

# Clasificación de señales encefalográficas mediante redes neuronales artificiales

Roberto Sepúlveda<sup>1</sup>, Oscar Montiel<sup>1</sup>, Gerardo Díaz<sup>1</sup>, Daniel Gutierrez<sup>1</sup> y Oscar Castillo<sup>2</sup>

<sup>1</sup> Instituto Politécnico Nacional-CITEDI, Tijuana, B.C., México

<sup>2</sup> Instituto Tecnológico de Tijuana, Tijuana, B.C., México

rsepulvedac@ipn.mx, oross@ipn.mx, gdiaz@citedi.mx, dgutierrez@citedi.mx, ocastillo@tectijuana.mx

**Resumen.** Para la clasificación de las señales del parpadeo y dolor muscular en el brazo derecho ocasionado por un agente externo, se proponen dos modelos de arquitecturas de redes neuronales artificiales, específicamente del tipo perceptrón multicapa y sistema de inferencia neurodifuso adaptativo, ambos modelos utilizan aprendizaje supervisado. Se utilizan series de tiempo obtenidas del parpadeo y electroencefalografías de 15 personas en el rango de 23 a 25 años de edad, para generar una base de datos que se divide en dos conjuntos de datos: entrenamiento y prueba. Los resultados experimentales en el dominio del tiempo y de la frecuencia, de 50 pruebas aplicadas a cada modelo de red, muestran que ambas propuestas de arquitecturas de redes neuronales producen resultados exitosos.

**Palabras clave.** EEG, BCI, interface cerebro-computadora, parpadeo, red neuronal artificial, FFT.

## Classification of Encephalographic Signals using Artificial Neural Networks

**Abstract.** For the signal classification of eye blinking and muscular pain in the right arm caused by an external agent, two models of artificial neural network architectures are proposed, specifically, the perceptron multilayer and an adaptive neurofuzzy inference system. Both models use supervised learning. The ocular and electroencephalographic time-series of 15 people in the range of 23 to 25 years of age are used to generate a data base which was divided into two sets: a training set and a test set. Experimental results in the time and frequency domain of 50 tests applied to each model show that both neural network architecture proposals for classification produce successful results.

**Keywords.** EEG, BCI, brain-computer interface, blink, artificial neural network, FFT.

## 1. Introducción

Durante generaciones, los seres humanos han soñado con la posibilidad de comunicarse e interactuar con las máquinas a través del pensamiento independiente o para crear dispositivos que pueden mirar en el interior de la mente y los pensamientos de las personas. Después de que Hans Berger en 1929 desarrollara el dispositivo que ahora se conoce como electroencefalograma, que tiene la capacidad de grabar potenciales eléctricos generados por la actividad cerebral, ha existido un gran interés en el desarrollo de métodos de comunicación entre cerebro y computadora basados en la actividad neuronal generada por el cerebro, nervios periféricos y músculos, dando con ello lugar al nacimiento de lo que hoy se conoce como interfaces computadora-cerebro "Brain-Computer Interface" (BCI) cuya meta principal es proporcionar al usuario canales de comunicación de la actividad cerebral que no dependa de los nervios periféricos y músculos con la finalidad de interpretar las emociones, sensaciones y deseos del usuario, lo cual tiene innumerables aplicaciones, siendo las más evidentes las dirigidas a personas con incapacidades motrices.

La electroencefalografía (EEG) se encarga de adquirir los datos generados por la actividad cerebral, regularmente se utiliza para el diagnóstico de

diversos trastornos psiquiátricos, trastornos metabólicos, trastornos del sueño, daños cerebrales, epilepsia, tumores, inflamación craneal, alcoholismo y algunos otros problemas que afectan directamente el cerebro humano.

Cada vez se hace más importante enfocarse en aplicaciones destinadas al mejoramiento de la salud del ser humano. Así mismo el empleo de conceptos de ingeniería han permitido alargar la vida de personas o en su defecto detectar a tiempo una variedad de enfermedades [1], debido a ésto y su potencial importancia, instituciones tales como la "National Center for Medical Rehabilitation Research" de el "National Institute of Child Health" han patrocinado talleres de trabajo para impulsar la tecnología BCI.

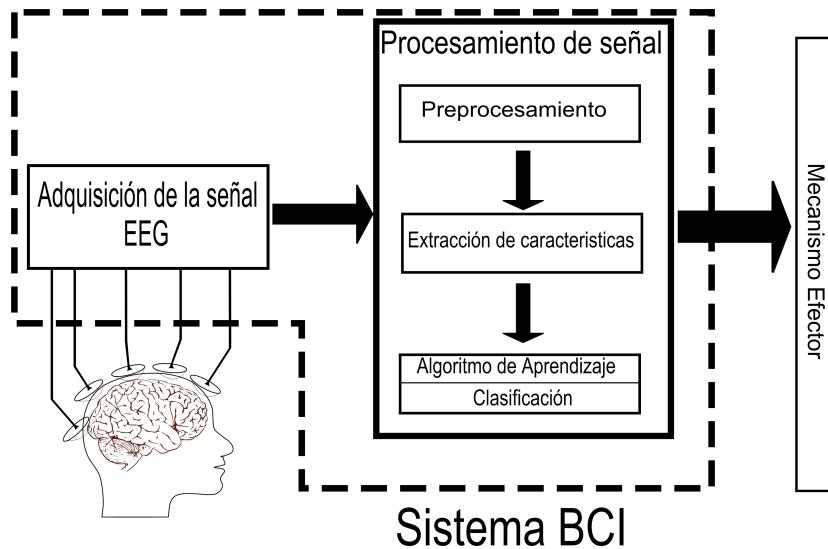
El uso de una BCI tiene un campo de aplicación muy amplio en la medicina, como lo son prótesis robóticas, estudios de patologías como la epilepsia, Alzheimer, y Parkinson, por mencionar sólo algunas. Esta tecnología, de manera general, se puede englobar en un sólo objetivo, crear canales de comunicación adecuados a las capacidades de estas personas. Muchas de ellas se ven con la problemática de no poder realizar ningún tipo de movimiento físico para poder comunicarse. Al presentar una discapacidad motriz las personas están aisladas de su entorno a pesar de que el cerebro goza de total funcionalidad, y es alrededor de estas características que se ha buscado dentro de un área muy específica para poder encontrar una solución, por medio de la interpretación de las señales EEG [8].

Es evidente que la detección oportuna del dolor en personas incapacitadas para expresarlo es de suma importancia, por lo que uno de los objetivos de este trabajo es estudiar y desarrollar un método que realice tal detección eficientemente, para ésto se desarrollan una serie de experimentos en donde se proponen dos arquitecturas de Redes Neuronales Artificiales (RNAs) y se prueban distintos algoritmos de entrenamiento con la finalidad de determinar los que realicen la clasificación más eficientemente. Para el desarrollo de los experimentos, en los sujetos de prueba, se provoca la señal EEG de dolor mediante un estímulo externo (pinchazo con un alfiler) directamente sobre una

extremidad corporal (brazo derecho). De investigaciones previas, se conoce que existe un incremento en las bandas de frecuencia denominadas alfa, beta, delta y theta, cuando el sujeto de prueba es sometido a algún tipo de dolor [4] [5]; por ejemplo, al aplicar una inyección de solución cásica se induce un dolor constante, provocando un aumento en magnitud de las señales EEG [13] [6]. En términos generales, se pretende que los resultados obtenidos sirvan de base para la clasificación eficiente y automatizada de señales de dolor, y que en un futuro puedan aplicarse en pacientes víctimas del dolor y que no poseen la capacidad de comunicarse de forma oral y/o corporal.

En la clasificación de señales EEG, se considera de suma importancia la identificación del parpadeo ocular, porque es uno de los artefactos más comunes que interfiere en el análisis encefalográfico. Como se indica en [21] se requiere tener un conocimiento adecuado de los artefactos en el análisis de EEG, porque puede repercutir en una pobre adquisición de la señal, refiriéndose a la señal EEG, y además de un fallo en el diagnóstico en el estudio de ciertas patologías como lo son Alzheimer y la epilepsia [18]. Para realizar una prueba se debe mantener al sujeto en un mismo estado con los ojos cerrados o abiertos para evitar incrementos de amplitud que afecten la prueba. Dado que la lectura EEG cambia drásticamente con los ojos abiertos y cerrados, es necesario mantener el estatus de los ojos de manera uniforme durante todo el experimento [4]. Por su peculiaridad de una amplitud constante en las lecturas EEG, el parpadeo es un artefacto que persiste en las lecturas encefalográficas, debido a que es un movimiento que ocurre naturalmente para realizar una lubricación del ojo y oxigenación de la córnea [16].

Uno de los impactos positivos que se ha tenido del estudio del parpadeo, es en los patrones del sueño, tal es el caso de los accidentes automovilísticos provocados por la fatiga del conductor como se menciona en [3], en el cual, utilizando el parpadeo (blink), el movimiento de cabeza y la expresión facial como entradas, se puede estimar el estado conciente del conductor, y por tanto anticipar un accidente carretero. Unas aplicaciones interesantes que se han realizado son entre otras, el considerar el parpadeo para hacer codificación



**Fig. 1.** El esquema funcional básico de cualquier sistema BCI, consiste de tres elementos principales: 1) el cerebro, 2) el sistema BCI que contiene un módulo de adquisición de señales y un módulo de procesamiento de las mismas, 3) el sistema que queremos actuar



**Fig. 2.** El sistema consiste de tres módulos: Adquisición, Procesamiento, y Clasificación de señales

de comandos para mecanismos efectores, tal es el caso de un teclado virtual [2], y el movimiento de una silla de ruedas [12].

