contiene 585 instancias con 16 características, calculadas sobre ventanas superpuestas

3.4.3 Pruebas y validación.

Como se describió anteriormente, se aplicó la técnica *Three- fold cross-validation* para cada archivo de entrenamiento. De cada archivo se generaron tres archivos, a los cuales se les aplicó el proceso de entrenamiento y la validación. Los resultados se describen en el capítulo IV. De igual forma, con esta técnica se obtuvieron las matrices de confusión correspondientes a cada uno de los experimentos y que permiten evaluar la eficiencia de la red.

3.4.4 Diseño de los experimentos realizados

Los experimentos se dividen en dos grandes casos. Cada uno de los casos se describe en las siguientes líneas.

Caso 1

El primer caso está compuesto de experimentos que utilizan bases de datos en las que la extracción de características se realizó utilizando el segmento completo de 10 segundos de cada señal de EEG. En el experimento uno se utilizaron 24 características, mientras que en el experimento dos se utilizan 16 características.

- Características sobre 10 segundos
  - **Experimento 1.** Se realizó utilizando de 5 a 20 neuronas para el archivo "archivoDeEntrenamiento Numeros45.txt".
    - **Experimento 1.1.** Se realizó en un ciclo de 10 veces para el archivo denominado "archivoDeEntrenamientoNumeros45.txt".
  - **Experimento 2.** Se realizó utilizando de 5 a 20 neuronas para el archivo "ArchivoComparacion_45x19_E2.txt".
    - **Experimento 2.1.**Se realizó en un ciclo de 10 veces para el archivo denominado "ArchivoComparacion_45x19_E2.txt".

El primer experimento consistió en probar la eficiencia de la red neuronal para una sola corrida del *3-fold-validation*, usando el archivo de entrenamiento "archivoDeEntrenamientoNumeros45.txt." Este archivo se dividió en tres partes, donde cada parte se construyó seleccionando renglones de manera aleatoria. Los archivos se nombraron "Archivo_1", "Archivo_2", "Archivo_3". Este experimento se realizó usando un rango de neuronas en las capas internas del clasificador de 5 a 20. En cada caso se calculó la eficiencia del sistema y la matriz de confusión, misma que se guardó en un archivo .txt para su posterior análisis.

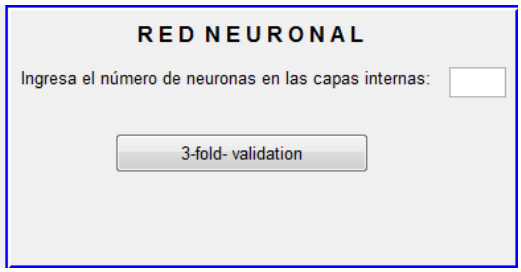


Figura 3.14 Aplicación del 3- *fold- validation* con n número de variables.

Experimento 1.1

En el experimento 1.1 se realizó el algoritmo 3- *fold- validation* 10 veces para los tres archivos generados a partir del archivo de entrenamiento "archivoDeEntrenamientoNumeros45.txt" con 16 neuronas, puesto que ese fue el mejor caso en el experimento anterior. Esto se realizó con el objetivo de que la red se re-entrenara 10 veces, y calcular un promedio de la eficiencia ocurrida en cada ejecución. Lo anterior se hace para eliminar en la evaluación, el sesgo generado por la aleatoriedad en los pesos iniciales en el entrenamiento de las redes neuronales. Al igual que en el caso anterior, se calculó la eficiencia y la matriz de confusión en cada ejecución.

Experimento 2. El segundo experimento consistió en probar la eficiencia de la red neuronal para una sola corrida del 3- *fold- validation* usando un archivo de entrenamiento llamado ArchivoComparacion_45x19_E2.txt, para los tres archivos que se generaron con datos aleatorios, denominados "Archivo_1", "Archivo_2",

"Archivo_3". El experimento se realizó para un rango de neuronas en las capas internas de 5 a 20, y para cada neurona además de su eficiencia, también se obtuvo la matriz de confusión, misma que se guardó en un archivo .txt para su posterior análisis.

Experimento 2.1. En el experimento 2.1 se realizó el algoritmo 3- *fold- validation* 10 veces para los tres archivos generados a partir del archivo de entrenamiento "ArchivoComparacion_45x19_E2.txt". Esto se realizó con el objetivo de que la red se re- entrenará 10 veces y de esta forma eliminar el sego de la aleatoriedad, como en el experimento 1.1. Los resultados de cada ciclo (entrenamiento) fueron las eficiencias, y la matriz de confusión final.

Caso 2

El segundo gran caso se caracteriza porque los experimentos realizados se hicieron con las bases de datos en las que la extracción de características se realizó con ventanas entrecortadas de 32 puntos. En el experimento 3 se utilizan 24 características, mientras que en el experimento 4 se utilizan 16 características.

Características sobre ventanas entrecortadas

- **Experimento 3.** Se realizó en un ciclo de 10 veces para el archivo denominado "archivoDeEntrenamientoNumeros585.txt".
- **Experimento 4.** Se realizó en un ciclo de 10 veces para el archivo denominado "archivoEntrenamiento585x19_E2.txt"

Experimento 3. En el tercer experimento se realizó el algoritmo 3-*fold-validation* 10 veces para los tres archivos generados a partir del archivo de entrenamiento "archivoDeEntrenamientoNumeros585.txt", con el mejor número de neuronas del experimento 1.

Experimento 4. En el cuarto experimento se realizó el algoritmo 3-*fold-validation* 10 veces para los tres archivos generados a partir del archivo de entrenamiento "archivoEntrenamiento585x19_E2.txt" con el mejor número de neuronas del experimento 2. Los resultados de cada ciclo (entrenamiento) fueron las eficiencias, y la matriz de confusión final.

3.5 Movimiento del robot.

Como prototipo de la silla de ruedas y con el objetivo de ver en acción el movimiento clasificado por la red neuronal, se construyó un perrito con el Lego Mindstorms NXT (Figura 4.3). El diseño del perrito se basó en las instrucciones proporcionadas por (NXT, 2011).

Figura 3.15. Perro Lego Mindstorms NXT construido para visualizar los experimentos.

Si el robot está en estado de reposo y la red neuronal identifica la clase "ambos ojos" (clase 001) entonces el perro avanza hacia adelante. El robot da un giro de 90° hacia la derecha cuando la red neuronal reconoce la clase 010 (ojo derecho), y gira 90° hacia la izquierda cuando reconoce la clase 100 (correspondiente al parpadeo del ojo izquierdo).

La programación del Lego Mindstorms NXT se llevó a cabo con el uso de la toolbox de matlab: RWTH, vía bluetooth (Vision, 2015). Después de que la red neuronal clasifica los movimientos, a través de un botón se le mandan al robot las instrucciones que tiene que ejecutar. La Figura 4.4 muestra un ejemplo de instrucciones enviadas al robot, después de ser identificadas por el clasificador.

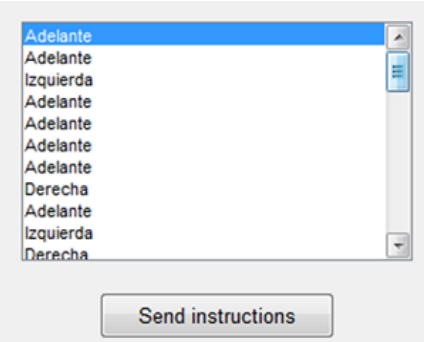


Figura 3.16. Instrucciones que se le envían al robot.

3.5.1 Especificaciones del robot

El robot camina en cuatro llantas, como lo haría una silla de ruedas, sin embargo, en el caso de nuestro robot, los motores están únicamente en las dos llantas traseras. El motor A corresponde a la llanta izquierda y el motor C a la llanta derecha; las dos llantas de enfrente están adaptadas a moverse solas, de acuerdo al empuje de las dos llantas traseras.

Cuando el robot recibe la instrucción de caminar hacia adelante, los dos motores activan el movimiento de sus respectivas llantas en la misma dirección al mismo tiempo, sin embargo, cuando el "lego" recibe la instrucción de dar vuelta a la derecha 90°, el único que se mueve es el motor A, mientras el motor C permanece estático. Por último, el "lego" logra moverse a la izquierda manteniendo el motor A sin movimiento y siendo el eje de la traslación el motor C.

Capítulo IV

Resultados

En la presente sección se muestran los resultados obtenidos a partir de los dos grandes casos de experimentos descritos en el capítulo anterior. El primer caso corresponde a experimentos realizados con extracción de características de 10 segundos para los EEG. En el segundo caso, la extracción de características en las bases de datos utilizadas se realizó con ventanas entrecortadas, con un retraso de 32 puntos. Para cada ejecución se calculó la eficiencia del clasificador, a través de la fórmula:

$$eficiencia = \frac{Número\ de\ clasificaciones\ correctas}{Número\ de\ clasificaciones\ totales} \quad \textbf{4.1}$$

Por otra parte, para cada ejecución del clasificador se calculó una matriz de confusión, la cual describe en las filas la clase real, y en las columnas, la clase que la red clasificó.

4.1 Evaluación de resultados

En los siguientes apartados se muestran los mejores resultados obtenidos de cada uno de los experimentos, es decir, su eficiencia y sus respectivas matrices de confusión.

4.1.1 Caso 1- Experimento 1

La figura 4.1 muestra los resultados del experimento 1- caso 1. En este experimento el mejor caso fue con 20 neuronas, puesto que resultó una eficiencia de 0.777, sin embargo, el peor caso se dio con 5 neuronas, y dicha eficiencia fue de 0.444.

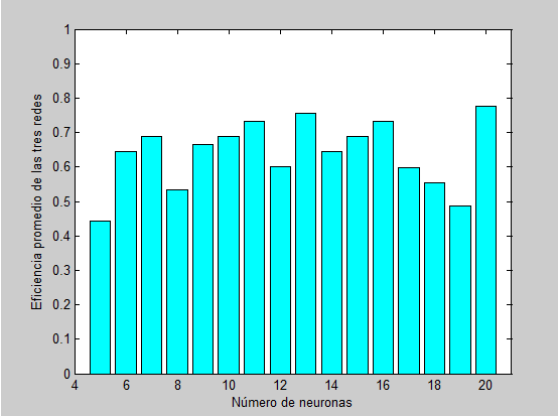


Figura 4.1 Eficiencias promedio del "archivoDeEntrenamientoNumeros45.txt" para diferente número de neuronas.

En la Figura A.1 del apéndice A, los incisos de a) - o) muestran las matrices de confusión de cada experimento realizado con el número de neuronas en las capas internas. La figura 4.2 presenta las matrices de confusión del mejor caso. En estas figuras el título "Red 1" corresponde al resultado obtenido en el primer *"fold,"* "Red 2"

	1	2	3	
1	1	1	3	5
2	0	5	0	5
3	0	0	5	5
	1	6	8	11

o.1) Red 1

	1	2	3	
1	3	0	2	5
2	0	5	0	5
3	0	1	4	5
	3	6	6	12

o.2) Red 2

	1	2	3	
1	3	0	2	5
2	0	5	0	5
3	1	0	4	5
	4	5	6	12

o.3) Red 3

Eficiencia promedio = 0.777

a) Con 20 neuronas.

