Pattern Identification by an Artificial Neural Network implemented in a DSP using a touchscreen

M. Méndez[1], J. Buendía[2], J. Galindo[2], M. Rodríguez[2], J. L. Mata-Machuca[1], L. Fonseca[1]

[1] UPIITA-IPN Academia de Mecatrónica, Av. Instituto Politécnico Nacional 2580, Barrio La Laguna Ticomán, Gustavo A. Madero, 07340, México .D.F.
mmendezma{ lfonseca, jmatam}@ipn.mx

[2] UPIITA-IPN Ingeniería Mecatrónica, alumnos PIFI, Av. Instituto Politécnico Nacional 2580, Barrio La Laguna Ticomán, Gustavo A. Madero, 07340, México .D.F.
jbuendiar0900@alumno.ipn.mx,
jgalindoo0900{mrodriguezm0809}@ipn.mx
Paper received on 11/29/13, Accepted on 01/19/14.

Abstract. This paper describes an application of pattern recognition by using an artificial neural network (ANN) employing a resistive touch screen as input method. The processing related to the ANN was implemented in a TMS320C6713 DSP Starter Kit (DSK). Some results obtained in a DSK6713 shown that the proposed methodology is able to recognize among three different patterns drawn in a resistive touch panel.

Keywords: artificial neural network, patterns, Bayesian network, back propagation, digital signal processor

1 Introduction

An artificial neural network is a system based on the operation of biological neural networks, in other words, is an emulation of a biological neural system (Hilera-González, 2000; Isasi and Galván, 2004). A neural network has at least two physical components, namely, the processing elements and the connections between them. The processing elements are called neurons, and the connections between the neurons are known as links. Every link has a weight parameter associated with it. Each neuron receives stimulus from the neighboring neurons connected to it, processes the information, and produces an output. Neurons that receive stimuli from outside the network (i.e., not from neurons of the network) are called input neurons. Neurons whose outputs are used externally are called output neurons. Neurons that receive stimuli from other neurons and whose output is a stimulus for other neurons in the neural network are known as hidden neurons. There are different ways in which the information can be processed by a neuron, and different forms of connecting neurons to another one. Different neural network structures can be constructed by using different processing elements and by the specific manner in which they are connected (Bishop, 2006; Chen, 2010).

One of the main applications for artificial neural networks is to recognize among different patterns (Goltsev, 2012; Jeong and Lee, 2012). In order to carry out this task it is needed a processing tool according to the computational load.

Nowadays microcontrollers have a limited mathematical processing capacity (Ibrahim, 2008; Tremberger et al., 2012), however, DSP's were made specifically for signal processing as its name means "Digital Signal Processor". These were developed by the U.S. semiconductor company Texas Instruments (TI), at present there are different companies that have developed their own versions of DSP as Motorola, Analog Devices and others.

The main contribution of this work is to tackle the pattern recognition problem applying an artificial neural network where the implementation is carried out in a TMS320C6713 DSP Starter Kit (DSK) employing a resistive touch screen as input method. For simplicity, the proposed technique implemented in a DSK6713 is able to recognize among three different patterns drawn in a resistive touch panel. The paper presents an implementation of an artificial neural network in a digital signal processor chip, for identification of the alphabet letters: A, M, and W. Although the processing level of the DSK6713 far exceeds the requirements of this application, the process of identifying patterns may contain, as future work, more challenging letters, for instance E and F or M and N, and others symbols

Some DSPs have the advantage of execute different processes in parallel that is why they can process much more information than other devices, so the applications of DSP are very diverse as communications, motor control, image processing, voice recognition, digital cameras or camcorders, MP3 players, high definition televisions (HDTV), etc. (Lee et al., 2010; Madisetti, 2010; Yolacan, 2011).

2 Problem statement and solution

The present paper describes the way we trained and applied an artificial neural network in a DSK6713 in order to be able to recognize among three different patterns drawn in a resistive touch panel.

A resistive touch screen were used for the acquisition of patterns, this screen has five terminals, two for power supply, two for x-axis and y-axis selection, and an analog output which is proportional to the pressed position. Subsequently, the analog output is converted by a 4-bit ADC and interpreted by the DSK (see Fig. 1).

The screen was discretized at a resolution of 9 x 9, which means, each pattern has a total of 81 elements.

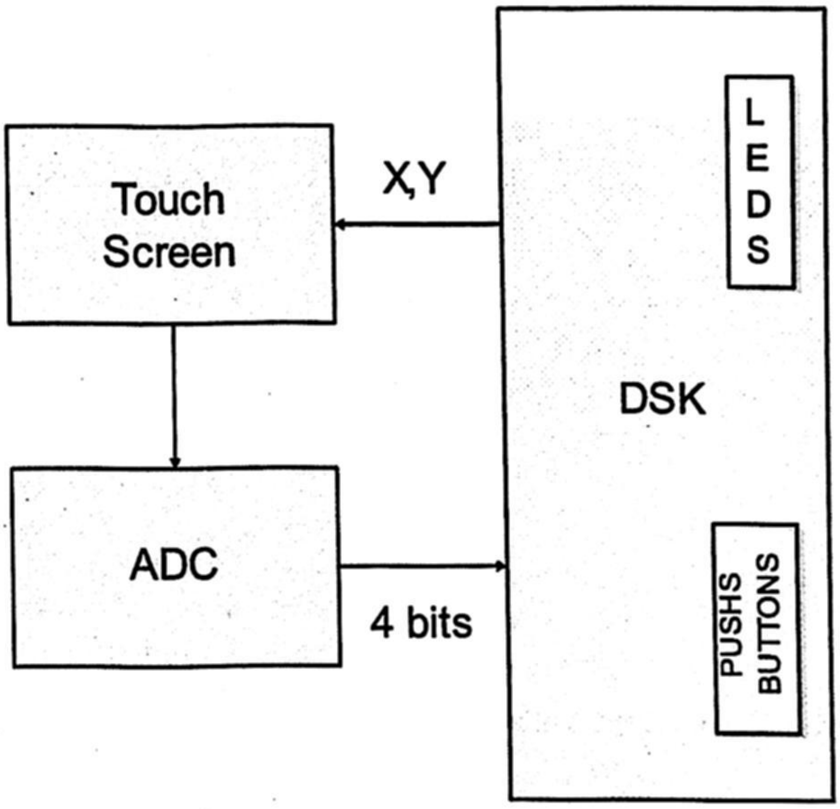


Fig. 1 Block diagram of the proposed prototype .

Within the DSK, each element pressed from the screen in a matrix form (XY) is interpreted as 1 and introduced a vector of 81 elements, the rest of them are 0. This vector is used as input to the ANN recognition stage.