Dada la importancia que representa el estudio del parpadeo, un segundo objetivo de este trabajo es el estudio y clasificación de la señal del parpadeo ocular. Hay antecedentes para desarrollar técnicas para su clasificación con cómputo inteligente, como es el caso de las RNAs [17], en donde se realiza una clasificación del parpadeo teniendo como entradas un análisis estadístico del mismo, para así poder reconocer este artefacto de una manera más intuitiva sin necesidad de tener a un especialista en la materia, como lo es un

neuroólogo. Hay que remarcar que a pesar de que existen estudios dedicados al parpadeo, que se involucran en la investigación de la señal EOG (Electrooculograma), ésta tiene una gran diferencia, con respecto a la adquisición de la señal de las señales EEG. En este estudio no se está tomando en cuenta los movimientos sacádicos del ojo, y sólo se toma el parpadeo del ojo y se clasifica, además que las lecturas se realizan con las referencias de los sensores del sistema 10-20, no con la del EOG [8].

Algunas de las técnicas actuales de clasificación empleadas en el estudio de las señales EEG son

las técnicas estadísticas, RNAs, Modelos Ocultos de Markov, y Máquinas de soporte vectorial [14].

Este trabajo está organizado como sigue: la sección 2 explica de manera general los elementos básicos de un sistema BCI, así como el esquema funcional que utilizamos para realizar la presente investigación, siendo el análisis de las señales electroencefalográficas el punto central, y se presenta en forma de pseudocódigo el procedimiento general que se siguió para realizar los experimentos mostrados en este trabajo. En la sección 3 se hace referencia a las arquitecturas de RNAs utilizadas, el método de aprendizaje, y los algoritmos de adaptación utilizados en el desarrollo de los experimentos, en esta sección también se proporcionan datos estadísticos de los resultados obtenidos de las pruebas realizadas. Finalmente en la sección 4 se dan las conclusiones de este trabajo.

## 2. Plataforma Experimental BCI

Las interfaces humano computadora o “human computer interfaces” (HCI) son de gran ayuda en la vida diaria de los seres humanos, algunos de estos ejemplos son: ratón de computadora, teclado de escritura, pluma digital, entre otros; sin embargo, existe un área en donde se necesita realizar una comunicación directa entre el cerebro y la computadora, a esta técnica se le conoce como interfaz cerebro-computadora o “brain computer interface” (BCI), se trata de un método de comunicación basado en la actividad neuronal generada por el cerebro y es independiente de las vías de salida normales de los nervios periféricos y músculos. La Fig. 1 describe de forma general el esquema funcional básico de cualquier sistema BCI, en ella puede apreciarse que existe un sistema de adquisición que sensa las señales encefalográficas mediante electrodos conectados ya sea al cuero cabelludo, superficie cortical, o dentro del cerebro, éstas se procesan para extraer características especiales mediante las cuales se pueda determinar alguna emoción, sensación o intención del usuario. Las características de las señales EEG son extraídas y traducidas a comandos interpretables por una computadora para ejecutar alguna acción. La actividad neuronal utilizada en la BCI puede

registrarse mediante el uso de técnicas invasivas o no invasivas [6]. Una ventaja de las BCI's es que proporcionan a los sujetos con discapacidad motriz canales de comunicación no invasivos que no dependan directamente de los nervios periféricos y los músculos [7].

La plataforma experimental utilizada está basada en el esquema funcional básico descrito anteriormente, es un sistema flexible que facilita el proceso de investigación y desarrollo, en la Fig. 2 se muestra el esquema, en él se identifican los siguientes módulos,

1. **Adquisición de Señales.** Se realiza mediante la diadema especializada “Emotiv EEG neuro-headset” contenida en el paquete “Research Edition” para Windows, que es un sistema de adquisición multicanal de alta resolución de señales EEG, tiene 14 bits de resolución ( $1 \text{ LSB}=0.51\mu\text{V}$ ), las señales se digitalizan a una razón de 128 muestras/segundo, el ancho de banda de la diadema es de 0.2 a 45 Hz, tiene un filtro digital rechazo de banda con alto factor de calidad para frecuencias entre los 50 y 60 hz. La diadema cuenta con 14 canales tomados del estándar internacional 10-20, los cuales son: AF3, F7, F3, FC5, T7, P7, O1, O2, P8, T8, FC6, F4, F8, AF4. Además tiene las referencias CMS/DRL que generalmente se posicionan en P3/P4 del mismo sistema internacional. Se genera un archivo con terminación “.edf” en donde se guardan las lecturas tomadas. La comunicación hacia la computadora se realiza en forma inalámbrica. Es compatible con el toolbox de distribución gratuita EEGLAB.
2. **Procesamiento de Señales.** En esta etapa se realiza la interpretación de la señal EEG, por medio del toolbox para Matlab EEGLAB, el cual se utiliza para análisis de señales electroencefalográficas de manera OFF-LINE. Este software es muy útil para determinar análisis frecuencial y dipolar de las señales EEG. Ya teniendo la interpretación de la señales, se procede a pasar el archivo de manera matricial, donde se representa la señal EEG muestreada por la diadema en formato de Excel.

**Algoritmo 1** Análisis y procesamiento de señales EEG.

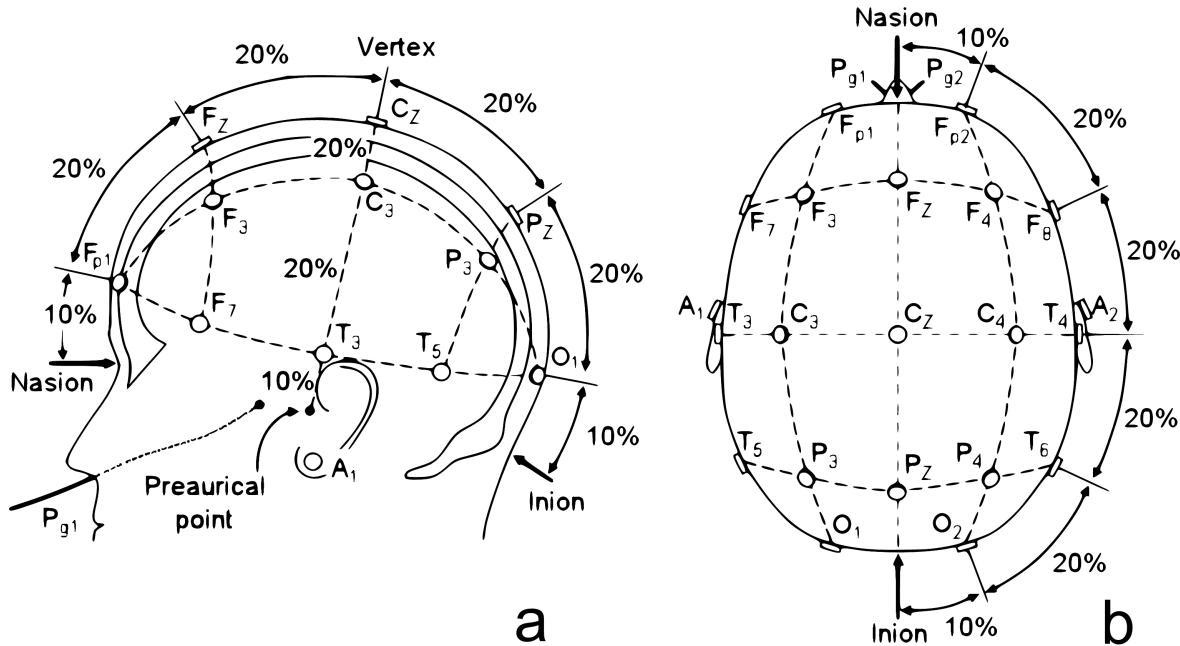
---

```

1: Initialize variables
2: Create InputElectrode.mat from EEG readings
3:  $N = \text{length}(\text{InputElectrode})$   $\triangleright N :=$  Número de muestras
4: Select electrodes to be analyzed
5: Apply FFT to InputElectrode.mat
6: Get Power Spectra Density
7: Normalize InputElectrode.mat already transformed
8: Select training, target and validation data to ANN training.
9: while  $(i) < \frac{N}{2}$  do
10:    $\text{TrainData}_{i+1} := \text{TrainData}_i$ 
11:    $i := i + +$ 
12: end while
13: while  $(i) < N$  do
14:    $\text{TestData}_{i+1} := \text{TestData}_i$ 
15:    $i := i + +$ 
16: end while
17: while  $j < M$  do  $\triangleright M :=$  Número de patrones en los que se divide la prueba
18:   while  $n < k$  do  $\triangleright k :=$  tamaño de Pi
19:      $s\text{Train}_j(n) := \text{TrainData}(m)$   $\triangleright$  Se le asigna un valor a cada muestra
20:      $n := n + +$ 
21:      $m := m + +$ 
22:   end while
23:    $p\text{Train}_j \leftarrow [s_j(n), s_j(n - 1), \dots, s_j(n - k)]$ 
24:    $j := j + +$ 
25:    $n := 0$ 
26: end while
27: while  $j < M$  do  $\triangleright M :=$  Número de patrones en los que se divide la prueba
28:   while  $n < k$  do  $\triangleright k :=$  tamaño de Pi
29:      $s\text{Test}_j(n) := \text{TestData}(m)$   $\triangleright$  Se le asigna un valor a cada muestra
30:      $n := n + +$ 
31:      $m := m + +$ 
32:   end while
33:    $p\text{Test}_j \leftarrow [s_j(n), s_j(n - 1), \dots, s_j(n - k)]$ 
34:    $j := j + +$ 
35:    $n := 0$ 
36: end while
37:  $P_{\text{Train}} \leftarrow [p\text{Train}_1, p\text{Train}_2, \dots, p\text{Train}_j]$ 
38:  $P_{\text{Test}} \leftarrow [p\text{Test}_1, p\text{Test}_2, \dots, p\text{Test}_j]$   $\triangleright$  values classes
39:  $\text{Target} \leftarrow [-1, 1]$   $\triangleright$  Entrenamiento de la red.
40: Define  $[\varepsilon_0, \text{maxepoch}]$   $\triangleright \varepsilon_0 :=$  error de convergencia  $\triangleright$  Entrenamiento de la red.
41: Call RNA( $P, \text{maxepoch}, \varepsilon_0, \text{Target}$ )  $\triangleright$  Entrenamiento de la red.  $\triangleright$  Puede ser una red MLP, o
   ANFIS.  $\triangleright$  El argumento  $P$  representa un lote de patrones.  $\text{maxepoch}$  es el número máximo
   de épocas de entrenamiento permitido.  $\varepsilon_0$  es el valor objetivo de convergencia (criterio de paro).  $\text{Target}$ 
   son los valores esperados para los patrones de entrenamiento. Dependiendo de la arquitectura que se
   quiera probar, se puede llamar a Backpropagation o a ANFIS.
42: Call TestRNA( $P_{\text{Test}}$ )  $\triangleright$  Una vez entrenado el sistema, la RNA puede probarse usando los patrones de
   entrenamiento, los de prueba, u otros.