Figura 4.2 Matrices de confusión del mejor caso. Caso 1- experimento 1.

4.1.2 Caso 1- Experimento 1.1

La figura A.2 del apéndice A muestra las matrices de confusión con sus respectivas eficiencias, resultantes de la red con el mejor resultado del experimento anterior (20 neuronas), entrenándola 10 veces. Esto se realizó con el fin de eliminar de los experimentos el sesgo que se genera en el desempeño de redes neuronales, debido a que su entrenamiento parte de números al azar. Como se recordará, los tres archivos que se trabajaron en este experimento fueron generados a partir del archivo de entrenamiento "archivoDeEntrenamientoNumeros45.txt". El promedio de las eficiencias generadas en cada corrida es de: 0.677, con una desviación estándar de 0.0675.

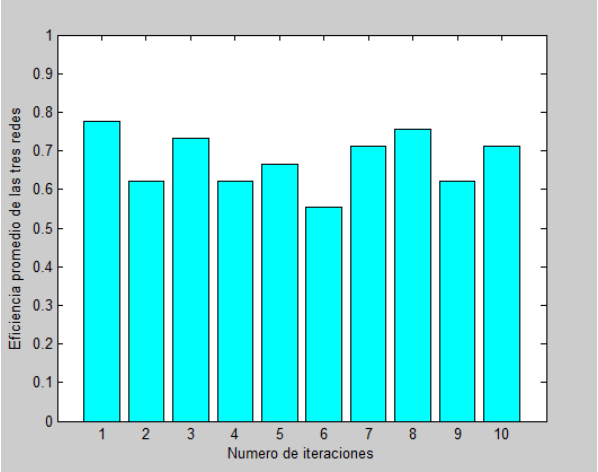


Figura 4.3 Eficiencias promedio generadas a partir del archivo de entrenamiento "archivoDeEntrenamientoNumeros45.txt" para 10 iteraciones.

	1	2	3	
1	3	1	1	5
2	0	5	0	5
3	1	0	4	5
	4	6	5	12

a.1) Red 1

	1	2	3	
1	3	0	2	5
2	0	5	0	5
3	0	0	5	5
	3	5	7	13

a.2) Red 2

	1	2	3	
1	1	0	4	5
2	0	5	0	5
3	1	0	4	5
	2	5	8	10

a.3) Red 3

Eficiencia promedio = 0.777

a) Iteración 1.

Figura 4.4 Matrices de confusión del mejor caso – caso 1- experimento 1.1.

4.1.3 Caso 1- Experimento 2

La figura A.3 del apéndice A, los incisos a) – o) muestra las matrices de confusión de cada experimento realizado con el número de neuronas en las capas internas. La figura 4.5 muestra las eficiencias promedio para diferente número de neuronas, y la figura 4.6 presentan las matrices de confusión del mejor caso.

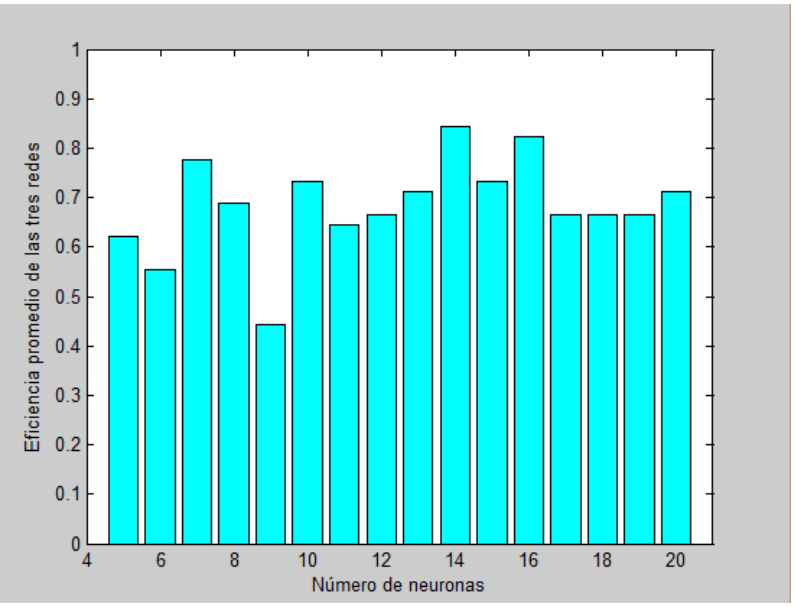


Figura 4.5. Eficiencias promedio del "ArchivoComparacion_45x19_E2.txt" para diferente número de neuronas.

	1	2	3	
1	4	0	1	5
2	0	5	0	5
3	1	0	4	5
	5	5	5	13

j.1) Red 1

	1	2	3	
1	5	0	0	5
2	0	5	0	5
3	1	1	3	5
	6	6	3	13

j.2) Red 2

	1	2	3	
1	4	0	1	5
2	0	5	0	5
3	1	1	3	5
	5	6	4	12

j.3) Red 3

Eficiencia promedio = 0.844

j) Con 14 neuronas.

Figura 4.6 Matrices de confusión del mejor caso- caso 1- experimento 2.

4.1.4 Caso 1- Experimento 2.1

La figura 4.7 muestra las eficiencias resultantes de la red en 10 veces que se re-entrenó. Los tres archivos que se trabajaron en este experimento fueron generados a partir del archivo de entrenamiento "ArchivoComparacion_45x19_E2.txt" con 14 neuronas, puesto que ese fue el mejor caso en el experimento 2.

El promedio de las eficiencias generadas en cada corrida es de: 0.6374, con una desviación estándar de 0.0242.

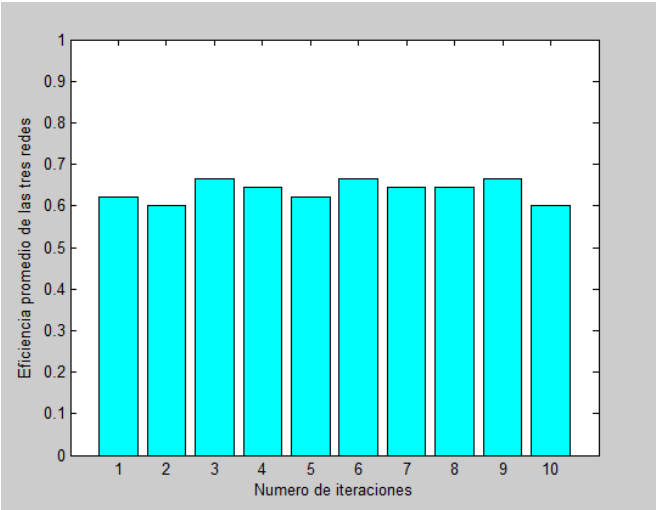


Figura 4.7 Eficiencias a partir del archivo de entrenamiento "ArchivoComparacion_45x19_E2.txt" para 10 iteraciones.

La Figura A.4 del apéndice A, muestra las matrices de confusión de cada una de las iteraciones, realizadas con 14 neuronas, puesto que este fue el mejor número de neuronas en el experimento 2.

Las matrices de confusión del mejor caso en las iteraciones de este experimento se muestran en la figura 4.8.

	1	2	3	
1	2	1	2	5
2	0	5	0	5
3	1	1	3	5
	3	7	5	10

c.1) Red 1

	1	2	3	
1	5	0	0	5
2	0	5	0	5
3	1	4	0	5
	6	9	0	10

c.2) Red 2

	1	2	3	
1	5	0	0	5
2	0	5	0	5
3	1	4	0	5
	6	9	0	10

c.3) Red 3

Eficiencia promedio = 0.666

c) Iteración 3.

	1	2	3	
1	3	1	1	5
2	1	4	0	5
3	1	3	1	5
	5	8	2	8

f.1) Red 1

	1	2	3	
1	5	0	0	5
2	0	5	0	5
3	1	4	0	5
	6	9	0	10

f.2) Red 2

	1	2	3	
1	4	0	1	5
2	0	5	0	5
3	1	1	3	5
	5	6	4	12

f.3) Red 3

Eficiencia promedio = 0.666

f) Iteración 6.

	1	2	3	
1	1	2	2	5
2	0	4	1	5
3	1	0	4	5
	2	6	7	9

i.1) Red 1

	1	2	3	
1	4	1	0	5
2	0	5	0	5
3	1	4	0	5
	5	10	0	9

i.2) Red 2

	1	2	3	
1	5	0	0	5
2	0	5	0	5
3	1	2	2	5
	6	7	2	12

i.3) Red 3

Eficiencia promedio = 0.666

i) Iteración 9.

Figura 4.8 Matrices de confusión del mejor caso– caso 1- experimento 2.1.

4.1.5 Caso 2- Experimento 3

La figura 4.9 muestra las eficiencias resultantes de la red en 10 veces que se re-entrenó. Los tres archivos que se trabajaron en este experimento fueron generados a partir del archivo de entrenamiento "archivoDeEntrenamientoNumeros585.txt", es decir, cada uno de los archivos generados contenía una matriz de números de 195 filas y 27 columnas. El promedio de eficiencias es: 0.750, con una desviación estándar de 0.0136.

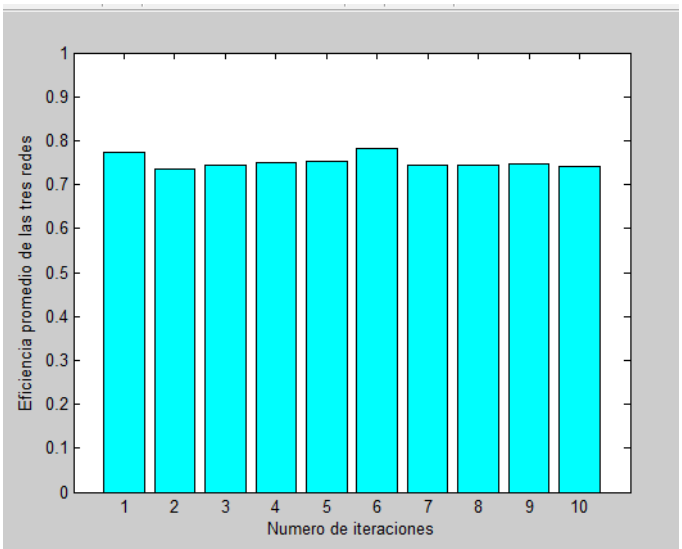


Figura 4.9 Eficiencias a partir del archivo de entrenamiento "archivoDeEntrenamientoNumeros585.txt" para 10 iteraciones.

La Figura A.5 del apéndice A, muestra las matrices de confusión de cada una de las iteraciones, realizadas con 20 neuronas, puesto que este fue el mejor número de neuronas en el experimento 1.

Las matrices de confusión del mejor caso del experimento 3 se muestran en la figura 4.10.

