Brain Computer Interface Development Based on Recurrent Neural Networks and ANFIS Systems

Emanuel Morales-Flores[1], Juan Manuel Ramírez-Cortés[1], Pilar Gómez-Gil[2], and Vicente Alarcón-Aquino[3]

[1] Department of Electronics, National Institute of Astrophysics, Optics and Electronics, Tonantzintla, Puebla, Mexico
[2] Department of Computer Science, National Institute of Astrophysics, Optics and Electronics, Tonantzintla, Puebla, Mexico
[3] Department of Electronics, University of the Americas, Cholula, Puebla, Mexico

Abstract. Brain Computer Interfaces (BCI) is the generic denomination of systems aiming to establish communication between a human being and an automated system, based on the electric brain signals detected through a variety of modalities. Among these, electroencephalographic signals (EEG) have received considerable attention due to several factors arising on practical scenarios, such as noninvasiveness, portability, and relative cost, without lost on accuracy and generalization. In this chapter we discuss the characteristics of a typical phenomenon associated to motor imagery and mental tasks experiments, known as event related synchronization and desynchronization (ERD/ERS), as well as its energy distribution in the time-frequency space. The typical behavior of ERD/ERS phenomenon has led proposal of different approaches oriented to the solution of the identification problem. In this work, an architecture based on adaptive neuro-fuzzy inference systems (ANFIS) assembled to a recurrent neural network, applied to the problem of mental tasks temporal classification, is presented. The electroencephalographic signals (EEG) are pre-processed through band-pass filtering in order to separate the set of energy signals in alpha and beta bands. The energy in each band is represented by fuzzy sets obtained through an ANFIS system, and the temporal sequence corresponding to the combination to be detected, associated to the specific mental task, is entered into a recurrent neural network. Experimentation using EEG signals corresponding to mental tasks exercises, obtained from a database available to the international community for research purposes, is reported. Two recurrent neural networks are used for comparison purposes: Elman network, and a fully connected recurrent neural network (FCRNN) trained by RTRL-EKF (real time recurrent learning – extended Kalman filter). A classification rate of 88.12 % in average was obtained through the FCRNN during the generalization stage.

P. Melin, O. Castillo (Eds.): *Soft Computing Appli. in Optimization,* STUDFUZZ 294, pp. 215–236.
DOI: 10.1007/978-3-642-35323-9_9

1 Introduction

Brain Computer Interfaces are systems aiming to translate the electrical brain signals generated by a human being as a results of some thoughts, in commands able to perform some control actions in computerized mechanisms. In other words BCIs measure brain activity, process it, and produce control signals that reflect the user's intent. Brain activity produces several physical phenomena which can be measured using a variety of sensing equipment. Among these phenomena, which can be of significant relevance for BCI development, are electrical potentials and hemodynamic measurements. Electrical potential measurements include action and field potentials which can be sensed through invasive methods, such as electro-corticography, and non-invasive, such as electroencephalography and magneto-encephalography techniques. Hemodynamic measurements include functional magnetic resonance imaging (fMRI), positron emission tomography (PET), and functional near-infrared brain monitoring (fNIRS). Among these, electroencephalographic signals (EEG) have received considerable attention due to several factors arising on practical scenarios, such as noninvasiveness, cost effectiveness, portability, ease of acquisition, and time resolution, which are ideal attributes for the development of practical brain computer interface applications. There are three main stages which can be distinguished in a BCI system: detection of the neural signals from the brain, an algorithm for decoding these signals, and a methodology for mapping decoded signals into some predefined activities. The general scheme of a BCI is shown in Fig. 1.

In recent years, there has been a growing interest in the research community on signal processing techniques oriented to solve the multiple challenges involved in BCI applications [1-3]. An important motivation to develop BCI systems, among

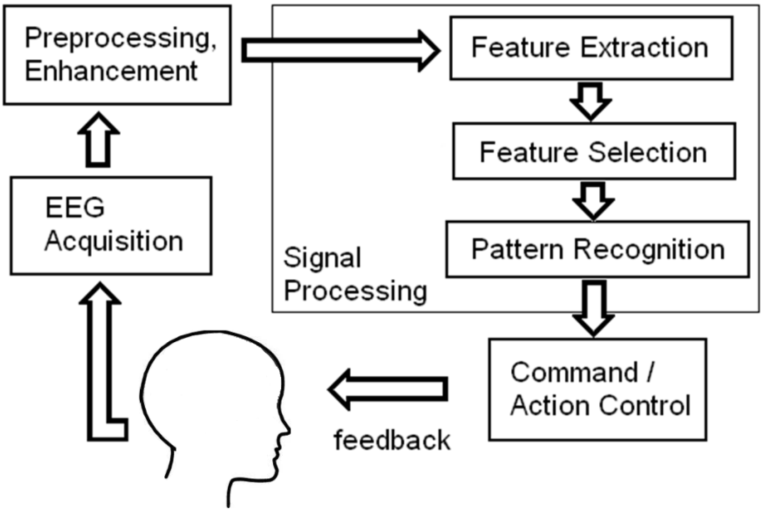


Fig. 1 General scheme of a Brain Computer Interface system

some others, would be to allow an individual with motor disabilities to have control over specialized devices such as computers, speech synthesizers, assistive appliances or neural prostheses.

A dramatic relevance arises when thinking about patients with severe motor disabilities such as locked-in syndrome, which can be caused by amyotrophic lateral sclerosis, high-level spinal cord injury or brain stem stroke. BCIs would increase an individual's independence, leading to an improved quality of life and reduced social costs. Electroencephalography (EEG) refers to recording electrical activity from the scalp with electrodes. A BCI based on EEG analyzes ongoing electric brain activity for brain patterns that originate from specific brain areas. To get consistent recordings from specific regions of the head, scientists rely on a standard system for accurately placing electrodes, which is called the International 10–20 System [4], generally used in clinical EEG recording and EEG research as well as BCI field. The name 10–20 indicates that the most commonly used electrodes are positioned 10, 20, 20, 20, 20, and 10% of the total naison-inion distance. Fig. 2 shows the electrode positions and denominations used in the international 10-20 system.

Measuring brain activity effectively is a critical step for brain–computer communication. However, measuring activity is not enough, because a BCI can only detect and classify specific patterns of activity in the ongoing brain signals that are associated with specific events. What the BCI user makes to produce these patterns is determined by the neurological mechanisms or processes that BCI system employs.