Input / output peripherals of the DSK were used, LED's to indicate that DSK is in acquisition mode or to indicate the result of the pattern recognition and DIP switches to select between the 2 stages of the process.

While pressing DIP switch 0 of DSK LED0 flashes, during this time the DSK is receiving data from the ADC and touch screen in order to fill the input vector. When DIP switch 3 is pressed ANN is implemented to this vector and the results are displayed by activating one of the remaining 3 LEDs, each of these refers to a pattern.

The implemented ANN is Bayesian type and it was trained by back-propagation method (Fig. 2). It consists of 3 layers; the first layer (the input layer) has 81 inputs and 81 neurons, which output feeds the second layer (the hidden layer) which consists of 30 neurons, which output is the input for the last layer of 3 neurons (the output layer) which gives us the result of the ANN as is shown in Fig. 3.

The training of the ANN was performed in MATLAB ®. For these three training patterns (letters M, A and W), we have used three variants for each one, giving a total of 9 training vectors. These training vectors are shown in a graphical representation in Fig. 4.

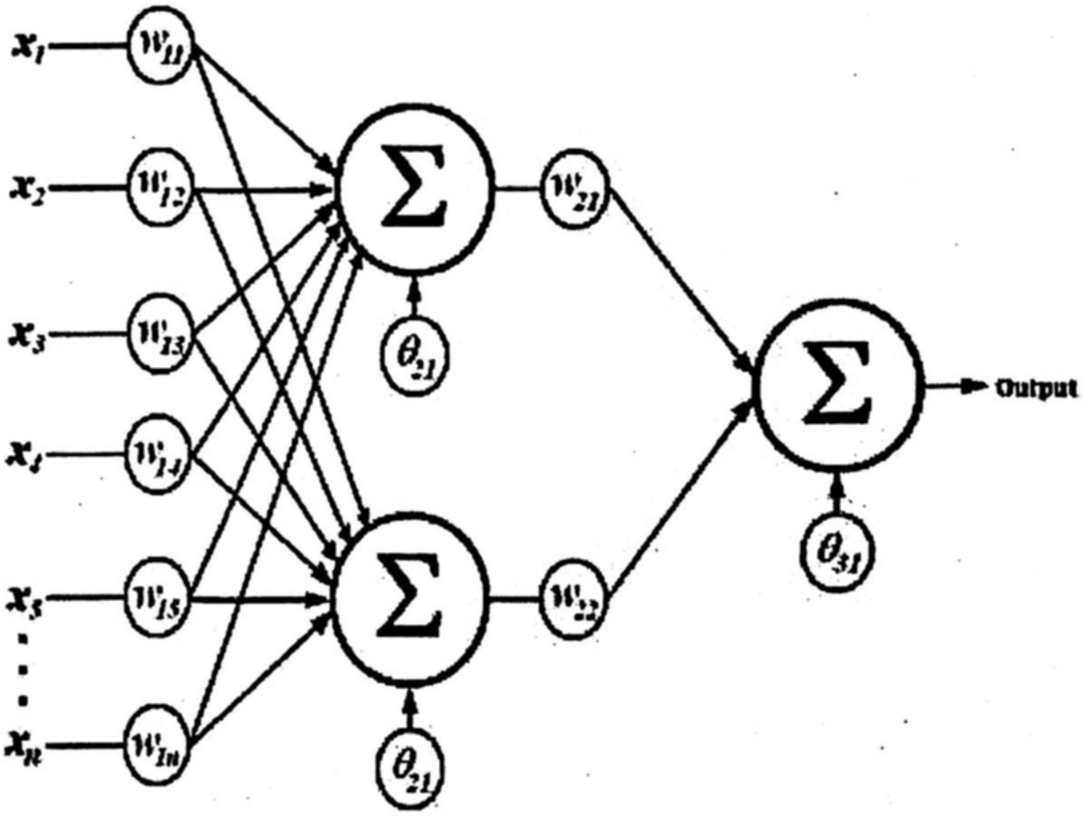


Fig. 2. Graphic representation of an artificial neural network.

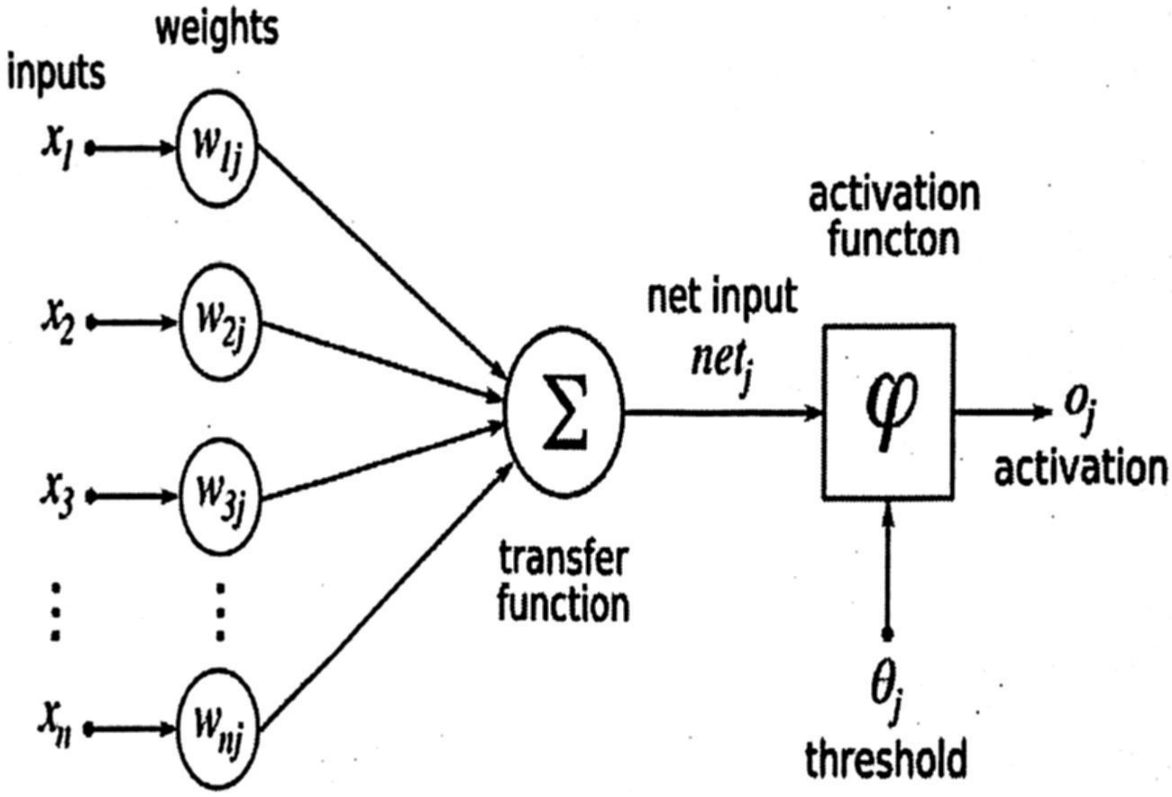


Fig. 3. Graphic representation of a neuron.