```

---



**Fig. 3.** Posición de los electrodos acorde al Sistema Internacional 10-20. Figura vista desde (a) lado izquierdo y (b) sobre la cabeza [15]

3. Clasificación. Por último ya teniendo la matriz de representación de las lecturas tomadas, se procede a realizar la identificación de patrones, ésto realizando un preprocesamiento de la señal y después una clasificación por medio de RNAs utilizando el toolbox de Matlab "Neural Networks".

En el Algoritmo 1 se explica de manera muy general, los pasos seguidos en el análisis y procesamiento de una señal EEG; el código llama un archivo con formato "xlsx" en donde se encuentra la lectura de la señal EEG, para ser procesadas se crea la matriz de datos "InputElectrode.mat" en la línea 2. Dentro de esa matriz se descartan los electrodos que no tienen participación en la prueba y se seleccionan solamente los que influyen en el estímulo (parpadeo o dolor). Si la señal se procesa en el tiempo, "inputElectrode.mat" queda sin modificarse; en caso de que la señal se procese en frecuencia, se utiliza la Transformada Discreta de Fourier (DFT) implementada mediante la Transformada Rápida de Fourier (FFT del inglés Fast Fourier Transform) (1) en donde  $N$  es el número de

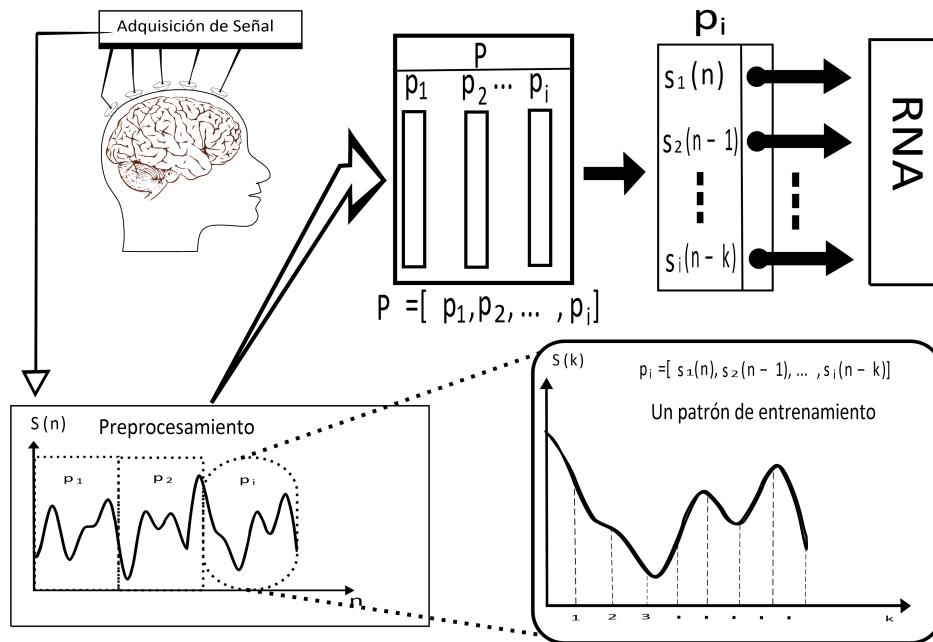
muestras (longitud de la FFT),  $W_N^{kn} = e^{-j(2\pi/N)kn}$ ,  $n = 2r$  para  $n$  par, y  $n = 2r + 1$  para  $n$  impar. La densidad expectral de potencia se obtiene calculando el peridiograma de Welch (2)

$$X[k] = \sum_{r=0}^{(N/2)-1} x[2r]W_{N/2}^{rk} + W_N^k \sum_{r=0}^{(N/2)-1} x[2r+1]W_{N/2}^{rk} \quad (1)$$

$$\widehat{P}_{welch}(\omega) = \frac{1}{K} \left| \sum_{l=0}^{K-1} \widehat{P}_{x_l}(\omega) \right| \quad (2)$$

en donde se está considerando que se tienen  $K$  segmentos de longitud  $N$  traslapados  $D$  muestras, es decir  $x_l[n] = x[n+lD]$  para  $n = 0, \dots, N-1$  y  $l = 0, \dots, K-1$ ; y el peridiograma de cada segmento se calcula usando (3),  $w[n]$  es una ventana y  $U$  esta dada por (4).

$$\widehat{P}_{x_l}(\omega) = \frac{1}{NU} \left| \sum_{n=0}^{N-1} x_l[n]w[n]e^{-j\omega n} \right| \quad (3)$$



**Fig. 4.** La señal EEG y el artefacto ocular parpadeo se muestran utilizando la diadema especializada, se forma una serie de tiempo que se divide en lotes de patrones de entrenamiento  $P$ , cada patrón está formado por varias secuencias de datos que servirán ya sea para entrenar la red o verificar su funcionamiento, pueden ser patrones en el dominio del tiempo o en el dominio de la frecuencia, según sea el caso. En la figura se muestran los de entrenamiento, los de prueba se obtienen de otra sección de la misma serie de tiempos

$$U = \frac{1}{N} \sum_{n=0}^{N-1} w^2[n] \quad (4)$$

La ecuación (3) puede escribirse en términos de la FFT usando (1), quedando como se indica en (5) para  $\omega = \frac{2\pi k}{N}$ , ver línea 6 del Algoritmo 1.

$$\hat{P}_{x_l}(w) = \frac{1}{NU} \left| T \sum_{n=0}^{N-1} X(\omega) w[n] \right| \quad (5)$$

En el Algoritmo 2, considerando una MLP de  $L$  capas, en donde  $l = [0, 1, \dots, L]$ , siendo  $l = L - 1$  las capas ocultas, se describe la regla de aprendizaje conocida como "Retropropagación del error" (Error Backpropagation learning Algorithm), el cual está originalmente basado en la "Regla de aprendizaje Delta", en donde se considera que el ajuste sináptico está dado por  $\partial \xi(i) / \partial w_{kj}(i)$ , en donde  $\xi(i) = \frac{1}{2} \sum_{k \in C} e_k^2(i)$  es el error cuadrático medio calculado en la neurona  $k$  de un conjunto de

