Int. J. Appl. Math. Comput. Sci., 2005, Vol. 15, No. 4, 561–576

NEURO–FUZZY MODELLING BASED ON A DETERMINISTIC ANNEALING APPROACH

ROBERT CZABAŃSKI

Department of Automatic Control, Electronics and Computer Sciences
Silesian University of Technology
ul. Akademicka 16, 44–100 Gliwice, Poland
e-mail: robert.czabanski@polsl.pl

This paper introduces a new learning algorithm for artificial neural networks, based on a fuzzy inference system ANBLIR. It is a computationally effective neuro-fuzzy system with parametrized fuzzy sets in the consequent parts of fuzzy if-then rules, which uses a conjunctive as well as a logical interpretation of those rules. In the original approach, the estimation of unknown system parameters was made by means of a combination of both gradient and least-squares methods. The novelty of the learning algorithm consists in the application of a deterministic annealing optimization method. It leads to an improvement in the neuro-fuzzy modelling performance. To show the validity of the introduced method, two examples of application concerning chaotic time series prediction and system identification problems are provided.

Keywords: fuzzy systems, neural networks, neuro-fuzzy systems, rules extraction, deterministic annealing, prediction

1. Introduction

When we create a model of a real process using only precise information, we frequently encounter a lot of unsolvable difficulties due to the highly complex nature of the world. On the other hand, humans have already used imprecise information in the form of linguistic terms of the natural language to describe all phenomena for thousands of years. This observation resulted in a novel way of characterizing nonprobabilistic uncertainties: fuzzy sets (Zadeh, 1965). Fuzzy set theory is a mathematical tool which incorporates vague information, expressed in a natural, humancomprehensible form to describe complex real world processes. A fundamental of fuzzy systems is a set of conditional if-then statements (rule base) with linguistically interpreted propositions. The ability to define fuzzy sets in premise and conclusion parts of fuzzy if-then rules is crucial for the use of fuzzy systems. Fuzzy modelling is an important tool in diverse areas, including various engineering fields, such as automatic control, signal processing, time-series prediction, identification, pattern recognition, information retrieval, data mining, consumer electronics, etc.

Methods of the extraction of fuzzy if-then statements can be broadly categorized into two families (Czogała and Łęski, 1999): (i) those determined from the knowledge of a human expert, or (ii) those obtained automatically from numerical data which describe input/output system characteristics. Methods from the first family have some disadvantages: the appointed rule set is often incomplete, subjective, or even contradictory. However, they reveal high effectiveness, particularly in automatic control problems (Mamdani, 1974; 1976; 1977; Mamdani and Assilian, 1975). Early methods from the second family still require information from a human expert (Kosko, 1987; Yager and Filev, 1984; Zadeh, 1971). Succeeding procedures can be characterized by either the necessity of applying heuristic procedures (Zadeh, 1973) or limited applicability (Pedrycz, 1984b). A true breakthrough in automatic knowledge acquisition is the invention of methods which use the learning capability of artificial neural networks. The integration of neural networks and fuzzy models leads to the so-called neuro-fuzzy systems. Systems of this kind are usually represented as multilayer feedforward neural networks (Cho and Wang, 1996; Czogała and Łęski, 1999; Jang, 1993; Jang and Sun, 1995; Mitra and Pal, 1995; Rutkowska, 2001). Radial basis function networks constitute an important class of feedforward neural networks with one hidden layer. They have some useful properties that make them particularly interesting for the extraction of fuzzy if-then rules (Cho and Wang, 1996; Czogała and Łęski, 1996; 1999; Jang and Sun, 1995; Yen *et al.*, 1998). Radial basis function networks are functionally equivalent to fuzzy systems (Jang and Sun, 1993). This equivalence resulted in the construction of the Takagi-Sugeno-Kang (TSK) type of the Adaptive Network based Fuzzy Inference System (ANNFIS) (Jang and Sun, 1993; 1995). The way of improving the

interpretability of TSK fuzzy models by combining global and local learning was presented by Yen *et al.* (1998). A similar approach was described by (Rao *et al.*, 1997; Rao and Rose, 1999; Rose, 1991; 1998). They proposed an algorithm based on a deterministic annealing (DA) optimization method that avoids many local minima on a cost surface during the estimation process of parameters of radial functions.

Fuzzy systems can be divided into two main classes. The first group (fuzzy systems based on the Mamdani as well as the logical approach) is based on conditional if-then statements whose antecedents and consequents utilize fuzzy sets. The second group (Takagi-Sugeno-Kang type systems) use a rule structure that has fuzzy antecedent and functional consequent parts. Both of them can be obtained as a particular case of the Artificial Neural Network Based Fuzzy Inference System (ANNBFIS), with parameterized consequents of fuzzy if-then rules (Czogała and Łęski, 1996). The equivalence of approximate reasoning results using logical and conjunctive interpretations of if-then rules which occurs under some respective circumstances was shown in a series of works by Czogała and Łęski (1999; 2001). This observation led to a more generalized structure of ANNBFIS–ANBLIR (Artificial neural Network Based on Logical Interpretation of fuzzy if-then Rules), a computationally effective system with parameterized consequents based on both conjunctive and logical interpretations of fuzzy rules (Czogała and Łęski, 1999). The ANBLIR system can be successfully applied to solve many practical problems such as classification, control, digital channel equalization, pattern recognition, prediction, signal compression and system identification (Czogała and Łęski, 1999). Originally, its learning procedure was based on a hybrid method which uses a combination of the steepest-descent and least-squares methods (Czogała and Łęski, 1999). However, it may lead to a local minimum in the case of a multimodal criterion function.

In this paper a modification of the ANBLIR learning algorithm is presented. It consists in the application of a deterministic annealing method adopted to the neuro-fuzzy system with parameterized consequents. To show the validity of the proposed method, the described neuro-fuzzy system is applied to the prediction of a chaotic time series generated through the solution of the Mackey-Glass equation (Schuster, 1984) and to the system identification problem based on Box and Jenkins' data (1976).

The remainder of this article is as follows: In Section 2, the structure of a neuro-fuzzy system based on logical as well as conjunctive interpretation of if-then rules is presented. Section 3 introduces a new learning algorithm based on the deterministic annealing method adopted to the neuro-fuzzy modeling problem. In Section 4, a learning algorithm that combines the deterministic annealing approach and the least-squares method is outlined. Section 5 presents an initialization procedure for the learning algorithm based on fuzzy clustering of training data. Examples of applications of the new learning algorithm are provided in Section 6. Section 7 concludes the paper and points out future directions.

2. Neuro–Fuzzy System with Parameterized Consequents

A fuzzy system with parameterized consequents generates inference results based on fuzzy if-then rules. Every fuzzy conditional statement from a rule base may be written in the following form (Czogała and Łęski, 1999):

$$R^{(i)}: \text{if } \operatorname*{and}_{j=1}^{t}\left(X_j \text{ is } A_j^{(i)}\right) \text{ then } Y \text{ is } B^{(i)}\left(y, \underline{\theta}\right), \quad (1)$$

$i = 1, \ldots, I$, where I denotes the number of fuzzy if-then rules, t is the number of inputs, X_j are input linguistic variables of the fuzzy system, Y is an output linguistic variable of the system, $A_j^{(i)}$ and $B^{(i)}\left(y, \underline{\theta}\right)$ are linguistic values (terms) of fuzzy sets in antecedents and consequents, respectively, and $\underline{\theta}$ is a set of parameters which define the consequence fuzzy set.

During the inference process, crisp numerical data from a training set are mapped to fuzzy sets using singleton fuzzifiers. Then the i-th if-then rule has the form (Czogała and Łęski, 1999):

$$R^{(i)}: \text{if } \operatorname*{and}_{j=1}^{t}\left(x_{0j} \text{ is } A_j^{(i)}\right) \text{then } Y \text{ is } B^{(i)}\left(y, \underline{x}_0\right), \quad (2)$$

where x_{0j} is the j-th element of the input vector of fuzzy singletons $\underline{x}_0 = [x_{01}, x_{02}, \ldots, x_{0t}]$.

If we assume that fuzzy sets of linguistic values in rule antecedents have Gaussian membership functions, then we can evaluate the grade of membership for the i-th rule and j-th input x_{0j} based on the following formula (Czogała and Łęski, 1999):

$$A_j^{(i)}\left(x_{0j}\right) = \exp\left[-\frac{1}{2}\left(\frac{x_{0j} - c_j^{(i)}}{s_j^{(i)}}\right)^2\right], \quad (3)$$

where $c_j^{(i)}$ and $s_j^{(i)}$ for $i = 1, 2, \ldots, I$ and $j = 1, 2, \ldots, t$ are membership function parameters, centre and dispersion, respectively.

From the membership functions of premise components, we can get a firing strength of rules:

$$F^{(i)}\left(\underline{x}_0\right) = A_1^{(i)}\left(x_{01}\right) \wedge A_2^{(i)}\left(x_{02}\right) \wedge \cdots \wedge A_t^{(i)}\left(x_{0t}\right), \quad (4)$$

where $\wedge$ stands for the t-norm, which represents the explicit connective 'and' of multi-input rule predicates.

Assuming the t-norm $\wedge$ to be the algebraic product, we get the firing strength of the i-th rule in the form

$$F^{(i)}(\underline{x}_0) = \prod_{j=1}^{t} A_j^{(i)}(x_{0j}) = \exp\left[-\frac{1}{2}\sum_{j=1}^{t}\left(\frac{x_{0j}-c_j^{(i)}}{s_j^{(i)}}\right)^2\right]. \quad (5)$$

$i = 1, \ldots, I$.

