A NEW ARTIFICIAL NEURAL NETWORK BASED FUZZY INFERENCE SYSTEM WITH MOVING CONSEQUENTS IN IF-THEN RULES

J. Łęski, E. Czogała
Technical University of Silesia
Akademicka 16, 44-101 Gliwice, Poland

Abstract

In this paper a new artificial neural network based fuzzy inference system (ANNBFIS) has been described. The novelty of the system consists in the moving fuzzy consequent in if-then rules. This system also automatically generates rules from numerical data. The application to medical pattern recognition is considered in this paper as well.

1. Introduction

In literature several methods of automatic fuzzy rule generation from given numerical data have been described [1,2,4,5,6]. The simplest method of rule generation is based on a clustering algorithm and estimation of the proper fuzzy relations from a set of numerical data. Kosko's fuzzy associative memory (FAM) [5] can store such fuzzy relations and process fuzzy inference simultaneously. This approach, however, causes some difficulties because of conflicts appearing among the generated rules.

Wang et al. [9] proposed a method for generating fuzzy rules from numerical data without conflicting rules. However, they used too many heuristic procedures and a trial-and-error choice of membership functions.

Another type of methods which use the learning capability of neural networks and the fact that both fuzzy systems and neural nets are universal approximators, has been successfully applied to various tasks. The problem here is the difficulty in understanding the identified fuzzy rules since they are implicitly acquired into the network itself.

Mitra et al. [6] have proposed a fuzzy multilayer perceptron generating fuzzy rules from the connection weights.

Several methods of extracting rules from the given data are based on a class of radial basis function networks (RBFNs). The fact that there is a functional equivalence between RBFNs and the fuzzy system has been used by Jang et al. [3] to construct Sugeno type of adaptive network based fuzzy inference system (ANFIS) which is trained by the back propagation algorithm. More general fuzzy reasoning schemes in ANFIS are employed by Horikawa et al. [2]. Such a developed radial basis function based adaptive fuzzy systems has been described by Cho and Wang [1] and applied to system identification and prediction.

The aim of this paper is the theoretical description and structure presentation of a new artificial neural network based fuzzy inference system ANNBFIS. The novelty of the system consists in the introduction of the moving fuzzy consequent in if-then rules. The described system is applied to pattern recognition problems

The paper is organized as follows. Some introductory remarks and the main goal of the paper are formulated in section 1. Section 2 introduces the basics of fuzzy systems. In section 3 the structure of ANNBFIS and the adaptation of the parameters are shown . Section 4. illustrates the theoretical considerations by means of application of the system to the pattern recognition problem. Finally, concluding remarks are gathered in section 5.

2. Fundamentals of fuzzy systems

In approximate reasoning realized in fuzzy systems the if-then fuzzy rules or fuzzy conditional statements play an essential and up to now the most important role. Often they are also used to capture the human ability of decision making and/or control in an uncertain and imprecise environment. In this section we will use such fuzzy rules to recall the important approximate reasoning methods which are basic in our further considerations.

Assume that m numbers of n-input and one-output (MISO) fuzzy implicative rules or fuzzy conditional statements are given. The i-th rule may be written in the following forms:

$$R^{(i)}:\ if\ X_1\ is\ A_1^{(i)}\ and ... and\ X_n\ is\ A_n^{(i)}\ then\ Y = f^{(i)}(X_1,..X_n) \qquad (1)$$

or in pseudo-vector notation

$$R^{(i)}:\ if\ X\ is\ A^{(i)}\ then\ Y = f^{(i)}(X) \qquad (2)$$

where:

$$X = [X_1\ X_2\ \dots\ X_n]^T \qquad (3)$$

$X_1,...,X_n$ and Y are linguistic variables which may be interpreted as inputs of fuzzy system ($X_1,...,X_n$) and the output of that system (Y). $A_1^{(i)},...,A_n^{(i)}$ are linguistic values of the linguistic variables $X_1,...X_n$ and $f^{(i)}$ is a function of the variables $X_1, ...,X_n$.