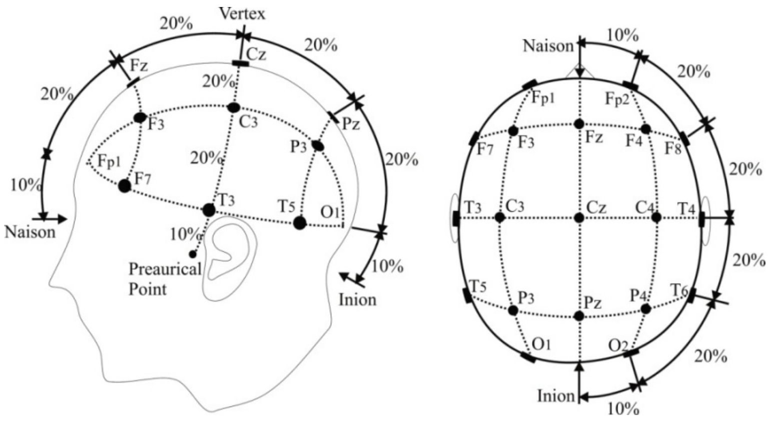


Fig. 2 EEG electrodes international 10-20 system

Current research on BCI systems distinguishes seven main categories according to the neurological mechanisms or processes involved: sensorimotor activity [5,6], P300 [7,8], visual evoked potentials [9,10], slow cortical potentials [11], activity of neural cell and response to mental tasks [12], as well as multiple neuro-mechanisms, which use a combination of two or more of the previous (see [2] for a review). Each category constitutes a paradigm which can be used for developing BCI systems in practical scenarios. P300 evoked potentials occur with latency around 300 milliseconds in response to target stimuli that occur unexpectedly. In a P300 controlled experiment, subjects are usually instructed to respond in a specific way to some stimuli, which can be auditory, visual, or somatosensory. P300 signals come from the central-parietal region of the brain and can be found more or less throughout the EEG on a number of channels. The P300 is an important signature of cognitive processes such as attention and working memory and an important clue in the field of neurology to study mental disorders and other psychological dysfunctions [8]. Another neurological mechanism widely studied for developing BCI systems is motor imagery (MI), which is obtained from the sensory motor brain activity. In general, two types of patterns are usually present in this mechanism: event related potentials (ERP), detected as energy changes in α (8-13 Hz), and β (14-20 Hz) bands generated when a voluntary movement is performed, and movement related potentials (MRP), which are low frequency patterns that initially appear between 1–1.5 s before the corresponding movement. In the first case, the event related potentials consist, in general terms, in decrements or increments of the energy on the ongoing EEG signal at certain frequency bands, which are described in the literature as the ERD/ERS phenomenon (Event Related Desynchronization and Synchronization) [13,14]. A crucial issue is to successfully estimate and translate the ERD/ERS phenomenon into a meaningful feature vector which can be used as input to some pattern recognition scheme. The analysis should be able to capture the spectral dynamic of the signal contained in the temporal evolution of the involved spectral bands. Several feature extraction techniques have been used for that purposes, such as: amplitude values of EEG [15], band power [16], power spectral density [17,18], auto-regressive (AR) and adaptive auto-regressive models (AAR) [19], windowed Fourier analysis, cross correlation, and some others. As these ERPs are locked in time but not in phase and they are highly non-stationary [20], the detection of these patterns turns into a difficult task in which some approaches oriented to follow the time evolution of the signals, such as time series prediction, and recurrent neural networks, could provide adequate results.

Another neurological mechanism in which ERD/ERS phenomenon is also present is the neural activity obtained in response to mental tasks. Mental task-based BCI systems have captured the attention of the research community, in part due to their independence of additional interfaces such as the screen of alphanumeric characters used in VEP, or the arrows and symbols used in motor imagery experiments, as well as the relative flexibility of the user to carry out some mental tasks at his /her own will. Several feature extraction methods for mental task-based BCI design have been reported, most of them based on parametric, such as autoregressive or adaptive models [21], non-parametric models based on several schemes of spectral analysis such as Wavelet transform or Stockwell transform [22,23], or fuzzy sets [24]. In this sense, it has been shown that information contained in spectral bands α (8-13 Hz), β (14-20 Hz), γ (24-37 Hz), or even in higher frequencies [25], can be used to detect neural activity directly related to specific mental tasks. Time-frequency analysis can be carried out using different approaches such as Wavelet analysis [22], filter bank [26], empirical mode decomposition [27], and others. Those approaches reflect only the estimated power across a range of frequencies. In a number of reported works, non-linear classifiers such as neural network and support vector machine algorithms are used [28]. Recently, there have been several studies oriented to capture temporal behavior through predictive schemes and recurrent neural networks with good results, which encourage further research in that direction [29-30].

To achieve the goal of translating brain activity into commands for computers there are two main approximations: regression and classification algorithms. Using classification algorithms is the most popular approach to identify patterns of brain activity. Most brain patterns used to control BCI are related to time variations of EEG in specific frequency bands. The time course of EEG signals has to be taken into account during feature extraction and one alternative is using a dynamical classifier. To obtain temporal information it is necessary to extract features from several time segments in order to build a temporal sequence. In this work we present a temporal classification approach on a two-state mental task experiment applying, for comparison purposes, two recurrent neural networks: Elman and Fully Connected Recurrent Neural Network (FCRNN). The proposed scheme performs the feature extraction based on an Adaptive Neuro-fuzzy Inference System (ANFIS), previous to the temporal classification stage.

The rest of the chapter is organized as follows: Section 2 describes theory related to ANFIS. Section 3 presents mathematical background associated to recurrent neural networks. Section 4 describes the proposed methodology on temporal classification of the mental task experiment. Section 5 presents and analyzes the obtained results. Section 6 presents some concluding remarks, perspectives, and future direction of this research oriented to the implementation of a BCI system.

2 Adaptive Neuro-Fuzzy Inference System (ANFIS)

Adaptive Neuro Fuzzy Inference Systems (ANFIS) combine the learning capabilities of neural networks with the approximate reasoning of fuzzy inference algorithms. Embedding a fuzzy inference system in the structure of a neural network has the benefit of using known training methods to find the parameters of a fuzzy system. Specifically, ANFIS uses a hybrid learning algorithm to identify the membership function parameters of Takagi-Sugeno type fuzzy inference systems. The task of the learning algorithm for this architecture is to tune all the modifiable parameters defining the fuzzy partitions and making the ANFIS output match the training data. In this work, the ANFIS model included in the MATLAB toolbox has been used for experimentation purposes. A combination of least-squares and backpropagation gradient descent methods is used for training the FIS membership function parameters to model a given set of input/output data through a multilayer neural network. ANFIS systems have been recently used for optimization, modeling, prediction, and signal detection, among others [31,32]. The ANFIS architecture (type-3 ANFIS) is shown in Fig. 3.