During the next stage of fuzzy inference, we evaluate the resulting conclusions of each rule. The kind of executed operations depends on the chosen way of interpreting if-then rules. We can introduce the general form of the conclusion membership function before aggregation as follows (Czogała and Łęski, 1999):

$$B^{(i)\prime}(y, \underline{x}_0) = \Psi\left[F^{(i)}(\underline{x}_0), B^{(i)}(y, \underline{x}_0)\right]. \quad (6)$$

We can apply different classes of membership functions of fuzzy sets in consequents, including the most frequently used ones, such as triangular, trapezoidal or Gaussian. In what follows, we assume symmetric triangular membership functions. This choice is dictated by the computational effectiveness of the system. A symmetric triangular membership function can be defined using two parameters: the width of the triangle base $w^{(i)}$ and the location of the centre of gravity $y^{(i)}(\underline{x}_0)$ determined by a linear combination of fuzzy system inputs:

$$y^{(i)}(\underline{x}_0) = p_0^{(i)} + p_1^{(i)} x_{01} + \cdots + p_t^{(i)} x_{0t} = \underline{p}^{(i)T} \underline{x}_0'. \quad (7)$$

The above dependence defines the so-called moving (parameterized) consequent (Czogała and Łęski, 1996; 1999).

The membership function of the resulting conclusions for the i-th rule after the inference process, but before aggregation, can be written as

$$B^{(i)\prime}(y, \underline{x}_0) = \Phi\left[F^{(i)}(\underline{x}_0), w^{(i)}, y^{(i)}(\underline{x}_0)\right], \quad (8)$$

where Φ stands for the fuzzy implication (for a logical interpretation of if-then rules) or the t-norm (for a conjunctive interpretation of if-then rules).

The output fuzzy set is derived from the aggregation process:

$$B'(y) = \bigoplus_{i=1}^{I} B^{(i)\prime}(y, \underline{x}_0), \quad (9)$$

where $\bigoplus$ denotes the aggregation operation.

The resulting fuzzy set has a non-informative part, i.e., there are elements of the fuzzy set $y \in \mathbf{Y}$ whose membership values are equal in the whole space $\mathbf{Y}$. Therefore, the following modified indexed centre of gravity defuzzifier (MICOG) has to be used (Czogała and Łęski, 1999):

$$y_0 = \frac{\int y\,(B'(y) - \alpha)\,\mathrm{d}y}{\int (B'(y) - \alpha)\,\mathrm{d}y}, \quad (10)$$

where y_0 denotes the crisp output value, and $\alpha \in [0, 1]$ describes the interdeterminancy that goes together with information. If we assume additionally the normalized arithmetic mean as the aggregation

$$\bigoplus_{i=1}^{I} B^{(i)\prime}(y, \underline{x}_0) = \frac{1}{I}\sum_{i=1}^{I} B^{(i)\prime}(y, \underline{x}_0), \quad (11)$$

then the final crisp output value of the fuzzy system can be evaluated from the formula

$$y_0 = \frac{\int \frac{y}{I}\sum_{i=1}^{I}\left(B^{(i)\prime}(y, \underline{x}_0) - \alpha_i\right)\mathrm{d}y}{\int \frac{1}{I}\sum_{i=1}^{I}\left(B^{(i)\prime}(y, \underline{x}_0) - \alpha_i\right)\mathrm{d}y} = \frac{\sum_{i=1}^{I}\int y\left(B^{(i)\prime}(y, \underline{x}_0) - \alpha_i\right)\mathrm{d}y}{\sum_{i=1}^{I}\int\left(B^{(i)\prime}(y, \underline{x}_0) - \alpha_i\right)\mathrm{d}y}. \quad (12)$$

The location of the modified indexed centre of gravity for each fuzzy rule is defined as

$$y^{(i)}(\underline{x}_0) = \frac{\int y\left(B^{(i)\prime}(y, \underline{x}_0) - \alpha_i\right)\mathrm{d}y}{\int\left(B^{(i)\prime}(y, \underline{x}_0) - \alpha_i\right)\mathrm{d}y}. \quad (13)$$

Combining (12) and (13) yields

$$y_0 = \frac{\sum_{i=1}^{I}\left[\int\left(B^{(i)\prime}(y, \underline{x}_0) - \alpha_i\right)\mathrm{d}y\right] y^{(i)}(\underline{x}_0)}{\sum_{i=1}^{I}\int\left(B^{(i)\prime}(y, \underline{x}_0) - \alpha_i\right)\mathrm{d}y}. \quad (14)$$

The integral $\int\left(B^{(i)\prime}(y, \underline{x}_0) - \alpha_i\right)\mathrm{d}y$ defines the area under the curve corresponding to the membership function of the consequent of the i-th rule after removing the non-informative part. For a symmetric triangular function, it is a function of the firing strength of the rule $F^{(i)}(\underline{x}_0)$ and width of the triangle base $w^{(i)}$:

$$\int\left(B^{(i)\prime}(y, \underline{x}_0) - \alpha_i\right)\mathrm{d}y = g\left(F^{(i)}(\underline{x}_0), w^{(i)}\right). \quad (15)$$

Finally, the crisp output value of the fuzzy system takes the form

$$y_0 = \sum_{i=1}^{I} G^{(i)}\left(\underline{x}_0\right) y^{(i)}\left(\underline{x}_0\right), \tag{16}$$

where

$$G^{(i)}\left(\underline{x}_0\right) = \frac{g\left(F^{(i)}\left(\underline{x}_0\right), w^{(i)}\right)}{\sum_{k=1}^{I} g\left(F^{(k)}\left(\underline{x}_0\right), w^{(k)}\right)}. \tag{17}$$

The function $g\left(F^{(i)}\left(\underline{x}_0\right), w^{(i)}\right)$ depends on the fuzzy implication we use. The respective formulae for selected fuzzy implications are included in Table 1. For notational simplicity, we write $B \triangleq B^{(i)}\left(y, \underline{x}_0\right)$, $F \triangleq F^{(i)}\left(\underline{x}_0\right)$ and $w \triangleq w^{(i)}$.

It was proved (Czogała and Łęski, 1999; 2001) that the neuro-fuzzy system with parameterized consequents based on Łukasiewicz and Reichenbach implications produces inference results equivalent to the inference obtained from Mamdani and Larsen fuzzy relations, respectively.

To establish a rule base of the fuzzy system with parameterized consequents, the following set of unknown parameters has to be estimated:

- centres of Gaussian membership functions of fuzzy sets from premises: $c_j^{(i)}$ for $i = 1, 2, \ldots, I$ and $j = 1, 2, \ldots, t$,
- dispersions of Gaussian membership functions of fuzzy sets from premises: $s_j^{(i)}$ for $i = 1, 2, \ldots, I$ and $j = 1, 2, \ldots, t$,
- parameters determining the locations of fuzzy sets from consequents: $p_j^{(i)}$ for $i = 1, 2, \ldots, I$ and $j = 0, 1, 2, \ldots, t$,
- parameters determining the widths of fuzzy sets from consequents: $w^{(i)}$ for $i = 1, 2, \ldots, I$.

The number of rules I is also unknown. We assume that it is pre-set arbitrarily. The number of antecedents t is defined by the size of the input training vector directly. The described fuzzy system with parameterized consequents can be treated as a radial basis function neural network (Czogała and Łęski, 1999). Consequently, the unknown neuro-fuzzy system parameters can be estimated using learning algorithms of neural networks.

Several solutions to this problem have been introduced in the literature (Czogała and Łęski, 1996; 1999; Łęski, 2003). In this work, a new learning procedure which combines deterministic annealing and least-squares methods is presented.

In the following, we assume that we have N examples of input vectors $\underline{x}_0(n) \in \mathbb{R}^t$ and the same number of known output values $t_0(n) \in \mathbb{R}$. They form the so-called training set:

$$Tr^{(N)} = \left\{\underline{x}_0(n), t_0(n)\right\}, \quad n = 1, 2, \ldots, N. \tag{18}$$

3. Deterministic Annealing

Our goal is the extraction of the set of fuzzy if-then rules that represent the knowledge of the phenomenon under consideration. The extraction process consists in estimating membership function parameters of both antecedents and consequents. To solve this task, we use a supervised learning algorithm based on the minimization of the following error (cost) function measured over the training set:

$$E = \sum_{n=1}^{N} d\left(t_0(n) - y_0(n)\right), \tag{19}$$

where $d(\cdot)$ is a distortion measure.

To increase the ability of avoiding many local minima that trap descent methods, we employ the technique of deterministic annealing (Rose, 1991; 1998; Rao *et al.*, 1997; Rao and Rose, 1999) adapted to the neuro-fuzzy system with learning parameterized consequents. However, it is not guaranteed that a global optimum of the cost will be found (Rao and Rose, 1999).

The deterministic annealing method was proposed by Rose in his Ph.D. dissertation (Rose, 1991). Its extensions to clustering, classification, regression and parsimonious modelling were described in (Rao and Rose, 1999; Rao *et al.*, 1997; Rose, 1998). The deterministic annealing is a simulated annealing (Kirkpatrick *et al.*, 1983; Metropolis *et al.*, 1953) based method, which replaces computationally intensive stochastic simulations by straightforward deterministic optimization of the modelled system error energy (Rao *et al.*, 1997). The algorithm reduces to the minimization of the cost function while simultaneously controlling the entropy level of the current solution.

From (17) we see that

$$\sum_{i=1}^{I} G^{(i)}\left(\underline{x}_0\right) = 1. \tag{20}$$

Equation (16) defines the neuro-fuzzy system as a mixture of experts (models). Its global output is expressed as a linear combination of I outputs $y^{(i)}\left(\underline{x}_0\right)$ of local models, each represented by a single fuzzy conditional statement. The weight $G^{(i)}\left(\underline{x}_0\right)$ may be interpreted as the possibility of associating the i-th local model with input data $\underline{x}_0$.