A collection of the above written rules for i=1,2,...m, creates a rule base which may be activated (fired) under the singleton inputs:

$$X_1 \text{ is } x_{10} \text{ and...and } X_n \text{ is } x_{n0} \tag{4}$$

or

$$X \text{ is } x_0 \tag{5}$$

It can easily be noticed that such a type of reasoning, where the inferred value of i-th rule output for crisp inputs (singletons) may be written in the form:

$$R_i(x_{10},...,x_{n0}) \Rightarrow f^{(i)}(x_{10},...,x_{n0}) = R_i(x_0) \Rightarrow f^{(i)}(x_0) \tag{6}$$

where: $\Rightarrow$ stands for fuzzy implication represented by e.g. minimum, product etc.,

$$R_i(x_{10},...,x_{n0}) = R_i(x_0) = A_1^{(i)}(x_{10}) \; and...and A_n^{(i)}(x_{n0}) = A^{(i)}(x_0) \tag{7}$$

denotes the degree of activation (the firing strength) of the i-th rule with respect to minimum ($\wedge$) or product ($\cdot$) representing explicite connective (AND) of the predicates X_l is $A_l^{(i)}$ (l=1,...,n) in the antecedent of a rule if-then.

A crisp value of the output for Larsen's fuzzy implication (product) and aggregation (sum) can be evaluated from the formula [1]:

$$y_0 = \frac{\sum_{i=1}^{m} R_i(x_{10},...,x_{n0}) \cdot f^{(i)}(x_{10},...,x_{n0})}{\sum_{i=1}^{m} R_i(x_{10},...,x_{n0})} = \frac{\sum_{i=1}^{m} R_i(x_0) \cdot f^{(i)}(x_0)}{\sum_{i=1}^{m} R_i(x_0)}$$

Taking into account that the function $f^{(i)}$ is of the form:

$$f^{(i)}(x_{10},\ldots,x_{n0}) = f^{(i)}(\mathbf{x_0}) = p_0^{(i)} \tag{9}$$

where $p_0^{(i)}$ is crisply defined constant in the consequent of the i-th rule.

Such a model is called a zero-order Sugeno fuzzy model.

The more general first-order Sugeno fuzzy model is of the form:

$$f^{(i)}(x_{10},\ldots,x_{n0}) = p_0^{(i)} + p_1^{(i)} x_{10} + \ldots p_n^{(i)} x_{n0} \tag{10}$$

where $p_0^{(i)}$, $p_1^{(i)}$,...,$p_n^{(i)}$ are all constants.
In vector notation it takes the form:

$$f^{(i)}(\mathbf{x_0}) = \mathbf{p}^{(i)T} \mathbf{x_0'} \tag{11}$$

where x_0' denotes an extended input vector

$$\mathbf{x_0'} = \begin{bmatrix} 1 \\ \mathbf{x_0} \end{bmatrix} \tag{12}$$

Notice that in both models the consequent is crisp. The above recalled method is called Takagi-Sugeno-Kang method of reasoning.

Now let us consider a more general form of MISO fuzzy rules, i.e. the rules in which the consequent is represented by a linguistic variable Y:

$$R^{(i)}\text{: if } X_1 \text{ is } A_1^{(i)} \text{ and}\ldots\text{and } X_n \text{ is } A_n^{(i)}\text{, then } Y \text{ is } B^{(i)} \tag{13}$$

Membership functions of fuzzy sets $B^{(i)}$ can be represented by the parameterized functions in the form:

$$B^{(i)} \sim f^{(i)}(Area(B^{(i)}), y^{(i)}) \tag{14}$$

where $y^{(i)}$ is the center of gravity (COG) location of the fuzzy set $B^{(i)}$:

$$y^{(i)} = COG(B^{(i)}) = \frac{\int y\, B^{(i)}(y)\, dy}{\int B^{(i)}(y)\, dy} \tag{15}$$

Next we will consider the constructive type of systems with Larsen's product operation as fuzzy implication of the fuzzy rule and sum as aggregation.
A general form of final output value can be put in the form:

$$y_0 = \frac{\sum_{i=1}^{m} y^{(i)} \cdot \mathrm{Area}(B^{(i)})}{\sum_{i=1}^{m} \mathrm{Area}(B^{(i)})} \tag{16}$$

For symmetric triangle (isosceles triangle) fuzzy values we can write a formula:

$$y_0 = \frac{\sum_{i=1}^{m} \frac{w^{(i)} R_i(x_0)}{2} y^{(i)}}{\sum_{i=1}^{m} \frac{w^{(i)} R_i(x_0)}{2}} \tag{17}$$

when $w^{(i)}$ is the width of the triangle base.

