International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:6, No:4, 2012

Trajectory Estimation and Control of Vehicle using Neuro-Fuzzy Technique

B. Selma, S. Chouraqui

***Abstract*—**Nonlinear system identification is becoming an important tool which can be used to improve control performance. This paper describes the application of adaptive neuro-fuzzy inference system (ANFIS) model for controlling a car. The vehicle must follow a predefined path by supervised learning. Back-propagation gradient descent method was performed to train the ANFIS system. The performance of the ANFIS model was evaluated in terms of training performance and classification accuracies and the results confirmed that the proposed ANFIS model has potential in controlling the non linear system.

***Keywords*—**Adaptive neuro-fuzzy inference system (ANFIS); Fuzzy logic; neural network; nonlinear system, control

I. Introduction

THE design of control systems is currently driven by a large number of requirements posed by increasing competition, environmental requirements, energy and material costs and the demand for robust, fault-tolerant systems. These considerations introduce extra needs for effective process modeling techniques. Many systems are not amenable to conventional modeling approaches due to the lack of precise, formal knowledge about the system, due to strongly nonlinear behavior, high degree of uncertainty, or time-varying characteristics.

Neuro-fuzzy modeling has been recognized as a powerful tool which can facilitate the effective development of models by combining information from different sources, such as empirical models, heuristics and data. Neuro-fuzzy models describe systems by means of fuzzy if–then rules represented in a network structure, to which learning algorithms known from the area of artificial neural networks can be applied.

Thanks to this structure, neuro-fuzzy models are to a certain degree transparent to interpretation and analysis, i.e. they can be better used to explain solutions to users than completely black-box models such as neural networks.

Artificial neural networks (ANNs) have been used as computational tools for pattern classification including diagnosis of diseases because of the belief that they have greater predictive power than signal analysis techniques [1].

B.Selma is with University of Science and Technology "Mohamed Boudiaf" USTO, Department of Computer Science. Faculty of Science. Oran. 31000, Algeria (phone: +213-77665-3511; e-mail: selma.boumediene@yahoo.fr).

S.Chouraqui is with University of Science and Technology "Mohamed Boudiaf" USTO, Department of Computer Science. Faculty of Science. Oran. 31000, Algeria (e-mail: s_chouraqui@yahoo.fr).

However, fuzzy set theory plays an important role in dealing with uncertainty when making decisions in medical applications. Therefore, fuzzy sets have attracted the growing attention and interest in modern information technology, production technique, decision making, pattern recognition, diagnostics, data analysis, etc. [5] - [6] and [7]. Neuro-fuzzy systems are fuzzy systems, which use ANNs theory in order to determine their properties (fuzzy sets and fuzzy rules) by processing data samples. Neuro-fuzzy systems harness the power of the two paradigms: fuzzy logic and ANNs, by utilizing the mathematical properties of ANNs in tuning rule-based fuzzy systems that approximate the way humans process information. A specific approach in neuro-fuzzy development is the adaptive neuro-fuzzy inference system (ANFIS), which has shown significant results in modeling nonlinear functions. In ANFIS, the membership function parameters are extracted from a data set that describes the system behavior. The ANFIS learns features in the data set and adjusts the system parameters according to a given error criterion [12]- [17]. Successful implementations of ANFIS in biomedical engineering have been reported, for classification [13]- [8]- [3]- [10]- [11] and data analysis [14].

In this study, a new approach based on ANFIS was presented for control of the car to follow his path. The proposed technique involved training the two ANFIS model to get the two classes of the car positions. Indeed, this problem is until now the subject of several studies. Also, several methods have to be developed and used.

The ANFIS model was trained with the backpropagation gradient descent method. We used the data described in reference [15], which is publicly available. after learning of ANFIS in several iterations there is an error between the calculated position and the position of the reference which is minimal.

The correct training rates and convergence rates of the proposed ANFIS model were examined and then performance of the ANFIS model was reported in the results obtained.

The simulation results showed that the neuro-fuzzy technique presented good results and that this controller is very robust and efficient.