Table 1. Function $g\left(F^{(i)}\left(\underline{x}_0\right), w^{(i)}\right)$ for selected fuzzy implications.

Fuzzy implication $\Psi[F,B]$	α	$g(F,w)$
Fodor $\begin{cases} 1, & \text{if } F \le B \\ \max(1-F,B), & \text{otherwise} \end{cases}$	$1-F$	$\begin{cases} \frac{w}{2}\left(1-2F+2F^2\right), & F \ge \frac{1}{2} \\ wF(1-F), & F < \frac{1}{2} \end{cases}$
Gödel $\begin{cases} 1, & \text{if } F \le B \\ B, & \text{otherwise} \end{cases}$	0	$\frac{w}{2}\left(2-2F+F^2\right)$
Gougen $\min\left(\frac{B}{F}, 1\right), F \ne 0$	0	$\frac{w}{2}(2-F)$
Kleene-Dienes $\max(1-F,B)$	$1-F$	$\frac{w}{2}F^2$
Łukasiewicz $\min(1-F+B,1)$	$1-F$	$\frac{w}{2}F(2-F)$
Reichenbach $1-F+FB$	$1-F$	$\frac{w}{2}F$
Rescher $\begin{cases} 1, & \text{if } F \le B \\ 0, & \text{otherwise} \end{cases}$	0	$w(1-F)$
Zadeh $\max\{1-F, \min(F,B)\}$	$1-F$	$\begin{cases} \frac{w}{2}(2F-1), & F \ge \frac{1}{2} \\ 0, & F < \frac{1}{2} \end{cases}$

For every local model we have to determine a set of its parameters

$$\underline{p}^{(i)} = \left[p_0^{(i)}, p_1^{(i)}, \ldots, p_t^{(i)}\right]^T, \tag{21}$$

as well as assignments $G^{(i)}\left(\underline{x}_0\right)$ that minimize the criterion (19). Deterministic annealing is a method that minimizes the squared-error cost

$$E = \sum_{n=1}^{N} E_n = \sum_{n=1}^{N} \frac{1}{2}\left(t_0(n) - y_0(n)\right)^2, \tag{22}$$

where N is the size of the training set.

In practice, we look for the following set of optimum values of the membership function parameters of antecedents and parameterized consequents:

$$\underline{\zeta}^{(i)} = \left[c_j^{(i)}, s_j^{(i)}, w^{(i)}, \underline{p}^{(i)^T}\right]^T, \tag{23}$$

$i = 1, \ldots, I$ and $j = 1, \ldots, t$.

The randomness of the association between data and local models can be measured using the Shannon entropy:

$$S = -\sum_{n=1}^{N}\sum_{i=1}^{I} G^{(i)}\left(\underline{x}_0(n)\right) \log G^{(i)}\left(\underline{x}_0(n)\right). \tag{24}$$

In deterministic annealing, the objective is the minimization of the cost E for a fixed level of entropy S_0:

$$\min E \text{ subject to } S = S_0. \tag{25}$$

The procedure involves a series of iterations while the randomness level is gradually reduced. To attain a global optimum of the cost, the framework of the simulated annealing method is used.

The constrained optimization is equivalent to unconstrained minimization of the Lagrangian (Rao *et al.*, 1997):

$$L = E - T\left(S - S_0\right), \tag{26}$$

where T is the Lagrange multiplier.

A connection between (26) and the annealing of a solid is essential here. The quantity L can be identified as the Helmholtz free energy of a physical system with the 'energy' E, 'entropy' S and 'temperature' T (Rao *et al.*, 1997).

At a high level of pseudo-temperature T, the minimization of the Lagrange function L amounts to entropy maximization of associating data and models. In other words, we seek a set of local models that are equally associated to each input data point — the set of local models which cooperate to work out a desired output (it can be noticed that, as $T \to \infty$, we get the uniform distribution of $G^{(i)}(\underline{x}_0)$ and, therefore, identical local models). As pseudo-temperature is lowered, more emphasis is placed on reducing the square error. This also leads to a decrease in entropy. We get more and more competitive local models, each associated with given data more closely. We cross gradually from cooperation to competition. Finally, at $T = 0$, the optimization is conducted regardless of the entropy level and the cost is minimized directly.

The deterministic annealing algorithm (DA) can be summarized as follows (Rao *et al.*, 1997):

1. Set the parameters: the initial solution $\underline{\zeta}$, initial pseudo-temperature $T_{\max}$, final pseudo-temperature $T_{\min}$ and annealing schedule function $q(T)$. Set $T = T_{\max}$.
2. Minimize the Lagrangian L using the steepest descent method:
$$\frac{\partial L}{\partial \underline{\zeta}} = \frac{\partial E}{\partial \underline{\zeta}} - T\frac{\partial S}{\partial \underline{\zeta}}. \tag{27}$$
3. Decrement the pseudo-temperature according to the annealing schedule $T \leftarrow q(T)$.
4. If $T < T_{\min}$, STOP. Otherwise, go to Step 2.

The annealing schedule function determines the pseudo-temperature reduction procedure. In the sequel, we assume the following decrement rule:

$$T \leftarrow q\,T, \tag{28}$$

where $q \in (0,1)$ is a pre-set parameter.

At each level of temperature we minimize the Lagrangian L iteratively. The parameters of the neuro-fuzzy system are

$$\underline{\zeta}(k+1) = \underline{\zeta}(k) - \eta \left.\frac{\partial L}{\partial \underline{\zeta}}\right|_{\underline{\zeta}=\underline{\zeta}(k)}, \tag{29}$$

where η is the learning rate, and k denotes the iteration index.

The Lagrange function (26) can be written in the form

$$L = \sum_{n=1}^{N} L_n, \tag{30}$$

where

$$L_n = E_n + T\sum_{i=1}^{I} G^{(i)}(\underline{x}_0)\log G^{(i)}(\underline{x}_0) + \frac{T}{N}S_0. \tag{31}$$

For notational simplicity, we introduce the following symbols:

$$\Xi^{(i)}(\underline{x}_0(n)) = [y_0(n) - t_0(n)]\; y^{(i)}(\underline{x}_0(n)) + T\,\log G^{(i)}(\underline{x}_0(n)), \tag{32}$$

$$\overline{\Xi}(\underline{x}_0(n)) = \sum_{i=1}^{I} G^{(i)}(\underline{x}_0(n))\;\Xi^{(i)}(\underline{x}_0(n)). \tag{33}$$

Then the gradients $\partial L_n/\partial\underline{\zeta},\ n = 1,\ldots,N$ may be expressed as

$$\frac{\partial L_n}{\partial c_j^{(i)}} = \frac{x_{j0} - c_j^{(i)}}{\left(s_j^{(i)}\right)^2}\frac{F^{(i)}(\underline{x}_0)}{g\left(F^{(i)}(\underline{x}_0), w^{(i)}\right)} \times \frac{\partial g\left(F^{(i)}(\underline{x}_0), w^{(i)}\right)}{\partial F^{(i)}(\underline{x}_0)} G^{(i)}(\underline{x}_0) \times \left[\Xi^{(i)}(\underline{x}_0) - \overline{\Xi}(\underline{x}_0)\right]\Big|_{\underline{x}_0=\underline{x}_0(n)}, \tag{34}$$

$$\frac{\partial L_n}{\partial s_j^{(i)}} = \frac{\left(x_{j0} - c_j^{(i)}\right)^2}{\left(s_j^{(i)}\right)^3}\frac{F^{(i)}(\underline{x}_0)}{g\left(F^{(i)}(\underline{x}_0), w^{(i)}\right)} \times \frac{\partial g\left(F^{(i)}(\underline{x}_0), w^{(i)}\right)}{\partial F^{(i)}(\underline{x}_0)} G^{(i)}(\underline{x}_0) \times \left[\Xi^{(i)}(\underline{x}_0) - \overline{\Xi}(\underline{x}_0)\right]\Big|_{\underline{x}_0=\underline{x}_0(n)}, \tag{35}$$

$$\frac{\partial L_n}{\partial p_j^{(i)}} = \frac{\partial E}{\partial p_j^{(i)}} = \begin{cases} \left[y_0(n) - t_0(n)\right] G^{(i)}(\underline{x}_0(n))\; x_{j0}(n) & \text{for } j \neq 0, \\ \left[y_0(n) - t_0(n)\right] G^{(i)}(\underline{x}_0(n)) & \text{for } j = 0, \end{cases} \tag{36}$$

$$\frac{\partial L_n}{\partial w^{(i)}} = \frac{1}{g\left(F^{(i)}\left(\underline{x}_0\right), w^{(i)}\right)} \frac{\partial g\left(F^{(i)}\left(\underline{x}_0\right), w^{(i)}\right)}{\partial w^{(i)}} \times G^{(i)}(\underline{x}_0)\left[\Xi^{(i)}(\underline{x}_0) - \overline{\Xi}\left(\underline{x}_0\right)\right]\Big|_{\underline{x}_0=\underline{x}_0(n)}. \quad (37)$$