	1	2	3	
1	53	1	11	65
2	0	64	1	65
3	8	30	27	65
	61	95	39	144

f.1) Red 1

	1	2	3	
1	54	6	5	65
2	0	60	5	65
3	11	16	38	65
	65	82	48	152

f.2) Red 2

	1	2	3	
1	57	3	5	65
2	0	63	2	65
3	6	18	41	65
	63	84	48	161

f.3) Red 3

Eficiencia promedio = 0.781

f) Iteración 6.

Figura 4.10 Matrices de confusión del mejor caso – caso 2- experimento 3.

4.1.6 Caso 2- Experimento 4

La figura 4.11 muestra las eficiencias resultantes de la red en 10 veces que se re-entreno. Los tres archivos que se trabajaron en este experimento fueron generados a partir del archivo de entrenamiento "archivoEntrenamiento585x19_E2.txt", cuya matriz de datos es de 585 filas y 19 columnas. El promedio de estás eficiencias es: 0.772 y su desviación estándar es de: 0.200.

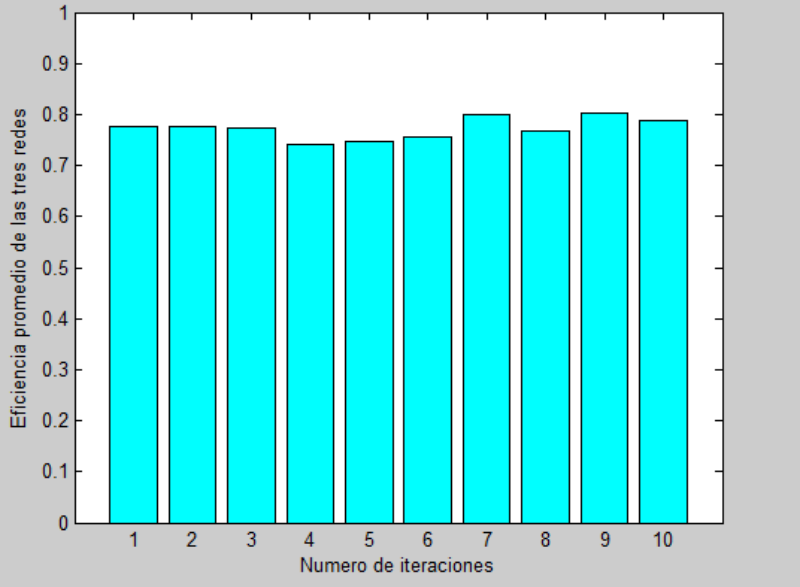


Figura 4.11 Eficiencias a partir del archivo de entrenamiento "archivoEntrenamiento585x19_E2.txt" para 10 iteraciones.

La Figura A.6 del apéndice A, muestra las matrices de confusión de cada una de las iteraciones, realizadas con 14 neuronas, puesto que este fue el mejor número de neuronas en el experimento 2.

Las matrices de confusión del mejor caso del experimento 4 se muestran en la figura 4.12.

	1	2	3	
1	49	0	16	65
2	0	65	0	65
3	11	13	41	65
	60	78	57	155

i.1) Red 1

	1	2	3	
1	63	0	2	65
2	0	65	0	65
3	7	25	33	65
	70	90	35	161

i.2) Red 2

	1	2	3	
1	55	0	10	65
2	0	64	1	65
3	12	18	35	65
	67	82	46	154

i.3) Red 3

Eficiencia promedio = 0.803

i) Iteración 9.

Figura 4.12 Matrices de confusión – caso 2- experimento 4.

4.2 Análisis de resultados

De los cuatro grandes experimentos que se realizaron, se puede observar que la red neuronal clasificó mejor el parpadeo correspondiente al ojo derecho, y confundió en más ocasiones la clase del ojo izquierdo con la clase del parpadeo de ambos ojos. Analizando los datos crudos, nos dimos cuenta de que los parámetros arrojados son muy parecidos, y esto se debe a que cuando se tomaron los EEG, los sujetos a los que se les puso la diadema Emotiv- EPOC en la etapa de recolección de datos podían realizar sin problemas la sesión de parpadeo de ojo derecho, sin embargo, cuando se les pedía parpadear el ojo izquierdo, por inercia y sin voluntad alguna de hacerlo, parpadeaban un poco el ojo derecho también, y no podían únicamente parpadear el ojo izquierdo sin provocar movimiento en el ojo derecho.

De igual manera, para el parpadeo de ambos ojos, los individuos a los que se les tomaron los EEG lo realizaban sin problemas, sin embargo, algunas de las matrices de confusión muestran que la red neuronal confunde la clase de ambos ojos, con la del ojo izquierdo, pero esto sucede en menor medida que la confusión de la clase de ojo izquierdo con ambos ojos, y esto se debe a lo descrito en el párrafo anterior.

De acuerdo a lo mencionado anteriormente, se puede deducir que el porcentaje erróneo en la clasificación, no únicamente es cuestión de la red neuronal, sino también de los movimientos del ojo derecho que por inercia se producían cuando se intentaba parpadear únicamente el ojo izquierdo en la toma de los EEG.

Capítulo V

Conclusiones y trabajo futuro

En este capítulo se presentan las conclusiones e ideas para trabajo futuro obtenidas a partir del trabajo de tesis presentado en este documento.

5.1 Conclusiones

En el presente trabajo se propuso aumentar de manera potencial, la capacidad de comunicación y movimiento de las personas cuadripléjicas a través de la detección del parpadeo de los ojos, con la finalidad de producir el desplazamiento de una silla de ruedas en direcciones específicas. El método propuesto se basa principalmente en cuatro etapas: la primera etapa es la adquisición de la señal, la segunda es la extracción de características, la tercera es la clasificación, y la última es el movimiento del robot.

La adquisición de la señal se llevó a cabo con la diadema Emotiv-EPOC, debido a que es un dispositivo de bajo costo y sirve muy bien para los fines que perseguimos, es decir, tomar EEG y después, realizar su posterior análisis. Posteriormente, en la extracción de características se utilizaron 3 sub-bandas de frecuencia: *delta*, *theta* y *alpha*, con cuatro medidas estadísticas: media, mediana, varianza y entropía. De igual forma, para la extracción de características se utilizó la Transformada Discreta Wavelet, cuyo resultado fue un conjunto de vectores característicos, mismos que sirvieron como entrada a la red neuronal de retro-propagación para clasificar tres clases: clase 001 (ambos ojos- si el robot está en estado de reposo avanza hacia adelante, y si se encuentra en movimiento se detiene.), clase 010 (ojo derecho- vuelta a la derecha), clase 100 (ojo izquierdo- vuelta a la izquierda). De acuerdo a los resultados obtenidos en la clasificación con la extracción de características para los dos grandes casos que aplicamos, demostraron que las medidas estadísticas fueron elegidas de manera correcta para este problema, y que las bases de

datos con ventanas entrecortadas y reducidas mejoraron notablemente los resultados del primer caso. El mejor caso se obtuvo utilizando una red neuronal alimentada con 16 características y con 14 nodos escondidos, la cual obtuvo una eficiencia promedio de 0.772 con una desviación estándar de 0.096. Como prototipo de la silla de ruedas, armamos un robot en forma de perro haciendo uso del Lego Mindstorms NXT, para simular los movimientos, de acuerdo a la clasificación de la red neuronal.

Este trabajo de investigación aporta evidencia de que las medidas estadísticas y el uso de la Transformada Discreta Wavelet en la extracción de características, así como el uso de la red neuronal ayudan a reconocer el parpadeo de los ojos de una persona, y que tratándose de un individuo cuadripléjico, podemos contribuir a que mueva una silla de ruedas, y por tanto, logre llegar a ser un ser humano más independiente.

5.2 Trabajo futuro

Existen varias tareas en el método propuesto en las cuales se puede seguir trabajando. La adquisición de señales en este trabajo de tesis se realizó con la diadema Emotiv- EPOC, sin embargo, esta etapa se podría realizar con equipo más sofisticado que pueda captar las señales EEG con mayor precisión.

La extracción de características de este trabajo de tesis se realizó con las bandas *delta*, *theta* y *alpha* y con las medidas estadísticas: media, mediana, entropía y varianza. Sin embargo, se pueden utilizar otras medidas estadísticas para obtener vectores característicos que den paso para hacer una buena clasificación a la hora de reconocer el parpadeo de cada uno de los ojos.

La clasificación de los movimientos para que la silla de ruedas realice la trayectoria deseada por el usuario en tiempo real, sería una actividad de mucha utilidad, puesto que la persona cuadripléjica podría desplazarse en el mismo instante que parpadee el ojo deseado. Implementar un sistema de clasificación en tiempo real implica mayor infraestructura de software y hardware, así como un análisis detallado del procesamiento de la señal en el tiempo.

Finalmente, el movimiento del robot que se realiza en este trabajo de investigación, sería posible llevarlo a cabo en una silla de ruedas real con el equipo necesario.

Bibliografía

Adam, I. (2013). *Complex Wavelet Transform: application to denoising.*

Agatonovic-Kustrin, S. (2000). Basic concepts of artificial neural network (ANN) modeling and its application in pharmaceutical research. *ELSEVIER*, 717-727.

Burrus Sidney, C., Gopinath, R. A., & Guo, H. (1998). *Introducion to Wavelets and Wavelet Transform.* Houston, Texas: Prentice Hall.

Campbell, A., Choudhury, T., Hu, S., Lu, H., K. Mukerjee, M., Rabbi, M., & Raizada , R. (2010). NeuroPhone: brain-mobile phone interface using a wireless EEG headset. *MobiHeld 2010*.

D. Paarmann, L. (2001). *Design and Analysis of Analog Filters. A Signal Processing Perspective.* Springer.

Emotiv. (n.d.). *TestBench User Manual.* Retrieved from TestBench User Manual: http://energylab.hpa.edu/public/brain/Emotiv%20Research%20Edition%20SDK%20v2.0.0.20/doc/TestBench%20Manual.pdf

Emotiv, I. (2014). *Emotiv* . Retrieved from Emotiv : https://emotiv.com/epoc.php

g.tec medical engineering. (2015). Retrieved from g.tec medical engineering: http://www.gtec.at/

Gabor, D. (1946). Theory of communication. Part 1: The analysis of information. Electrical. *Journal of the Institution of Radio and Communication Engineering*, 429-441.

Galvão Barbosa, A. O., Zacarias Freitas, D., de Moura Guedes, J. Q., & Meggiolaro, M. A. (2013). Implementation of a wheelchair control using a four-command brain computer interface. *22nd International Congress of Mechanical Engineering*.

Gómez Gil, P. (2014). *Transformadas Wavelets* . http://ccc.inaoep.mx/~pgomez/cursos/pds/index_archivos/Page611.htm.

Informer Technologies, I. (2015). *Emotiv TestBench*. Retrieved from Emotiv TestBench: http://emotiv-testbench.software.informer.com/

Juárez Guerra, E., Alarcon Aquino, V., & Gómez Gil, P. (2014). Epilepsy Seizure Detection in EEG Signals Using Wavelet Transforms and Neural Networks. In K. Elleithy, & T. Sobh, *New Trends in Networking, Computing, E-learning, Systems Sciences, and Engineering* (pp. 261-269).