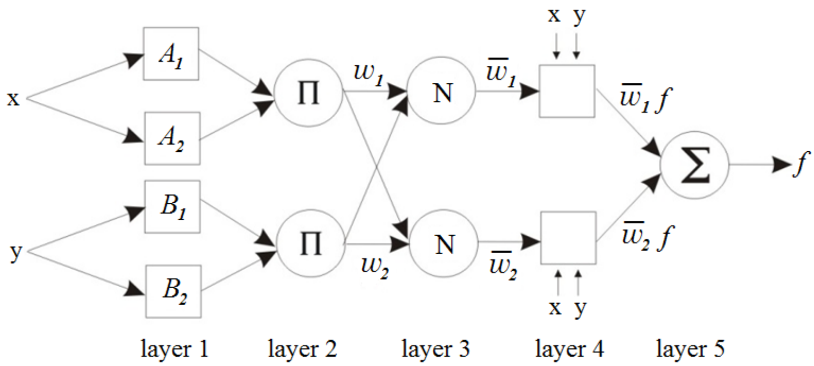


Fig. 3 ANFIS architecture

In this figure x and y are inputs to the node i in layer 1. A_i and B_i are linguistic labels e.g. (small, medium, large, etc.). In other words, the output of each node is the membership function of A_i and B_i, and specifies the degree to which the given x or y satisfies the quantifier A_i and B_i respectively. The output of each node in this layer is described as follows:

$$O_i^1 = \mu_{A_i}(x)$$

Every node in layer 2 is a circle node labeled which multiplies the incoming signals and sends the product out.

$$\omega_i = \mu_{A_i}(x) \times \mu_{B_i}(x)$$

In layer 3 each node is a circle node labeled N. The ith node calculates the ratio of the ith rule's firing strength to the sum of all rules' firing strengths:

$$\overline{\omega} = \frac{\omega_i}{\omega_1 + \omega_2}, i = 1, 2$$

Every node in layer 4 is a square node that performs the following function:

$$O_i^4 = \overline{\omega}_i f_i = \overline{\omega}_i (p_i x + q_i y + r_i) \quad ,$$

where $\{p_i, q_i, r_i\}$ is the parameter set.

The single node in the 5 layer is a circle node labeled $\sum$ that computes the overall output as the summation of all incoming signals

$$O_i^1 = \sum_i \overline{\omega}_i f_i = \frac{\sum_i \omega_i f_i}{\sum_i \omega_i}$$

The architecture presented is functionally equivalent to a type-3 fuzzy inference system. For detailed information see reference [33].

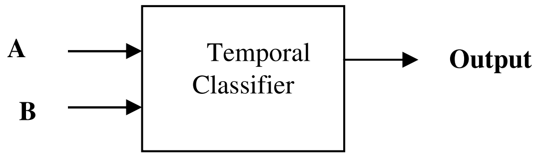
3 Neural Network Classifiers

Nowadays, artificial neural networks are a popular tool to tackle complex classification problems. Specifically, the ability of recurrent neural networks (RNN) to model nonlinear dynamical systems has been widely proved [34]. Therefore, it is fairly common to use RNN for several kinds of temporal information processing, as in prediction, control systems and temporal classification systems [35].

Next, we present a brief description of the problem of temporal classification and the solution applied in this research using two architectures of RNN to build the temporal classifier required for mental task-based BCI systems.

3.1 Temporal Classification

Temporal classification refers to the assignation of a class, based on features obtained in different time periods. Such features are represented as vectors forming a temporal sequence of components. Temporal classification is a difficult task because, in order to obtain the correct class, it is mandatory to consider not only the values of the features but also the order in which they appear in a specific time period. The definition of the size of time that must be considered in order to get the right classification is also a challenge. Fig. 4 illustrates a simple temporal classification problem. Suppose that we want to identify if the sequence {1,2} is sensed in input A when the sequence {2,1} is sensed in input B. If so, the expected classification outcome is "yes", otherwise it is "No". The table therein Fig. 4 illustrates the desired outputs of such classifier in the first 10 time periods.



time	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*
Input A	1	1	2	2	1	2	1	1	2	1
Input B	1	2	1	2	2	2	1	2	1	2
Output	No	No	Yes	No	No	No	No	No	Yes	No

Fig. 4 A simple temporal classification problem

In this example, the classifier must be able to "remember" the last two inputs, in order to identify the sequences correctly. Looking this table, it is fairly easy to figure out that the sequences defining the involved classes have a size of two.

However, this is not the case for more complicated problems as the one presented in this research, in which a human mental state has to be identified by a sequence of features occurring in a EEG. For such cases the classifier would have to automatically model a dynamics memorizing the feature sequences using the right size of past events. In other words, time has to be implicitly represented in the model. In this research a temporal classifier is used as the last component of the system classifying mental tasks (see Fig. 6). The classifier has to find out if the involved mental task occurs or it does not, that is, it works as a binary classifier.

3.2 Adaptive Temporal Classifiers

The building of a classifier able to label sequences requires several steps. The most important decisions to resolve during its design are: the definition of the structure of a feature vector representing the information of the sequence, the mathematical model used for the classifier and the training strategy used in such model. Section 4 describes how the structure of the feature vector for the classifier of mental tasks was built in this research. With respect to the mathematical model of the classifier, we chose to use RNN for two reasons: first, RNN are able to build internal representations involving time and second, most recurrent neural architectures are able to model chaos [36]. This last reason refers to the fact that the dynamics in an EEG is chaotic, according to several authors (for example see [37]).

Regarding to the selection of a right RNN and training algorithm, there are many choices when they are used for building temporal classifiers. The most versatile models are the ones proposed by Jordan [38], Elman [39], Werbos [40] and Williams and Zipser [41]. Other works have used more sophisticated structures, for example [42].

For the results presented here, we built and tested the performance of two classifiers using two types of recurrent neural networks: a Simple Recurrent Network (SRN), also known as “Elman network” [39] and a fully connected recurrent neural network (FCRNN) with external inputs, similar to the one described in [40,47]. SRN was trained using the algorithm “Back Propagation through time” (BPTT) [40] and FCRNN was trained with the algorithm “Real Time Recurrent Learning – Extended Kalman filter” (RTRL-EKF) [43,44] using the implementation proposed in [48]. These architectures and algorithms are briefly described next.

3.3 Simple Recurrent Network (SRN or Elman Network)

Time can be represented in several ways in recurrent neural networks. In a SRN, time is implicitly represented using a context layer. This model was introduced by Elman [39], which in spite of being rather simple, is able to memorize previous states of a sequence. SRN architecture has 4 layers: an input layer, a hidden layer, an output layer and a context layer. (see Fig. 5). The representation of past events is achieved because nodes in the context layer memorize the outputs of nodes in hidden layer coming from a previous time. This context layer is able to create a map of some temporal properties of the system.

In general, the state-space model of a RNN can be described by the following equations [44]:

$$\mathbf{x}_{n+1} = \mathbf{a}(\mathbf{x}_n, \mathbf{u}_n) \tag{1}$$

$$\mathbf{y}_n = \mathbf{B}\mathbf{x}_\mathbf{n} \tag{2}$$

where:

$\mathbf{y}_n$ represents the output of the system (all neurons in the network),

$\mathbf{u}_n = \{u_n, u_{n-1} \ldots u_{n-q+1}\}$ is a vector of the exogenous inputs in different steps,

$\mathbf{x}_\mathbf{n}$ is the output of a bank of q unit-time delays, q being the number of nodes in the input layer.

$\mathbf{a}(\cdot, \cdot)$ is a nonlinear function characterizing the hidden layer.

B is the matrix of synaptic weights characterizing the output layer.