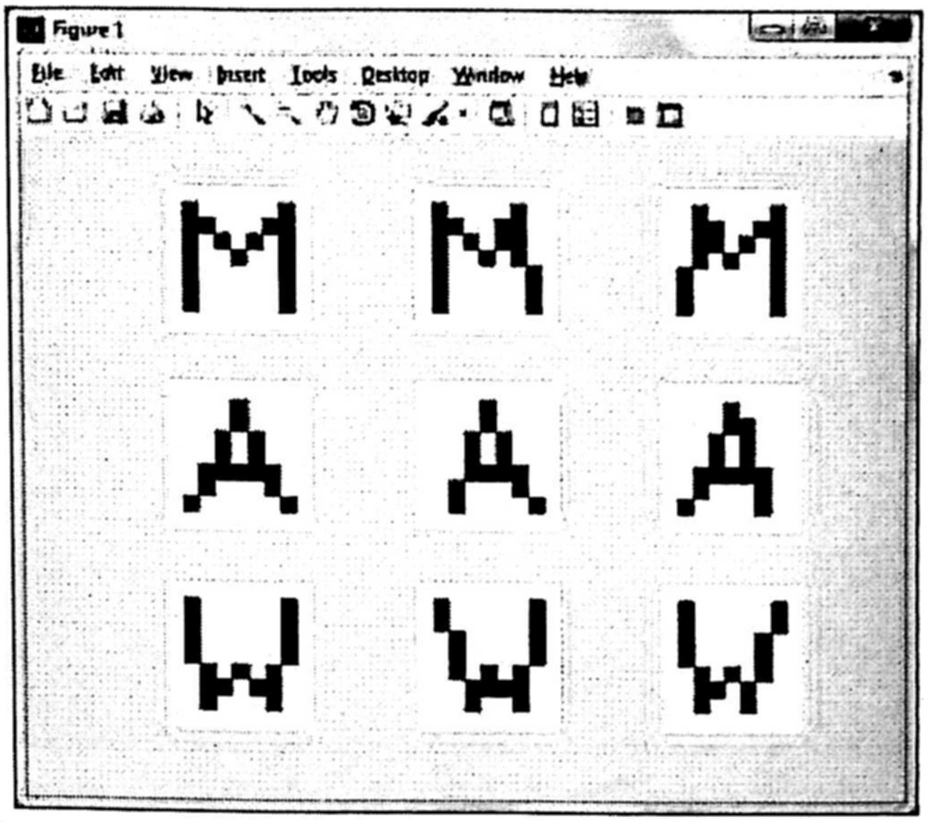


Fig. 4. Target vectors used for training.

For each pattern a target vector was defined, whose length is the number of neurons in the output layer, see Table 1.

Table 1. Patterns and output layers

	Pattern		
Output layer	'M'	'A'	'W'
Neuron 1	1	0	0
Neuron 2	0	1	0
Neuron 3	0	0	1

As a validation method for the ANN, it was introduced into the ANN the same training vectors previously described in Fig. 4. The results obtained from the validation are the expected one in each case.

Pattern 'M' outputs

1.0000
0.0000

0.0000

Pattern 'A' outputs

0.0000
1.0000
0.0000

Pattern 'W' outputs

0.0000
0.0000
1.0000

In this way we tested that the ANN responds appropriately to the training vectors.

3 Results

In order to demonstrate that the current project is working in a proper manner, around 100 different testing patterns were tested; these patterns and there results were stored in the DSK in order to be analyzed later. Figure 5 depicts the experimental platform.

The results were satisfactory in 96% of cases, since the patterns drawn on the touch screen were properly detected. Twelve of these patterns are shown in Table 2.

The results in Table 2 show the values of the output layer of the ANN, which means that the closer to 1, the output of neuron 1 pattern is identified by the network as an 'M' and so on. This is evident if compared with the target vector.

According to the tests the best identified pattern is the 'W' followed by 'A' and finally the pattern 'M'. Although some trials have values in the other two neurons, these are insignificant because they do not affect to the result.

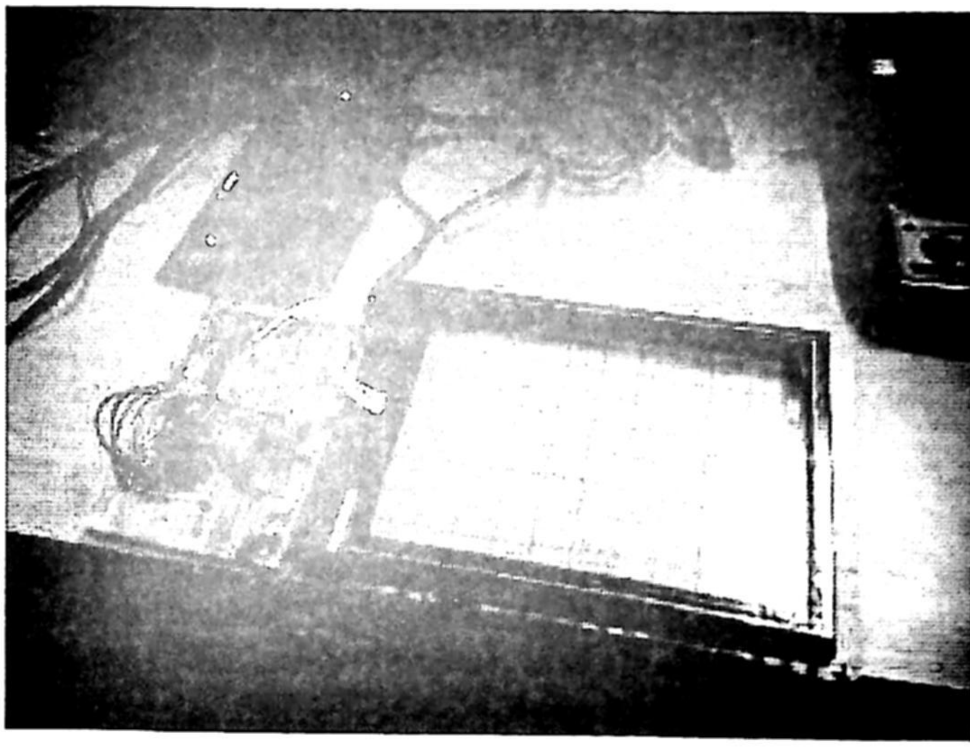
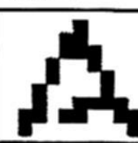
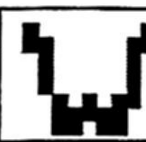
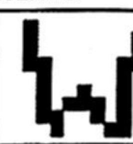
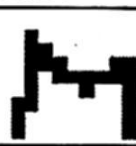
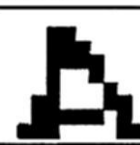
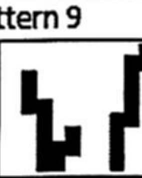


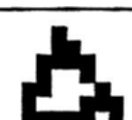
Fig. 5. Experimental implementation of the proposed prototype.