The partial derivatives with respect to unknown parameters for all data from the training set may be written in the following form:

$$\frac{\partial L}{\partial c_j^{(i)}} = \frac{1}{\left(s_j^{(i)}\right)^2} \sum_{n=1}^{N} \left[x_{j0}(n) - c_j^{(i)}\right] \times \frac{F^{(i)}\left(\underline{x}_0(n)\right)}{g\left(F^{(i)}\left(\underline{x}_0(n)\right), w^{(i)}\right)} \times \frac{\partial g\left(F^{(i)}\left(\underline{x}_0(n)\right), w^{(i)}\right)}{\partial F^{(i)}\left(\underline{x}_0(n)\right)} \times G^{(i)}\left(\underline{x}_0(n)\right)\left[\Xi^{(i)}\left(\underline{x}_0(n)\right) - \overline{\Xi}\left(\underline{x}_0(n)\right)\right], \quad (38)$$

$$\frac{\partial L}{\partial s_j^{(i)}} = \frac{1}{\left(s_j^{(i)}\right)^3} \sum_{n=1}^{N} \left[x_{j0}(n) - c_j^{(i)}\right]^2 \times \frac{F^{(i)}\left(\underline{x}_0(n)\right)}{g\left(F^{(i)}\left(\underline{x}_0(n)\right), w^{(i)}\right)} \times \frac{\partial g\left(F^{(i)}\left(\underline{x}_0(n)\right), w^{(i)}\right)}{\partial F^{(i)}\left(\underline{x}_0(n)\right)} \times G^{(i)}(\underline{x}_0(n))\left[\Xi^{(i)}\left(\underline{x}_0(n)\right) - \overline{\Xi}\left(\underline{x}_0(n)\right)\right], \quad (39)$$

$$\frac{\partial L}{\partial p_j^{(i)}} = \frac{\partial E}{\partial p_j^{(i)}} = \begin{cases} \left[y_0(n) - t_0(n)\right] \sum_{n=1}^{N} G^{(i)}\left(\underline{x}_0(n)\right) x_{j0}(n) \\ \qquad \text{for } j \neq 0, \\ \left[y_0(n) - t_0(n)\right] \sum_{n=1}^{N} G^{(i)}\left(\underline{x}_0(n)\right) \\ \qquad \text{for } j = 0, \end{cases} \quad (40)$$

$$\frac{\partial L}{\partial w^{(i)}} = \sum_{n=1}^{N} \frac{1}{g\left(F^{(i)}\left(\underline{x}_0(n)\right), w^{(i)}\right)} \times \frac{\partial g\left(F^{(i)}\left(\underline{x}_0(n)\right), w^{(i)}\right)}{\partial w^{(i)}} \times G^{(i)}\left(\underline{x}_0(n)\right)\left[\Xi^{(i)}\left(\underline{x}_0(n)\right) - \overline{\Xi}\left(\underline{x}_0(n)\right)\right]. \quad (41)$$

If we introduce the notation

$$\underline{d}\left(\underline{x}_0\right) = \left[G^{(1)}\left(\underline{x}_0\right)\underline{x}_0'^T, G^{(2)}\left(\underline{x}_0\right)\underline{x}_0'^T, \ldots, G^{(I)}\left(\underline{x}_0\right)\underline{x}_0'^T\right], \quad (42)$$

$$\underline{P} = \left[\underline{p}^{(1)T}, \underline{p}^{(2)T}, \ldots, \underline{p}^{(I)T}\right]^T, \quad (43)$$

where

$$\underline{x}_0' = \begin{bmatrix} 1 \\ \underline{x}_0 \end{bmatrix} \quad (44)$$

is the extended input vector

$$\underline{p}^{(i)T} = \left[p_0^{(i)}, p_1^{(i)}, \ldots, p_t^{(i)}\right], \quad (45)$$

then Eqn. (16) defining the crisp output value of the neuro-fuzzy system is given by (Czogała and Łęski, 1999):

$$y_0 = \underline{d}\left(\underline{x}_0\right)^T \underline{P}. \quad (46)$$

Thus, the parameters $\underline{P}$ of consequents may be estimated using the least-squares (LS) method (Czogała and Łęski, 1999; Jang *et al.*, 1997; Sugeno and Kang, 1988). The least-squares method accelerates the convergence of the learning method (Czogała and Łęski, 1999). There are two approaches to solve the LS problem, namely global and local ones (Łęski, 2003). In what follows, we adopt the local one. It enables us to tune each local model (rule) independently. Hence, we have to solve I independent weighted LS problems, one for each fuzzy conditional statement (Łęski, 2003). To avoid the matrix inverse operation, the recurrent least-squares method can be applied (Czogała and Łęski, 1999).

The integration of the least-squares algorithm used for estimating the parameters of linear combinations in the fuzzy sets of the consequents and the deterministic annealing procedure used for estimating the remaining parameters of the neuro-fuzzy system leads to a hybrid learning method.

4. Learning Algorithm

The integration of the least-squares procedure with the deterministic annealing method leads to a learning method where the parameters of the fuzzy sets from antecedents and consequents of fuzzy if-then rules are adjusted separately. The antecedent parameters $c_j^{(i)}, s_j^{(i)}, i = 1, 2, \ldots, I$ and $j = 1, 2, \ldots, t$, as well as the triangle base widths $w^{(i)}, i = 1, 2, \ldots, I$ of fuzzy sets in consequents are estimated by means of the deterministic annealing method, whereas the parameters of the linear equations from consequents are adjusted using the least-squares algorithm $\underline{p}^{(i)^T}, i = 1, 2, \ldots, I$. The proposed

method can be summarized in the following steps:

1. Set the parameters: the initial solution $\underline{\zeta}$, initial pseudo-temperature $T_{\max}$, final pseudo-temperature $T_{\min}$ and annealing schedule function $q\,(T)$. Set $T = T_{\max}$.
2. Minimize the Lagrangian L using the steepest-descent method (27).
3. Estimate the parameters of linear equations from the consequents $\underline{P}$ (43) by means of the least-squares method.
4. Check the equilibrium condition $|\delta S| = \left|S^{[k-1]} - S^{[k]}\right| / \left|S^{[k-1]}\right| > \delta$ or the stopping condition $k \leq k_{\max}$, where k denotes the iteration index, δ is a pre-set parameter and $k_{\max}$ denotes the maximum number of iterations at a given level of pseudo-temperature. If one of them is fulfilled, go to Step 2.
5. Lower pseudo-temperature according to the annealing schedule $T \leftarrow q\,T$.
6. If $T \geq T_{\min}$, go to Step 2.
7. Perform a zero entropy iteration, i.e., set $T = 0$ and minimize the square error using the steepest-descent and least-squares methods.
8. Stop the algorithm.

Another problem is the initialization of the learning algorithm. Its solution is described in the subsequent sections.

5. Initialization of the Learning Algorithm

The problem of estimating initial values for the parameters of membership functions for antecedents can be solved by means of preliminary clustering of the input training data (Czogała and Łęski, 1999). For this task we use the fuzzy c-means (FCM) clustering method (Bezdek, 1982). The quality of the FCM method as the initialization procedure was confirmed in (Czogała and Łęski, 1996; 1999). Clustering is based on the partition of the input vectors $\underline{x}_0\,(n)$ into c classes represented by the prototypes (cluster centres) $\underline{v}_i \in \underline{V} \subset \mathbb{R}^t,\ \forall i = 1, 2, \ldots, c$. The certainty of the assignment of the n-th sample in the i-th class is measured by the grade of membership $u_{in} \in [0, 1]$. The $(c \times N)$–dimensional partition matrix $\underline{U} = [u_{in}]$ is a fuzzy c-partition in the set $\mathcal{M}_{fc}$ defined as (Bezdek, 1982):

$$\mathcal{M}_{fc} = \left\{ \underline{U} \in \underline{V}_{cN} \,\middle|\, u_{in} \in [0,1], \sum_{i=1}^{c} u_{in} = 1,\ \forall\, 1 \leq n \leq N,\ 0 < \sum_{n=1}^{N} u_{in} < N,\ \forall\, 1 \leq i \leq c \right\}. \quad (47)$$

In the FCM method we seek a partition that minimizes the criterion function given by (Bezdek, 1982):

$$J_m(\underline{U}, \underline{V}) = \sum_{i=1}^{c} \sum_{n=1}^{N} (u_{in})^m\, d_{in}^2, \quad (48)$$

where $d_{in}^2 = \|\underline{x}_0\,(n) - \underline{v}_i\|^2$ is the distance (the most frequent Euclidean distance) between the i-th prototype $\underline{v}_i$ and the n-th data point $\underline{x}_0\,(n)$, $m \in [1, \infty)$ is the weighted exponent (usually $m = 2$), and the cluster centres are defined as

$$\underline{v}_i = \frac{\sum\limits_{n=1}^{N} (u_{in})^m\, \underline{x}_0\,(n)}{\sum\limits_{n=1}^{N} (u_{in})^m}, \qquad i = 1, \ldots, c. \quad (49)$$

The iterative scheme leading to either a local minimum or a saddle point of the objective function $J_m\,(\underline{U}, \underline{V})$ is a series of commutative modifications of both the partition matrix and prototypes. If we fix the values of the parameters m and c, and for each $n = 1, \ldots, N$ we define the sets

$$\mathcal{I}_n = \{i \mid 1 \leq i \leq c,\, d_{in} = 0\},$$
$$\overline{\mathcal{I}}_n = \{1, 2, \ldots, c\} \setminus \mathcal{I}_n, \quad (50)$$

then using the technique of Lagrange multipliers we can get updating equations for partition ed matrix elements (Bezdek, 1982):

$$u_{in} = \begin{cases} \left(\sum\limits_{j=1}^{c} \left(\frac{d_{in}}{d_{jn}} \right)^{\frac{2}{m-1}} \right)^{-1} & \text{if } \ \mathcal{I}_n = \emptyset, \\ 0 & \text{if } \ i \in \overline{\mathcal{I}}_n, \\ & \text{and } \ \mathcal{I}_n \neq \emptyset, \\ \sum\limits_{i \in \mathcal{I}_n} u_{in} = 1 & \text{if } \ i \in \mathcal{I}_n \neq \emptyset, \end{cases} \quad (51)$$

where $i = 1, \ldots, c$ and $n = 1, \ldots, N$.