II. Fuzzy Systems and Neural Networks

Both neural networks and fuzzy systems are motivated by imitating human reasoning processes. In fuzzy systems, relationships are represented explicitly in the form of if–then rules. In neural networks, the relations are not explicitly given, but are ‘coded’ in the network and its parameters. In contrast

to knowledge-based techniques, no explicit knowledge is needed for the application of neural networks.

Neuro-fuzzy systems combine the semantic transparency of rule-based fuzzy systems with the learning capability of neural networks. This section gives the background on nonlinear input–output modeling, fuzzy systems and neural networks, which is essential for understanding the rest of this article.

A. Fuzzy Models

A mathematical model which in some way uses fuzzy sets is called a fuzzy model. In system identification, rule-based fuzzy models are usually applied. In these models, the relationships between variables are represented by means of if– then rules with imprecise (ambiguous) predicates, such as: If heating is high then temperature increase is fast.

This rule defines in a rather qualitative way the relationship between the heating and the temperature in a room, for instance. To make such a model operational, the meaning of the terms 'high' and 'fast' must be defined more precisely.

This is done by using fuzzy sets, i.e. sets where the membership is changing gradually rather than in an abrupt way.

Fuzzy sets are defined through their membership functions (denoted by μ) which map the elements of the considered universe to the unit interval $[0,1]$. The extreme values 0 and 1 denote complete membership and non-membership, respectively, while a degree between 0 and 1 means partial membership in the fuzzy set. Depending on the structure of the if–then rules, two main types of fuzzy models can be distinguished: the Mamdani (or linguistic) model and the Takagi–Sugeno (which is used in this work).

1. Takagi–Sugeno model

The Mamdani model is typically used in knowledge-based (expert) systems. In data-driven identification, the model due to Takagi and Sugeno has become popular. In this model, the antecedent is defined in the same way as above, while the consequent is an affine linear function of the input variables:

$$R_i: \text{if } x \text{ is } A_i \text{ then } y_i = a_i^T x + b_i \qquad (1)$$

Where a_i is the consequent parameter vector, b_i is a scalar offset and $i=1,\ldots,K.$ This model combines a linguistic description with standard functional regression: the antecedents describe fuzzy regions in the input space in which the consequent functions are valid. The output y is computed by taking the weighted average of the individual rules' contributions:

$$y = \frac{\sum_{i=1}^{k} B_i(x) y_i}{\sum_{i=1}^{k} B_i(x)} = \frac{\sum_{i=1}^{k} B_i(x)(a_i^T x + b_i)}{\sum_{i=1}^{k} B_i(x)} \qquad (2)$$

where $B_i(x)$ is the degree of fulfillment of the i th rule.

For the rule (1), $B_i(x) = \mu_{A_i}(x)$ but it can also be a more complicated expression, as shown later on. The antecedent fuzzy sets are usually defined to describe distinct, partly overlapping regions in the input space. The parameters are then (approximate) local linear models of the considered nonlinear system.

Note that the antecedent and consequent variables may be different.

B. Artificial neural networks

Artificial neural networks (ANNs), originally inspired by the functionality of biological neural networks can learn complex functional relations by generalizing from a limited amount of training data. Neural networks can thus serve as black-box models of nonlinear, multivariable static and dynamic systems and can be trained by using input–output data observed on the system. The most common ANNs consist of several layers of simple processing elements called neurons, interconnections among them and weights assigned to these interconnections. The information relevant to the input–output mapping of the net is stored in the weights.

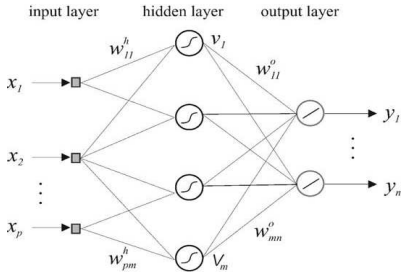

Fig. 1 A multi-layer neural network (MNN) with one hidden layer

1. Multi-layer neural network

A feedforward multi-layer neural network (MNN) has one input layer, one output layer and an number of hidden layers between them. For illustration purposes, consider a MNN with one hidden layer (Fig. 1).