Table 2. Test Patterns.

	Pattern 1		Pattern 2		Pattern 3	
Neuron 1	0.9999		0.0116		0.0000	
Neuron 2	0.0000		0.9357		0.0000	
Neuron 3	0.0000		0.0000		1.0000	

	Pattern 4		Pattern 5		Pattern 6	
Neuron 1	1.0000		0.0000		0.0002	
Neuron 2	0.0000		0.9960		0.0000	
Neuron 3	0.0000		0.0000		0.9986	

	Pattern 7		Pattern 8		Pattern 9	
Neuron 1	0.9951		0.0000		0.0000	
Neuron 2	0.0000		1.0000		0.0000	
Neuron 3	0.0000		0.0000		1.0000	

	Pattern 10		Pattern 11		Pattern 12	
Neuron 1	0.9982		0.0000		0.0000	
Neuron 2	0.0001		0.9999		0.0000	
Neuron 3	0.0000		0.0000		1.0000	

4 Conclusion

Based on the performed tests, it can be determined that the system is able to successfully distinguish among the three patterns, even if the pattern has significant changes. Considering that the processing level of the DSK used far exceeds the requirements of this application, the process of identifying patterns may contain, as future work, all letters of the alphabet and others symbols, it also would be possible to increase the screen resolution for more detailed patterns.

5 Acknowledgements

M. Méndez, J. Mata-Machuca, L. Fonseca thank to the Secretaría de Investigación y Posgrado, Instituto Politécnico Nacional. J. Buendía, J. Galindo and M. Rodríguez are grateful for the support provided by the Instituto Politécnico Nacional through the PIFI scholarship program with the project SIP 20131714.

References

1. Bishop C (2006) Pattern recognition and machine learning. Springer, New York, USA.
2. Chen C(ed) (2010) Handbook of pattern recognition and computer vision, 4th Edition, World Scientific Publishing Co, Danvers, MA, USA
3. Goltsev A (2012) Investigation of efficient features for image recognition by neural networks, Neural Networks 28: 15-23
4. Hilera-González J (2000) Redes neuronales artificiales: fundamentos, modelos y aplicaciones, Alfaomega, México
5. Ibrahim D (2008) Advanced PIC microcontroller projects in C: form USB to Zigbee with the PIC 18F Series, Newnes, Elsevier, MA, USA
6. Isasi P, Galván I (2004) Redes neuronales artificiales: un enfoque práctico, Pearson-Prentice Hall, Madrid, España
7. Jeong S, Lee M (2012) Adaptive object recognition model using incremental feature representation and hierarchical classification, Neural Networks 25: 130-140
8. Lee H, Chakrabarti C, Mudge T (2010) A low-power DSP for wireless communications, IEEE Transactions on Very Large Scale Integration (VLSD) Systems 18(9): 1310-1322
9. Madisetti V (2010) The digital signal processing handbook, second edition, Digital signal processing fundamentals, CRC Press, Taylor & Francis Group, FL, USA
10. Ruge I, Alvarado D (2013) FPGA-based neural network evaluation system for image recognition, Tecnura 17(36): 87-95
11. Tremberger G, Armendariz R, Takai H, Holden T, Austin S, Johnson L P, Cheung T (2012). Applications of Arduino microcontroller in student projects in a community college. In American Society for Engineering Education. American Society for Engineering Education
12. Yolacan E., Aydin S, Ertunc H (2011) Real time DSP based PID and state feedback control of a brushed DC motor, In XXIII International Symposium, Information, Communication and Automation Technologies (ICAT), pp. 1-6

Muscle Pain and Blink Classification using a Brain Computer Interface

Oscar Montiel, Roberto Sepúlveda, Gerardo Díaz, Daniel Gutierrez, and Oscar Castillo

Instituto Politécnico Nacional, CITEDI. Tijuana, B.C., México
{oross,rsepulvedac}@ipn.mx
{gdiaz,dgutierrez}@citedi.mx
Instituto Tecnológico de Tijuana-ITT. Tijuana, B.C., México
{ocastillo}@tectijuana.mx
Paper received on 11/30/13, Accepted on 01/19/14.

Abstract. In this paper, two different architectures of artificial neural network (*ANN*) for the classification of blinking and arm pain caused by an external agent are used. The electroencephalographic (*EEG*) dataset is obtained from 20 people in the range of 23 to 30 years of age using a Brain Computer Interface (*BCI*), it is divided into a batch of necessary patterns to train and test the ANN. Experimental results using different training algorithms are shown.

Key words: EEG, BCI, ANN, FFT, Arm Pain, Blink, Adaptation Algorithm, MLP

1 Introduction

The human brain is a complex network of synaptic connections between neurons, which generate the electric impulses necessaries to develop human functions like movements, communication, language, feelings, memory, reasoning, etc.; these functions are represented by EEG signals [1]. Since Human Computer Interface (HCI) technology has allowed to read EEG signals in humans, it was thought for interpreting and using them as communication channels with auxiliary devices that can help people with mental and physical problems [7].

EEG signals are read and interpreted by a BCI system; these electrical signals are produced by the different stimulus as physical action (motion) or mental status as feeling, imagination, memory, etc. Using BCI devices have many applications in robotic prostheses, pattern recognition, studies of pathologies such as epilepsy, Alzheimer, Parkinson, etc.

This paper presents a methodology to classify EEG signals using a multilayer perceptron (*MLP*) trained with the backpropagation algorithm, to find patterns produced by different external stimulus, specifically muscle pain and eye blinking. It is organized as follows: In Section II, the Backpropagation algorithm is explained as a method of ANN training as well as an EEG overview for understanding how signals are interpreted and manipulated. Section III deals with the problem formulation, here the process used for implementing EEG Signal Processing and Classification is shown. In Section IV the methodology to present training patterns to the ANN is explained. In Section V and Section VI, the analysis of results and conclusions are shown, respectively.