Juárez Guerra, E., Alarcon Aquino, V., & Gómez Gil, P. (2014). Epilepsy Seizure Detection in EEG Signals Using Wavelet Transforms and Neural Networks. In K. Elleithy, & T. Sobh, *New Trends in Networking, Computing, E-learning, Systems Sciences, and Engineering* (pp. 261-269). Springer.

Lego Mindstorms. (2015). Retrieved from Lego Mindstorms: http://www.lego.com/es-ar/mindstorms/?domainredir=mindstorms.lego.com

Liu, Y., Jiang, X., Cao, T., Wan, F., Mak, P. U., Mak, P.-I., & I Vai, M. (2012, Julio 2-4). Implementation of SSVEP Based BCI with Emotiv EPOC. *Virtual Environments Human-Computer Interfaces and Measurement Systems (VECIMS)*, 34-37.

Mahajan, R., & I. Morshed, B. (2015). Unsupervised Eye Blink Artifact Denoising of EEG Data with Modified Multiscale Sample Entropy, Kurtosis, and Wavelet-ICA. *Biomedical and Health Informatics*, 158-165.

Marín, L., Vallés, M., Soriano, Á., Valera, Á., & Albertos, P. (2014). Event-Based Localization in Ackermann Steering Limited Resource Mobile Robots.

MathWorks, I. (2015). *MathWorks*. Retrieved from MathWorks: http://www.mathworks.com/

MechLab. (2013). Model Based Design of a Self-balancing Vehicle: A Mechatronic System Design Case Study. In T. Březina, & R. Jabloński, *Mechatronics 2013* (pp. 869-876).

Morales Flores, E., Ramírez Cortés, J. M., Gómez Gil, P., & Alarcón, A. V. (2012). Brain Computer Interface Development Based on Recurrent Neural Networks and ANFIS Systems. In P. Milin, & O. Castillo, *Soft Compting Applications in Optimization, Control and Recognition* (pp. 215-236). Springer.

Moreno, L. Á., Peña, C. A., & Gualdrón, O. E. (2014). Development of neuromarketing system using EMOTIV-EPOC device. *Redes de Ingeniería*, 6-15.

Morlet, J., Arens, G., Fourgeau, E., & Glard, D. (1982). Wave propagation and sampling theory—Part I: Complex signal and scattering in multilayered media. *Geophysics*, 203.

Nasreddine Belkacem, A., Shin, D., Kambara, H., Yoshimura, N., & Koike, Y. (2015). Online classification algorithm for eye-movement-basedcommunication systems using two temporal EEG sensors. *Elsevier*, 40-47.

Nazari Kousarrizi, M. R., Asadi Ghanbari, A., Teshnehlab, M., Aliyari, M., & Gharaviri, A. (2009). Feature Extraction and Classification of EEG Signals using Wavelet Transform, SMV and Artificial Neural Networks for Brain Computer Interface. *IEEE Computer Society*, 352-355.

NXT, L. M. (2011). *nxtprograms.com*. Retrieved from nxtprograms.com: http://www.nxtprograms.com/puppy/index.html

Pásztor, A. (2014). Gathering simulation of real robot swarm. *Tehnički vjesnik*, 1073-1080.

Pérez Águila, R. (2012). *Una Introducción al Cómputo Neuronal Artificial.* Argentina: El Cid.

Refaeilzadeh, P., Tang, L., & Liu, H. (2009). *Encyclopedia of Database Systems.* Springer.

Reyes Rueda, A. (2013, Julio). Interfaz cerebro computadora mediante la clasificación de señales electroencefalográficas. Colombia.

Rumelhart, D. E., Geoffrey, E. H., & R.J, W. (1986). *"Learning internal representations by Error Propagation," In Parallel Distributed Processing: Explorations in the Microstructure of Cognition.*

Sheoran, M., Kumar, S., & Kumar, A. (2014). Wavelet-ICA based Denoising of Electroencephalogram Signal. *International Journal of Information & Computation Technology*, 1205-1210.

Silveira Duarte, A., L'Erario, A., Takeshi Endo, A., Canhadas Genvigir, E., de Lima, F. C., Gonçalves, J. A., . . . da Silva Rocha, J. (2014). Teaching software project management by using Lego Mindstorms robot. *Information Systems and Technologies (CISTI), 2014 9th Iberian Conference on*, 1-4.

Svozil, D., KvasniEka, V., & Pospichal, J. (1997). Introduction to multi-layer feed-forward neural networks. *ELSEVIER*.

Talamillo García, T. (2011). Manual básico para enfermeros en electroencefalografía. *Enfermería Docente*, 29-33.

Tan, L. (2008). *Digital Signal Processing. Fundamentals and Applications.* Elsevier.

Vision, I. o. (2015). *RWTH- Mindstorms NXT Toolbox*. Retrieved from RWTH-Mindstorms NXT Toolbox: http://www.mindstorms.rwth-aachen.de/

Vourvopoulos, A., & Liarokapis, F. (2011). Brain-controlled NXT Robot: Tele-operating a robot through brain electrical activity. *Third International Conference on Games and Virtual Worlds for Serious Applications IEEE*, 140-143.

Werbos, P. (1974). *Beyond Regression: New tools for prediction and Analysis in the Behavioral Sciences.* Harvard University: Ph.D. thesis.

Apéndices

Apéndice A

Matrices de confusión

Caso 1- Experimento 1

	1	2	3	
1	0	5	0	5
2	0	5	0	5
3	0	5	0	5
	0	15	0	5

a.1) Red 1

	1	2	3	
1	1	4	0	5
2	0	5	0	5
3	0	5	0	5
	1	14	0	6

a.2) Red 2

	1	2	3	
1	1	4	0	5
2	0	5	0	5
3	0	2	3	5
	1	11	3	9

a.3) Red 3

Eficiencia promedio = 0.444

a) Con 5 neuronas.

	1	2	3	
1	3	0	2	5
2	0	3	2	5
3	1	0	4	5
	4	3	8	10

b.1) Red 1

	1	2	3	
1	2	0	3	5
2	2	2	1	5
3	0	1	4	5
	4	3	8	8

b.2) Red 2

	1	2	3	
1	4	1	0	5
2	0	5	0	5
3	2	1	2	5
	6	7	2	11

b.3) Red 3

Eficiencia promedio= 0.644

b) Con 6 neuronas.

	1	2	3	
1	3	2	0	5
2	0	5	0	5
3	1	0	4	5
	4	7	4	12

c.1) Red 1

	1	2	3	
1	3	0	2	5
2	2	2	1	5
3	0	0	5	5
	5	2	8	10

c.2) Red 2

	1	2	3	
1	3	1	1	5
2	0	3	2	5
3	1	1	3	5
	4	5	6	9

c.3) Red 3

Eficiencia promedio = 0.688

c) Con 7 neuronas.

	1	2	3	
1	1	2	2	5
2	1	0	4	5
3	1	1	3	5
	3	3	9	4

d.1) Red 1

	1	2	3	
1	2	1	2	5
2	1	3	1	5
3	0	0	5	5
	3	4	8	10

d.2) Red 2

	1	2	3	
1	3	2	0	5
2	0	4	1	5
3	1	1	3	5
	4	7	4	10

d.3) Red 3

Eficiencia promedio = 0.533

d) Con 8 neuronas.

	1	2	3	
1	3	1	1	5
2	0	5	0	5
3	1	0	4	5
	4	6	5	12

e.1) Red 1

	1	2	3	
1	2	0	3	5
2	0	5	0	5
3	1	0	4	5
	3	5	7	11

e.2) Red 2

	1	2	3	
1	1	1	3	5
2	0	4	1	5
3	2	1	2	5
	3	6	6	7

e.3) Red 3

Eficiencia promedio = 0.666

e) Con 9 neuronas.

	1	2	3	
1	3	0	2	5
2	0	5	0	5
3	2	0	3	5
	5	5	5	11

f.1) Red 1

	1	2	3	
1	3	0	2	5
2	0	5	0	5
3	0	1	4	5
	3	6	6	12

f.2) Red 2

	1	2	3	
1	0	2	3	5
2	0	5	0	5
3	1	1	3	5
	1	8	6	8

f.3) Red 3

Eficiencia promedio = 0.688

f) Con 10 neuronas.

	1	2	3	
1	4	0	1	5
2	0	5	0	5
3	0	2	3	5
	4	7	4	12

g.1) Red 1

	1	2	3	
1	3	0	2	5
2	0	3	2	5
3	0	0	5	5
	3	3	9	11

g.2) Red 2

	1	2	3	
1	4	0	1	5
2	0	4	1	5
3	2	1	2	5
	6	5	4	10

g.3) Red 3

Eficiencia promedio = 0.733

g) Con 11 neuronas.

	1	2	3	
1	3	2	0	5
2	0	5	0	5
3	1	3	1	5
	4	10	1	9

h.1) Red 1

	1	2	3	
1	3	0	2	5
2	2	3	0	5
3	1	1	3	5
	6	4	5	9

h.2) Red 2

	1	2	3	
1	1	1	3	5
2	0	5	0	5
3	0	2	3	5
	1	8	6	9

h.3) Red 3

Eficiencia promedio = 0.600

h) Con 12 neuronas.

	1	2	3	
1	3	1	1	5
2	0	4	1	5
3	0	1	4	5
	3	6	6	11

i.1) Red 1

	1	2	3	
1	3	0	2	5
2	0	5	0	5
3	0	0	5	5
	3	5	7	13

i.2) Red 2

	1	2	3	
1	3	1	1	5
2	0	4	1	5
3	2	0	3	5
	5	5	5	10

i.3) Red 3

Eficiencia promedio = 0.755

i) Con 13 neuronas.

	1	2	3	
1	3	2	0	5
2	0	5	0	5
3	1	2	2	5
	4	9	2	10

j.1) Red 1

	1	2	3	
1	2	1	2	5
2	0	5	0	5
3	1	0	4	5
	3	6	6	11

j.2) Red 2

	1	2	3	
1	0	1	4	5
2	0	3	2	5
3	0	0	5	5
	0	4	11	8

j.3) Red 3

Eficiencia promedio = 0.644

j) Con 14 neuronas.

	1	2	3	
1	3	1	1	5
2	0	4	1	5
3	0	1	4	5
	3	6	6	11

k.1) Red 1

	1	2	3	
1	2	0	3	5
2	0	5	0	5
3	0	2	3	5
	2	7	6	10

k.2) Red 2

	1	2	3	
1	1	1	3	5
2	0	5	0	5
3	0	1	4	5
	1	7	7	10

k.3) Red 3

Eficiencia promedio = 0.688

k) Con 15 neuronas.

	1	2	3	
1	3	1	1	5
2	1	4	0	5
3	1	2	2	5
	5	7	3	9

l.1) Red 1

	1	2	3	
1	2	1	2	5
2	0	4	1	5
3	0	0	5	5
	2	5	8	11

l.2) Red 2

	1	2	3	
1	4	0	1	5
2	0	5	0	5
3	1	0	4	5
	5	5	5	13

l.3) Red 3

Eficiencia promedio = 0.733

l) Con 16 neuronas.