The input-layer neurons do not perform any computations; they merely distribute the inputs x_i to the weights w_{ij}^h of the hidden layer. In the neurons of the hidden layer, first the weighted sum of the inputs is computed:

$$z_j = \sum_{i=1}^{p} w_{ij}^h x_i = (w_j^h)^T x, \qquad j = 1, m. \qquad (3)$$

It is then passed through a nonlinear activation function, such as the sigmoid:

$$V_j = \frac{e^{Z_j}}{e^{Z_j}+1} = \frac{1}{1+e^{Z_j}}, \qquad j = 1, m \qquad (4)$$

Other typical activation functions are the threshold function (hard limiter) and the sigmoidal function. The neurons in the

output layer are linear, i.e. only compute the weighted sum of their inputs:

$$y_l = \sum_{j=1}^{h} w_{jl}^{o} v_j = (w_j^o)^T x, \qquad l = 1, n. \tag{5}$$

Training is the adaptation of weights in a multi-layer network such that the error between the desired output and the network output is minimized.

Two steps are distinguished in this procedure:

- Feedforward computation. From the network inputs xi, the outputs of the first hidden layer are first computed.

Then using these values as inputs to the second hidden layer, the outputs of this layer are computed, etc. Finally, the output of the network is obtained.

- Weight adaptation. The output of the network is compared to the desired output. The difference of these two values, the error, is then used to adjust the weights first in the output layer, then in the layer before, etc., in order to decrease the error (gradient-descent optimization). This backward computation is called error backpropagation [16].

A network with one hidden layer is sufficient for most approximation tasks. More layers can give a better fit, but the training takes longer. Choosing the right number of neurons in the hidden layer is essential for a good result. Too few neurons give a poor fit, while too many neurons result in over-training of the net (poor generalization to unseen data).

A compromise is usually sought by trial and error methods.

2. Error back propagation

Consider for simplicity a MNN with one output. A set of N input–output data pairs $\{(x_k, y_k^*) \setminus k = 1, 2, \ldots, n\}$ is available. We represent this set as a matrix $x \in \mathbf{R}^{N \times P}$, having the input vectors x_k in its rows, and a column vector $y^* \in \mathbf{R}^N$, containing the desired outputs $y^*{}_k$:

$$X = [X_1, \ldots, X_N]^T \qquad y^* = [y_1^*, \ldots, y_N^*]^T \tag{6}$$

The difference between the desired output y^* and the output of the network y is called the error. This error is used to adjust the weights in the net via the minimization of the following cost function:

$$j = \frac{1}{2} \sum_{k=1}^{N} e_k^2 \qquad \text{with} \qquad e_k = y_k^* - y_k \tag{7}$$

Note that the network's output y is nonlinear in the weights w (for notational convenience, all the weights are lumped in a single vector w). The training of a MNN is thus a nonlinear optimization problem to which various methods can be applied; it's used Error backpropagation (first-order gradient). First-order gradient methods are based on the following general update rule for the weights:

$$w(n+1) = w(n) - \alpha(n) \nabla J(w(n)) \tag{8}$$

where $w(n)$ is the weight vector at iteration $n, \alpha(n)$ is a (variable) learning rate (a parameter) and $\nabla J(w)$ is the Jacobian of the network:

$$\nabla J(w) = \left[\frac{\partial J(w)}{\partial w1} + \frac{\partial J(w)}{\partial w2}, \ldots, \frac{\partial J(w)}{\partial w3} \right]^T \tag{9}$$

The nonlinear optimization problem is thus solved by using the first term of its Taylor series expansion (the gradient).

III. ADAPTIVE NEURO-FUZZY INFERENCE SYSTEM (ANFIS)

A. Architecture of ANFIS

The ANFIS is a fuzzy Sugeno model put in the framework of adaptive systems to facilitate learning and adaptation [12]-[17]. Such framework makes the ANFIS modeling more systematic and less reliant on expert knowledge.