3. Moving consequent fuzzy set

In equation (17), the value describing the location of COG's consequent fuzzy set in if-then rules is constant and equals $y^{(i)}$ for i-th rule. A natural extension of the above described situation is an assumption that the location of the consequent fuzzy set is a linear combination of all inputs for i-th rule:

$$y^{(i)}(x_0) = p^{(i)T} x_0^{'} \tag{18}$$

Hence we get the final output value in the form:

$$y_0 = \frac{\sum_{i=1}^{m} \frac{w^{(i)} R_i(x_0)}{2} p^{(i)T} x_0^{'}}{\sum_{i=1}^{m} \frac{w^{(i)} R_i(x_0)}{2}} \tag{19}$$

Additionally, we assume that $A_1^{(i)},\dots,A_n^{(i)}$ have Gaussian membership functions:

$$A_j^{(i)}(x_{j0}) = \exp\left(- \frac{(x_{j0} - c_j^{(i)})^2}{2 s_j^{(i)^2}} \right) \tag{20}$$

where $c_j^{(i)}$, $s_j^{(i)}$; j=1,2,...,n ; i=1,2,...,m are the parameters of the membership functions.
On the basis of (7), and for explicit connective AND taken as product we get:

$$A^{(i)}(x_0) = \prod_{j=1}^{m} A_j^{(i)}(x_{j0}) \tag{21}$$

Hence, on the basis of (20) we have:

$$R_i(x_0) = \exp\left(- \sum_{j=1}^{m} \frac{(x_{j0} - c_j^{(i)})^2}{2 s_j^{(i)^2}} \right) \tag{22}$$

For n inputs and m rules if-then we have to establish the following unknown parameters:
- $c_j^{(i)}$, $s_j^{(i)}$; j=1,2,...,n; i=1,2,...,m, the parameters of membership functions of input sets,
- $p_j^{(i)}$; j=0,1,...,n; i=1,2,...,m; the parameters determining the location of output sets,
- $w^{(i)}$; i=1,2,...,m; the parameters of output sets.

Obviously, the number of rules if-then is unknown. Equations (19), (22) describe a radial neural network. The unknown parameters (except the number of rules m) are estimated by means of gradient method performing the steepest descent on a surface in the parameter space. Therefore the so called learning set is necessary, i.e. a set of inputs for which the output values are known. This is the set of pair $(x_0(k), y_0(k))$; k=1,2,...,N. The measure of the error of output value may be defined for a single pair from training set:

$$E = \frac{1}{2}(t_0 - y_0)^2 \tag{23}$$

where t_0 - the desired (target) value of output.

The minimization of error E is made iteratively (for parameter α):

$$(\alpha)_{new} = (\alpha)_{old} - \eta \left.\frac{\partial E}{\partial \alpha}\right|_{\alpha = (\alpha)_{old}} \tag{24}$$

where η - learning rate.

Now we derive the negative partial derivatives of error E according to the unknown parameters:

$$-\frac{\partial E}{\partial c_j^{(i)}} = (t_0 - y_0)\frac{y^{(i)}(x_0) - y_0}{\sum_{i=1}^{m} \frac{w^{(i)}}{2} R_i(x_0)} \frac{w^{(i)}}{2} R_i(x_0) \frac{x_{j0} - c_j^{(i)}}{s_j^{(i)2}} \tag{25}$$

$$-\frac{\partial E}{\partial s_j^{(i)}} = (t_0 - y_0)\frac{y^{(i)}(x_0) - y_0}{\sum_{i=1}^{m} \frac{w^{(i)}}{2} R_i(x_0)} \frac{w^{(i)}}{2} R_i(x_0) \frac{(x_{j0} - c_j^{(i)})^2}{s_j^{(i)3}} \tag{26}$$

$$-\frac{\partial E}{\partial p_j^{(i)}} = (t_0 - y_0)\frac{\frac{w^{(i)}}{2} R_i(x_0)}{\sum_{i=1}^{m} \frac{w^{(i)}}{2} R_i(x_0)} x_{j0} \tag{27}$$

$$-\frac{\partial E}{\partial p_0^{(i)}} = (t_0 - y_0)\frac{\frac{w^{(i)}}{2} R_i(x_0)}{\sum_{i=1}^{m} \frac{w^{(i)}}{2} R_i(x_0)} \tag{28}$$

$$-\frac{\partial E}{\partial w^{(i)}} = (t_0 - y_0)\frac{y^{(i)}(x_0) - y_0}{\sum_{i=1}^{m} \frac{w^{(i)}}{2} R_i(x_0)} \frac{R_i(x_0)}{2} \tag{29}$$

The unknown parameters may be modified on the basis of (24), after the input of one data collection into the system or after the input of all data collections (cummulative method). Additionally, the following heuristic rules for changes of η parameter have been applied. If in four sequential iterations mean square error has diminished for the whole learning set, then the learning parameter is increased (multiplied by n_I). If in four sequential iterations the error commutatively has been increased and decreased then the learning parameter was decreased (multiplied by n_D).