The FCM algorithm can be summarized as follows:

1. Fix the number of classes c, the weighted exponent value m. Initialize the membership matrix $\underline{U}^{(0)} \in \mathcal{M}_{fc}$.
2. Construct c prototypes using (49).
3. Compute the value of the criterion function $J_m^{(k)}(\underline{U}^{(k)}, \underline{V}^{(k)})$, where k is the iteration index.
4. Update the membership u_{in} according to (51).
5. Compare the last two values of the objective function $J_m^{(k)}(\underline{U}^{(k)}, \underline{V}^{(k)})$ and $J_m^{(k+1)}(\underline{U}^{(k+1)}, \underline{V}^{(k+1)})$. If the change was less than a predefined value, terminate the algorithm. Otherwise, go to Step 2.

Since the algorithm leads to a local minimum of the performance index (48), the computations are repeated for various random realizations of the initial partition matrix. They are stopped if a maximum number of iterations is achieved or when the change in the objective function is less than a predefined value. To evaluate cluster validity, we use the Xie-Beni validity index (Xie and Beni, 1991):

$$\mathfrak{v}_{XB}\left(\underline{U},\underline{V}\right)=\frac{1}{N}\frac{J_2(\underline{U},\underline{V})}{\operatorname{sep}\left(\underline{V}\right)}, \tag{52}$$

where

$$\operatorname{sep}\left(\underline{V}\right)=\min_{i\neq j}\left\|\underline{v}_i-\underline{v}_j\right\|^2 \tag{53}$$

is the separation measure between cluster centres.

The centre and dispersion parameters of Gaussian membership functions of the neuro-fuzzy system can be initialized using the clustering results (Czogała and Łęski, 1999):

$$c_j^{(i)}=\frac{\sum\limits_{n=1}^{N}\left(u_{in}\right)^m x_{0j}\left(n\right)}{\sum\limits_{n=1}^{N}\left(u_{in}\right)^m} \tag{54}$$

and

$$\left(s_j^{(i)}\right)^2=\frac{\sum\limits_{n=1}^{N}\left(u_{in}\right)^m\left(x_{0j}\left(n\right)-c_j^{(i)}\right)^2}{\sum\limits_{n=1}^{N}\left(u_{in}\right)^m} \tag{55}$$

for each $i=1,\ldots,I$ and $j=1,\ldots,t$.

6. Numerical Experiments

To validate the introduced method of neuro-fuzzy modelling, two numerical experiments using benchmark databases were conducted. The first one concerns a problem of chaotic time-series prediction generated by means of the Mackey-Glass differential delay (Schuster, 1984):

$$\frac{\mathrm{d}x\left(t\right)}{\mathrm{d}t}=ax\left(t\right)+\frac{bx\left(t-\tau\right)}{1+\left[x\left(t-\tau\right)\right]^{10}}.$$

We considered the benchmark database generated by Jang (Jang and Sun, 1995) to be able to draw a comparison with the results reported in the literature. To obtain a solution, Jang applied the fourth-order Runge-Kutta method with the following values of parameters: $a=-0.1$, $b=0.2$, $x(0)=0.1$, $\tau=17$. From the Mackey-Glass time series $x(t)$, 1000 input data pairs were extracted in the following form (Jang and Sun, 1995):

$$\left[x\left(t\right),x\left(t-6\right),x\left(t-12\right),x\left(t-18\right),x(t+6)\right],$$

where $t=118$ to 1117. All data were divided into two subsets of equal cardinalities: the training set consisting of the first 500 input-output pairs and the testing set which contains the remaining data. The goal is the prediction of a future value $x(k+6)$ (system output) using past values combined in the embedded input vector

$$\left[x\left(k\right)\ x\left(k-6\right)\ x\left(k-12\right)\ x\left(k-18\right)\right]^T.$$

The learning process (DA+LS) was conducted for the most frequently used fuzzy implications (Fodor, Gödel, Gougen, Kleene-Dienes, Łukasiewicz, Reichenbach, Rescher and Zadeh) using the following parameter values: $\eta=0.01$, $T_{\max}\in\left\{10^{-2},10^{-3},\ldots,10^{-10}\right\}$, $T_{\min}=10^{-5}T_{\max}$, $q=0.95$, $k_{\max}=5$, and the number of zero entropy iterations equal to 500. The prediction results obtained from Łukasiewicz and Reichenbach implications are equivalent to inference results obtained on the basis of Mamdani and Larsen fuzzy relations, respectively. The number of fuzzy if-then rules I was changed from 2 to 6. The initial values of membership functions of antecedents were computed using FCM clustering results obtained for $m=2$. The clustering was stopped if the maximum number (500) of iterations was achieved or when in sequential iterations the change in the criterion function $J_m(\underline{U},\underline{V})$ was less than 10^{-5}. The partition process was repeated 25 times for different random initializations of the partition matrix. As a reference procedure, we used the original ANBLIR learning procedure. To get similar computational burdens of 2750 iterations of the steepest descent procedure combined with the least-squares method (SD+LS). Moreover, two heuristic rules for changes in the learning rate were applied in the reference learning algorithm (Czogała and Łęski, 1999; Jang *et al.*, 1997): (i) if in four sequential iterations the value of the error function was reduced for the whole learning set, then the learning parameter was increased (multiplied by 1.1), (ii) if in four sequential iterations the value of the error function increased and decreased alternately for the whole learning set, then the learning parameter was decreased (multiplied by 0.9). The prediction quality and the generalization ability were evaluated on the basis of the mean-square-error values obtained for the training (MSE_{trn}) and testing (MSE_{tst}) sets, respectively. All numerical experiments were conducted in the MATLAB® environment. The prediction results are presented in Tables 2–6.

Clearly, deterministic annealing based learning yields a consistent improvement in neuro-fuzzy modelling quality. Only two examples ($I=3$, Kleene-Dienes and Reichenbach implications) did not produce a decrease in the prediction error. The DA+LS method leads to a better generalization ability compared with the SD+LS learning as well. Only for one example ($I=3$, Reichenbach implication) there was no increase in the generalization ability.

The prediction errors for learning and testing data decrease as the number of fuzzy if-then rules for all im-

Table 2. MSE of the predition ($I = 2$).

Fuzzy implication (relation)	DA+LS learning			SD+LS learning	
	$T_{\max}$	MSE$_{trn}$	MSE$_{tst}$	MSE$_{trn}$	MSE$_{tst}$
Fodor	$\mathbf{10^{-4}}$	4.1370 e-5	5.0309 e-5	6.4907 e-5	7.9482 e-5
Gödel	10^{-9}	6.5435 e-5	8.3436 e-5	7.6207 e-5	9.7570 e-5
Gougen	10^{-2}	5.4552 e-5	6.3871 e-5	5.7061 e-5	6.4697 e-5
Kleene-Dienes	10^{-2}	8.0767 e-5	8.6786 e-5	9.9971 e-5	13.270 e-5
Łukasiewicz (Mamdani)	10^{-5}	4.7168 e-5	5.2120 e-5	5.9349 e-5	6.4673 e-5
Reichenbach (Larsen)	10^{-2}	6.9900 e-5	8.6171 e-5	8.3648 e-5	8.8223 e-5
Rescher	10^{-3}	5.8582 e-5	6.7172 e-5	6.0191 e-5	6.7677 e-5
Zadeh	10^{-5}	14.259 e-5	14.857 e-5	37.681 e-5	41.771 e-5

Table 3. MSE of the predition ($I = 3$).

Fuzzy implication (relation)	DA+LS learning			SD+LS learning	
	$T_{\max}$	MSE$_{trn}$	MSE$_{tst}$	MSE$_{trn}$	MSE$_{tst}$
Fodor	10^{-3}	1.9907 e-5	2.6856 e-5	3.2878 e-5	4.0671 e-5
Gödel	10^{-4}	1.7003 e-5	2.0325 e-5	2.6040 e-5	2.9445 e-5
Gougen	10^{-15}	1.5855 e-5	1.8599 e-5	1.5102 e-5	1.7598 e-5
Kleene-Dienes	10^{-4}	1.0977 e-5	1.4574 e-5	4.1052 e-5	4.5181 e-5
Łukasiewicz (Mamdani)	10^{-4}	1.1280 e-5	1.4606 e-5	1.8137 e-5	2.1673 e-5
Reichenbach (Larsen)	$\mathbf{10^{-4}}$	1.0763 e-5	1.4520 e-5	1.0656 e-5	1.5116 e-5
Rescher	10^{-4}	1.7849 e-5	2.2996 e-5	1.8999 e-5	2.4887 e-5
Zadeh	10^{-2}	3.9871 e-5	4.8860 e-5	39.399 e-5	298.92 e-5

Table 4. MSE of the predition ($I = 4$).

Fuzzy implication (relation)	DA+LS learning			SD+LS learning	
	$T_{\max}$	MSE$_{trn}$	MSE$_{tst}$	MSE$_{trn}$	MSE$_{tst}$
Fodor	10^{-10}	0.9843 e-5	1.4486 e-5	1.4618 e-5	2.0496 e-5
Gödel	10^{-2}	0.9904 e-5	1.2865 e-5	1.4771 e-5	1.9524 e-5
Gougen	10^{-8}	0.9611 e-5	1.1839 e-5	1.1134 e-5	1.3758 e-5
Kleene-Dienes	10^{-3}	0.7862 e-5	1.1910 e-5	1.3541 e-5	1.6517 e-5
Łukasiewicz (Mamdani)	$\mathbf{10^{-10}}$	0.6814 e-5	0.9603 e-5	0.8464 e-5	1.1667 e-5
Reichenbach (Larsen)	10^{-5}	0.6948 e-5	1.0150 e-5	0.7382 e-5	1.0948 e-5
Rescher	10^{-3}	1.0571 e-5	1.2771 e-5	1.1050 e-5	1.3608 e-5
Zadeh	10^{-2}	1.9340 e-5	2.7107 e-5	42.390 e-5	50.744 e-5

Table 5. MSE of the predition ($I = 5$).