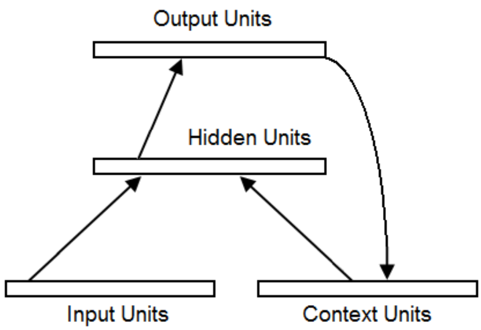


Fig. 5 The Simple Recurrent Network [39]

Notice that in this model, the hidden layer is non-linear (equation 1) and the output layer is linear (equation 2). A SRN is a special case of this model, where the connection weights and the output layer may also be non-linear. In the results reported here, a hyperbolic tangent sigmoid transfer function was used for the nodes in hidden layer ('tansig' Matlab function) and a logarithmic sigmoid transfer function ('logsig' Matlab function) was used for the output layers.

SRN may be trained in different ways. For the experiments reported in this chapter, we used a gradient descent back propagation algorithm with adaptive learning rate. Function 'calcgbtt', provided by the neural network toolbox of Matlab V6.0. was used as the gradient function, which calculates the bias and weight performance gradients using the back-propagation through time algorithm (BPTT) [40]. BPTT is a supervised learning algorithm originally proposed by Werbos [46] and independently discover by Rumelhart and collaborators [47], that attempts to minimize the output error of the network obtained over a period of time. This error is calculated as:

$$\mathbf{E} = \sum_{t=1}^{T} (\mathbf{D}_{t,\mathbf{n}} - \mathbf{y}_{t,\mathbf{n}})^2 \tag{3}$$

where $\mathbf{D}_{t,\mathbf{n}}$ is the desired output of the neurons in the network where an output is required at time t, and T is the size of the sequence being used to train the network. The core of back-propagation is a efficient method for the calculation of derivatives that allow to minimize the error described in equation 3. BPTT constructs a feed-forward network with identical behavior over a particular time interval that the involved RNN. The main drawback of BPTT is that it requires to use the complete training sequence for each training epoch in order to calculate the gradient. For a detailed explanation of BPTT see [44].

3.4 Fully-Connected Recurrent Neural Network

A fully-connected recurrent neural network with one input layer, one hidden layer and one output layer was also used in this work to build a temporal classifier. The term "fully connected" means that all neurons in the network are connected each other. The input layer is formed by neurons receiving an external input; the output layer is formed by nodes whose outputs are considered the output of the system; the training sequence contains the desired values for such outputs (corresponding class). As occurring with other layered neural network architectures, the number of neurons in the hidden layer depends upon the complexity of the problem and the appropriate number of them requires to be defined by experimentation.

As we explained before, there are several algorithms to train recurrent neural networks. BPTT has been very popular during many years, but currently it is known that very useful algorithms for training recurrent neural networks are based on Kalman Filtering (KF) [48]. KF is a common method to estimate unknown variables of a system based on the observations of measurements across time. KF is based on the idea that the involved dynamical system of the problem is hidden and can only be observed or measured through some time series (sequences). In KF the dependency among two consecutive states, measurements and the state process is assumed linear [49]. Therefore, an Extended Kalman Filtering (EKF) is required when nonlinear systems are involved, as in the case of recurrent neural networks. In EKF, a linearization around the current working point is applied before that standard KF is performed. EKF has been widely studied and applied using different strategies to train RNN, for example in [49-51]. It also has been combined with other algorithms, for example with “back-truncated propagation through time” [51] and with RTRL [43].

For the experiments presented in this research, we used a combination of the RTRL and KF proposed by [43]. The training algorithm RTRL contains two main steps (see [44,52]): gradient calculation and weights adjustments. RTRL is used to calculate the derivatives of the gradients and EKF is used for modifying the weights. According to [44], the state-space model of this network, when training, is defined by two models:

1) The system model, described by:

$$\mathbf{w}_{n+1} = \mathbf{w}_n + \omega_n \quad , \tag{4}$$

where:

$\mathbf{w}_n$ is the weight (state) vector

ω_n is a white Gaussian noise.

2) The measurement model, described by:

$$\mathbf{d}_n = \mathbf{b}(\mathbf{w}_n, \mathbf{v}_n, \mathbf{u}_n) + \mathbf{v}_n \quad , \tag{5}$$

where:

$\mathbf{d}_n$ is the desirable response o the system, playing the role of the “observable”,

$\mathbf{v}_n$ represents the recurrent node activities inside the network,

$\mathbf{u}_n$ denotes the input signal to the network and

$\mathbf{v}_n$ is a vector denoting measurement noise corrupting $\mathbf{d}_n$.

EKF allows the estimation of the value of the correction in the state space model, updating weights as follows:

$$\hat{\mathbf{w}}_n = \hat{\mathbf{w}}_{n-1} + \mathbf{G}_n \boldsymbol{\alpha}_n \tag{6}$$

$$\boldsymbol{\alpha}_n = \mathbf{d}_n - \mathbf{b}(\hat{\mathbf{w}}_{n-1}, \mathbf{v}_n, \mathbf{u}_n) \quad , \tag{7}$$

where:

$\mathbf{G}_n$ is the Kalman gain, calculated using:

$$\mathbf{G}_n = \mathbf{P}_{n-1}\mathbf{B}_n^T [\mathbf{B}_n \mathbf{P}_{n-1} \mathbf{B}_n^T + \mathbf{Q}_{\nu,n}]^{-1} \tag{7}$$

$$\mathbf{P}_n = \mathbf{P}_{n-1} - \mathbf{G}_n \mathbf{B}_n \mathbf{P}_{n-1} + \mathbf{Q}_{\omega,n} \tag{8}$$

$\mathbf{B}_n$ is the Jacobian of the partial derivatives with respect to the state, that is, the weights, which is calculated using RTRL algorithm.

$\mathbf{Q}_{\omega,n}$ is the covariance matrix of the dynamic noise ω_n,

$\mathbf{P}_n$ is the prediction error covariance matrix, and

$\mathbf{Q}_{\nu,n}$ is the covariance matrix of the measurement noise $\mathbf{v}_n$.

The calculation of partial derivatives $\mathbf{B}_n$ is defined as:

$$\mathbf{B}_n = \begin{bmatrix} \frac{\partial y_1}{\partial w_1} & \frac{\partial y_1}{\partial w_2} & \cdots \frac{\partial y_1}{\partial w_m} \\ \frac{\partial y_2}{\partial w_1} & \frac{\partial y_2}{\partial w_2} & \cdots \frac{\partial y_2}{\partial w_m} \\ \cdot & & \\ \frac{\partial y_p}{\partial w_1} & \frac{\partial y_p}{\partial w_2} & \cdots \frac{\partial y_q}{\partial w_m} \end{bmatrix} , \tag{9}$$

where q is the total number of neurons in the networks and m is the total number of weights. Using RTRL, derivatives in $\mathbf{B}_n$ are calculated as [53]:

$$\frac{\partial y_i(n+1)}{\partial w_{kl}} = \sigma'(x_i(n)) \left[\sum_{j=1}^{m} w_{ij} \frac{\partial y_j(n)}{\partial w_{kl}} + \delta_{ik} z_l(n) \right] \tag{10}$$

$\sigma'(\cdot)$ is the derivative of the neuron transfer function $\sigma(\cdot)$; $x_i(n) = \sum_{j=1}^{m} w_{ij} z_j(n)$ is the input to each neuron, δ_{ik} is the Kronocker delta. For further details, see [44,52,53].