Another problem is the estimation of the number m of if-then rules. This task is solved by means of commutative generation of new if-then rules (increasing of number m) and estimation of the parameters usin the gradient method. A new rule is generated if the condition is fulfilled for any k:

$$\min_i \max_j A_j^{(i)}(x_{j0}(k)) < A_0 \tag{30}$$

where A_0 - is a constant determining the sensitivity of the method for generation of new rules. For a new generated rule we take the following initial parameter values for Gaussian membership function: $c_j^{(i)} = x_{j0}(k)$, $s_j^{(i)} = s_{0j}$, where s_{0j} - is a predefined constant implied from the determining of membership function number for a given input (mfn_j):

$$s_{0j} = \frac{\max\limits_k x_{0j}(k) - \min\limits_k x_{0j}(k)}{2\, mfn_j \sqrt{2 \ln\left(\frac{1}{A_0}\right)}} \tag{31}$$

4. Application of ANNBFIS to pattern recognition

The fuzzy system described in the previous section can be applied to pattern recognition. If patterns from a learning set belong to classes ω_1 and ω_2 then we can build a fuzzy system whose output takes positive values for patterns from class ω_1 and negative values for class ω_2. If we denote fuzzy system as $y_0 = FNN(x_0)$, we get:

$$y_0(k) = FNN[x_0(k)] \begin{cases} > 0, & if\, x_0(k) \in \omega_1 \\ \le 0, & if\, x_0(k) \in \omega_2 \end{cases} \tag{32}$$

During the learning process of a classifier we take $t_0(k) = 1$ for pattern $x_0(k)$ from class ω_1 and $t_0(k) = -1$ for pattern from class ω_2. For a bigger number of class ($p > 2$) we use an extension class-rest (p classifiers) or class-class (p(p-1)/2 classifiers) [8].

A numerical example was presented using Ripley data [7]. These data are taken on diagnostic tests on patients with Cushing's syndrome, a hypersensitive disorder associated with over-secretion of cortisol by the adrenal gland. These data on Fig. 1 consist of three classes of the syndrome represented as '+' (adenoma), '*' (bilateral hyerplasia) and 'O' (carcinoma). The observations are logaritm of urinary excretion rates (mg/24h) of the steroid metabolites tetrahydrocortisone and pregnanetriol. The classes of pattern are determined histopathologically.

The learning process was evaluated for structure class-rest (3 classifiers). It is assumed that: $m\dot{f}n_1 = m\dot{f}n_2 = 4$, $\eta = 0.01$, $n_I = 1.1$, $n_D = 0.9$, $A_0 = 0.5$. The obtained discrimination functions after 100 iterations are presented in Fig. 1. From this figure we can infer that the created discrimination functions separate perfectly the considered class of patterns.