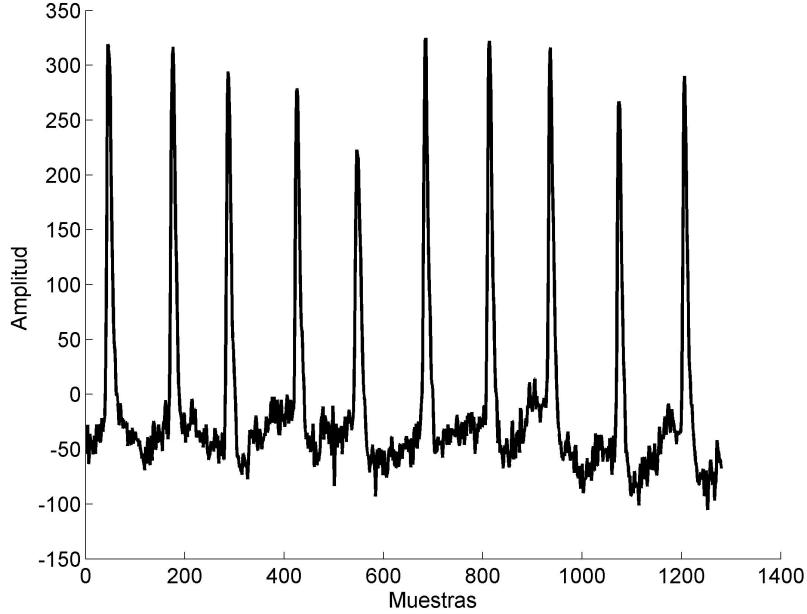
$C$  neuronas en la capa  $L$  para el patrón  $p_i$ . El error en la capa de salida, para el caso de una neurona  $Net_L$  de salida, se tiene

$$e_L(i) = (t_i(i) - Net_L(i)) \quad (6)$$

es decir el error en la neurona  $Net_L$  se calcula con el valor deseado y la salida proporcionada por el perceptrón  $Net_L$  en la salida. Siendo prácticamente imposible medir el error en las neuronas en las capas ocultas, es necesario realizar una estimación del mismo utilizando una retropropagación del error de las capas posteriores; con lo anterior puede deducirse fácilmente que

$$\Delta w_{kj} = -\eta \frac{\partial \xi(i)}{\partial w_{kj}(i)} \quad (7)$$

en donde  $w_{kj}$  es uno de los pesos que conecta la neurona  $j$  a la  $k$ ,  $\eta$  es la razón de aprendizaje, y el signo menos indica un descenso y/o minimización, es decir está relacionado con el gradiente



**Fig. 5.** Prueba EEG donde se muestra el parpadeo durante 10 segundos, con intervalo de un segundo cada uno

descendente. Este algoritmo considera dos casos: a) cuando la neurona  $k$  está en la capa de salida, entonces el error se calcula usando (6), b) si la neurona está en la capa oculta, entonces es necesario calcular el gradiente local  $\delta$  en base a los errores retropropagados [9], por lo que el cálculo de la línea 15 depende de la

$$\begin{aligned} w_{kj}^{(l)}(i+1) &= w_{kj}^{(l)}(i) + \alpha[w_{kj}^{(l)}(i-1)] \\ &\quad + \eta\delta_j^{(l)}(i)y_k^{(l-1)}(i) \end{aligned} \quad (8)$$

Finalmente, el error promedio al pasar todos los patrones  $p_i$  por la red se calcula usando 9,

$$\xi_{av} = \frac{1}{N} \sum_1^N \xi(n) \quad (9)$$

Usando esta misma estructura general de algoritmo es posible definir otras, por ejemplo la de “Retropropagación del error con algoritmo de aprendizaje LMS (Least Mean Squared) con término de momento  $\alpha$  usado para evitar oscilaciones y acelerar la convergencia, en este caso

se modifica la línea 15, quedando como se indica en (8). Para el caso “Resilient Backpropagation (RP)” que es de especial interés en este trabajo, se tiene que en la línea 15  $\Delta w_{kj}$  se modifica usando (10) como sigue [20],

$$\Delta w_{kj} = -\alpha_{kj} \text{sgn}\left(\frac{\partial \xi(w_{kj}(n))}{\partial w_{kj}}\right) \quad (10)$$

en donde  $\alpha$  se calcula usando (11)

$$\alpha_{kj} = \begin{cases} \min(\alpha\alpha_{kj}(i-1), \alpha_{max}) & \text{si } S_{kj}(i)S_{kj}(i-1) > 0 \\ \max(b\alpha_{kj}(i-1), \alpha_{min}) & \text{si } S_{kj}(i)S_{kj}(i-1) < 0 \\ \alpha_{kj}(i-1) & \text{de otra forma} \end{cases} \quad (11)$$

en donde,

$$S_{kj} = \frac{\partial \xi(w_{kj}(i))}{\partial w_{kj}} + \eta w_{kj}(i) \quad (12)$$

Para el método de Levenberg-Marquardt la línea 15 se modifica como se indica en (13)

$$\mathbf{w}_{kj}^{(l)}(i+1) = \mathbf{w}_{kj}^{(l)}(i) - (\mathbf{J}_k^T \mathbf{J}_k + \mu \mathbf{I})^{-1} \mathbf{J}_k^T \mathbf{e} \quad (13)$$

en donde **w**, **J**, **I**, y **e** son arreglos matriciales que corresponden a la matriz de pesos sinápticos **w** que conectan la capa *j* con la *k*, la matriz jacobiana **J** que contiene información obtenida de las derivadas parciales de los errores con respecto a los pesos correspondientes en la capa indicada, la matriz identidad **I**, y la matriz de errores **e**, y el valor de  $\mu$  es un escalar que se utiliza para mantener la dirección de descenso y en consecuencia la estabilidad del método.

Similarmente pueden obtenerse todas las reglas de actualización de pesos mencionadas en la Tabla 1 que indica los algoritmos de aprendizaje utilizados en este trabajo [19].

El Algoritmo 3 muestra que el entrenamiento de un sistema ANFIS puede realizarse mediante el algoritmo híbrido propuesto por Jang [11], que consiste de dos pasos. En el primer paso, conocido como paso hacia adelante, se utiliza el estimador de mínimos cuadrados para identificar los parámetros del consecuente, manteniendo fijos los parámetros de la premisa. En el segundo paso, conocido como paso hacia atrás, los errores se propagan en sentido inverso (retropropagación del error) con la intención de actualizar las premisas mediante el método de gradiente descendente.

La matriz de datos “inputElectrode.mat” se divide en dos partes, una para crear los datos de entrenamiento *TrainData<sub>i+1</sub>* (líneas 8 a 11) y otra para crear los datos de prueba *TestData<sub>i+1</sub>* (líneas 12 a 15). Ya que tenemos las tramas de datos de entrenamiento y de prueba, ahora procedemos a dividirlas en diferentes patrones de entrenamiento para aprovechar toda la lectura de datos. En las líneas 17 a 26 se crean todos los patrones de la etapa de entrenamiento *pTrainj(n)*, cabe mencionar que la asignación de cada una de las muestras se realiza en la línea 19, donde *n* representa el número de patrones obtenidos de los datos de entrenamiento (*TrainData<sub>i+1</sub>*). La asignación de los datos de prueba se realiza en las líneas 27 a 36 siguiendo el mismo proceso pero con los datos de entrenamiento (*TestData<sub>i+1</sub>*). Una vez asignados los patrones de cada etapa, se hace un llamado a la función *BackPropagation()*, en la cual se realizará el entrenamiento (línea 41), mientras que en la línea 42 se realiza la prueba de la RNA previamente entrenada.

**Tabla 1.** Acrónimos para algoritmos de adaptación

Acrónimo	Algoritmo de adaptación
LM	Levenberg Marquardt
OSS	One Step Secant
BFGS	BFGS Quasi-Newton
RP	Resilient Backpropagation
GD	Gradient Descent

**Tabla 2.** Arquitectura de MLP para clasificadores de estímulos fisiológicos

Estímulos	Arquitectura de RNA
parpadeo (ambos ojos)	120:20:10:5:1
Dolor (Brazo derecho)	1280:20:10:5:1

Para que la comunicación entre el cerebro y la computadora exista, las BCI adquieren la señal análoga del cerebro por medio de la colocación específica de electrodos elaborados de plata y oro siguiendo el estándar del sistema internacional 10-20 para la colocación de electrodos en el cuero cabelludo, tal como se muestra en la Fig. 3.

La posición de los electrodos se identifica mediante dos parámetros, la letra y número que describen el lóbulo y el hemisferio cerebral, respectivamente. Las letras son F, T, C, P y O para identificar frontal, temporal, central, parietal y occipital. Cabe mencionar que el lóbulo central “C” no existe; sin embargo, se utiliza como punto de referencia; la letra “Z” se utiliza para identificar un electrodo localizado en una línea intermedia. Para identificar el hemisferio derecho se utilizan los números pares 2, 4, 6, 8 y los impares 1, 3, 5, 7 para el hemisferio izquierdo [10].

Siendo, la EEG una técnica que mide la distribución espacial de los campos de voltaje en el cuero cabelludo y su variación en el tiempo. El origen de esta actividad es debido a la suma de las fluctuaciones de los potenciales postsinápticos que generan las neuronas. Desde otro punto de vista la técnica de EEG es utilizada para cuantificar la actividad neuronal por medio de la colocación de electrodos en el cuero cabelludo. Dentro de las características necesarias para el análisis de la EEG se encuentra la frecuencia, voltaje, localización, morfología, polaridad, estado, reactividad, simetría y artefactos involucrados [10].