	1	2	3	
1	0	1	4	5
2	4	0	1	5
3	0	0	5	5
	4	1	10	5

m.1) Red 1

	1	2	3	
1	2	0	3	5
2	0	5	0	5
3	0	1	4	5
	2	6	7	11

m.2) Red 2

	1	2	3	
1	3	0	2	5
2	0	5	0	5
3	1	1	3	5
	4	6	5	11

m.3) Red 3

Eficiencia promedio = 0.599

m) Con 17 neuronas.

	1	2	3	
1	4	0	1	5
2	0	3	2	5
3	3	0	2	5
	7	3	5	9

n.1) Red 1

	1	2	3	
1	2	1	2	5
2	0	1	4	5
3	0	0	5	5
	2	2	11	8

n.2) Red 2

	1	2	3	
1	1	1	3	5
2	0	3	2	5
3	1	0	4	5
	2	4	9	8

n.3) Red 3

Eficiencia promedio = 0.555

n) Con 18 neuronas.

	1	2	3	
1	2	1	2	5
2	0	0	5	5
3	0	2	3	5
	2	3	10	5

ñ.1) Red 1

	1	2	3	
1	0	2	3	5
2	0	2	3	5
3	0	0	5	5
	0	4	11	7

ñ.2) Red 2

	1	2	3	
1	3	2	0	5
2	0	5	0	5
3	2	1	2	5
	5	6	4	10

ñ.3) Red 3

Eficiencia promedio= 0.488

ñ) Con 19 neuronas.

	1	2	3	
1	1	1	3	5
2	0	5	0	5
3	0	0	5	5
	1	6	8	11

o.1) Red 1

	1	2	3	
1	3	0	2	5
2	0	5	0	5
3	0	1	4	5
	3	6	6	12

o.2) Red 2

	1	2	3	
1	3	0	2	5
2	0	5	0	5
3	1	0	4	5
	4	5	6	12

o.3) Red 3

Eficiencia promedio = 0.777

o) Con 20 neuronas.

Figura A.1 Matrices de confusión para el número de neuronas en las capas internas- caso 1- experimento

Caso 1- Experimento 1.1

	1	2	3	
1	3	1	1	5
2	0	5	0	5
3	1	0	4	5
	4	6	5	12

a.1) Red 1

	1	2	3	
1	3	0	2	5
2	0	5	0	5
3	0	0	5	5
	3	5	7	13

a.2) Red 2

	1	2	3	
1	1	0	4	5
2	0	5	0	5
3	1	0	4	5
	2	5	8	10

a.3) Red 3

Eficiencia promedio = 0.777

a) Iteración 1.

	1	2	3	
1	1	3	1	5
2	0	5	0	5
3	0	2	3	5
	1	10	4	9

b.1) Red 1

	1	2	3	
1	3	0	2	5
2	0	3	2	5
3	1	0	4	5
	4	3	8	10

b.2) Red 2

	1	2	3	
1	4	0	1	5
2	0	3	2	5
3	3	0	2	5
	7	3	5	9

b.3) Red 3

Eficiencia promedio = 0.622

b) Iteración 2.

	1	2	3	
1	2	0	3	5
2	1	3	1	5
3	0	0	5	5
	3	3	9	10

c.1) Red 1

	1	2	3	
1	2	1	2	5
2	0	5	0	5
3	0	0	5	5
	2	6	7	12

c.2) Red 2

	1	2	3	
1	1	1	3	5
2	0	5	0	5
3	0	0	5	5
	1	6	8	11

c.3) Red 3

Eficiencia promedio = 0.733

c) Iteración 3.

	1	2	3	
1	4	1	0	5
2	0	5	0	5
3	2	3	0	5
	6	9	0	9

d.1) Red 1

	1	2	3	
1	3	0	2	5
2	0	4	1	5
3	1	1	3	5
	4	5	6	10

d.2) Red 2

	1	2	3	
1	1	0	4	5
2	0	4	1	5
3	1	0	4	5
	2	4	9	9

d.3) Red 3

Eficiencia promedio = 0.622

d) Iteración 4.

	1	2	3	
1	2	2	1	5
2	0	3	2	5
3	1	2	2	5
	3	7	5	7

e.1) Red 1

	1	2	3	
1	3	0	2	5
2	0	5	0	5
3	0	1	4	5
	3	6	6	12

e.2) Red 2

	1	2	3	
1	3	0	2	5
2	0	5	0	5
3	2	0	3	5
	5	5	5	11

e.3) Red 3

Eficiencia promedio = 0.666

e) Iteración 5.

	1	2	3	
1	4	1	0	5
2	0	4	1	5
3	0	3	2	5
	4	8	3	10

f.1) Red 1

	1	2	3	
1	3	0	2	5
2	2	3	0	5
3	0	4	1	5
	5	7	3	7

f.2) Red 2

	1	2	3	
1	4	1	0	5
2	0	3	2	5
3	4	0	1	5
	8	4	3	8

f.3) Red 3

Eficiencia promedio = 0.555

f) Iteración 6.

	1	2	3	
1	4	1	0	5
2	0	5	0	5
3	0	2	3	5
	4	8	3	12

g.1) Red 1

	1	2	3	
1	3	0	2	5
2	2	3	0	5
3	1	1	3	5
	6	4	5	9

g.2) Red 2

	1	2	3	
1	4	0	1	5
2	0	5	0	5
3	2	1	2	5
	6	6	3	11

g.3) Red 3

Eficiencia promedio = 0.711

g) Iteración 7.

	1	2	3	
1	3	2	0	5
2	0	4	1	5
3	3	0	2	5
	6	6	3	9

h.1) Red 1

	1	2	3	
1	3	0	2	5
2	0	5	0	5
3	0	1	4	5
	3	6	6	12

h.2) Red 2

	1	2	3	
1	4	0	1	5
2	0	5	0	5
3	1	0	4	5
	5	5	5	13

h.3) Red 3

Eficiencia promedio = 0.755

h) Iteración 8.

	1	2	3	
1	3	1	1	5
2	1	0	4	5
3	1	0	4	5
	5	1	9	7

i.1) Red 1

	1	2	3	
1	2	0	3	5
2	0	3	2	5
3	0	0	5	5
	2	3	10	10

i.2) Red 2

	1	2	3	
1	3	1	1	5
2	0	4	1	5
3	1	0	4	5
	4	5	6	11

i.3) Red 3

Eficiencia promedio = 0.622

i) Iteración 9.

	1	2	3	
1	2	1	2	5
2	0	5	0	5
3	0	0	5	5
	2	6	7	12

j.1) Red 1

	1	2	3	
1	3	0	2	5
2	1	4	0	5
3	1	1	3	5
	5	5	5	10

j.2) Red 2

	1	2	3	
1	4	0	1	5
2	0	3	2	5
3	2	0	3	5
	6	3	6	10

j.3) Red 3

Eficiencia promedio = 0.711

j) Iteración 10.

Figura A.2 Matrices de confusión – caso 1- experimento 1.1.

Caso 1- Experimento 2

	1	2	3	
1	3	0	2	5
2	0	3	2	5
3	1	0	4	5
	4	3	8	10

a.1) Red 1

	1	2	3	
1	2	1	2	5
2	0	5	0	5
3	2	1	2	5
	4	7	4	9

a.2) Red 2

	1	2	3	
1	5	0	0	5
2	1	4	0	5
3	3	2	0	5
	9	6	0	9

a.3) Red 3

Eficiencia promedio = 0.622

a) Con 5 neuronas.

	1	2	3	
1	2	1	2	5
2	4	1	0	5
3	4	0	1	5
	10	2	3	4

b.1) Red 1

	1	2	3	
1	5	0	0	5
2	0	5	0	5
3	1	4	0	5
	6	9	0	10

b.2) Red 2

	1	2	3	
1	5	0	0	5
2	0	5	0	5
3	1	3	1	5
	6	8	1	11

b.3) Red 3

Eficiencia promedio= 0.555

b) Con 6 neuronas.

	1	2	3	
1	2	1	2	5
2	0	5	0	5
3	0	0	5	5
	2	6	7	12

c.1) Red 1

	1	2	3	
1	5	0	0	5
2	1	4	0	5
3	1	3	1	5
	7	7	1	10

c.2) Red 2

	1	2	3	
1	4	0	1	5
2	0	5	0	5
3	1	0	4	5
	5	5	5	13

c.3) Red 3

Eficiencia promedio = 0.777

c) Con 7 neuronas.

	1	2	3	
1	4	0	1	5
2	0	2	3	5
3	1	0	4	5
	5	2	8	10

d.1) Red 1

	1	2	3	
1	3	0	2	5
2	0	5	0	5
3	0	2	3	5
	3	7	5	11

d.2) Red 2

	1	2	3	
1	3	0	2	5
2	0	5	0	5
3	1	2	2	5
	4	7	4	10

d.3) Red 3

Eficiencia promedio = 0.688

d) Con 8 neuronas.

	1	2	3	
1	3	0	2	5
2	0	3	2	5
3	1	2	2	5
	4	5	6	8

e.1) Red 1

	1	2	3	
1	2	1	2	5
2	4	1	0	5
3	5	0	0	5
	11	2	2	3

e.2) Red 2

	1	2	3	
1	2	0	3	5
2	0	5	0	5
3	1	2	2	5
	3	7	5	9

e.3) Red 3

Eficiencia promedio= 0.444

e) Con 9 neuronas.

	1	2	3	
1	3	0	2	5
2	0	3	2	5
3	0	0	5	5
	3	3	9	11

f.1) Red 1

	1	2	3	
1	4	0	1	5
2	1	1	3	5
3	1	0	4	5
	6	1	8	9

f.2) Red 2

	1	2	3	
1	4	0	1	5
2	0	5	0	5
3	1	0	4	5
	5	5	5	13

f.3) Red 3

Eficiencia promedio = 0.733

f) Con 10 neuronas.

	1	2	3	
1	1	2	2	5
2	0	5	0	5
3	0	3	2	5
	1	10	4	8

g.1) Red 1

	1	2	3	
1	4	0	1	5
2	0	5	0	5
3	1	3	1	5
	5	8	2	10

g.2) Red 2

	1	2	3	
1	4	0	1	5
2	0	5	0	5
3	1	2	2	5
	5	7	3	11

g.3) Red 3

Eficiencia promedio = 0.644

g) Con 11 neuronas.

	1	2	3	
1	2	0	3	5
2	0	5	0	5
3	0	1	4	5
	2	6	7	11

h.1) Red 1

	1	2	3	
1	4	0	1	5
2	0	2	3	5
3	2	0	3	5
	6	2	7	9

h.2) Red 2

	1	2	3	
1	2	0	3	5
2	0	5	0	5
3	0	2	3	5
	2	7	6	10

h.3) Red 3

Eficiencia promedio = 0.666

h) Con 12 neuronas.