For the experiments showed here we used an implementation of RTRL-EKF created by [52], which is itself based on the Matlab functions created by [45]. A very good description of the data structures used in such software is given by [48]. In that reference, the interested lector can find a very good algorithm to implement RTRL-EKF using GPA architecture.

4 Proposed Methodology

A block diagram of the proposed scheme is represented in Fig. 6. The algorithm is described as follows: preprocessing of the EEG signals obtained from P4 electrode includes a blind source separation through Independent Component Analysis (ICA) in order to remove eye blink and other artifacts. The signal is then filtered in order to obtain the *alpha* and *beta* bands, and the power signal for each band is computed. The power signal in each band is partitioned into 5 windows with a 50 % overlapping as a feature reduction process. The signal is passed through an

ANFIS system in order to obtain a representation in fuzzy sets corresponding to the evolution in time of the estimated power across both spectral bands alpha and beta. Temporal sequences corresponding to the combination of energy bands for each mental task are input into a recurrent neural network, which is trained to deliver a classification decision on the corresponding mental task.

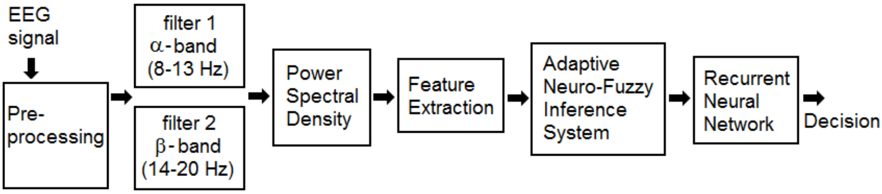


Fig. 6 Block diagram of the proposed architecture for mental tasks classification

Preprocessing EEG data in order to eliminate the artifacts added during the recording sessions is an essential task to facilitate accurate classification. The most corruptive of the artifacts is due to eye blinks because it produces a high amplitude signal called electrooculogram (EOG) that can be many times greater than the EEG signals of interest.

The use of ICA for blind source separation of EEG data is based on an assumption that EEG data recorded from multiple scalp sensors are linear sums of temporally independent components arising from spatially fixed, distinct or overlapping brain networks [54]. The goal of ICA is to recover statistically independent sources given only sensor observations that are unknown linear mixtures of the unobserved independent source signals. ICA reduces the statistical dependencies of the signals, attempting to make the signals as independent as possible which make ICA capable of separating artifact components from EEG data since they are usually independent of each other [55].

As mentioned before, $x_i(t)$ are assumed to be the result of linear combinations of the independent sources, as expressed in:

$$x_i(t) = a_{i1}s_i(t) + a_{i2}s_2(t) + \cdots + a_{in}s_n(t)$$

Or in matrix form:

$$\mathbf{x} = \mathbf{A}\mathbf{s}$$

where:

A is a matrix containing mixing parameters and
s is the source signals.

The goal of ICA is to calculate the original source signals from the mixture by estimating a de-mixing matrix U that gives:

$$\hat{\mathbf{s}} = \mathbf{U}\mathbf{x}$$

Both the mixing matrix **A** and the matrix containing the sources **S** are unknown. The non mixing matrix **U** is found by optimizing a cost function. Several different cost functions can be used for performing ICA, e.g. kurtosis, negentropy, etc., therefore, different methods exist to estimate U. For that purpose the source signals are assumed to be non-Gaussian and statistically independent. The requirement of non-Gaussianity stems from the fact that ICA relies on higher order statistics to separate the variables, and higher order statistics of Gaussian signals are zero. In this way, ICA is applied to EEG signal from P4 electrode in order to remove eye blink artifacts. For additional information see [54]. The result of preprocessing EEG data is shown in Fig. 7.

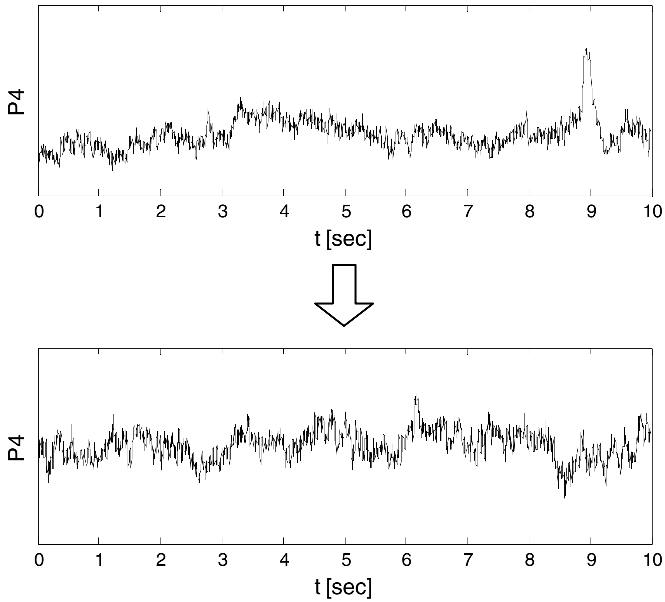


Fig. 7 EEG data before and after preprocessing

Elliptic filters of five order were used in order to obtain the *alpha* and *beta* bands. After filtering EEG data, the power for each band is computed squaring the amplitude of samples; then, the power signal in each band is partitioned into 5 windows with a 50 % overlapping as a feature reduction process. The signal is passed through an ANFIS system in order to obtain a representation in fuzzy sets corresponding to the evolution in time of the estimated power across both spectral bands alpha and beta. Temporal sequences corresponding to the combination of energy bands for each mental task are input into a recurrent neural network, which is trained to deliver a classification decision on the corresponding mental task.

5 Experimental Results

EEG data were obtained previously by Keirn and Aunon [56] and are available on line for research purposes. Ten trials for each mental task resulted in a total of 20 patterns. Details of the procedure followed to detect the signals can be consulted in the cited reference. A brief description is as follows: an Electro-Cap elastic electrode cap was used to record data from positions C3, C4, P3, P4, O1, and O2 defined by the 10-20 system of electrode placement. In the original data set, there were seven subjects performing five different mental tasks and one subject performing two different mental tasks. Signals were recorded for ten seconds during the task at a sampling frequency of 250 Hertz, and each task was repeated five times per session. Subjects attended two sessions recorded on different weeks, resulting in a total of ten trials for each task. The two mental tasks are described as follows. In the task described as mental letter composing, the subjects were instructed to mentally compose a letter to a friend or relative without vocalizing. The second mental task described as visual counting, was constructed by asking the subjects to imagine numbers being written sequentially on a blackboard, with the previous number erased before the next number was written. Experiments were executed using MATLAB version 7.6 in a personal computer with a 2.0 GHz AMD Turion processor and 3GB RAM. Figure 8 shows an example of the normalized power signal corresponding to alpha and beta bands for each mental task.