La EEG se divide principalmente en cuatro bandas de frecuencia; delta (0.4Hz a 4 Hz), theta (4 Hz a 8Hz), alpha (8Hz a 13 Hz) y beta (13Hz a 30Hz); siendo considerado como ruido o artefacto cualquier señal que exceda la frecuencia de los 30 Hz. El ruido que comúnmente se detecta en el análisis de la EEG es el de la corriente doméstica (60 Hz). La Fig. 10 muestra el análisis en el dominio de la frecuencia de una prueba EEG donde un paciente es sometido al dolor; se puede apreciar ruido cuando existe actividad después de los 30 Hz, especialmente el que proveniente de la corriente doméstica de 60 Hz.

La señal EEG en una ventana de tiempo, puede considerarse como un proceso estocástico estacionario ya que no existe un patrón de señales específico para cada uno de los estímulos que son generados por el músculo cerebral, algunos parámetros como la densidad espectral de potencia y la frecuencia son utilizados para el análisis y clasificación de la EEG. La actividad cerebral y eléctrica pasa del dominio del tiempo al dominio de la frecuencia con el objetivo de eliminar artefactos que afecten el análisis de la onda EEG.

### 3. Desarrollo Experimental

Se presentan dos modelos de arquitecturas de RNAs para la clasificación de las señales producidas por el artefacto parpadeo ocular y las señales EEG de dolor muscular, en ambos modelos de arquitectura se utiliza aprendizaje supervisado.

La primera de las arquitecturas de RNA es del tipo perceptrón multicapa (MLP, del inglés Multi-Layer Perceptron) con funciones de activación tangente hiperbólica, los experimentos se realizaron utilizando los algoritmos de adaptación mencionados en la Tabla 1. La Tabla 2 muestra las arquitecturas MLP utilizadas para la clasificación del artefacto parpadeo y de las señales EEG para la detección del dolor.

La segunda arquitectura de RNA es del tipo ANFIS, que consiste en un sistema difuso del tipo Sugeno en donde se identifican premisas (funciones de membresía), consecuentes, y un conjunto de reglas que relaciona las premisas con los consecuentes; y se utiliza un método de aprendizaje

híbrido de dos pasos, como se explica en el Algoritmo 3.

En la Fig. 4 se ilustra en términos generales la forma en que se generaron los diversos patrones de entrenamiento y se presentaron en forma de lote a la RNA (MLP o ANFIS).

Para la realización de los experimentos, la información se recabó de 15 personas de las cuales se tomaron patrones de manera arbitraria por cada estímulo a identificar (parpadeo y dolor), con el propósito de tener una gran diversidad de patrones. En general se tomaron para cada persona 30 patrones de entrenamiento por cada estímulo y 50 patrones de validación.

Para realizar la clasificación se considera que los elementos que pertenecen a la clase 1 son los que han sido clasificados correctamente, y a la clase -1 los que no; también se considera para la clase 1 un umbral de decisión de 0.9. En todos los experimentos, se utiliza el procedimiento de prueba proporcionado en Algorithm 1, y se establece un error de convergencia de  $1 \times 10^{-3}$ .

Se utiliza una computadora con procesador I7 920A a 2.67Gh, con 6GB de memoria RAM y Sistema Operativo Windows 7 de 64 bits.

Se presentan dos experimentos, y se determinan las características particulares de los modelos de RNAs que clasifican adecuadamente las señales del artefacto y EEG. Se realizan experimentos y comparaciones.

#### 3.1. Experimento 1: Parpadeo

Se realizó la identificación de parpadeo de ambos ojos. En la Fig. 5 se presenta la actividad eléctrica cuando existen diez parpadeos, la cual servirá como patrón de entrenamiento para la RNA. En la Fig. 6 se muestran ocho de los ejemplos utilizados como patrón de entrenamiento pertenecientes a la clase 1, mientras que en la Fig. 7 se muestran ocho de los ejemplos utilizados como patrón de entrenamiento pertenecientes de la clase -1.

En el caso de MLP, se uso una arquitectura 120:20:10:5:1; en la Tabla 3 se concentran los resultados estadísticos en cuanto a tiempo de entrenamiento obtenidos con 50 experimentos para cada algoritmo de adaptación.

**Algoritmo 2** Algoritmo de Retropropagación del error.

---

```

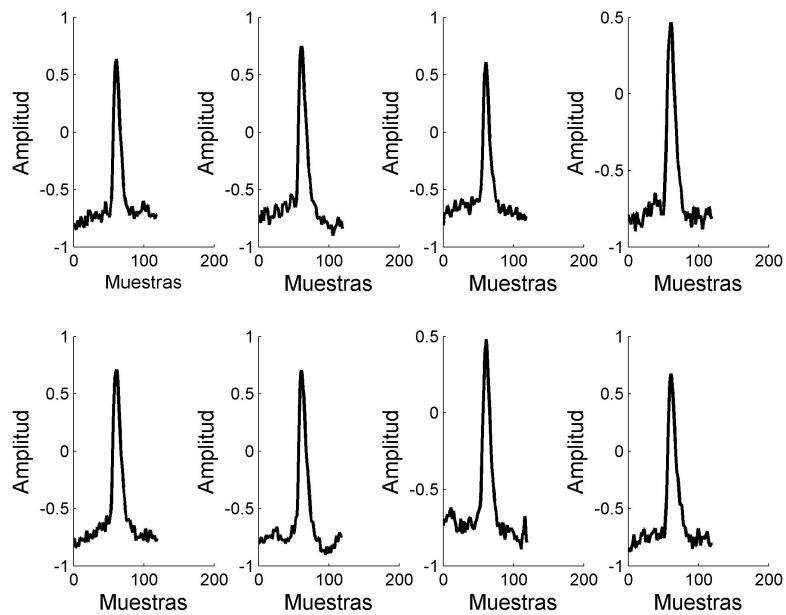
1: procedure BACKPROPAGATION( $P, maxepoch, \varepsilon_0, Target$ ) ▷
    $P = [p_1, \dots, p_i, \dots, p_N], Target = [t_1, \dots, t_i, \dots, t_N]$ 
2:   Iniciar con un conjunto aleatorio de pesos sinápticos
3:   while  $|\xi_{av} < \varepsilon_0|$  OR Número de épocas  $\leq maxepoch$  do ▷  $\varepsilon_0 :=$ Error de convergencia
4:     for all  $p_i$  do ▷ Usar Target como patrón de referencia para entrenamiento supervisado
5:       Propagar la entrada  $p_i$  a través de la red para calcular la salida  $y(n)$ 
6:       a) Capa de salida L:
7:          $e_L(i) = t_i(i) - Net_L(i)$  ▷ Calcular el error en la neurona de salida
8:          $\xi(i) = \frac{1}{2}e_k^2(i)$  ▷ Error cuadrático medio de entrenamiento de cada patrón
9:       b) En las capas ocultas, en donde  $l = L - 1$ :
10:      for cada capa  $l$  do
11:        Retropropagar el error de la capa posterior ( $l + 1$ )
12:        Calcular el ajuste sináptico  $\Delta w_{kj}^{(l)}(i)$  de los pesos de las neuronas en la capa  $l$ 
13:        Ajustar los pesos sinápticos usando:
14:
15:         $w_{kj}^{(l)}(i + 1) = w_{kj}^{(l)}(i) + \Delta w_{kj}^{(l)}(i)$ 
16:
17:      end for
18:    end for
19:     $\xi_{av} = \frac{1}{N} \sum_1^N \xi(i)$  ▷ Error promedio
20:  end while
21: end procedure