To present the ANFIS architecture, two fuzzy if-then rules based on a first order Sugeno model are considered:

Rule 1: if (x is A_1) and (y is B_1) then $(f_1 = p_1 x + q_1 y + r_1)$

Rule 2: If (x is A_2) and (y is B_2) then $(f_2 = p_2 x + q_2 y + r_2)$

where x and y are the inputs, A_i and B_i are the fuzzy sets, f_i are the outputs within the fuzzy region specified by the fuzzy rule, p_i, q_i and r_i ri are the design parameters that are determined during the training process. The ANFIS architecture to implement these two rules is shown in Fig. 2, in which a circle indicates a fixed node, whereas a square indicates an adaptive node.

In the first layer, all the nodes are adaptive nodes. The outputs of layer 1 are the fuzzy membership grade of the inputs, which are given by:

$$O_i^1 = \mu_{A_i}(x) \; i = 1, 2 \tag{10}$$

$$O_i^1 = \mu_{B_i}(x) \; i = 1, 2 \tag{11}$$

Where $\mu_{A_i}(x)$, $\mu_{B_i}(x)$ can adopt any fuzzy membership function. For example, if the bell shaped membership

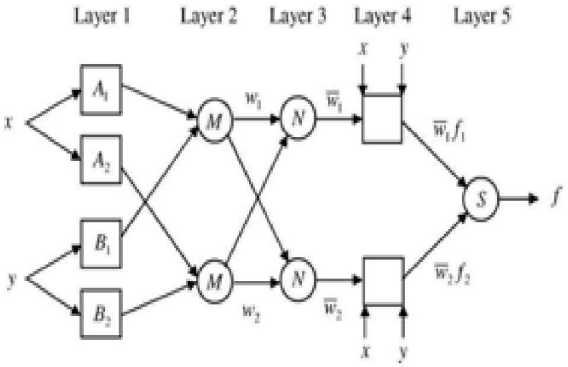

Fig. 2 ANFIS architecture

function is employed, $\mu_{A_i}(x)$ is given by:

$$\mu_{A_i}(x) = \frac{1}{1+\left\{\frac{x-c_i}{a_i}\right\}^{b_i}} \tag{12}$$

where a_i, b_i and c_i are the parameters of the membership function, governing the bell shaped functions accordingly. In the second layer, the nodes are fixed nodes. They are labeled with Fig. 2 (M), indicating that they perform as a simple multiplier.
The outputs of this layer can be represented as:

$$O_i^2 = w_i = \mu_{A_i}(x)\mu_{B_i}(y) \quad i = 1,2 \tag{13}$$

which are the so-called firing strengths of the rules.

In the third layer, the nodes are also fixed nodes. They are labeled with Fig. 2 (N), indicating that they play a normalization role to the firing strengths from the previous layer.
The outputs of this layer can be represented as:

$$O_i^3 = \overline{w}_i = \frac{w_i}{w_1 + w_2} \quad i = 1,2 \tag{14}$$

which are the so-called normalized firing strengths.

In the fourth layer, the nodes are adaptive nodes. The output of each node in this layer is simply the product of the normalized firing strength and a first order polynomial (for a first order Sugeno model). Thus, the outputs of this layer are given by:

$$O_i^4 = \overline{w}_i f_i = \overline{w}_i (p_i x + q_i y + r_i) \quad i = 1,2 \tag{15}$$

In the fifth layer, there is only one single fixed node labeled with Fig. 2 (S). This node performs the summation of all incoming signals. Hence, the overall output of the model is given by:

$$O_i^5 = \sum_{i=1}^{2} \overline{w}_i f_i = \frac{\sum_{i=1}^{2} w_i f_i}{w_1 + w_2} \tag{16}$$

It can be observed that there are two adaptive layers in this ANFIS architecture, namely the first layer and the fourth layer. In the first layer, there are three modifiable parameters $\{a_i, b_i, c_i\}$, which are related to the input membership functions. These parameters are the so-called premise parameters. In the fourth layer, there are also three modifiable parameters $\{p_i, q_i, r_i\}$, pertaining to the first order polynomial. These parameters are so-called consequent parameters [12]- [17].