According to the proposed procedure previously described, feature extraction is performed on the power signals by a window-averaging with a 50% window overlap. Fig. 9 shows an example of the feature vectors obtained through the

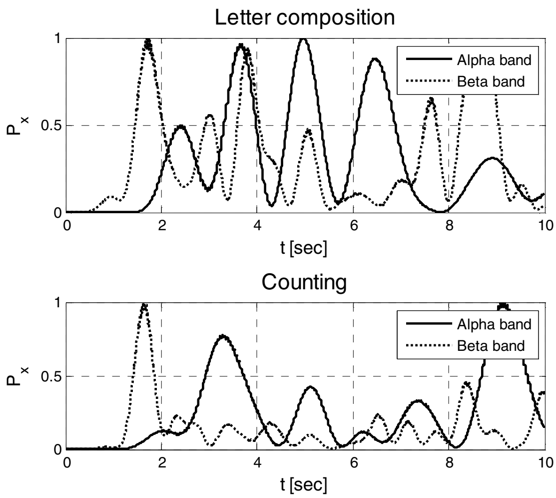


Fig. 8 Alpha and beta band power for letter composition and counting task

described procedure, corresponding to the referred mental tasks. As Fig. 9 illustrates, the power representation of alpha and beta bands presents variations associated to temporal evolution of power bands following each mental task. Since the power in bands shows variations for each subject and trial, we propose the use of an adaptive system allowing the assignment of membership functions in an automatic way in order to represent the configuration of bands through fuzzy sets, translating each experiment into a simple sequence that preserve the temporal evolution of the performed mental task. Fig. 10 shows an example of the state assignment corresponding to the case of letter composition task.

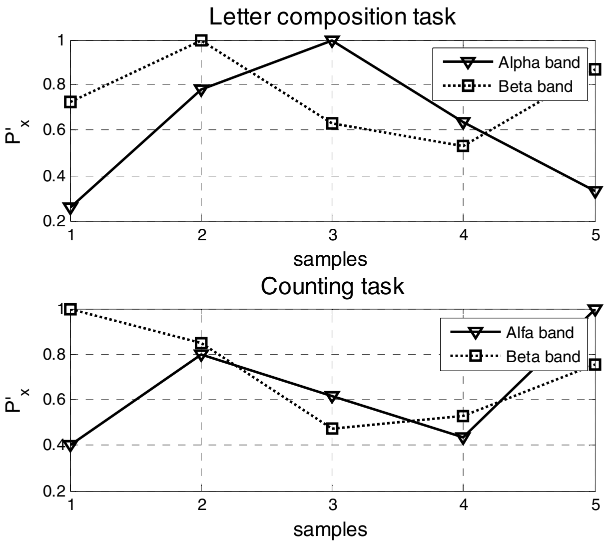


Fig. 9 Result of feature extraction process for two different mental tasks

The feature extraction process is then applied to each trial in the mental tasks database, obtaining some sequences representing the state transitions of power band configurations and corresponding to each mental task. The ANFIS system was trained with the features extracted over all trials, considering an input representation with eight membership functions. Fig. 11 shows an example of the results obtained from the ANFIS training for the two mental tasks. Temporal classification of the obtained feature vectors representing each mental task was performed using a recurrent neural network. In this paper we compare the performance of two models previously described: a simple recurrent neural network or Elman network and a Full Connected Recurrent Neural Network FCRNN. In both cases, the architecture of the recurrent neural networks was: 1 node in the input layer, 10 nodes in the hidden layer and 1 node in the output layer. The architecture was determined by experimentation, with the best results obtained using the described configuration.

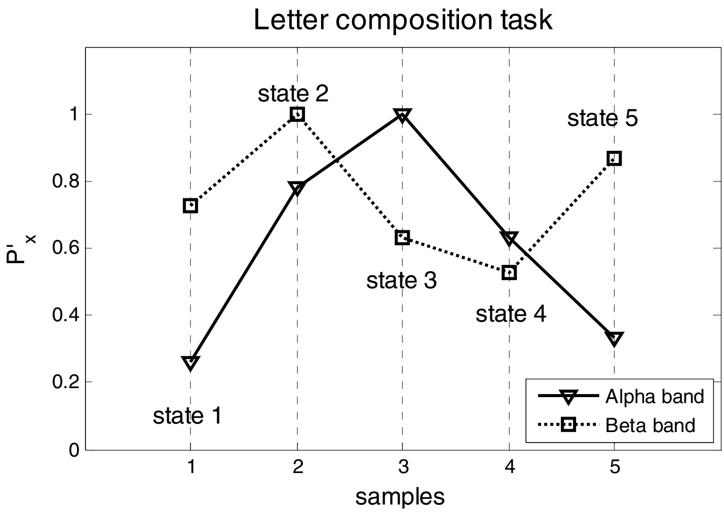


Fig. 10 State assignment for letter composition task

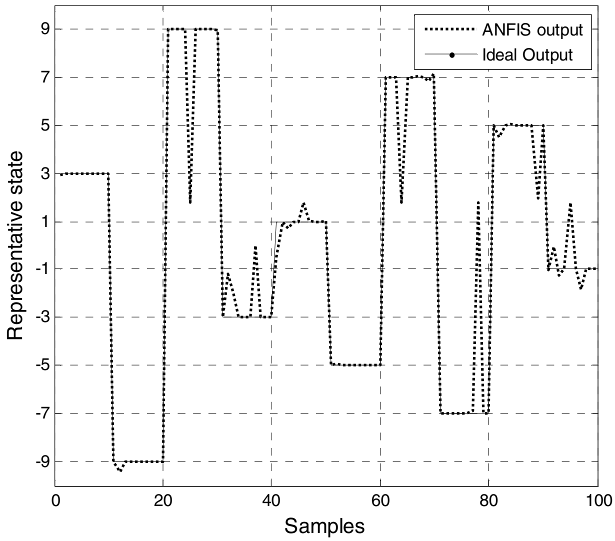


Fig. 11 Result of ANFIS training

Temporal classification results are reported based on a leave-one-out (LOO-CV) cross-validation. LOO-CV is typically used in the analysis of small datasets, where the relatively high variance of the estimator is offset by the stability resulting from the greater size of the training partition than is possible using conventional k-fold cross-validation [57].

Ten trials for each mental task result in a total of 20 patterns. The dataset was partitioned in 5 folds with 4 trials each one. LOO-CV was performed using four folds for training and the remaining one for testing. Table 1 summarizes the temporal classification results obtained in average from both, training and testing cases, with the two recurrent neural networks previously described.

Table 1 Results on temporal classification; training and testing

RNN	**Training 500 epochs**			**Testing**	
	MSE	**Performance**	**time**	**MSE**	**Performance**
Elman	0.0328	91.75%	3’49’’	0.0401	90.16%
FCRNN	0.0121	94.61%	1’ 12’’	0.0528	88.12%.