```

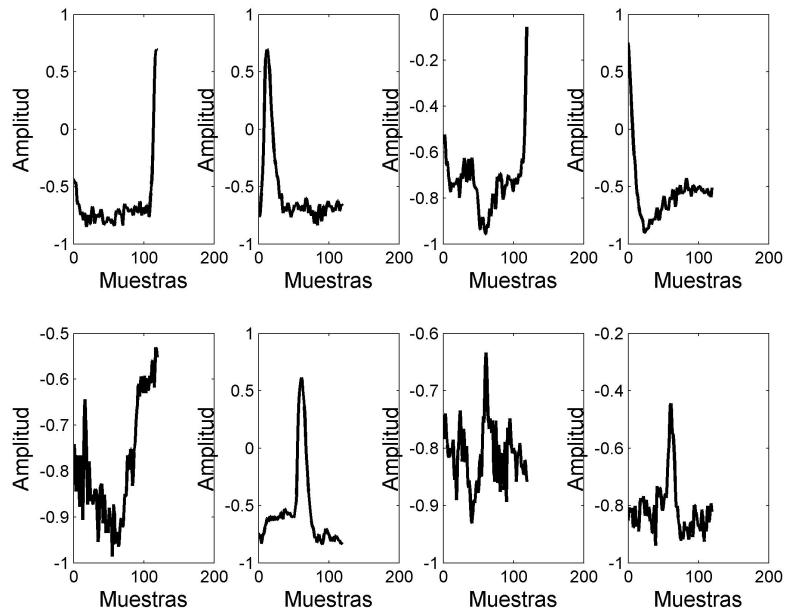
---

**Algoritmo 3** Algoritmo de aprendizaje híbrido que utiliza ANFIS.

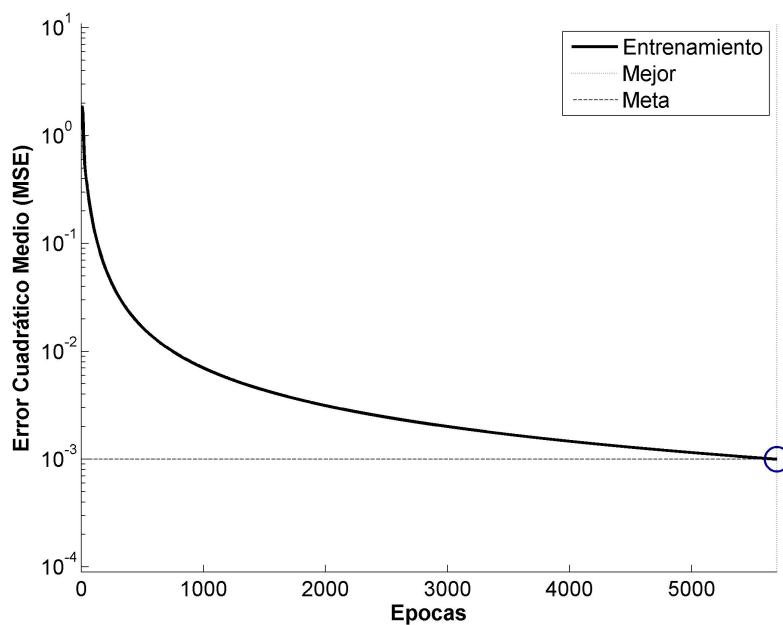
- 
- 1: **Paso 1: Hacia adelante. Las señales son las salidas de los nodos.**
  - 2: 1.a) Dejar los parámetros de las premisas fijos.
  - 3: 1.b) Adaptar los parámetros de los consecuentes usando el estimador de mínimos cuadrados
  - 4: **Paso 2: Hacia atrás. Las señales son señales de error retropropagadas en sentido inverso.**
  - 5: 2.a) Dejar los parámetros de los consecuentes fijos.
  - 6: 2.b) Adaptar las premisas mediante el método de gradiente descendente.
-



**Fig. 6.** Ocho ejemplos de parpadeo, pertenecientes a la clase 1



**Fig. 7.** Ocho ejemplos diferentes a un parpadeo, pertenecientes a la clase -1

**Fig. 8.** Entrenamiento de Red para parpadeo con MLP**Tabla 3.** Resultados de entrenamientos de MLP con diferentes algoritmos de adaptación para parpadeo. Todos los tiempos están en segundos

Algoritmos de adaptación	Tiempo promedio	Tiempo mínimo	Tiempo máximo	Desv. Std	Promedio épocas
LM	9.433	2	28	5.894	10.933
OSS	0.835	0.494	1.7035	0.261	27.2
BFGS	145.967	59	269	60.327	18.8
RP	0.505	0.362	2.793	0.433	26.867
GD	22.433	10	55	10.197	3825.533

**Tabla 4.** Comparativa de clasificación MLP vs. ANFIS para parpadeo

Clasificador	índice de Clasificación( %)	Patrones clasificados	Detección de falsos
MLP(RP)	96.424	48	2
ANFIS(G-bell)	94.010	47	3

En la Tabla 6 se ilustra el algoritmo de adaptación utilizado, y los resultados de clasificación obtenidos con los datos de entrenamiento y los datos de prueba.

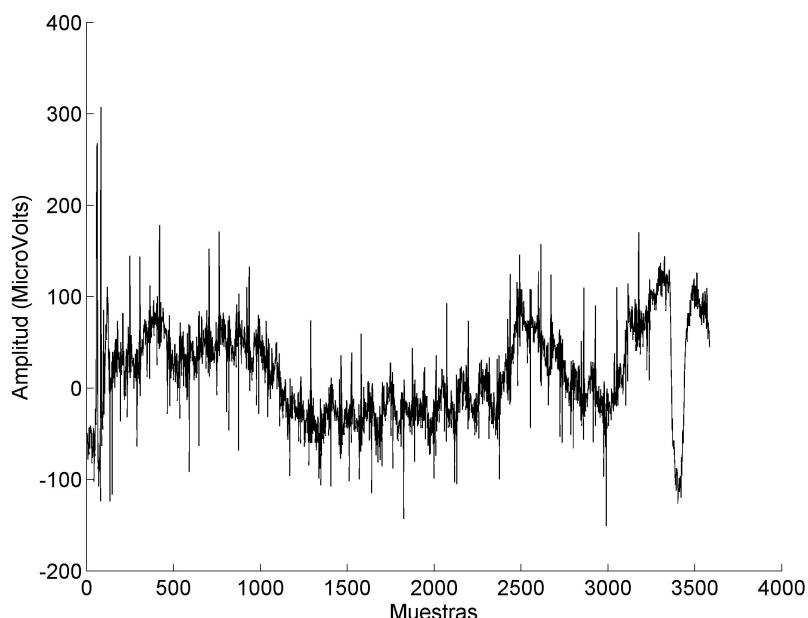
En la Fig. 8 se muestra el rendimiento obtenido del clasificador MLP cuando se utiliza gradiente

descendente como algoritmo de adaptación.

En el caso de ANFIS, se utilizaron funciones de membresía del tipo “Campana Generalizada” conocidas como G-bell. En la Tabla 4 se muestra una comparación de los resultados obtenidos con MLP y ANFIS. Puede observarse que en este

**Tabla 5.** Resultados de entrenamientos de MLP con diferentes algoritmos de adaptación para dolor de brazo derecho

Algoritmo de adaptación	Tiempo promedio	Tiempo mínimo	Tiempo máximo	Desv. Std.	Promedio épocas
LM	1167	420	2700	468.527	33343.9
OSS	1864.433	840	5687	1012.525	36462.567
BFGS	1403.6	600	4140	706.334	33677.233
RP	1568	840	4800	771.395	48982.2
GD	1290	720	4140	728.868	39677.233

**Fig. 9.** Señal EEG en el dominio del tiempo, correspondiente al canal AF3 en una prueba de dolor por un pinchazo

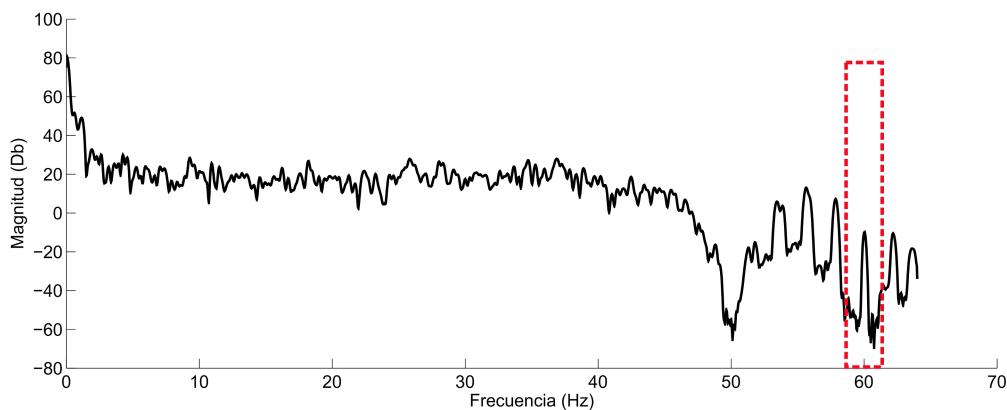
caso, los mejores resultados se obtuvieron con MLP y el algoritmo de adaptación RP, en términos generales, MLP clasificó más patrones con menor clasificación de falsos positivos.

**Tabla 6.** Resultados del clasificador de Parpadeo

Algoritmo de adaptación	Datos de entrenamiento	Datos de prueba
LM	98.710	93.461
OSS	99.338	94.101
BFGS	99.357	92.987
RP	99.529	96.424
GD	99.755	88.775

**Tabla 7.** Resultados del clasificador de Dolor brazo derecho

Algoritmo de adaptación	Datos de entrenamiento	Datos de prueba
LM	99.713	96.861
OSS	96.594	95.426
BFGS	97.363	93.123
RP	95.327	92.987
GD	98.721	96.776



**Fig. 10.** Señal AF3 de un paciente sometido al dolor en el brazo izquierdo. Notar que a simple vista no es posible apreciar un cambio el cual ocurre entre los 0 y 30 Hz, sin embargo, el ruido de 60 Hz si puede observarse.

**Tabla 8.** Comparativa de clasificación MLP vs. ANFIS para dolor brazo derecho

Clasificador	índice de Clasificación( %)	Patrones clasificados	Detección de falsos
MLP(LM)	96.861	48	2
ANFIS(G-bell)	98.000	49	1

### 3.2. Experimento 2: Dolor muscular

Se realizaron experimentos para detectar clasificar el dolor muscular inducido por un agente externo, en todos los experimentos se utilizó el esquema mostrado en la Fig. 2.

En el caso de MLP, se utiliza una arquitectura de 1280:20:10:5:1; en la Tabla 5 se proporciona un concentrado estadístico en cuanto a tiempo de entrenamiento y cantidad de épocas requeridas para realizar la adaptación de los pesos en la red, los resultados se obtienen realizando 50 experimentos para cada algoritmo de adaptación.