2 Artificial Neural Network Overview

An Artificial Neural Network (*ANN*) is an assembly of interconnected and hierarchical organized simple processing elements; its functionality is inspired by the biological nervous system. The processing ability of the network is contained in the strength of the interconnection (weights) of its units, which is obtained through a process of adaptation of its parameters; the idea is to learn a set of patterns, which are the training examples [18] [20]. An ANN is a machine learning method inspired in how the brain works to solve any kind of problems by the association of neural information.

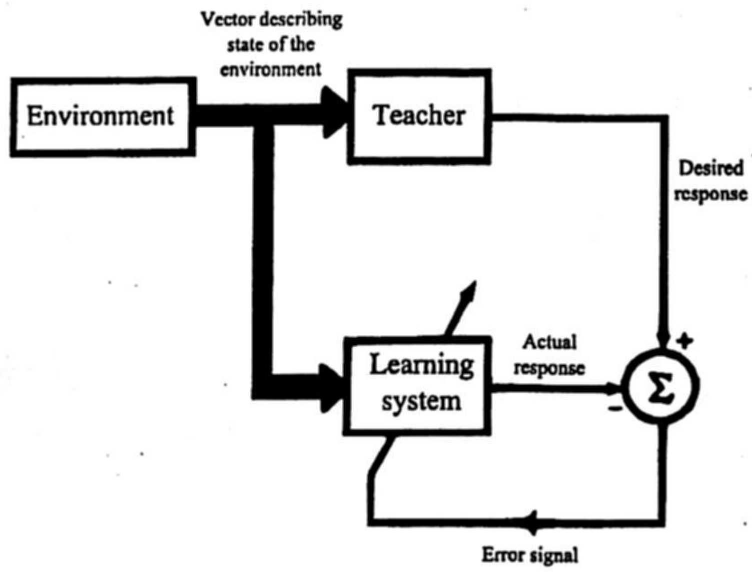


Fig. 1. Model representation of supervised mode training. This figure shows the different elements that conform this learning process.

2.1 Backpropagation Algorithm

This algorithm is a method of ANN training; it basically consists in using an error signal calculated by the difference between the actual $y_j(n)$ and desired output $d_j(n)$ of the network to correct the weights, in this algorithm the error is retropropagated from the output to the input then an optimization algorithm modifies the weights with the aim to reduce the error $e_j(n)$ at the individual neurons, hence the global output error called Mean Squared Error (MSE), the training finished when the MSE is less than the convergence error ε_0, see

Algorithm 1. Once trained the network, it has the ability to focus on the characteristics of an arbitrary input that resembles other previously seen, regardless any noise signals affecting the patterns [17].

The Backpropagation Algorithm could be used in two modes of training; supervised and non-supervised mode, respectively. The first one is defined as a type of network learning with a teacher, which is the expert problem solver, the knowledge is provided by tagged pairs of inputs and outputs to achieve the training, see Fig. 1. In the non-supervised mode, the ANN learns with no teacher, in this mode, there are no tagged examples of a function to be learn [18] [17].

The experiments shown in this paper were achieved using the supervised training mode; so, the teacher presents to the network the training dataset. In Algorithm 1, the pseudocode for implementing the Backpropagation Algorithm is shown [17].

Algorithm 1 Backpropagation Algorithm

procedure BACKPROPAGATION(*pattern*,*maxepoch*,ε_0,*Target*) ▷ ε_0 :=convergency error
  while $|MSE < \varepsilon_0|$ **do**
    repeat
      Calculate for each node
      $w_{ji} := d_j(n) + y_j(n)$
      $MSE(n) := \frac{1}{2}\sum_j \mid e_j(n) \mid^2$
      $w_{ji}(n+1) := d_j(n+1) + y_j(n+1)$
      $MSE(n+1) := \frac{1}{2}\sum_j \mid e_j(n+1) \mid^2$
      $epoch := epoch + 1$
    until *maxepoch* $\|$ ε_0 achieved
  end while
  return y
end procedure

2.2 EEG Overview

The technique of electroencephalography (*EEG*) is used to analyze the brain activity, which is manifested in electric waves. To accomplish EEG technique, an array of electrodes are placed on the scalp over multiple areas of the brain to detect and record the patterns of electrical activity. The electrodes are placed on the scalp according to the international 10-20 system of electrode position [10].

It is important to know that the brain is divided by sections, two hemispheres (left and right) and four lobules (frontal, parietal, temporal and occipital), where each section is related to specific sections of the human body. For example, to analyze a stimulus on the left side of the body the right hemisphere has to be analyzed, and viceversa. The scheme of a BCI presented in the Fig. 2 is divided into three main sections: In the first section, the electric activity generated by the brain as well as the interface to acquire the activity is shown. In the second

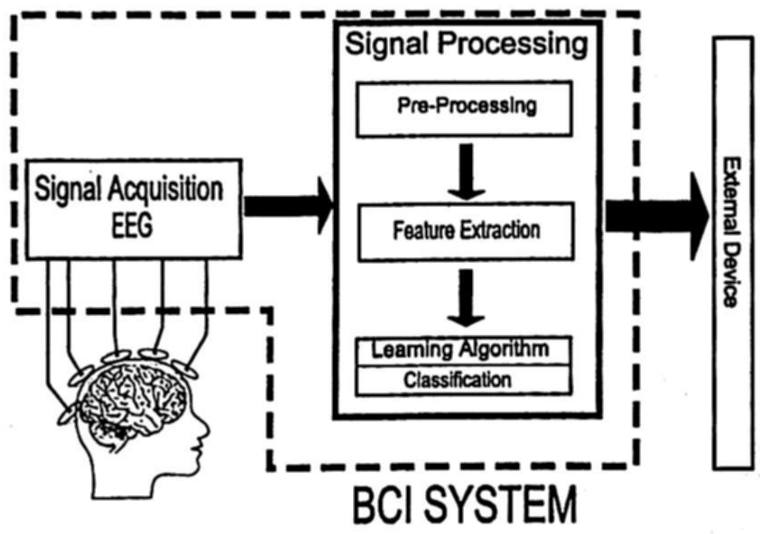


Fig. 2. Scheme of a Brain Computer Interface (BCI) system

section, the signal-processing block is in charge of conditioning and manipulating data using methods as correlation, Artificial Neural Networks, ANFIS, etc., to accomplish a task. The third section consists in the application, here a variety of algorithms can be implemented, for example, wheel chair control by mind, health assistant, rehabilitation, epilepsy detection, mind control of a prosthesis [7] [12] [8] [15].