B. Learning algorithm of ANFIS

The task of the learning algorithm for this architecture is to tune all the modifiable parameters, namely $\{a_i, b_i, c_i\}$ and $\{p_i, q_i, r_i\}$, to make the ANFIS output match the training data.

When the premise parameters a_i, b_i and c_i of the membership function are fixed, the output of the ANFIS model can be written as:

$$f = \frac{w_1}{w_1 + w_2} f_1 + \frac{w_2}{w_1 + w_2} f_2 \tag{17}$$

Substituting (11) into (14) yields:

$$f = \overline{w}_1 f_1 + \overline{w}_2 f_2 \tag{18}$$

Substituting the fuzzy if-then rules into (15), it becomes:

$$f = \overline{w}_1 (p_1 x + q_1 y + r_1) + \overline{w}_2 (p_2 x + q_2 y + r_2) \tag{19}$$

After rearrangement, the output can be expressed as:

$$\begin{aligned} f = {} & (\overline{w}_1 x) p_1 + (\overline{w}_1 y) q_1 + (\overline{w}_1) r_1 + (\overline{w}_2 x) p_2 + (\overline{w}_2 y) q_2 \\ & + (\overline{w}_2) r_2 \end{aligned} \tag{20}$$

which is a linear combination of the modifiable consequent parameters p_1, q_1, r_1, p_2, q_2 and r_2. The least squares method can be used to identify the optimal values of these parameters easily. When the premise parameters are not fixed, the search space becomes larger and the convergence of the training becomes slower. A hybrid algorithm combining the least squares method and the gradient descent method is adopted to solve this problem. The hybrid algorithm is composed of a forward pass and a backward pass. The least squares method (forward pass) is used to optimize the consequent parameters with the premise parameters fixed. Once the optimal consequent parameters are found, the backward pass starts

immediately. The gradient descent method (backward pass) is used to adjust optimally the premise parameters corresponding to the fuzzy sets in the input domain. The output of the ANFIS is calculated by employing the consequent parameters found in the forward pass. The output error is used to adapt the premise parameters by means of a standard backpropagation algorithm. It has been proven that this hybrid algorithm is highly efficient in training the ANFIS [12]- [17].

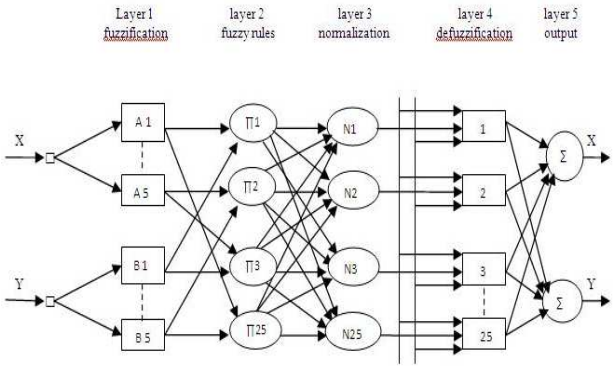

Fig. 3 Architecture of ANFIS proposed model

C. Controller learning "ANFIS network training"

Our controller learning is done by through the retro propagation algorithm to determine the parameters of premises (fit parameters related to membership functions) and parameter estimation by the consequential least squares method. Which has term "blended learning".

The ANFIS network training is made by an algorithm to two days when we first estimate parameters consistent with a technique least squares and then the weights of the network by a gradient descent. Each time the drive includes a phase front and rear phase. During the phase before the input patterns are used to determine the outputs of neurons layer by layer, for determining the values of parameters consistent account at the end. During Phase back, the algorithm error back propagation is applied to resolve weight of the various layers. During the backward pass, the back fitting algorithm error propagation is applied to update the weight of history rules. In the algorithm of ANFIS Jang, it also optimizes many parameters as those of history consistent. During the phase before, the parameters consequents are adapted so that the parameters history are held constant; during the back stage, the roles are exchanged. The result of learning process is shown in Fig. 3