	1	2	3	
1	3	1	1	5
2	0	5	0	5
3	0	2	3	5
	3	8	4	11

i.1) Red 1

	1	2	3	
1	4	0	1	5
2	0	5	0	5
3	1	4	0	5
	5	9	1	9

i.2) Red 2

	1	2	3	
1	2	0	3	5
2	0	5	0	5
3	0	0	5	5
	2	5	8	12

i.3) Red 3

Eficiencia promedio = 0.711

i) Con 13 neuronas.

	1	2	3	
1	4	0	1	5
2	0	5	0	5
3	1	0	4	5
	5	5	5	13

j.1) Red 1

	1	2	3	
1	5	0	0	5
2	0	5	0	5
3	1	1	3	5
	6	6	3	13

j.2) Red 2

	1	2	3	
1	4	0	1	5
2	0	5	0	5
3	1	1	3	5
	5	6	4	12

j.3) Red 3

Eficiencia promedio = 0.844

j) Con 14 neuronas.

	1	2	3	
1	2	2	1	5
2	0	5	0	5
3	1	0	4	5
	3	7	5	11

k.1) Red 1

	1	2	3	
1	5	0	0	5
2	0	5	0	5
3	2	0	3	5
	7	5	3	13

k.2) Red 2

	1	2	3	
1	3	0	2	5
2	0	5	0	5
3	0	4	1	5
	3	9	3	9

k.3) Red 3

Eficiencia promedio = 0.733

k) Con 15 neuronas.

	1	2	3	
1	2	1	2	5
2	0	4	1	5
3	1	0	4	5
	3	5	7	10

l.1) Red 1

	1	2	3	
1	5	0	0	5
2	0	5	0	5
3	0	1	4	5
	5	6	4	14

l.2) Red 2

	1	2	3	
1	3	0	2	5
2	0	5	0	5
3	0	0	5	5
	3	5	7	13

l.3) Red 3

Eficiencia promedio = 0.822

l) Con 16 neuronas.

	1	2	3	
1	2	1	2	5
2	0	2	3	5
3	2	0	3	5
	4	3	8	7

m.1) Red 1

	1	2	3	
1	3	0	2	5
2	0	5	0	5
3	1	1	3	5
	4	6	5	11

m.2) Red 2

	1	2	3	
1	5	0	0	5
2	0	5	0	5
3	1	2	2	5
	6	7	2	12

m.3) Red 3

Eficiencia promedio = 0.666

m) Con 17 neuronas.

	1	2	3	
1	1	1	3	5
2	0	4	1	5
3	1	0	4	5
	2	5	8	9

n.1) Red 1

	1	2	3	
1	5	0	5	5
2	0	5	0	5
3	1	1	3	5
	6	6	3	13

n.2) Red 2

	1	2	3	
1	3	1	1	5
2	0	5	0	5
3	1	4	0	5
	4	10	1	8

n.3) Red 3

Eficiencia promedio = 0.666

n) Con 18 neuronas.

	1	2	3	
1	2	0	3	5
2	0	4	1	5
3	1	0	4	5
	3	4	8	10

ñ.1) Red 1

	1	2	3	
1	4	0	1	5
2	0	5	0	5
3	2	2	1	5
	6	7	2	10

ñ.2) Red 2

	1	2	3	
1	2	1	2	5
2	0	5	0	5
3	1	1	3	5
	3	7	5	10

ñ.3) Red 3

Eficiencia promedio = 0.666

ñ) Con 19 neuronas.

	1	2	3	
1	1	1	3	5
2	0	4	1	5
3	1	0	4	5
	2	5	8	9

o.1) Red 1

	1	2	3	
1	5	0	0	5
2	0	5	0	5
3	1	3	1	5
	6	8	1	11

o.2) Red 2

	1	2	3	
1	5	0	0	5
2	0	5	0	5
3	1	2	2	5
	6	7	2	12

o.3) Red 3

Eficiencia promedio = 0.711

o) Con 20 neuronas.

Figura A.3 Matrices de confusión para el número de neuronas en las capas internas- caso 1- experimento 2.

Caso 1- Experimento 2.1

	1	2	3	
1	4	0	1	5
2	0	3	2	5
3	0	0	5	5
	4	3	8	12

a.1) Red 1

	1	2	3	
1	1	3	1	5
2	0	4	1	5
3	0	4	1	5
	1	11	3	6

a.2) Red 2

	1	2	3	
1	4	0	1	5
2	0	5	0	5
3	1	3	1	5
	5	8	2	10

a.3) Red 3

Eficiencia promedio = 0.622

a) Iteración 1.

	1	2	3	
1	3	1	1	5
2	0	5	0	5
3	1	0	4	5
	4	6	5	12

b.1) Red 1

	1	2	3	
1	4	0	1	5
2	0	4	1	5
3	1	3	1	5
	5	7	3	9

b.2) Red 2

	1	2	3	
1	1	4	0	5
2	0	5	0	5
3	0	5	0	5
	1	14	0	6

b.3) Red 3

Eficiencia promedio = 0.600

b) Iteración 2.

	1	2	3	
1	2	1	2	5
2	0	5	0	5
3	1	1	3	5
	3	7	5	10

c.1) Red 1

	1	2	3	
1	5	0	0	5
2	0	5	0	5
3	1	4	0	5
	6	9	0	10

c.2) Red 2

	1	2	3	
1	5	0	0	5
2	0	5	0	5
3	1	4	0	5
	6	9	0	10

c.3) Red 3

Eficiencia promedio = 0.666

c) Iteración 3.

	1	2	3	
1	1	2	2	5
2	0	2	3	5
3	1	0	4	5
	2	4	9	7

d.1) Red 1

	1	2	3	
1	5	0	0	5
2	0	5	0	5
3	1	2	2	5
	6	7	2	12

d.2) Red 2

	1	2	3	
1	3	0	2	5
2	0	3	2	5
3	0	1	4	5
	3	4	8	10

d.3) Red 3

Eficiencia promedio = 0.644

d) Iteración 4.

	1	2	3	
1	1	2	2	5
2	1	4	0	5
3	0	2	3	5
	2	8	5	8

e.1) Red 1

	1	2	3	
1	5	0	0	5
2	0	5	0	5
3	1	3	1	5
	6	8	1	11

e.2) Red 2

	1	2	3	
1	2	0	3	5
2	1	4	0	5
3	1	1	3	5
	4	5	6	9

e.3) Red 3

Eficiencia promedio = 0.622

e) Iteración 5.

	1	2	3	
1	3	1	1	5
2	1	4	0	5
3	1	3	1	5
	5	8	2	8

f.1) Red 1

	1	2	3	
1	5	0	0	5
2	0	5	0	5
3	1	4	0	5
	6	9	0	10

f.2) Red 2

	1	2	3	
1	4	0	1	5
2	0	5	0	5
3	1	1	3	5
	5	6	4	12

f.3) Red 3

Eficiencia promedio = 0.666

f) Iteración 6.

	1	2	3	
1	1	0	4	5
2	0	4	1	5
3	1	0	4	5
	2	4	9	9

g.1) Red 1

	1	2	3	
1	1	0	4	5
2	0	5	0	5
3	0	0	5	5
	1	5	9	11

g.2) Red 2

	1	2	3	
1	4	0	1	5
2	0	2	3	5
3	1	1	3	5
	5	3	7	9

g.3) Red 3

Eficiencia promedio = 0.644

g) Iteración 7.

	1	2	3	
1	3	1	1	5
2	0	2	3	5
3	0	0	5	5
	3	3	9	10

h.1) Red 1

	1	2	3	
1	4	0	1	5
2	0	5	0	5
3	1	1	3	5
	5	6	4	12

h.2) Red 2

	1	2	3	
1	2	0	3	5
2	0	2	3	5
3	1	1	3	5
	3	3	9	7

h.3) Red 3

Eficiencia promedio = 0.644

h) Iteración 8.

	1	2	3	
1	1	2	2	5
2	0	4	1	5
3	1	0	4	5
	2	6	7	9

i.1) Red 1

	1	2	3	
1	4	1	0	5
2	0	5	0	5
3	1	4	0	5
	5	10	0	9

i.2) Red 2

	1	2	3	
1	5	0	0	5
2	0	5	0	5
3	1	2	2	5
	6	7	2	12

i.3) Red 3

Eficiencia promedio = 0.666

i) Iteración 9.

	1	2	3	
1	2	2	1	5
2	0	2	3	5
3	1	2	2	5
	3	6	6	6

j.1) Red 1

	1	2	3	
1	5	0	0	5
2	0	5	0	5
3	1	4	0	5
	6	9	0	10

j.2) Red 2

	1	2	3	
1	5	0	0	5
2	0	5	0	5
3	1	3	1	5
	6	8	1	11

j.3) Red 3

Eficiencia promedio = 0.600

j) Iteración 10.

Figura A.4 Matrices de confusión – caso 1- experimento 2.1.

Caso 2- Experimento 3

	1	2	3	
1	56	1	8	65
2	1	60	4	65
3	11	23	31	65
	68	84	43	147

a.1) Red 1

	1	2	3	
1	57	1	7	65
2	0	61	4	65
3	16	21	28	65
	73	83	39	146

a.2) Red 2

	1	2	3	
1	59	1	5	65
2	0	65	0	65
3	9	21	35	65
	68	87	40	159

a.3) Red 3

Eficiencia promedio = 0.772

a) Iteración 1.

	1	2	3	
1	48	4	13	65
2	2	61	2	65
3	9	32	24	65
	59	97	39	133

b.1) Red 1

	1	2	3	
1	56	4	5	65
2	0	65	0	65
3	11	34	20	65
	67	103	25	141

b.2) Red 2

	1	2	3	
1	54	4	7	65
2	0	63	2	65
3	11	14	40	65
	65	81	49	157

b.3) Red 3

Eficiencia promedio= 0.736

b) Iteración 2.

	1	2	3	
1	42	3	20	65
2	2	53	10	65
3	7	17	41	65
	51	73	71	136

c.1) Red 1

	1	2	3	
1	46	2	17	65
2	0	54	11	65
3	9	13	43	65
	55	69	71	143

c.2) Red 2

	1	2	3	
1	52	3	10	65
2	0	60	5	65
3	8	13	44	65
	60	76	59	156

c.3) Red 3

Eficiencia promedio = 0.743

c) Iteración 3.

	1	2	3	
1	46	1	18	65
2	0	63	2	65
3	6	27	32	65
	52	91	52	141

d.1) Red 1

	1	2	3	
1	54	3	8	65
2	0	64	1	65
3	11	19	35	65
	65	86	44	153

d.2) Red 2

	1	2	3	
1	55	3	7	65
2	1	32	32	65
3	5	2	58	65
	61	37	97	145

d.3) Red 3

Eficiencia promedio = 0.750

d) Iteración 4.

	1	2	3	
1	44	1	20	65
2	1	61	3	65
3	11	17	37	65
	56	79	60	142

e.1) Red 1

	1	2	3	
1	55	4	6	65
2	1	64	0	65
3	13	30	22	65
	69	98	28	141

e.2) Red 2

	1	2	3	
1	52	2	11	65
2	0	64	1	65
3	5	19	41	65
	57	85	53	157

e.3) Red 3

Eficiencia promedio = 0.752

e) Iteración 5.