The brain activity has specific characteristics like time, frequency, amplitude, magnitude and kurtosis; some of them can be analyzed in time or frequency domain.

3 Problem Formulation

The experimental platform of Fig. 3 shows the modules used to process EEG signals; it is a flexible system based on the described block diagram shown in Fig. 2 that supports the process of research and development.

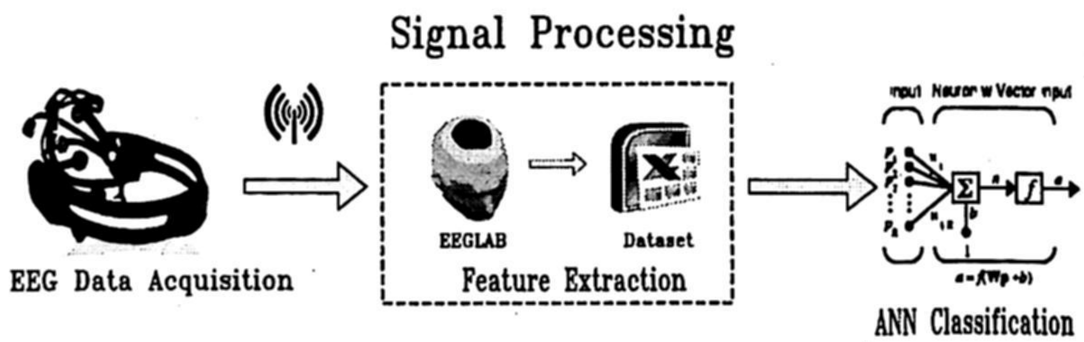


Fig. 3. EEG Signal Processing and Classification using a BCI system.

In the first section of Fig. 3, the EEG Data is acquired by the EPOC Neuroheadset and is transmitted using wireless communication. Once the data has been received by the computer system, it is recorded for postprocessing, then using a Feature Extraction method some aspects of the signal like, amplitude, time duration, form, magnitude, frequency bands, etc., are determined, to accomplish this EEGlab and Excel are used. The EEGlab is an open-source toolbox for Matlab, which is used to study the offline EEG data already recorded; this software package has different properties for analyzing dipole sources, Independent Component Analysis (ICA), Fast Fourier Transform (FFT), Wavelet Transform, etc. The EEGlab software can read samples from EPOC Neuroheadser and save them in Excel format. Then, the dataset is divided in patterns that are used for training and testing the Neural Network [21].

4 Proposed Methodology

Once the data has been processed, the next step is to select patterns based on a teacher heuristic. All EEG signals are divided in patterns to build the batch of patterns used as input of the ANN. Fig. 4 illustrates how the batch of patterns feed the ANN.

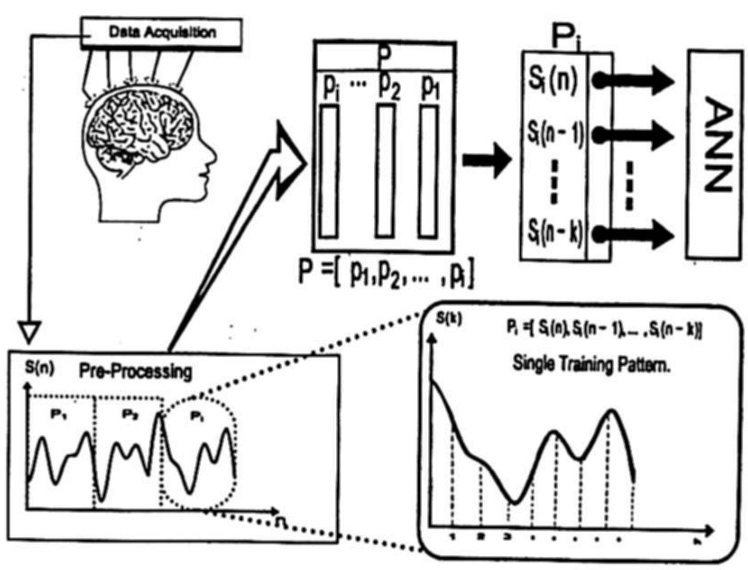


Fig. 4. The EEG signals are sampled using the EPOC neuroheadset; a time series is divided into lots of training patterns P, each pattern consists of multiple data streams that will serve either to train the network, or to verify the training.

Each pattern contains a number of samples to train and test the ANN; the number of samples for each pattern depends on the stimulus that required to be classified. In Algorithm 2, the pseudocode for EEG signal processing is shown.

4.1 Muscle Pain Classification

The first stimulus to classify is the pain induced by an external agent [4] [14], this stimulus is induced by a prick in the right arm; the subject must be in a relaxed status for two minutes approximately, and then, the pain is induced [6].

Algorithm 2 EEG Signal Processing general method proposal

Initialize variables
Select electrodes to be analyzed
Create Inputelectrode matrix from EEG readings ▷ **If it is necessary, apply FFT to Inputelectrode matrix**
Normalize Inputelectrode matrix already changed
Select TRAINDATA, TARGET and TESTDATA to train ANN
while Any data be minor than size of train dataset **do** ▷ Training Data
    TRAINDATA accumulate pattern for each iteration
end while
while Any data be minor than size of test dataset **do** ▷ Testing Data
    TRAINDATA accumulate pattern for each iteration
end while
Define ANN TARGET ▷ Depending activation function
Define ERROR, MAXEPOCH ▷ ANN training parameters
Train ANN
Test ANN

The EEG signal obtained after the prick is the most important information used for training and testing the ANN [5]. Fig. 5 explains the experimental process of pain activity classification.

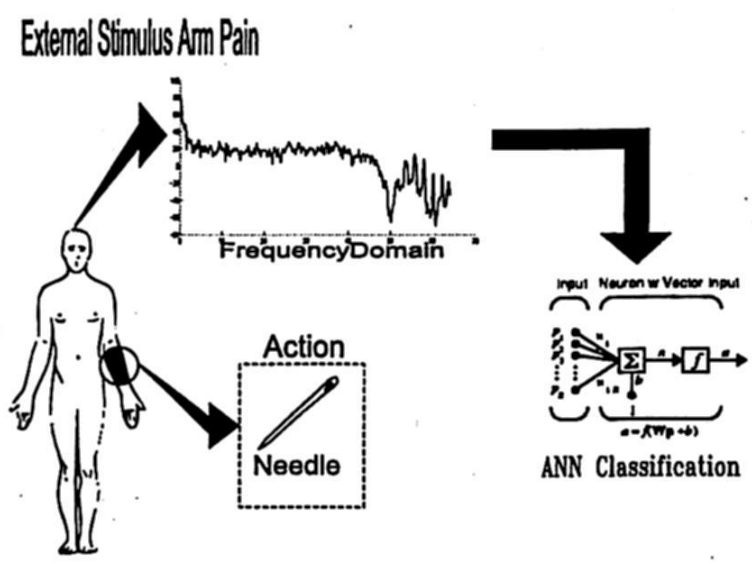


Fig. 5. Scheme for implementing EEG signal processing and classification of muscle pain.