IV. RESULTS AND DISCUSSION

In this section we examine the effects of this hybridization, illustrated by the following graphs which thus demonstrate the contribution of global research in improving learning outcomes classical neural networks. Once the neuro-fuzzy controller is designed and tested, it is applied to guide the car to follow her path and this controller allows automatic generation of fuzzy rule-based inference model of Sugeno. A numerical simulation is then performed. We took an example to show the results of our work, the application of the simulation is developed with NetBeans IDE 6.9.1 and it began by giving the reference path is shown in Fig. 4

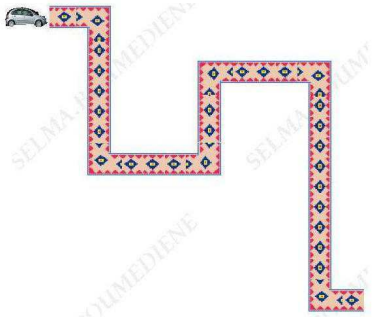

Fig. 4 Learning about this sample reference path

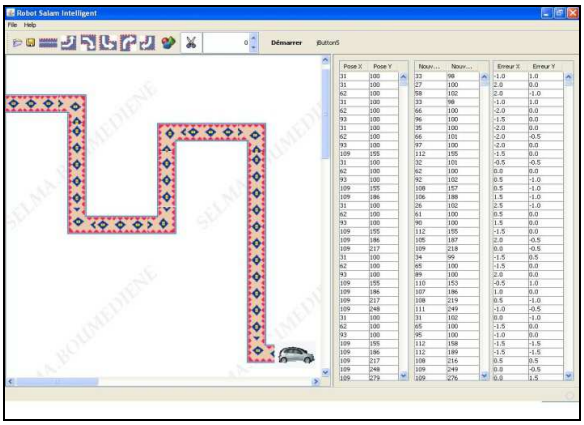

Fig. 5 Application interface after execution

We took the positions of path x, y integers, and since is plenty of points we took a few to make learning about them that are filled in the table I.

TABLE I

THE EXTRACTED (x, y) POSITIONS OF EXEMPLARY REFERENCE PATH

X	31	109	93	109	62	460	296
y	100	155	100	155	100	549	197

After applying the ANFIS controller, he obtained a new position followed by the car after rounding the new positions that are shown in Table II.

The results are obtained after 500 iterations.

TABLE II

THE VALUES OF THE NEW (x, y) POSITIONS

X	32,77	108,47	90,45	112,84	63,08	460,19	298,38
y	98,81	157,43	99.98	158,16	98,33	549,04	199,01

The Fig. 6 shows the error of our controller after learning by computing the new positions of the path and it was minimal compared to the results obtained.

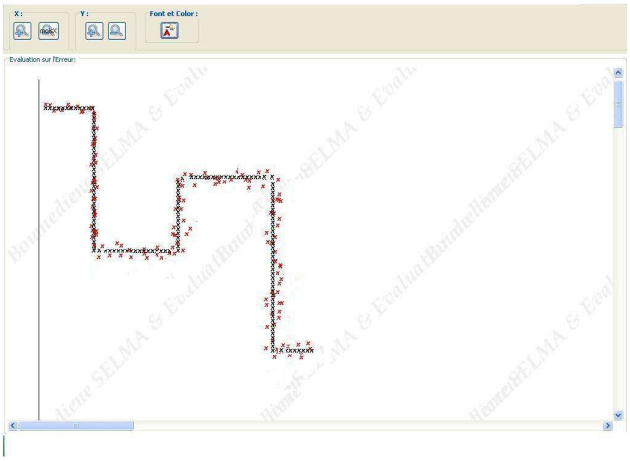

Fig. 6 Error between the reference positions and the calculated positions

The obtained simulation results showed that the neuro-fuzzy technique presented good results and that The ANFIS controller is very robust and efficient. The classification results and statistical measures were used for evaluating the ANFIS. The total classification accuracy of the ANFIS model was 98.24% which is number of correct decisions/total number of cases as it is shown in Table III.