En la Tabla 7 se relacionan los algoritmos de adaptación utilizados con los porcentajes de clasificación obtenidos con los datos de entrenamiento y los de prueba, puede observarse que el algoritmo que adaptó mejor a la red MLP para la clasificación de los datos de entrenamiento como los de prueba fue el de Levengerg-Marquardt (LM).

En la Fig. 9 se observa la actividad cerebral en el dominio del tiempo durante una prueba de dolor. En la Fig. 11 se observa la misma señal en el dominio de la frecuencia, cabe mencionar que existe actividad detectada hasta los 65 Hz. En base al

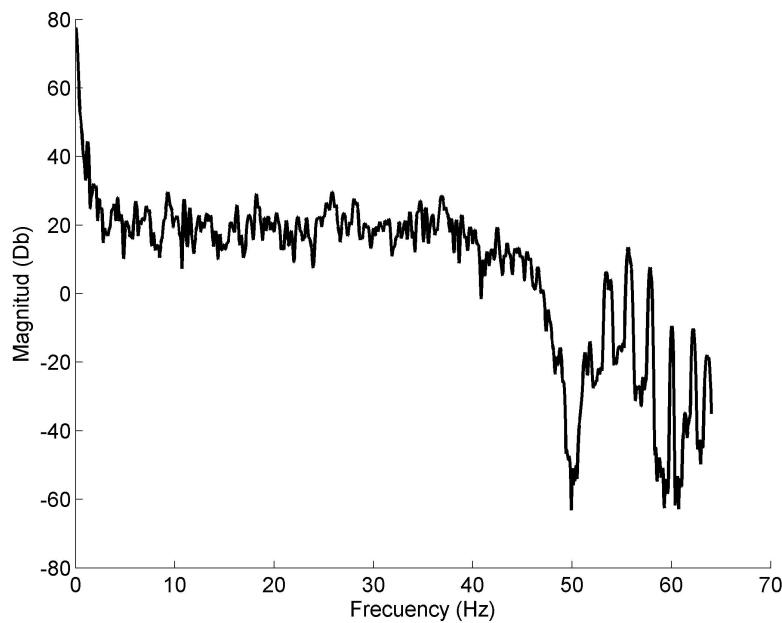
sistema internacional 10-20 el electrodo colocado en el hemisferio izquierdo y el lóbulo frontal es el AF3.

La Fig. 12 muestra el espectro de frecuencia cuando el sujeto se encuentra en estado de reposo y en la Fig. 13 se observa el espectro de frecuencia de una señal EEG cuando se induce dolor muscular externo al sujeto. Para eliminar cualquier señal que no provenga del encéfalo, se procede a descartar del estudio las ondas que tengan un espectro de frecuencia mayor a 30 Hz, es decir los artefactos.

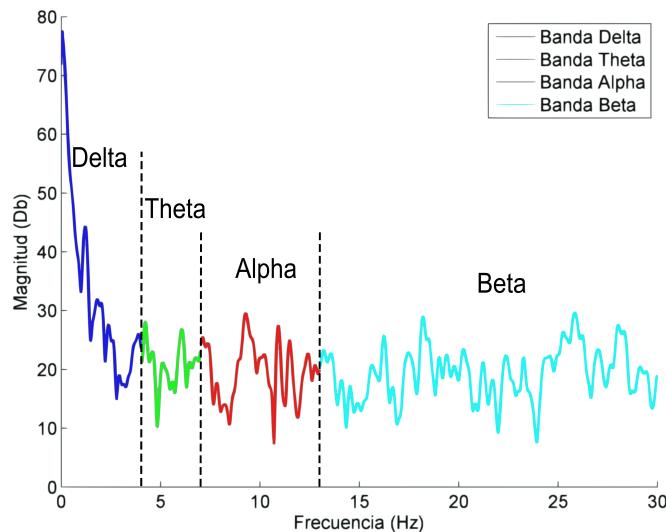
En la Tabla 8 se comparan los resultados obtenidos con MLP adaptado con LM y ANFIS. Puede observarse que a diferencia del caso anterior, ANFIS clasificó mejor el dolor muscular que MLP, de los 50 patrones presentados a la red ANFIS, clasificó correctamente 49 patrones, y sólo hubo un falso positivo clasificado.

### 4. Conclusiones

Se conoce que el uso de las RNAs es una herramienta poderosa y eficiente para la clasifica-



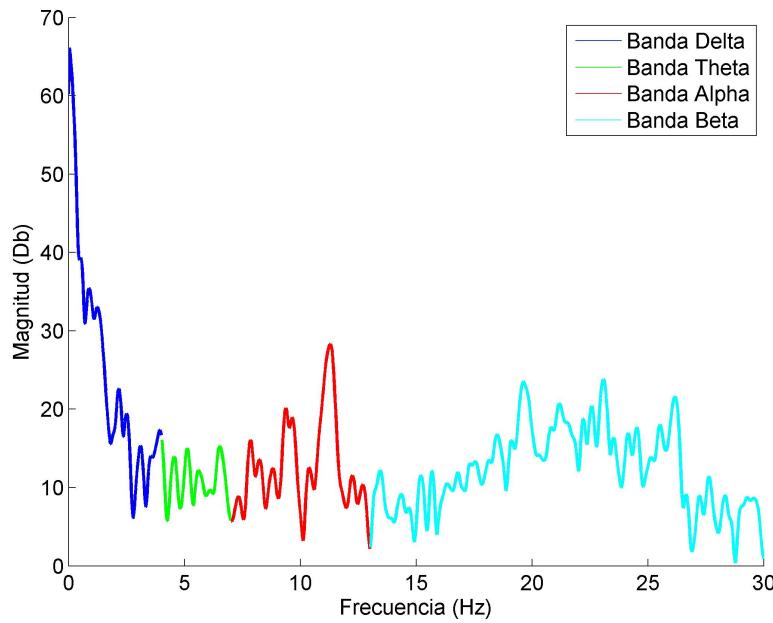
**Fig. 11.** Señal EEG en el dominio de la frecuencia, correspondiente al canal AF3 en una prueba de dolor por un pinchazo.



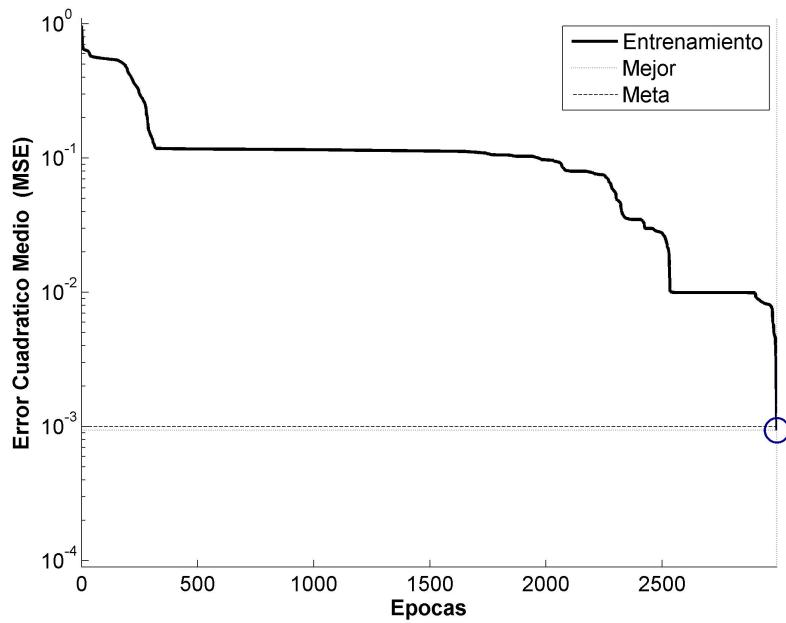
**Fig. 12.** Señal EEG correspondiente al canal F7 correspondiente a un sujeto en estado de relajación

ción de señales EEG, pero no existen suficientes trabajos de investigación enfocados a la detección eficiente del dolor producido por agentes externos, este trabajo aporta información en ese sentido. Se

presenta la metodología y arquitecturas de RNAs de tipo perceptrón multicapa y ANFIS utilizadas para la clasificación del dolor muscular y el artefacto ocular conocido como parpadeo, los cuales



**Fig. 13.** Señal EEG correspondiente al canal F7 correspondiente a un sujeto sometido a dolor en el brazo derecho



**Fig. 14.** Entrenamiento de Red para respuesta al Dolor

en combinación ofrecen una gran certeza de la existencia del dolor.

Se observa de los resultados estadísticos de los experimentos, que para la detección del parpadeo el algoritmo de adaptación de Gradiente descendente es el que reconoce mejor los datos de entrenamiento; sin embargo, es el que necesita realizar más épocas de entrenamiento, por otro lado el “Resilient Backpropagation” es el que tiene mayor poder de generalización con un 96.4% de éxito y el promedio de épocas de entrenamiento requeridas es bajo, por lo que se concluye en este punto que este último es el más conveniente para realizar esta clasificación, superando a ANFIS ya que sólo clasificó erróneamente dos patrones.

En cuanto a la detección de dolor muscular, en el caso de MLP con el algoritmo de adaptación “Levenberg-Marquardt” es el que demuestra tener las mejores cualidades para clasificar tanto los datos de entrenamiento como los de prueba con un 96.8.0 % de éxito; sin embargo, en esta prueba no supera a ANFIS con un porcentaje de éxito del 98.9% ya que sólo clasificó un patrón de forma errónea de los 50 presentados, correspondientes al grupo de datos de prueba.