6 Conclusions

In this chapter, an architecture based on adaptive neuro-fuzzy inference systems (ANFIS) assembled to recurrent neural networks, applied to the problem of mental tasks temporal classification, has been presented. Information on power signal obtained from Alpha and Beta bands constituted a good descriptor with an adequate separability, providing a good balance between complexity and classification rate. The feature vectors representing each mental task following a fuzzy-set paradigm, provided a good description about the temporal evolution of the power signal. A classification rate in training of 94.61 % in average was obtained through the FCRNN, with an 88.12 % of classification using leave-on-out cross validation in the testing stage. A comparison with the Elman Network indicates a better performance of the FCRNN during the training stage, with a slightly better performance of the Elman network on generalization. In both cases, an architecture of the neural network with 10 nodes in the hidden layer provided the better results. Further experimentation oriented to the construction of a database for BCI applications is currently in progress.

Acknowledgments. The first author acknowledges the financial support from the Mexican National Council for Science and Technology (CONACYT), scholarship No. 224304. This research has been partially supported by CONACYT Grant No. CB-2010-155250.

References

[1] Brunner, P., Bianchi, L., Guger, C., Cincotti, F., Schalk, G.: Current trends in hardware and software for brain–computer interfaces (BCIs). Journal of Neural Engineering 8, 025001 (2011)

[2] Bashashati, M., Fatourechi, R., Ward, K., Birch, G.E.: A survey of signal processing algorithms in brain-computer interfaces based on electrical brain signals. Journal of Neural Engineering 4(2), R32–R57 (2007)
[3] Berger, T.W., Chapin, J.K., Gerhardt, G.A., McFarland, D.J., Principe, J.C., Soussou, W.V., Taylor, D.M., Tresco, P.A.: WTEC Panel Report on International Assessment of Research and Development in Brain-Computer Interfaces. World Technology Evaluation Center, Inc. (2007), `http://www.wtec.org/bci/BCI-finalreport-26Aug2008-lowres.pdf`
[4] Hosni, S.M., Gadallah, M.E., Bahgat, S.F., AbdelWahab, M.S.: Classification of EEG signals using different feature extraction techniques for mental-task BCI. In: 2007 International Conference on Computer Engineering Systems, pp. 220–226 (2007)
[5] Neuper, C., Scherer, R., Wriessnegger, S., Pfurtscheller, G.: Motor imagery and action observation: Modulation of sensorimotor brain rhythms during mental control of a brain–computer interface. Clinical Neurophysiology 120(2), 239–247 (2009)
[6] Solis-Escalante, T., Muller-Putz, G., Brunner, C., Kaiser, V., Pfurtscheller, G.: Analysis of sensorimotor rhythms for the implementation of a brain switch for healthy subjects. Biomedical Signal Processing and Control 5(1), 15–20 (2010)
[7] McFarland, D.J., Sarnacki, W.A., Townsend, G., Vaughan, T., Wolpaw, J.R.: The P-300-based brain–computer interface (BCI): Effects of stimulus rate. Clinical Neurophysiology 122(4), 731–737 (2011)
[8] Ramirez-Cortes, J.M., Alarcon-Aquino, V., Rosas-Cholula, G., Gomez-Gil, P., Escamilla-Ambrosio, J.: Anfis-Based P300 Rhythm Detection Using Wavelet Feature Extraction on Blind Source Separated EEG Signals. In: Ao, S., Amouzegar, M., Rieger, B.B. (eds.) Intelligent Automation and Systems Engineering, ch. 27. LNEE, vol. 103, pp. 353–365. Springer, New York (2011)
[9] Shyu, K.K., Lee, P.L., Liu, Y.J., Sie, J.J.: Dual-frequency steady-state visual evoked potential for brain computer interface. Neuroscience Letters 483(1), 28–31 (2010)
[10] Horki, P., Solis-Escalante, T., Neuper, C., Muller-Putz, G.R.: Hybrid Motor Imagery and Steady-state Visual Evoked Potential Based BCI for Artificial Arm Control. In: Proceedings of the First Tools for Brain Computer Interaction Workshop, Graz, Austria, p. 46 (2010)
[11] Wang, H., Li, C.S., Li, Y.G.: Brain-computer interface design based on slow cortical potentials using Matlab/Simulink. In: Proceedings of the International Conference on Mechatronics and Automation, Changchun, China, pp. 1044–1048 (2009)
[12] Khare, V., Santhosh, J., Anand, S., Bhatia, M.: Performance comparison of three artificial neural network methods for classification of electroencephalograph signals of five mental tasks. J. Biomedical Science and Engineering 3, 200–205 (2010)
[13] Pfurtscheller, G.: Spatiotemporal ERD/ERS patterns during voluntary movement and motor imagery. Supplements to Clinical Neurophysiology 53, 196–198 (2000)
[14] Chiappa, S., Bengio, S.: HMM and IOHMM modeling of EEG rhythms for asynchronous BCI systems. In: European Symposium on Artificial Neural Networks, ESANN (2004)
[15] Millan, J.R., Mouriño, J.: Asynchronous BCI and local neural classifiers: an overview of the adaptive brain interface project. IEEE Transactions on Neural Systems and Rehabilitation Engineering 11, 159–161 (2003)

[16] Pfurtscheller, G., Neuper, C., Schlogl, A., Lugger, K.: Separability of EEG signals recorded during right and left motor imagery using adaptive autoregressive parameters. IEEE Trans. Rehabil. Eng. 6, 316–325 (1998)

[17] Pfurtscheller, G., Neuper, C., Flotzinger, D., Pregenzer, M.: EEG-based discrimination between imagination of right and left hand movement. Electroenceph. Clin. Neurophysiology 103, 642–651 (1997)

[18] Wang, T., He, B.: An efficient rhythmic component expression and weighting synthesis strategy for classifying motor imagery EEG in a brain–computer interface. J. Neural Eng. 1, 1–7 (2004)

[19] Wang, T., Denga, J., He, B.: Classifying EEG-based motor imagery tasks by means of time-frequency synthesized spatial patterns. Clinical Neurophysiology 115, 2744–2753 (2004)

[20] Durka, P.: Matching Pursuit and Unification in EEG Analysis. Artech House, Inc., Norwood (2007)

[21] Wang, J., Xu, G., Wang, L., Zhang, H.: Feature extraction of brain-computer interface based on improved multivariate adaptive autoregressive models. In: Proceedings of the 3rd International Conference on Biomedical Engineering and Informatics (BMEI), Yantai, China, pp. 895–898 (2010)

[22] Kołodziej, M., Majkowski, A., Rak, R.J.: A New Method of EEG Classification for BCI with Feature Extraction Based on Higher Order Statistics of Wavelet Components and Selection with Genetic Algorithms. In: Dobnikar, A., Lotrič, U., Šter, B. (eds.) ICANNGA 2011, Part I. LNCS, vol. 6593, pp. 280–289. Springer, Heidelberg (2011)

[23] Vijean, V., Hariharan, M., Saidatul, A., Yaacob, S.: Mental tasks classifications using S-transform for BCI applications. In: Proceedings of the IEEE Conference on Sustainable Utilization and Development in Engineering and Technology, Semenyih, Malaysia, pp. 69–73 (2011)