Fuzzy implication (relation)	DA+LS learning			SD+LS learning	
	$T_{\max}$	MSE$_{trn}$	MSE$_{tst}$	MSE$_{trn}$	MSE$_{tst}$
Fodor	10^{-4}	0.5848 e-5	0.8147 e-5	0.7817 e-5	1.2424 e-5
Gödel	10^{-8}	0.5859 e-5	0.8755 e-5	0.7004 e-5	1.0227 e-5
Gougen	10^{-4}	0.5014 e-5	0.7639 e-5	0.5486 e-5	0.8395 e-5
Kleene-Dienes	10^{-4}	0.4405 e-5	0.6759 e-5	0.5805 e-5	0.8470 e-5
Łukasiewicz (Mamdani)	$\mathbf{10^{-10}}$	0.3850 e-5	0.5989 e-5	0.5881 e-5	0.8158 e-5
Reichenbach (Larsen)	10^{-5}	0.4658 e-5	0.7074 e-5	0.7453 e-5	1.1598 e-5
Rescher	10^{-5}	0.4581 e-5	0.6893 e-5	0.5408 e-5	0.8325 e-5
Zadeh	10^{-2}	1.6342 e-5	2.1129 e-5	14.006 e-5	108.25 e-5

Table 6. MSE of the predition ($I = 6$).

Fuzzy implication (relation)	DA+LS learning			SD+LS learning	
	$T_{\max}$	MSE$_{trn}$	MSE$_{tst}$	MSE$_{trn}$	MSE$_{tst}$
Fodor	10^{-5}	0.4914 e-5	0.7860 e-5	0.6086 e-5	0.9420 e-5
Gödel	10^{-2}	0.3636 e-5	0.5979 e-5	0.4896 e-5	0.8442 e-5
Gougen	10^{-2}	0.4206 e-5	0.6776 e-5	0.5536 e-5	0.8385 e-5
Kleene-Dienes	$\mathbf{10}^{-5}$	0.3017 e-5	0.4597 e-5	0.3618 e-5	0.5636 e-5
Łukasiewicz (Mamdani)	10^{-4}	0.3120 e-5	0.4670 e-5	0.6618 e-5	0.9435 e-5
Reichenbach (Larsen)	10^{-4}	0.3300 e-5	0.5468 e-5	0.7362 e-5	1.0540 e-5
Rescher	10^{-8}	0.3336 e-5	0.5390 e-5	0.5461 e-5	0.8442 e-5
Zadeh	10^{-2}	1.0462 e-5	1.7056 e-5	8.4695 e-5	10.143 e-5

plications used increases. Different methods of interpreting if-then rules lead to different results. Nevertheless, it is difficult to qualify one of them as the best. Only for Zadeh fuzzy implications we did not get satisfactory quality of neuro-fuzzy modelling. Generally, the lowest values of the prediction error were achieved using the logical interpretation of fuzzy if-then rules based on Kleene-Dienes, Łukasiewicz and Reichenbach fuzzy implications, and, hence, a conjunctive interpretation for Mamdani and Larsen fuzzy relations, too. The best prediction quality (MSE$_{trn}$ = 0.3017 e-5, MSE$_{tst}$ = 0.3017 e-5) was obtained using the deterministic annealing algorithm combined with the least-squares method for $I = 6$ and $T_{\max} = 10^{-5}$.

The problem of chaotic time series prediction generated by means of the Mackey-Glass differential delay has been studied by many authors (Cho and Wang, 1996; Chung and Duan, 2000; Czogała and Łęski, 1999; Juang and Lin, 1998; Jang and Sun, 1995). Table 7 shows the comparison of performances (root mean square error values, RMSE) of fuzzy modelling methods reported in the literature.

Table 7. Comparison of chaotic time series prediction methods.

Model	I	RMSE$_{trn}$	RMSE$_{tst}$
Juang & Lin	4	0.0180	—
Chung & Duan	20	0.0174	0.0139
Cho & Wang	23	0.0096	0.0114
Jang & Sun	16	0.0016	0.0015
ANNBFIS	15	0.0011	0.0014
ANBLIR	15	0.0011	0.0014
DA+LS	14	0.0006	0.0010

The best results were obtained for the ANNBFIS and ANBLIR neuro-fuzzy systems (RMSE$_{trn}$ = 0.0011, RMSE$_{tst}$ = 0.0014, Reichenbach implication, $I = 15$). The modification of their learning algorithms using the deterministic annealing approach enables us to improve the prediction quality (RMSE$_{trn}$ = 0.0006, RMSE$_{tst}$ = 0.0010, Reichenbach implication, $T_{\max} = 10^{-5}$) while simultaneously reducing the number of if-then rules ($I = 14$). Figures 1 and 2 show the chaotic time series (con-

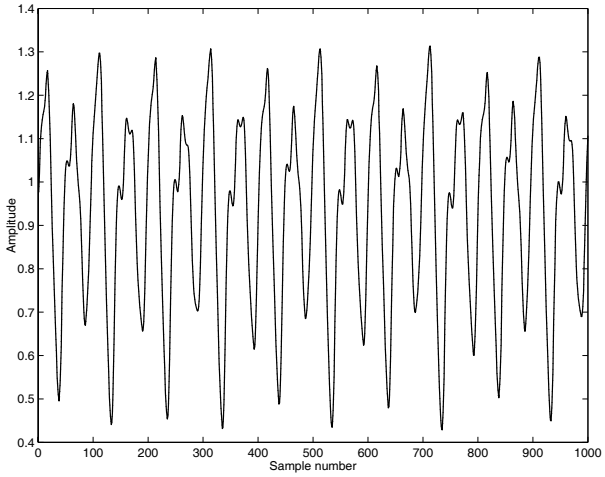

Fig. 1. Chaotic time series (continuous line) and predicted values (dotted line).

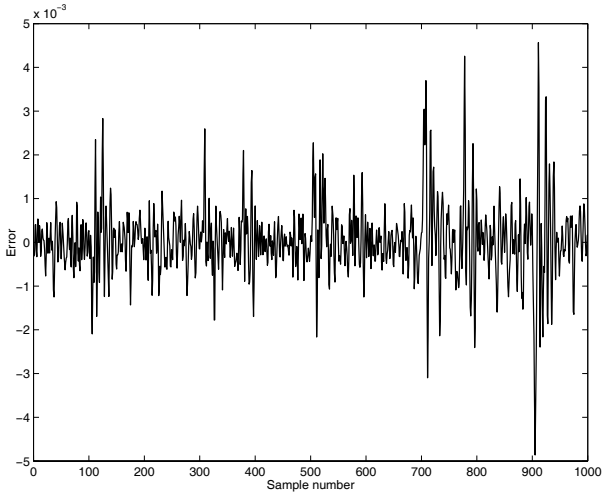

Fig. 2. Error values of chaotic time series prediction.

tinuous line), predicted values (dotted line) and the prediction error, respectively, obtained for $I = 14$, using the DA+LS learning procedure with the Reichenbach fuzzy implication.

The second numerical example—a system identification problem—is based on benchmark data originating from (Box and Jenkins, 1976). It concerns the identification of a gas oven. The input signal consists of measurements of a methane flow $x(k)$ [ft/min]. Methane is delivered into a gas oven together with air to form a mixture of gases containing carbon dioxide. The samples of the corresponding CO_2 percentage content form the output signal $y(k)$. The sampling period was 9 sec. To identify a model, the data set consisting of 290 pairs of the input vectors $[y(k-1)\dots y(k-4)x(k)\dots x(k-5)]^T$ and the output values $y(k)$ was used.

In much the same way as in to the previous example, the learning process (DA+LS) was conducted for the most frequently used fuzzy implications (Fodor, Gsödel, Gougen, Kleene-Dienes, Łukasiewicz, Reichenbach, Rescher and Zadeh). The specifications of the proposed learning algorithm and the reference procedure were defined in the same manner. However, the searching range of the initial pseudo-temperature values for the (DA+LS) method was changed to $T_{\max} \in \{10^3, 10^2, \dots, 10^{-4}\}$. The identification quality was evaluated on the basis of the mean square error values obtained for the data set (MSE). Tables (8)–(12) show the identification results. In this case, too, the deterministic annealing based method leads to higher learning quality. Only in four examples (Gougen, $I = \{2, 4\}$, Gödel, $I = 6$ and Rescher, $I = 5$) we observed an increase in the identification error compared with the reference procedure.

From the obtained results we can see that the identification error decreases as the number of fuzzy if-then rules for all implications used increases. Just as in the first numerical experiment, different methods of interpreting if-then rules lead to different learning results. All implications except for the Zadeh one lead to satisfactory identification quality. Generally, the lowest values of the identification error were achieved using the logical interpretation of fuzzy if-then rules based on the Kleene-Dienes fuzzy implication. The best learning quality (MSE = 1.5268 e-2) was obtained using the (DA+LS) method for $I = 6$ and $T_{\max} = 1$. Table 13 is provided for comparison with the RMSE results of some previous studies of the Box-Jenkins identification problem reported in the literature (Box and Jenkins, 1976; Chen *et al.*, 1998; Czogała and Łęski, 1999; Kim *et al.*, 1997; Lin and Cunningham, 1995; Pedrycz, 1984a; Sugeno and Yasukawa, 1993; Tong, 1980; Wang and Langari, 1995; Xu and Lu, 1987; Zikidis and Vasilakos 1996).

Table 8. MSE of the identification ($I = 2$).