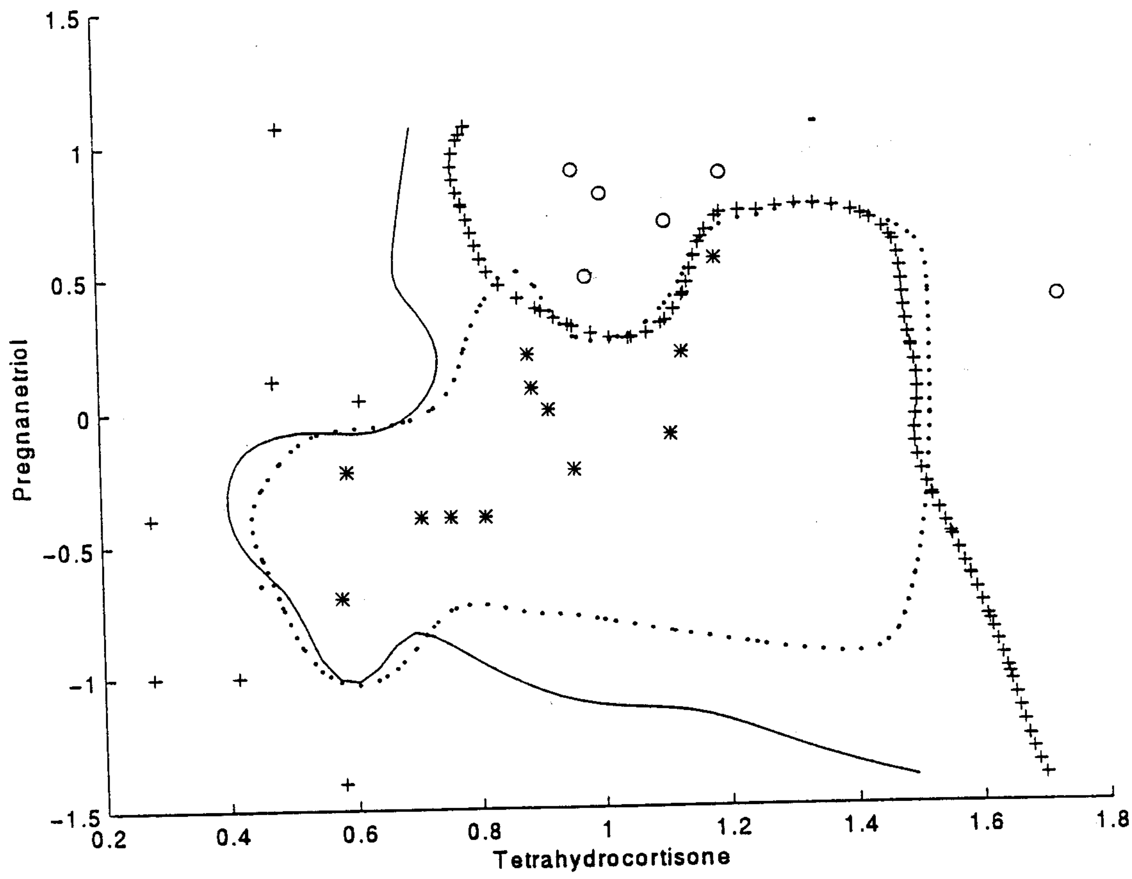

Fig. 1. Discrimination functions for Cushing's syndrome.

5. Conclusions

In this paper a new artificial neural network based fuzzy inference system (ANNBFIS) has been described. Such a presented system can be used for the automative if-then rule generation. The novelty of that system in comparison to the well known from literature is a whole moving fuzzy consequent. A particular case of our system is Jang's ANFIS (moving consequent considered as singleton) or Cho & Wang AFS with a constant fuzzy consequent. The gradient method of parameter optimization for ANNBFIS has been used. A promising application of the presented system to pattern recognition of Cushing's syndrom has been shown.

Literatura

[1] K. B. Cho, B. H. Wang, Radial basis function based adaptive fuzzy systems and their applications to system identification and prediction, Fuzzy Set & Systems 83, 1996, pp.325-339,

[2] S. Horikawa, T. Furuhashi, Y. Uchikawa, On fuzzy modeling using fuzzy neural networks with the back-propagation algorithm, IEEE Trans. Neur. Net., Vol. 4, 1992, pp.801-806,

[3] J. R. Jang, C. Sun, Functional equivalence between radial basis function and fuzzy inference systems, IEEE Trans. Neur. Net., Vol. 4, No. 1, 1993, pp.156-159,

[4] J. R. Jang, C. Sun, Neuro-fuzzy modeling and control, Proc. IEEE, Vol.83, No.3, 1995, 378-406,

[5] B. Kosko, Fuzzy associative memories, in: Kandel. Ed., Fuzzy Expert Systems, 1987,

[6] S. Mitra, S. K. Pal, Fuzzy multi-layer perceptron, inferencing and rule generation, IEEE Trans. Neur. Net., Vol. 6, 1995, pp.51-63,

[7] B. R.Ripley, Pattern Recognition and neural network, Cambridge University Press 1996,

[8] J. T. Tou, R. C. Gonzalez, Pattern recognition principles, Addison-Wesley 1974,

[9] L. Wang, J. M. Mendel, Generating fuzzy rules by learning from examples, IEEE Trans. Systems Man Cyber., Vol. 22, 1992, pp.1414-1427,