En términos generales, los resultados obtenidos son lo suficientemente buenos para que el médico conozca la condición del paciente, el resultado puede mejorarse positivamente simplemente desplazando el umbral de aceptación de 0.9 a un valor menor, lo cual requiere de un estudio clínico y representa un posible trabajo futuro.

## Agradecimientos

Los autores agradecen al Instituto Politécnico Nacional (IPN), al Instituto Tecnológico de Tijuana (ITT), y al Consejo Nacional de Ciencia y Tecnología (CONACYT) por su apoyo.

## Referencias

1. Arias, G. & Felipe, H. (2009). Detección y clasificación de artefactos en señales eeg. *Memorias de STSIVA '09*, Universidad Tecnológica De Pereira.
2. Chambayil, B., Singla, R., & Jha, R. (2010). Virtual keyboard bci using eye blinks in eeg. *Wireless and Mobile Computing, Networking and Communications (WiMob), 2010 IEEE 6th International Conference on*, pp. 466–470.
3. Chang, B.-C., Lim, J.-E., Kim, H.-J., & Seo, B.-H. (2007). A study of classification of the level of sleepiness for the drowsy driving prevention. *SICE, 2007 Annual Conference*, pp. 3084–3089.
4. Chang, P., Arendt-Nielsen, L., Graven-Nielsen, T., Svensson, P., & Chen, A. (2001). Different eeg topographic effects of painful and non-painful intramuscular stimulation in man. *Experimental Brain Research*, Vol. 141, pp. 195–203.
5. Chang, P.-F., Arendt-Nielsen, L., Graven-Nielsen, T., & Chen, A. C. (2003). Psychophysical and {EEG} responses to repeated experimental muscle pain in humans: Pain intensity encodes {EEG} activity. *Brain Research Bulletin*, Vol. 59, No. 6, pp. 533 – 543.
6. Chang, P. F., Arendt-Nielsen, L., Graven-Nielsen, T., Svensson, P., & Chen, A. C. (2004). Comparative eeg activation to skin pain and muscle pain induced by capsaicin injection. *International journal of psychophysiology*, Vol. 51, No. 2, pp. 117–126.
7. Chen, A. & Rappelsberger, P. (1994). Brain and human pain: Topographic eeg amplitude and coherence mapping. *Brain Topography*, Vol. 7, No. 2, pp. 129–140.
8. De la O Chavez, J. R. (2008). *BCI para el control de un cursor basada en ondas cerebrales*. Master's thesis, Universidad Autónoma Metropolitana.
9. Haykin, S. (1999). *Neural Networks. A Comprehensive Foundation. Second Edition*. Prentice Hall.
10. Hirsch, L. & Richard, B. (2010). *Atlas of EEG in Critical Care*, chapter EEG basics. Wiley, pp. 1–7.
11. Jyh-Shing Roger Jang, Chuen-Tsai Sun, & Sun, Eiji Mizutani (1997). *Neuro-Fuzzy and Soft Computing. A Computational Approach to Learning and Machine Intelligence*. Prentice Hall.
12. Lin, J.-S., Chen, K.-C., & Yang, W.-C. (2010). Eeg and eye-blinking signals through a brain-computer interface based control for electric wheelchairs with wireless scheme. *New Trends in Information Science and Service Science (NISS), 2010 4th International Conference on*, pp. 731–734.
13. Pera, D. L., Svensson, P., Valeriani, M., Watana-be, I., Arendt-Nielsen, L., & Chen, A. C. (2000). Long-lasting effect evoked by tonic muscle pain on

- parietal {EEG} activity in humans. *Clinical Neurophysiology*, Vol. 111, No. 12, pp. 2130 – 2137.
14. Pérez, M. & Luis, J. (2009). *Comunicación con Computador mediante Señales Cerebrales. Aplicación a la Tecnología de la Rehabilitación*. Ph.D. thesis, Universidad Politécnica de Madrid.
  15. Sharbrough F, Chatrian G-E, Lesser RP, Luders H, Nuwer M, & Picton TW (1991). American electroencephalographic society guidelines for standard electrode position nomenclature. *J. Clin. Neurophysiol*, Vol. 8, No. 200, pp. 2.
  16. Sovierzoski, M., Argoud, F., & De Azevedo, F. (2008). Identifying eye blinks in eeg signal analysis. *Information Technology and Applications in Biomedicine, 2008. ITAB 2008. International Conference on*, pp. 406–409.
  17. Sovierzoski, M., Argoud, F., & De Azevedo, F. (2008). Identifying eye blinks in eeg signal analysis. *Information Technology and Applications in Biomedicine, 2008. ITAB 2008. International Conference on*, pp. 406–409.
  18. Venkataraman, S. & Kalpakam, N. V. (2004). Aiding the detection of alzheimer's disease in clinical electroencephalogram recording by selective denoising of ocular artifacts. *Communications, Circuits and Systems, 2004. ICCCAS 2004. 2004 International Conference on*, volume 2, pp. 965–968.
  19. Wilamowski, B. (2009). Neural network architectures and learning algorithms. *Industrial Electronics Magazine, IEEE*, Vol. 3, No. 4, pp. 56–63.
  20. Wilamowski, B. M. (2003). Neural network architectures and learning algorithms. *International Conference on Industrial Technology*, pp. TU1–TU12.
  21. Wu, J., Ifeatchor, E., Allen, E., Wimalaratna, S., & Hudson, N. (1997). Intelligent artefact identification in electroencephalography signal processing. *Science, Measurement and Technology, IEE Proceedings -*, Vol. 144, No. 5, pp. 193–201.

**Roberto Sepúlveda Cruz** received the M.Sc. from the Tijuana Institute of Technology, Tijuana, México, and the D.Sc. degree in the Universidad Autónoma de Baja California, Tijuana, México, both in Computer Science, in 1999 and 2006 respectively. He has published papers about type-2 fuzzy systems, embedded type-1 and type-2 fuzzy logic controllers, mobile robot navigation. He works as a researcher at the Centro de Investigación y Desarrollo de Tecnología Digital from the IPN. His

research interests include type-2 fuzzy systems, intelligent systems, and robotics. Dr. Sepúlveda is co-founder and an active member of HAFSA (Hispanic American Fuzzy Systems Association), and the Mexican Chapter of the Computational Intelligence Society (IEEE), he is member of the International Association of Engineers (IANG), and member of the Mexican science foundation CONACYT.

**Oscar Humberto Montiel Ross** received the M.Sc. from Instituto Politécnico Nacional (IPN) in Digital Systems in 1999, a M.Sc. From the Tijuana Institute of Technology, Tijuana, México, and the D.Sc. degree in the Universidad Autónoma of Baja California, Tijuana, México, both in Computer Science, in 2000 and 2006 respectively. He has published papers about evolutionary computation, Mediative Fuzzy Logic, Ant Colonies, type-2 fuzzy systems, embedded systems, and mobile robotics. He works as a researcher at the Centro de Investigación y Desarrollo de Tecnología Digital (CITEDI) from the IPN. His research interests include optimization, intelligent systems, and robotics. Dr. Montiel is co-founder and an active member of HAFSA (Hispanic American Fuzzy Systems Association), and the Mexican Chapter of the Computational Intelligence Society (IEEE), he is member of the International Association of Engineers (IANG), and member of the Mexican science foundation CONACYT (Consejo Nacional de Ciencia y Tecnología)

**Gerardo Diaz** received his Engineering title in Electronic from Universidad Autónoma de Baja California (UABC). Currently he is a student of the M.Sc. in Digital Systems from the Centro de Investigación y Desarrollo de Tecnología Digital (CITEDI) from the Instituto Politécnico Nacional (IPN). He has been working on EEG Signal Processing and Neural Networks. His research interests include intelligent systems, biomedical signal processing, human interface devices and software engineering.

**Daniel Gutierrez** received his Engineering title in Mechatronics from Centro de Enseñanza Técnica y Superior (CETYS). Currently he is a student

of the M.Sc. in Digital Systems from the Centro de Investigación y Desarrollo de Tecnología Digital (CITEDI) from the Instituto Politécnico Nacional (IPN). His research interests include optimization, intelligent systems, neuroscience, biomedical signal processing, human interface devices, medical neurosciences and software engineering.

**Oscar Castillo López** holds the Doctor in Science degree (Doctor Habilitatus) in Computer Science from the Polish Academy of Sciences (with the Dissertation "Soft Computing and Fractal Theory for Intelligent Manufacturing"). He has published over 130 research papers, u authored books, 10

edited books, and 200 papers in conferences proceedings about Soft Computing. He is a Professor of Computer Science in the Graduate Division, Tijuana Institute of Technology, Tijuana, México. In addition, he is serving as Research Director of Computer Science and head of the research group on fuzzy logic and genetic algorithms. President of IFSA (International Fuzzy Systems Association); he is also Vice-Chair of the Mexican Chapter of the Computational Intelligence Society (IEEE).

*Article received on 04/10/2013, accepted on 26/11/2014.  
Corresponding author is Roberto Sepúlveda.*