Fuzzy implication (relation)	DA+LS learning		SD+LS learning
	$T_{\max}$	MSE	MSE
Fodor	10^0	3.5004 e-2	4.6156 e-2
Gödel	$\mathbf{10^0}$	3.4669 e-2	3.5272 e-2
Gougen	10^2	3.4828 e-2	3.5227 e-2
Kleene-Dienes	10^3	3.7239 e-2	4.6186 e-2
Łukasiewicz (Mamdani)	10^0	3.4815 e-2	4.6355 e-2
Reichenbach (Larsen)	10^0	3.4967 e-2	4.6375 e-2
Rescher	10^0	3.4683 e-2	3.4947 e-2
Zadeh	10^0	5.2824 e-2	5.5516 e-2

Table 9. MSE of the identification ($I = 3$).

Fuzzy implication (relation)	DA+LS learning		SD+LS learning
	$T_{\max}$	MSE	MSE
Fodor	10^0	3.1708 e-2	4.0168 e-2
Gödel	10^0	3.0840 e-2	3.2917 e-2
Gougen	10^0	3.0977 e-2	3.2303 e-2
Kleene-Dienes	$\mathbf{10^0}$	2.8372 e-2	3.9848 e-2
Łukasiewicz (Mamdani)	10^2	3.1805 e-2	4.0170 e-2
Reichenbach (Larsen)	10^2	3.1710 e-2	4.0170 e-2
Rescher	10^{-1}	3.0840 e-2	3.2109 e-2
Zadeh	10^0	5.4378 e-2	5.4378 e-2

Table 10. MSE of the identification ($I = 4$).

Fuzzy implication (relation)	DA+LS learning		SD+LS learning
	$T_{\max}$	MSE	MSE
Fodor	10^0	2.2425 e-2	4.0168 e-2
Gödel	10^0	2.7980 e-2	3.2917 e-2
Gougen	10^0	2.7922 e-2	3.2303 e-2
Kleene-Dienes	$\mathbf{10^2}$	2.2049 e-2	3.9848 e-2
Łukasiewicz (Mamdani)	10^0	2.2518 e-2	4.0170 e-2
Reichenbach (Larsen)	10^0	2.2374 e-2	4.0170 e-2
Rescher	10^{-1}	2.7838 e-2	3.2109 e-2
Zadeh	10^0	4.1911 e-2	5.4378 e-2

The best identification quality was obtained for the ANBLIR neuro-fuzzy system (RMSE = 0.1791, Rescher implication, $I = 3$). The learning algorithm using the deterministic annealing approach improved the identification results (RMSE = 0.1684, Kleene-Dienes implication, $T_{\max} = 1$) for the same number of if-then rules. Figures 3–5 show the evolution of the input, output (original—continuous line, modelled—dotted line) and identification error signals, respectively, obtained for $I = 6$, using the DA+LS learning procedure with the Kleene-Dienes fuzzy implication.

Table 11. MSE of the identification ($I = 5$).

Fuzzy implication (relation)	DA+LS learning		SD+LS learning
	$T_{\max}$	MSE	MSE
Fodor	10^{-3}	1.9130 e-2	2.0362 e-2
Gödel	10^{0}	2.2190 e-2	2.4808 e-2
Gougen	10^{0}	2.1980 e-2	2.2772 e-2
Kleene-Dienes	10^{0}	2.0042 e-2	2.2391 e-2
Łukasiewicz (Mamdani)	10^{-3}	1.9215 e-2	1.9845 e-2
Reichenbach (Larsen)	$\mathbf{10^{0}}$	1.8852 e-2	2.1900 e-2
Rescher	10^{0}	2.2172 e-2	2.2170 e-2
Zadeh	10^{3}	4.0008 e-2	5.2668 e-2

Table 12. MSE of the identification ($I = 6$).

Fuzzy implication (relation)	DA+LS learning		SD+LS learning
	$T_{\max}$	MSE	MSE
Fodor	10^{-1}	1.6119 e-2	2.0362 e-2
Gödel	10^{-4}	1.9361 e-2	2.4808 e-2
Gougen	10^{0}	2.1918 e-2	2.2772 e-2
Kleene-Dienes	$\mathbf{10^{-1}}$	1.5268 e-2	2.2391 e-2
Łukasiewicz (Mamdani)	10^{-1}	1.5530 e-2	1.9845 e-2
Reichenbach (Larsen)	10^{-1}	1.5946 e-2	2.1900 e-2
Rescher	10^{-3}	1.9122 e-2	2.2170 e-2
Zadeh	10^{2}	4.8292 e-2	5.2668 e-2

Table 13. Comparison of Box-Jenkins identification methods.

Model	I	Number of inputs	Number of parameters	RMSE
Tong	19	2	—	0.6848
Pedrycz	81	2	—	0.5656
Xu & Lu	25	2	—	0.5727
Box & Jenkins	—	6	10	0.4494
Sugeno & Yasukawa	6	3	96	0.4358
Chen *et al.*	3	2	—	0.2678
Lin & Cunningham	4	5	354	0.2664
Wang & Langari	2	6	110	0.2569
Zikidis&Vasilakos	2	6	—	0.2530
Kim *et al.*	2	6	110	0.2190
ANNBFIS	3	10	96	0.2004
ANBLIR	3	10	96	0.1791
DA+LS	3	10	96	0.1684

Summarizing, the combination of the deterministic annealing method and the least-squares procedure leads to an improvement in modelling results. However, it must be noted that the performance enhancement is achieved through a decrease in the computational effectiveness of the learning procedure. The computational burden of

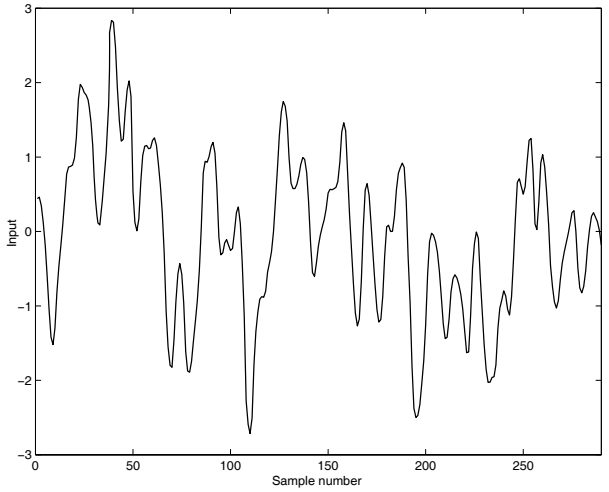

Fig. 3. Input signal for system identification data.

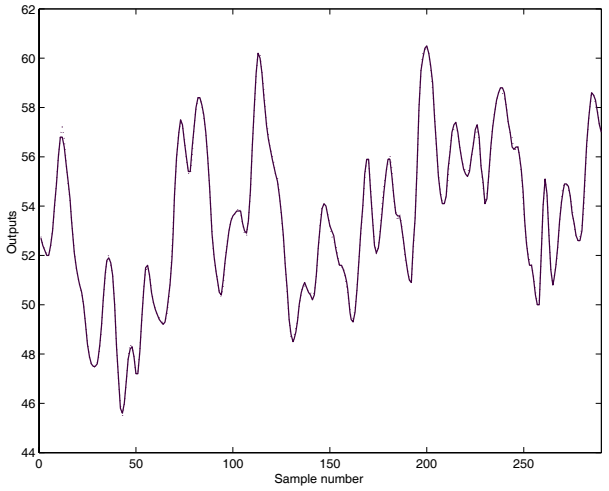

Fig. 4. Output signals for system identification data: original (continuous line) and modelled (dotted line).

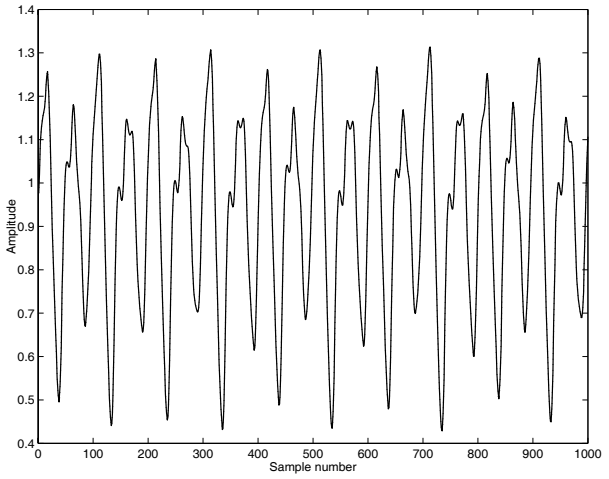

Fig. 5. Error signal for system identification data.

the proposed optimization method approximately doubles when compared with the original ANBLIR learning. Another disadvantage of the DA procedure is the necessity of an arbitrary selection of learning parameters. The value of the learning rate η was determined on the basis of a trial-and-error procedure. There is no standard method of determining the learning rate and application of the same heuristic rules as those employed in the reference procedure was unsuccessful. Other parameters exerting a strong influence on the learning results are the initial pseudo-temperature $T_{\max}$, the final pseudo-temperature $T_{\min}$, the annealing schedule parameter q and the number of iterations at each level of the pseudo-temperature $k_{\max}$.

The initial pseudo-temperature should be sufficiently high to ensure entropy maximization at the beginning of the optimization procedure. The final pseudo-temperature should be low enough to assure the minimization of the square error in the end. In our experiments a trial-and-error method was used to establish their values. We attempted to get satisfactory modelling results and the lowest possible number of iterations. A formula for the calculation of the annealing schedule parameter that guarantees finding a global minimum of the cost for the simulated annealing method was given in (German and German, 1984). However, there is no such a result for the deterministic annealing procedure. This method of computing the annealing schedule parameter leads to a significant increase in the number of steps needed to find optimal system parameters. Therefore, its value was set arbitrarily. Again, we tried to obtain acceptable modelling quality and a low number of iterations. The number of iterations at each level of the pseudo-temperature was determined on the basis of the entropy variation level (Rose, 1999). However, to ensure faster convergence, a limitation on the maximum number of criterion evaluations was added.