	1	2	3	
1	53	1	11	65
2	0	64	1	65
3	8	30	27	65
	61	95	39	144

f.1) Red 1

	1	2	3	
1	54	6	5	65
2	0	60	5	65
3	11	16	38	65
	65	82	48	152

f.2) Red 2

	1	2	3	
1	57	3	5	65
2	0	63	2	65
3	6	18	41	65
	63	84	48	161

f.3) Red 3

Eficiencia promedio = 0.781

f) Iteración 6.

	1	2	3	
1	41	1	23	65
2	0	63	2	65
3	9	28	28	65
	50	92	53	132

g.1) Red 1

	1	2	3	
1	56	4	5	65
2	0	39	26	65
3	12	6	47	65
	68	49	78	142

g.2) Red 2

	1	2	3	
1	55	4	6	65
2	0	60	5	65
3	6	13	46	65
	61	77	57	161

g.3) Red 3

Eficiencia promedio = 0.743

g) Iteración 7.

	1	2	3	
1	48	3	14	65
2	1	63	1	65
3	8	29	28	65
	57	95	43	139

h.1) Red 1

	1	2	3	
1	50	6	9	65
2	0	62	3	65
3	11	24	30	65
	61	92	42	142

h.2) Red 2

	1	2	3	
1	51	6	8	65
2	1	57	7	65
3	7	12	46	65
	59	75	61	154

h.3) Red 3

Eficiencia promedio = 0.743

h) Iteración 8.

	1	2	3	
1	50	0	15	65
2	1	62	2	65
3	5	34	26	65
	56	96	43	138

i.1) Red 1

	1	2	3	
1	51	5	9	65
2	1	63	1	65
3	11	28	26	65
	63	96	36	140

i.2) Red 2

	1	2	3	
1	52	7	6	65
2	0	64	1	65
3	7	15	43	65
	59	86	50	159

i.3) Red 3

Eficiencia promedio = 0.747

i) Iteración 9.

	1	2	3	
1	52	0	13	65
2	1	62	2	65
3	11	22	32	65
	64	84	47	146

j.1) Red 1

	1	2	3	
1	49	9	7	65
2	0	65	0	65
3	10	32	23	65
	59	106	30	137

j.2) Red 2

	1	2	3	
1	51	3	11	65
2	5	51	9	65
3	7	9	49	65
	63	63	69	151

j.3) Red 3

Eficiencia promedio= 0.741

j) Iteración 10.

Figura A.5 Matrices de confusión – caso 2- experimento 3.

Caso 2- Experimento 4

	1	2	3	
1	46	2	17	65
2	0	63	2	65
3	9	17	39	65
	55	82	58	148

a.1) Red 1

	1	2	3	
1	49	1	15	65
2	0	65	0	65
3	4	21	40	65
	53	87	55	154

a.2) Red 2

	1	2	3	
1	55	1	9	65
2	0	62	3	65
3	11	18	36	65
	66	81	48	153

a.3) Red 3

Eficiencia promedio = 0.777

a) Iteración 1.

	1	2	3	
1	45	1	19	65
2	0	57	8	65
3	7	15	43	65
	52	73	70	145

b.1) Red 1

	1	2	3	
1	57	1	7	65
2	0	65	0	65
3	6	25	34	65
	63	91	41	156

b.2) Red 2

	1	2	3	
1	53	3	9	65
2	0	62	3	65
3	11	15	39	65
	64	80	51	154

b.3) Red 3

Eficiencia promedio = 0.777

b) Iteración 2.

	1	2	3	
1	45	2	18	65
2	0	65	0	65
3	12	17	36	65
	57	84	54	146

c.1) Red 1

	1	2	3	
1	51	1	13	65
2	0	53	12	65
3	5	8	52	65
	56	62	77	156

c.2) Red 2

	1	2	3	
1	54	0	11	65
2	0	61	4	65
3	10	19	36	65
	64	80	51	151

c.3) Red 3

Eficiencia promedio = 0.774

c) Iteración 3.

	1	2	3	
1	52	2	11	65
2	0	64	1	65
3	13	26	26	65
	65	92	38	142

d.1) Red 1

	1	2	3	
1	42	6	17	65
2	0	65	0	65
3	4	31	30	65
	46	102	47	137

d.2) Red 2

	1	2	3	
1	53	2	10	65
2	0	62	3	65
3	11	15	39	65
	64	79	52	154

d.3) Red 3

Eficiencia promedio= 0.740

d) Iteración 4.

	1	2	3	
1	51	4	10	65
2	0	63	2	65
3	11	28	26	65
	62	95	38	140

e.1) Red 1

	1	2	3	
1	45	2	18	65
2	0	55	10	65
3	4	5	56	65
	49	62	84	156

e.2) Red 2

	1	2	3	
1	62	1	2	65
2	0	31	34	65
3	12	5	48	65
	74	37	84	141

e.3) Red 3

Eficiencia promedio = 0.747

e) Iteración 5.

	1	2	3	
1	47	3	15	65
2	1	64	0	65
3	10	20	35	65
	58	87	50	146

f.1) Red 1

	1	2	3	
1	46	2	17	65
2	1	64	0	65
3	5	27	33	65
	52	93	50	143

f.2) Red 2

	1	2	3	
1	52	4	9	65
2	0	64	1	65
3	10	18	37	65
	62	86	47	153

f.3) Red 3

Eficiencia promedio = 0.755

f) Iteración 6.

	1	2	3	
1	53	2	10	65
2	1	63	1	65
3	12	14	39	65
	66	79	50	155

g.1) Red 1

	1	2	3	
1	61	1	3	65
2	0	64	1	65
3	9	17	39	65
	70	82	43	164

g.2) Red 2

	1	2	3	
1	56	3	6	65
2	0	54	11	65
3	14	12	39	65
	70	69	56	149

g.3) Red 3

Eficiencia promedio = 0.800

g) Iteración 7.

	1	2	3	
1	53	2	10	65
2	4	61	0	65
3	18	41	6	65
	75	104	16	120

h.1) Red 1

	1	2	3	
1	60	2	3	65
2	0	64	1	65
3	6	13	46	65
	66	79	50	170

h.2) Red 2

	1	2	3	
	54	1	10	65
	0	65	0	65
	10	15	40	65
	64	81	50	159

h.3) Red 3

Eficiencia promedio= 0.767

h) Iteración 8.

	1	2	3	
1	49	0	16	65
2	0	65	0	65
3	11	13	41	65
	60	78	57	155

i.1) Red 1

	1	2	3	
1	63	0	2	65
2	0	65	0	65
3	7	25	33	65
	70	90	35	161

i.2) Red 2

	1	2	3	
1	55	0	10	65
2	0	64	1	65
3	12	18	35	65
	67	82	46	154

i.3) Red 3

Eficiencia promedio = 0.803

i) Iteración 9.

	1	2	3	
1	53	1	11	65
2	0	65	0	65
3	13	19	33	65
	66	85	44	151

j.1) Red 1

	1	2	3	
1	51	2	12	65
2	1	64	0	65
3	5	11	49	65
	57	77	61	164

j.2) Red 2

	1	2	3	
1	49	4	12	65
2	0	62	3	65
3	15	15	35	65
	64	81	50	146

j.3) Red 3

Eficiencia promedio = 0.788

j) Iteración 10.

Figura A.6 Matrices de confusión – caso 2- experimento 4.

Apéndice B

Manual de usuario

La ventana inicial después de realizar la ejecución del programa se puede apreciar en la figura B.1

Figura B.1 Ventana inicial del software.

Posteriormente, procedemos a dar clic en el botón de inicio (indicado en la figura anterior con una flecha), y aparecerá en nuestra pantalla una ventana como la de la figura B.1.

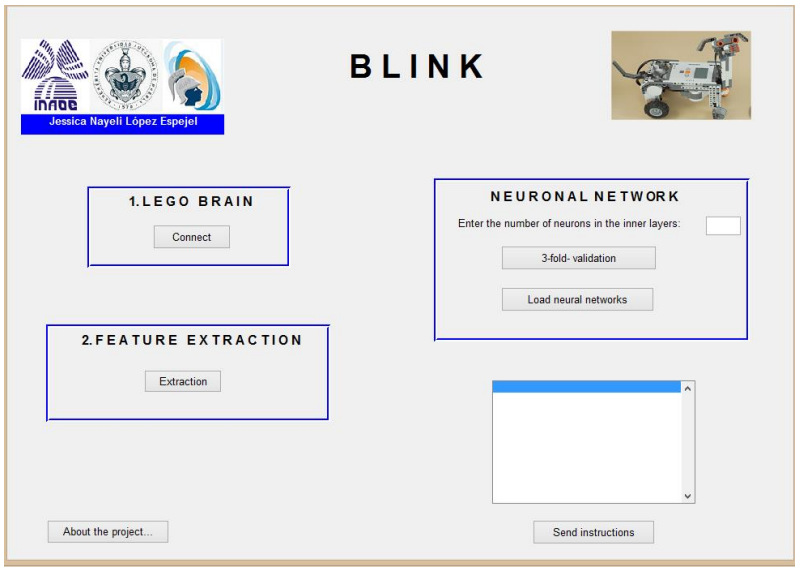


Figura B.2. Ventana de BLINK.

Comenzaremos la explicación de cada una de las funcionalidades del software por secciones, tal y como están indicadas en la Figura B.2.

1. Lego Brain

   En este paso, se oprime el botón indicado con la palabra "*Connect*" para realizar la conexión *bluetooth* con el Lego Mindstorms NXT. Después de pulsar el botón, aparecerá un cuadro de dialogo, como el de la Figura B.3 indicando que la conexión entre el lego y la computadora se está llevando a cabo, una vez que se ha terminado de conectar la barra de proceso estará completamente pintada de color verde, procedemos cerrarla y continuar con los siguientes pasos.

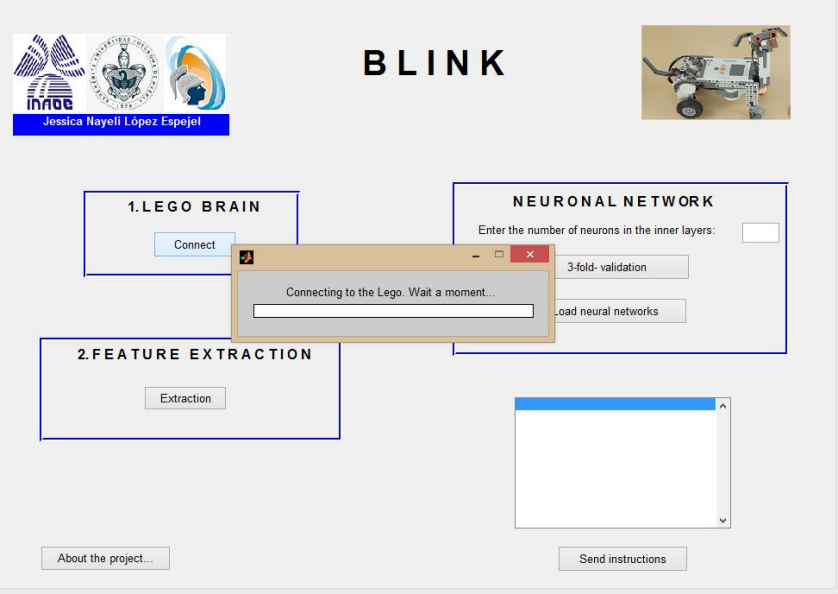


Figura B.3. Conectando la PC con Lego Mindstorms vía *bluetooth*.