It is worthwhile to mention that the pain cannot be appreciated in the time domain; therefore, the signal has to be converted to frequency domain using the Fast Fourier Transform, then a filter is applied to eliminate noise in the signal. All the experiments were saved in an Excel Table (dataset); it was divided into two sets; one set is used to train the ANN, and the second set for testing it, which is important for proving the network knowledge generalization capability.

4.2 Eye Blinking Classification

Fig. 6 shows the experimental process to achieve the eye blinking classification; although, it is considered an artifact, it is important to take care of it because this artifact is present in the whole EEG encephalographic readings. Blinking is a natural body movement that helps to maintain the eyes wet and protected from external elements [16] [11].

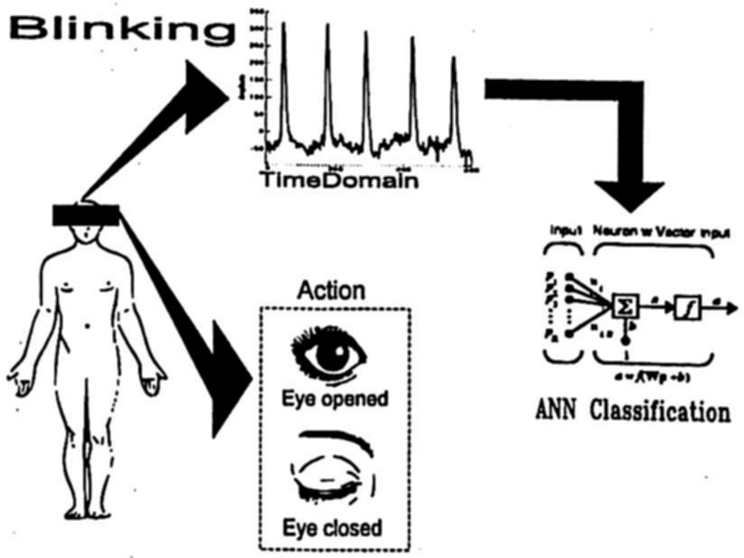


Fig. 6. Implementing eye blinking signal processing and classification.

As it is seen in Fig. 6 each time the eyelid changes from open to close, an increment of amplitude in microvolts appears in the readings, this data can be used to generate the dataset for the ANN [9].

It is important to achieve a proficient identification and classification of eye blinking for a good management of EEG signals, because it may affect the analysis of other EEG stimulus classifications [16].

5 Experimental Results

Two cases of study for the classification of EEG signals are presented using ANN multilayer perceptron type with hyperbolic tangent activation functions, it was trained in supervised mode with the backpropagation algorithm. For training, a +1 value is assigned to a valid stimulus, and -1 for a non-valid stimulus. In the operation mode, the ANN returns values between 1 and -1, to make the classification possible a threshold of +0.9 is applied, hence any value equals or over the threshold is considered a valid stimulus, otherwise it is a non-valid stimulus.

The first case shown in Fig. 5 consists in identifying muscle pain induced by an external agent. The second case is to identify the blink of both eyes as it is shown in Fig. 6; although the blink signals are not true encephalographic signals, they are artifacts and their study is considered important, because they interfere with the EEG results and their interpretation. In all the conducted experiments in this work, the scheme shown in Fig. 3 was used.

The adaptive algorithms have the purpose to accelerate the convergence of the error for neural network by optimizing the weights in the learning process. In this research, the adaptive algorithms shown in Table 1 were used to investigate which one have the best characteristics in this kind of classification.

Table 1. Adaptive algorithms used for experiments.

Acronym	Adaptive algorithm
LM	Levenberg Marquardt
OSS	One Step Secant
BFGS	BFGS Quasi-Newton
RP	Resilient Backpropagation
GD	Gradient Descent

The ANN architectures used for both experiments are summarized in Table 2. To establish a benchmark to compare different adaptive algorithms, a convergence error of $1e^{-3}$ was established. Fig. 4 illustrates in general terms, how the different training patterns were generated and used in batch mode. To perform the experiments, a computer with a processor I7 2.67Ghz 920A, with 6GB of RAM and OS Windows 7 of 64 bits was used.

Table 2. ANN architectures for external stimulus classification.

Stimulus	ANN Architectures
Blinking (both eyes)	120:20:10:5:1
Pain (right arm)	1280:20:10:5:1

Table 3 shows some parameters about training ANN for the pain arm stimulus. The LM algorithm has the best qualities for training and testing (i.e. knowledge generalization), 99.7% and 98.8%, respectively; moreover, it is the fastest algorithm in the training mode, it last a meantime of 1167 seconds.

Table 4 shows statistical values of time about training ANN for blinking. The RP last a meantime of 0.50 seconds, which means that RP was the fastest algorithm to train the blinking patterns. Also, the best rate of classification was obtained with this algorithm, for the training patterns a 99.5% of successful was achieved, and 96.4% for the test pattern.

Table 3. Statistical results of ANN training to classify pain in the right arm. Different adaptive learning algorithms were used.

Adaptive learning algorithm	Mean time(sec)	Min time(sec)	Max time(sec)	Desv. Std(sec)	Mean epochs
LM	1167.00	420.00	2700.00	468.53	33344
OSS	1864.43	840.00	5687.00	1012.52	36462
BFGS	1403.60	600.00	4140.00	706.33	33677
RP	1568.00	840.00	4800.00	771.39	48982
GD	1290.00	720.00	4140.00	728.87	39677

Table 4. Statistical results of ANN training to classify eye blinking. Different adaptive learning algorithms were used.

Adaptive learning algorithm	Mean time(sec)	Min time(sec)	Max time(sec)	Desv. Std(sec)	Mean epochs
LM	9.43	2.00	28.00	5.89	11
OSS	0.83	0.49	1.70	0.26	27
BFGS	145.97	59.00	269.00	60.32	18
RP	0.50	0.36	2.79	0.43	27
GD	22.43	10.00	55.00	10.20	3825

Table 5. Classification results for the blinking activity using the training and testing data. The values express the percentage of true classification.