7. Conclusions

A neuro-fuzzy modelling method based on the deterministic annealing approach has been presented. We have described a new learning procedure of the ANBLIR neuro-fuzzy system. In the proposed method, the parameters of fuzzy sets from antecedents and consequents of if-then rules are adjusted separately by means of deterministic annealing and the least-squares method, respectively. Experiments prove the usefulness of the proposed method in the extraction of fuzzy if-then rules for the chaotic time series prediction problem. The obtained results indicate an improvement in neuro-fuzzy modelling quality compared with selected fuzzy modelling methods reported in the literature. However, performance enhancement is achieved through an increased computational load of the learning procedure. Another problem is the necessity of an arbitrary selection of learning parameters. The determination of automated methods for their selection constitutes the principal direction of future investigations. Other interesting questions for the future are sa follows:

- How does the proposed learning procedure influence the generalization ability of the neuro-fuzzy system when ε-insensitive learning is considered?
- How does the learning performance change if different clustering algorithms are used for the estimation of initial values of membership functions of antecedents?

References

Bezdek J.C. (1982): *Pattern Recognition with Fuzzy Objective Function Algorithms*. — New York: Plenum Press.

Box G.E.P. and Jenkins G.M. (1976): *Time Series Analysis. Forecasting and Control*. — San Francisco: Holden–Day.

Chen J.Q., Xi Y.G. and Zhang Z.J. (1998): *A clustering algorithm for fuzzy model identification*. — Fuzzy Sets Syst., Vol. 98, No. 3, pp. 319–329.

Cho K.B. and Wang B.H. (1996): *Radial basis function based adaptive fuzzy systems and their applications to system identification and prediction*. — Fuzzy Sets Syst., Vol. 83, No. 3, pp. 325–339.

Chung F.L. and Duan J.C. (2000): *On multistage fuzzy neural network modeling*. — IEEE Trans. Fuzzy Syst., Vol. 8, No. 2, pp. 125–142.

Czogała E. and Łęski J. (1996): *A new fuzzy inference system with moving consequents in if-then rules. Application to pattern recognition*. — Bull. Polish Acad. Sci., Vol. 45, No. 4, pp. 643–655.

Czogała E. and Łęski J. (1999): *Fuzzy and Neuro-Fuzzy Intelligent Systems*. — Heidelberg: Physica-Verlag.

Czogała E. and Łęski J. (2001): *On equivalence of approximate reasoning results using different interpretations of if-then rules*. — Fuzzy Sets Syst., Vol. 117, No. 2, pp. 279–296.

German S. and German D. (1984): *Stochastic relaxation, Gibbs distribution and the Bayesian restoration in images*. — IEEE Trans. Pattern Anal. Mach. Intell., Vol. 6, No. 9, pp. 721–741.

Jang J.S.R. (1993): *ANFIS: Adaptive-Network-Based Fuzzy Inference System*. — IEEE Trans. Syst. Man Cybern., Vol. 23, No. 3, pp. 665–685.

Jang J.S.R. and Sun C.T. (1993): *Functional equivalence between radial basis function networks and fuzzy inference systems*. — IEEE Trans. Neural Netw., Vol. 4, No. 1, pp. 156–159.

Jang J.S.R. and Sun C.T. (1995): *Neuro-fuzzy modeling and control*. — Proc. IEEE, Vol. 83, No. 3, pp. 378–406.

Jang J.S.R., Sun C.T. and Mizutani E. (1997): *Neuro-Fuzzy and Soft Computing. A Computational Approach to Learning and Machine Intelligence*. — Upper Saddle River: Prentice-Hall.

Juang C. and Lin C. (1998): *An on-line self-constructing neural fuzzy inference network and its applications*. — IEEE Trans. Fuzzy Syst., Vol. 6, No. 1, pp. 12–32.

Kim E., Park M. and Ji S. (1997): *A new approach to fuzzy modeling*. — IEEE Trans. Fuzzy Syst., Vol. 5, No. 3, pp. 328–337.

Kirkpatrick S., Gelatt C. and Vecchi M. (1983): *Optimization by simulated annealing*. — Science, Vol. 220, No. 4598, pp. 671–680.

Kosko B. (1987): *Fuzzy associative memories* In: Fuzzy Expert Systems (A. Kandel, Ed.). — Boca Raton: CRC Press.

Łęski J. (2003): *ε–insensitive learning techniques for approximate reasoning systems*. — Int. J. Comput. Cognit., Vol. 1, No. 1, pp. 21–77.

Lin Y. and Cunningham G.A. (1995): *A new approach to fuzzy-neural modeling*. – IEEE Trans. Fuzzy Syst., Vol. 3, No. 2, pp. 190–197.

Mamdani E.H. (1974): *Applications of fuzzy algorithms for control of simple dynamic plant*. — Proc. IEEE, Vol. 121, No. 12, pp. 1585–1588.

Mamdani E.H. (1976): *Advances in the linguistic synthesis of fuzzy controller*. — Int. J. Man–Mach. Stud., Vol. 8, No. 6, pp. 669–678.

Mamdani E.H. (1977): *Applications of fuzzy logic to approximate reasoning using linguistic synthesis*. — IEEE Trans. Comput., Vol. 26, No. 12, pp. 1182–1191.

Mamdani E.H. and Assilian S. (1975): *An experiment in linguistic synthesis with a fuzzy logic controller*. — Int. J. Man-Mach. Stud., Vol. 7, No. 1, pp. 1–13.

Metropolis N., Rosenbluth A.W., Rosenbluth M.N., Teller A.H. and Teller E. (1953): *Equation of state calculation by fast computing machines*. — J. Chem. Phys., Vol. 21, No. 6, pp. 1087–1092.

Mitra S. and Pal S.K. (1995): *Fuzzy multi-layer perceptron, inferencing and rule generation*. — IEEE Trans. Neural Netw., Vol. 6, No. 1, pp. 51–63.

Pedrycz W. (1984a): *An identification algorithm in fuzzy relational systems*. — Fuzzy Sets Syst., Vol. 13, No. 2, pp. 153–167.

Pedrycz W. (1984b): *Identification in fuzzy systems*. — IEEE Trans. Syst. Man Cybern., Vol. 14, No. 2, pp. 361–366.

Rao A.V. and Rose K. (1999): *A deterministic annealing approach for parsimonious design of piecewise regression models*. — IEEE Trans. Pattern Anal. Mach. Intell., Vol. 21, No. 2, pp. 159–173.

Rao A.V., Miller D., Rose K. and Gersho A. (1997): *Mixture of experts regression modeling by deterministic annealing*. — IEEE Trans. Signal Process., Vol. 45, No. 11, pp. 2811–2820.

Rose K. (1991): *Deterministic Annealing, Clustering and Optimization*. — Ph.D. Thesis, California Inst. Technol, Pasadena.

Rose K. (1998): *Deterministic annealing for clustering, compression, classification, regression and related optimization problems*. — Proc. IEEE, Vol. 86, No. 11, pp. 2210–2239.

Rose K. (1999): *A deterministic annealing approach for parsimonious design of piecewise regression models*. — IEEE Trans. Pattern Anal. Mach. Intell., Vol. 21, No. 2, pp. 159–173.

Rutkowska D. (2001): *Neuro–Fuzzy Architectures and Hybrid Learning*. — Heidelberg: Physica–Verlag.

Schuster H.G. (1984): *Deterministic Chaos*. — Weinheim: VCH Verlagsgesellschaft.

Sugeno M. and Kang G.T. (1988): *Structure identification of fuzzy model*. — Fuzzy Sets Syst., Vol. 28, No. 1, pp. 15–33.

Sugeno M. and Yasukawa T. (1993): *A fuzzy-logic based approach to qualitative modeling*. — IEEE Trans. Fuzzy Syst., Vol. 1, No. 1, pp. 7–31.

Tong R.M. (1980): *The evaluation of fuzzy models derived from experimental data*. — Fuzzy Sets Syst., Vol. 4, No. 13, pp. 1–12.

Wang L. and Langari R. (1995): *Building Sugeno-type models using fuzzy discretization and orthogonal parameter estimation techniques*. — IEEE Trans. Fuzzy Syst., Vol. 3, No. 4, pp. 454–458.

Xie X.L. and Beni G. (1991): *A validity measure for fuzzy clustering*. — IEEE Trans. Pattern Anal. Mach. Intell., Vol. 13, No. 8, pp. 841–847.

Xu C.W. and Lu Y.Z. (1987): *Fuzzy model identification and self-learning for dynamic systems*. — IEEE Trans. Syst. Man Cybern., Vol. 17, No. 4, pp. 683–689.

Yager R.R. and Filev D.P. (1984): *Essentials of Fuzzy Modeling and Control*. — New York: Wiley.

Yen J., Wang L. and Gillespie C.W. (1998): *Improving the interpretability of TSK fuzzy models by combining global learning and local learning*. — IEEE Trans. Fuzzy Syst., Vol. 6, No. 4, pp. 530–537.

Zadeh L.A. (1965): *Fuzzy sets*. — Inf. Contr., Vol. 8, No. 3, pp. 338–353.

Zadeh L.A. (1971): *Towards a theory of fuzzy systems,* In: Aspects of Network and System Theory (R.E. Kalman and N. DeClaris, Ed.). — New York: Holt, Rinehart & Winston.

Zadeh L.A. (1973): *Outline of a new approach to the analysis of complex systems and decision processes*. — IEEE Trans. Syst. Man Cybern., Vol. 3, No. 1, pp. 28–44.

Zikidis K.C. and Vasilakos A.V. (1996): *ASAFES2: A novel, neuro-fuzzy architecture for fuzzy computing, based on functional reasoning*. — Fuzzy Sets Syst., Vol. 83, No. 1, pp. 63–68.

Received: 24 March 2005
Revised: 12 July 2005
Re-revised: 4 August 2005