TABLE III
THE VALUES OF STATISTICAL RESULTS

datasets	Statistical results	
	rate accuracy(%)	Total classification accuracy (%)
Set X	99.03	98.94
Set Y	98.86	

V. CONCLUSION

Fuzzy set theory plays an important role in dealing with uncertainty when making decisions in artificial intelligence applications. Using fuzzy logic enabled us to use the uncertainty in the classifier design and consequently to increase the credibility of the system output. This paper presented a new application of ANFIS for control of the nonlinear system. The presented ANFIS model combined the neural network adaptive capabilities and the fuzzy logic qualitative approach.; neuro-fuzzy system shows a good range of characterization and computational efficiency.Its robustness, speed and accuracy of its outputs it to avoid cases of indecision, neural networks with their ability to adapt to unfamiliar situations through learning, and fuzzy logic with its modeling capability of imprecise knowledge and uncertainty management.

The results obtained in our work encourage further research in this direction we can also consider improvements. This research work is by no means to condemn conventional methods. The approach presented is primarily enrich the family of methods for control of a nonlinear system.

REFERENCES

[1] Baxt WG. Use of an artificial neural network for data analysis in clinical decision making: the diagnosis of acute coronary occlusion. Neural Comput 1990;2:480–9.
[2] Miller AS, Blott BH, Hames TK. Review of neural network applications in medical imaging and signal processing. Med Biol Eng Comput 1992;30:449–64.
[3] Güler I˙, Übeyli ED. Detection of ophthalmic artery stenosis by leastmean squares backpropagation neural network. Comput Biol Med 2003;33(4):333–43.
[4] Übeyli ED, Güler I˙. Neural network analysis of internal carotid arterial Doppler signals: Predictions of stenosis and occlusion. Expert Syst Appl 2003;25(1):1–13.
[5] Dubois D, Prade H. An introduction to fuzzy systems. Clin Chim Acta 1998;270:3–29.
[6] Kuncheva LI, Steimann F. Fuzzy diagnosis. Artif Intell Med 1999;16:121–8.
[7] Nauck D, Kruse R. Obtaining interpretable fuzzy classification rules from medical data. Artif Intell Med 1999;16:149–69.
[8] Belal SY, Taktak AFG, Nevill AJ, Spencer SA, Roden D, Bevan S. Automaticdetection of distorted plethysmogram pulses in neonates and paediatric patients using an adaptive-network-based fuzzy inference system. Artif Intell Med 2002;24:149–65.
[9] Güler I˙, Übeyli ED. Application of adaptive neuro-fuzzy inference system for detection of electrocardiographic changes in patients with partial epilepsy using feature extraction. Expert Syst Appl 2004;27(3):323–30.
[10] Übeyli ED, Güler I˙. Automatic detection of erythemato-squamous diseases using adaptive neuro-fuzzy inference systems. Comput Biol Med 2005;35(5):421–33.
[11] Übeyli ED, Güler I˙. Adaptive neuro-fuzzy inference systems for analysis of internal carotid arterial Doppler signals. Comput Biol Med (2005), in press.
[12] Jang J-SR. Self-learning fuzzy controllers based on temporal backpropagation. IEEE Trans Neural Netw 1992;3(5):714–23.
[13] Usher J, Campbell D, Vohra J, Cameron J. A fuzzy logic-controlled classifier for use in implantable cardioverter defibrillators. Pace PacingClin Electrophysiol 1999;22:183–6.
[14] Virant-Klun I, Virant J. Fuzzy logic alternative for analysis in the biomedical sciences. Comput Biomed Res 1999;32:305–21.
[15] Andrzejak RG, Lehnertz K, Mormann F, Rieke C, David P, Elger CE. Indications of nonlinear deterministic and finite-dimensional structures in time seires of brain electrical activity: dependence on recording region and brain state. Phys Rev E 2001;64:061907.
[16] Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning internal representations by error propagation. In D. E. Rumelhart & J. L. McClelland(Eds.), Parallel distributed processing. Cambridge, MA: MIT Press.
[17] Jang J-SR. ANFIS: Adaptive-network-based fuzzy inference system.IEEE Trans Syst Man Cybern 1993;23(3):665–85.