[24] Lotte, F.: The use of fuzzy inference systems for classification in EEG-based brain-computer interfaces. In: Proceedings of the 3rd International Brain-Computer Interfaces Workshop and Training Course, Graz, Austria (2006)

[25] Zhang, L., He, W., He, C., Wang, P.: Improving mental task classification by adding high frequency band information. Journal of Medical Systems 34(1), 51–60 (2010)

[26] Palaniappan, R.: Utilizing Gamma band to improve mental task based brain-computer interface design. IEEE Transactions on Neural Systems and Rehabilitation Engineering 14(3), 299–303 (2006)

[27] Park, C., Looney, D., Kidmose, P., Ungstrup, M., Mandic, D.P.: Time-frequency analysis of EEG asymmetry using bivariate Empirical Mode Decomposition. IEEE Transactions on Neural Systems and Rehabilitation Engineering 19(4), 366–373 (2011)

[28] Kousarrizi, M.R.N., Ghanbari, A.A., Teshnehlab, M., Shorehdeli, M.A., Gharaviri, A.: Feature extraction and classification of EEG signals using Wavelet Transform, SVM and artificial neural networks for brain computer interfaces. In: Proceedings of the International Joint Conference on Bioinformatics, Systems Biology and Intelligent Computing, Shanghai, China, pp. 352–355 (2009)

[29] Forney, E.M., Anderson, C.W.: Classification of EEG during imagined mental tasks by forecasting with Elman recurrent neural networks. In: Proceedings of the International Joint Conference on Neural Networks, San Jose, California, USA, pp. 2749–2755 (2011)

[30] Coyle, D., McGinnity, T.M., Prasad, G.: Improving the separability of multiple EEG features for a BCI by neural-time-series-prediction-preprocessing. Biomedical Signal Processing and Control 5(3), 196–204 (2010)
[31] Chang, F., Chang, Y.: Adaptive neuro-fuzzy inference system for prediction of water level in reservoir. Advances in Water Resources 29(1), 1–10 (2006)
[32] Subasi, A.: Application of adaptive neuro-fuzzy inference system for epileptic seizure detection using wavelet feature extraction. Computers in Biology and Medicine 37(2), 227–244 (2007)
[33] Jang, J.S.R.: ANFIS: adaptive-network-based fuzzy inference system. IEEE Transactions on Systems Man and Cybernetics 23(3), 665–685 (1993)
[34] Mandic, D., Chambers, J.: Recurrent neural networks for prediction. John Wiley & Sons, Chinchester (2001)
[35] Fuchs, E., Gruber, C., Reitmaier, T., Sick, B.: Processing short-term and long-term information with a combination of polynomial approximation techniques and time-delay neural networks. IEEE Transactions on Neural Networks 20(9), 1450–1462 (2009)
[36] Gomez-Gil, P.: Long term prediction, chaos and artificial neural networks. Where is the meeting point? Engineering Letters 15(1), 1–5 (2007)
[37] Skarda, C., Freeman, W.: How brains make chaos in order to make sense of the world. Behavioral and Brain Sciences 10, 161–195 (1987)
[38] Jordan, M.: Serial order: a parallel distributed processing approach. Technical Report TR-8604. UC San Diego Institute for Cognitive Science, San Diego (1986)
[39] Elman, J.: Finding structure in time. Cognitive Science 14, 179–211 (1990)
[40] Werbos, P.: Backpropagation through time: what it does and how to do it. Proceedings IEEE 74(10), 1550–1560 (1990)
[41] Williams, R., Zipser, D.: A learning algorithm for continually running fully recurrent neural networks. Neural Computation 1, 270–280 (1989)
[42] Graves, A., Fernandez, S., Gomez, F., Schmidhuber, J.: Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In: Proceedings of the 23rd International Conference on Machine Learning, pp. 369–376. ACM, Pittsburgh (2006), doi:10.1145/1143844.1143891
[43] Williams, R.: Some observations on the use of the extended Kalman Filter as a recurrent network learning algorithm. Technical Report NU-CCS-92-1, Northeastern University, Boston, MA (1992)
[44] Haykin, S.: Neural Networks, 2nd edn. Prentice Hall, Upper Saddle River (1999)
[45] Cernansky, M.: Matlab functions for training recurrent neural networks RTRL-EKF (2008), `http://www2.fiit.stuba.sk/~cernans/main/download.html` (accessed January 2009)
[46] Werbos, P.: Beyond regression: new tools for prediction and analysis of the behavioral sciences. PhD Thesis, Cambridge, MA (1974)
[47] Rumelhart, D., Hinton, E., Williams, R.: Learning internal representations by error propagation. In: Parallel Distributed Processing: Explorations in the Microstructure of Cognition, vol. I. Bradford Books, Cambridge (1986)
[48] Čerňanský, M.: Training Recurrent Neural Network Using Multistream Extended Kalman Filter on Multicore Processor and Cuda Enabled Graphic Processor Unit. In: Alippi, C., Polycarpou, M., Panayiotou, C., Ellinas, G. (eds.) ICANN 2009, Part I. LNCS, vol. 5768, pp. 381–390. Springer, Heidelberg (2009)

[49] Ralaivola, L., d'Alché-Buc, F.: Nonlinear Time Series Filtering, Smoothing and Learning using the Kernel Kalman Filter. Technical Report, Universite Pierre et Marie Curie, Paris France (2004)
[50] Alanis, A., Sanchez, E., Loukianov, A.: Discrete-time adaptive backstepping nonlinear control via high-order neural networks. IEEE Transactions on Neural Networks 18(4), 1185–1195 (2007)
[51] Prokhorov, D.: Toyota prius hev neurocontrol and diagnostics. Neural Networks 21, 458–465 (2008)
[52] García-Pedrero, A.: Arquitectura neuronal apoyada en señales reconstruidas con wavelets para predicción de series de tiempo caóticas, M. Sc. Thesis, INAOE, Tonantzintla, Puebla (2009) (in spanish)
[53] Doka, K.: Handbook of brain theory and neural networks, 2nd edn. MIT Press, Cambridge (2002)
[54] Kachenoura, A., Albera, L., Senhadji, L., Comon, P.: ICA: A Potential Tool for BCI Systems. IEEE Signal Processing Magazine, 57–68 (January 2008)
[55] Keralapura, M., Pourfathi, M., Sirkeci-Mergen, B.: Impact of Contrast Functions in Fast-ICA on Twin ECG Separation. IAENG International Journal of Computer Science 38(1), 38–47 (2011)
[56] Keirn, Z.A., Aunon, J.I.: A new mode of communication between man and his surroundings. IEEE Trans. Biomed. Eng. 37(12), 1209–1214 (1990)
[57] Cawley, G.C.: Leave-One-Out Cross-Validation Based Model Selection Criteria for Weighted LS-SVMs. In: Proceedings of the International Joint Conference on Neural Networks, Vancouver, Canada, pp. 1661–1668 (2006)