Learning algorithm	Train data	Test data
LM	98.7	93.5
OSS	99.3	94.1
BFGS	99.3	93.0
RP	99.5	96.4
GD	99.7	88.8

Table 6. Classification results for the right arm pain using the training and testing data The values express the percentage of true classification.

Learning algorithm	Train data	Test data
LM	99.7	96.9
OSS	96.6	95.4
BFGS	97.4	93.1
RP	95.3	93.0
GD	98.7	96.8

6 Conclusions and Future Work

It is known that the use of ANNs is a powerful and efficient tool to classify EEG signals; however, there is not enough research works focused to pain detection caused by external agents; this work provides valuable information in this field. In addition, eye blinking signals classification has been included because at the present, it is well known its precise classification importance to improve EEG interpretation.

Several experiments with different learning algorithms for a multilayer perceptron type ANN were achieved. In all the experiments, we used a dataset consisting of 60 patterns of each person; 30 of them were used for training, and the rest for testing the network; a total of 20 people in the range of 23 to 30 years of age were used to complete the whole dataset.

The muscle pain classification experiments were performed by inducing pain with a needle in the right arm of a person. Statistical results demonstrate that the LM algorithm performs better than the others shown in Table 3, with this algorithm once trained the network, we obtain a 99.7% of reliability classifying the training dataset, and 96.9% using the test dataset which is fine because the patterns in the test set are unknown for the ANN. It is worth mentionion that the LM proved to be the fasted algorithm for training.

For eye blinking classification, the GD was the better recognizing the training patterns but the less reliable with the test pattern. The RP is almost as good as the GD recognizing training patterns, and it is the most reliable classifying the test patterns; with the RP, a reliability percentage of 99.5% with the training patterns and 96.4% of reliability with the test patterns was obtained.

With respect to the statistical results, hence the reliability percentages, it is important to mention that a 0.9 recognition umbral was used; therefore, these results might be improved by reducing the umbral value.

7 Acknowledgments

The authors thank to the "Instituto Politécnico Nacional (IPN)", to the "Instituto Tecnológico de Tijuana (ITT)", and the "Consejo Nacional de Ciencia y Tecnología (CONACYT)" for their support.

References

1. Arias, G. , Felipe, H.: Detección y clasificación de artefactos en señales eeg. In Memorias de STSIVA 09. Universidad Tecnológica De Pereira. (2009).
2. Balakrishnan, D.; Puthusserypady, S.: Multilayer perceptrons for the classification of brain computer interface data. Bioengineering Conference, 2005. Proceedings of the IEEE 31st Annual Northeast , pp.118,119, 2–3 (2005).
3. Chambayil, B., Singla, R., Jha, R.: Virtual keyboard bci using eye blinks in eeg. In Wireless and Mobile Computing, Networking and Communications (WiMob), 2010 IEEE 6th International Conference on. 466-470. (2010).

4. Chang, P., Arendt-Nielsen, L., Graven-Nielsen, T., Svensson, P., Chen, A.: Different eeg topographic effects of painful and non-painful intramuscular stimulation in man. Experimental Brain Research, 141, 195-203. (2001).
5. Chang, P. F., Arendt-Nielsen, L., Graven-Nielsen, T., Svensson, P., Chen, A. C.: Comparative eeg activation to skin pain and muscle pain induced by capsaicin injection. International journal of psychophysiology, 51(2), 117-126. (2004).
6. Chen, A. Rappelsberger, P.: Brain and human pain: Topographic eeg amplitude and coherence mapping. Brain Topography, 7(2), 129-140. (1994).
7. De la O Chavez, J. R.: BCI para el control de un cursor basada en ondas cerebrales. Masters thesis, Universidad Autónoma Metropolitana. (2008).
8. Erfanian, A.; Gerivany, M. .: EEG signals can be used to detect the voluntary hand movements by using an enhanced resource-allocating neural network. Engineering in Medicine and Biology Society, 2001. Proceedings of the 23rd Annual International Conference of the IEEE , vol.1, no., pp.721,724 vol.1 (2001).
9. Gabor AJ, Seyal M.: Automated interictal EEG spike detection using artificial neural networks. Electroencephalogr Clin Neurophysiol 1992; 83: 271-280 (1992).
10. Hirsch, L. Richard, B.: Atlas of EEG in Critical Care, chapter EEG basics. Wiley, 1-7. (2010).
11. Kutlu, Y.; Isler, Y.; Kuntalp, D.: Detection of Spikes with Multiple Layer Perceptron Network Structures. Signal Processing and Communications Applications, 2006 IEEE 14th, vol., no., pp.1, 4, 17–19. (2006).
12. Lin, J.-S., Chen, K.-C., Yang, W.-C.: Eeg and eye-blinking signals through a brain-computer interface based control for electric wheelchairs with wireless scheme. In New Trends in Information Science and Service Science (NISS), 2010 4th International Conference on. 731-734. (2010).
13. Papadourakis G, Vourkas M, Micheloyannis S, Jervis B.: Use of artificial neural networks for clinical diagnosis. Math Comput Simulation 1996; 40: 623-635 (1996).
14. Pera, D. L., Svensson, P., Valeriani, M., Watanabe, I., Arendt-Nielsen, L., Chen, A. C.: Long-lasting effect evoked by tonic muscle pain on parietal EEG activity in humans. Clinical Neurophysiology, 111(12), 2130-2137. (2000).
15. Pérez, M. Luis, J.: Comunicación con Computador mediante Señales Cerebrales. Aplicación a la Tecnología de la Rehabilitación. Ph.D. thesis, Universidad Politécnica de Madrid. (2009).
16. Sovierzoski, M., Argoud, F., De Azevedo, F.: Identifying eye blinks in eeg signal analysis. In Information Technology and Applications in Biomedicine, 2008. ITAB 2008. International Conference on. 406-409. (2008).
17. Mitchell, T.M.: Machine Learning, Chapter 4, McGraw-Hill. (1997).
18. Haykin, S.: Neural Networks A Comprehensive Foundation, Delhi, India: Pearson Prentice Hall. (1999).
19. Haselsteiner, E.: Using Time-Dependent Neural Networks for EEG Classification, IEEE TRANSACTIONS ON REHABILITATION ENGINEERING, vol. 8, no. 4, pp. 457–463, (2000).
20. Hagan, M. T.: Neural Network Design, Boston, MA: PWS Publishing Company, (1996).
21. Morchen, F.: Time series feacture extraction for data mining using DWT and DFT, Philipps University Marburg, Marburg, Germany, (2003).