2. Feature extraction.

   La Figura B.4 muestra el recuadro donde podremos hacer la extracción de características para los datos crudos de los EEG. Para iniciar con la extracción de características comenzamos oprimiendo el botón de "*Extraction*".

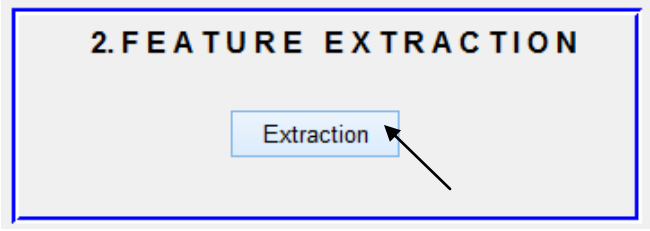
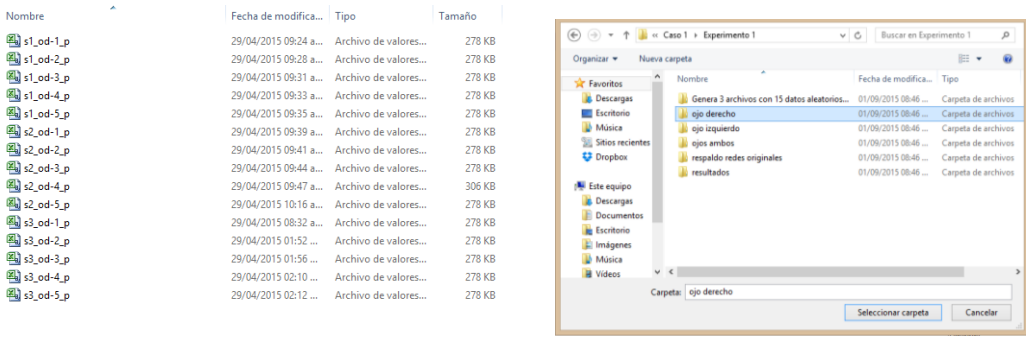


Figura B.4 Extracción de características.

Después de haber dado clic en el botón, aparecerá un cuadro de diálogo (Figura B.5 inciso b) en el que debemos elegir un directorio que tenga guardados únicamente archivos de los datos crudos de los EEG (Figura B.5 inciso a) a los cuales les deseemos extraer características.



a) Archivos de datos crudos de los EEG. b) Directorio elegido.

Figura B.5 Directorio de ojo derecho.

Una vez elegido el directorio con sus correspondientes archivos, nos pedirá que introduzcamos un nombre para nuestro nuevo archivo (Figura B.6). Este archivo que se procederá a guardar contiene los datos de la extracción de características de todos los archivos que estaban en nuestro directorio.

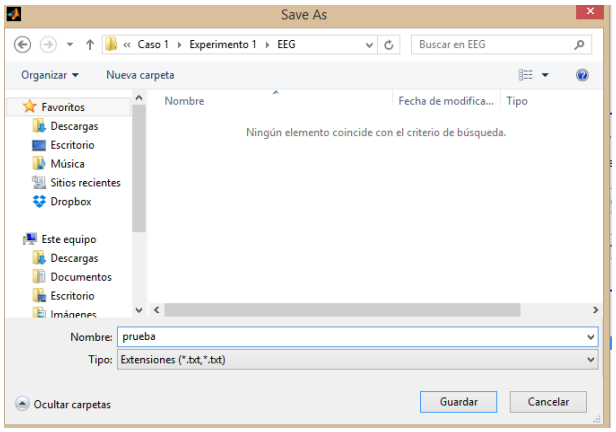


Figura B.6 Guardar nuevo archivo de la extracción de características.

Después de haber guardado el archivo, se podrá visualizar la información que contiene, y se mostrará algo parecido a la Figura B.7.

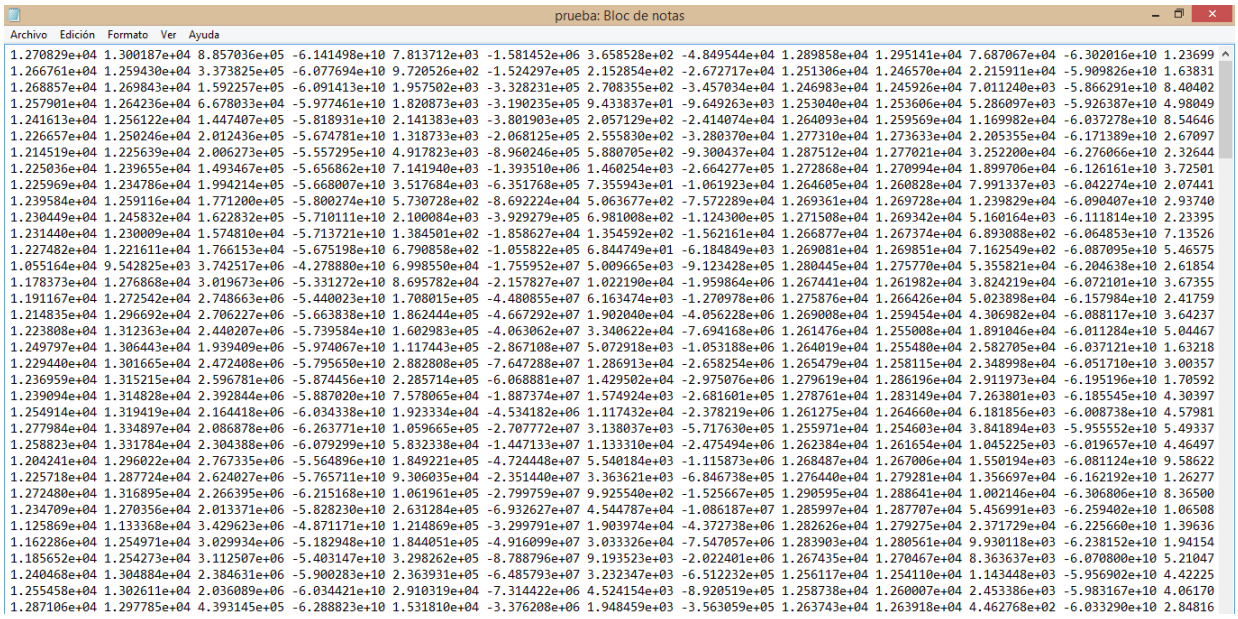


Figura B.7 Datos de la extracción de características.

3. Neural network.

   En este apartado como podemos ver en la Figura B.8 tenemos un cuadro en blanco donde podemos introducir el número de neuronas que se deseen para posteriormente entrenar la red neuronal utilizando el 3- *fold- validation* (primer botón del recuadro).

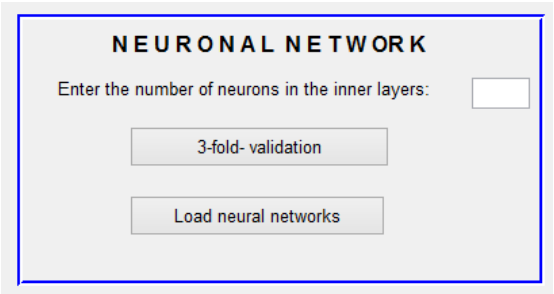


Figura B.8 Red neuronal

Si ya tenemos la red neuronal entrenada y guardada con los archivos deseados, entonces se puede presionar el botón indicado con la leyenda "*Load neural networks*", para cargar los datos de las redes que ya tenemos y proceder a realizar la clasificación de los comandos.

Después de haber entrenado a la red neuronal, oprimimos el botón de la parte inferior derecha indicado con la leyenda "*Send instructions*" (Figura B.9). Este botón no solamente realizará la clasificación de los comandos de la red neuronal, sino que también los mostrará en pantalla y el robot Lego Mindstorms realizará la trayectoria de las indicaciones que se clasificaron.

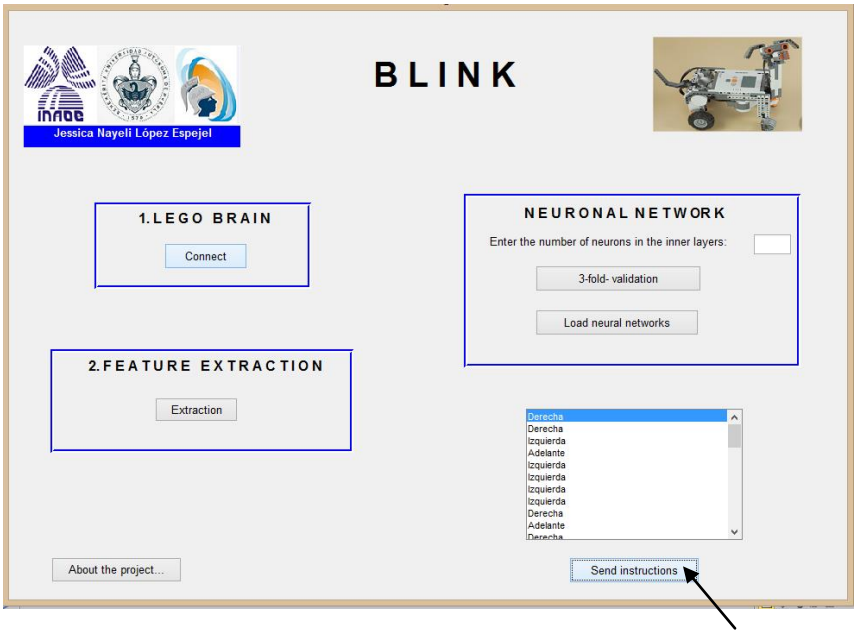


Figura B.9. Clasificador de comandos.

El robot que recibe las instrucciones y realiza la trayectoria indicada por la clasificación se muestra en la figura B.10.

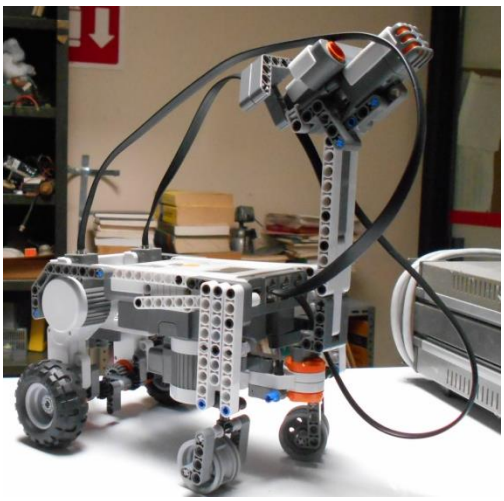


Figura B.10 Robot de Lego Mindstorms NXT.