BULLETIN OF THE POLISH ACADEMY OF SCIENCES
TECHNICAL SCIENCES
Vol. 52, No. 1, 2004

A new method for system modelling and pattern classification

L. RUTKOWSKI*

Technical University of Częstochowa, Department of Computer Engineering, 36 Armii Krajowej Ave., 42-200 Częstochowa, Poland

Abstract. In this paper we present a new class of neuro-fuzzy systems designed for system modelling and pattern classification. Our approach is characterized by automatic determination of fuzzy inference in the process of learning. Moreover, we introduce several flexibility concepts in the design of neuro-fuzzy systems. The method presented in the paper is characterized by high accuracy which outperforms previous techniques applied for system modelling and pattern classification.

Keywords: system modelling, pattern classification, neuro-fuzzy systems, flexible parameters, generalized triangular norms.

1. Introduction

Fuzzy sets and fuzzy logic, introduced in 1965 by Lotfi Zadeh [1], have been used in a wide range of problems, e.g. process control, image processing, pattern recognition and classification, management, economics and decision making. Specific applications include washing-machine automation, camcorder focusing, TV colour tuning, automobile transmissions and subway operations [2]. We have also been witnessing a rapid development in the area of neural networks (see e.g. [3, 4]). Both fuzzy systems [5, 6] and neural networks, along with probabilistic methods [7, 8], evolutionary algorithms [9], rough sets [10, 11] and uncertain variables [12, 13], constitute a consortium of soft computing techniques [8, 14, 15]. These techniques are often used in combination. For example, fuzzy inference systems are frequently converted into connectionist structures called neuro-fuzzy systems which exhibit advantages of neural networks and fuzzy systems. In literature various neuro fuzzy systems have been developed (see e.g. [16–32]). They combine the natural language description of fuzzy systems and the learning properties of neural networks.

In this paper we present a new class of neuro-fuzzy systems designed for system modelling and pattern classification. Our approach is characterized by automatic determination of fuzzy inference in the process of learning. Moreover, we introduce several flexibility concepts in the design of neuro-fuzzy systems. The method presented in the paper allows to perfectly represent patterns encoded in the data. Consequently, we achieve high accuracy of our method which outperforms previous techniques applied for system modelling and pattern classification.

2. Fuzzy reasoning and fuzzy implications

In this Section we present the idea of fuzzy reasoning with various fuzzy implications which will be useful for construction of neuro-fuzzy systems developed in the next sections. The basic rule of inference in classical logic is modus ponens. The compositional rule of inference describes a composition of a fuzzy set and a fuzzy relation. Fuzzy rule

$$\text{IF} \quad x \quad \text{is} \quad A \quad \text{THEN} \quad y \quad \text{is} \quad B \tag{1}$$

is represented by a fuzzy relation R. Having given input linguistic value A', we can infer an output fuzzy set B' by the composition of the fuzzy set A' and the relation R. The generalized modus ponens is the extension of the conventional modus ponens tautology, to allow partial fulfilment of the premises:

Premise	x	is	A'					
Implication	IF	x	is	A	THEN	y	is	B
Conclusion	y	is	B'					

where A, B, A', B' are fuzzy sets and x, y are linguistic variables. Applying the compositional rule of inference [5], we get

$$B' = A' \circ R = A' \circ (A \to B) \tag{2}$$

and

$$\mu_{B'}(y) = \mu_{A' \circ R}(y) = \sup_{\mathbf{x} \in \mathbb{X}} \left\{ \mu_{A'}(x) \overset{T}{*} \mu_R(x,y) \right\} \tag{3}$$

The problem is to determine the membership function of the fuzzy relation described by

$$\mu_R(x,y) = \mu_{A \to B}(x,y) \tag{4}$$

based on the knowledge of $\mu_A(x)$ and $\mu_B(y)$. We denote

$$\mu_{A \to B}(x,y) = I(\mu_A(x), \mu_B(y)) \tag{5}$$

where $I(\cdot)$ is a fuzzy implication given in Definition 1 (see Fodor [33]).

Definition 1. A fuzzy implication is a function $I : [0,1]^2 \to [0,1]$ satisfying the following conditions:

(I1) if $a_1 \leqslant a_3$, then $I(a_1, a_2) \geqslant I(a_3, a_2)$, for all $a_1, a_2, a_3 \in [0,1]$,

(I2) if $a_2 \leqslant a_3$, then $I(a_1, a_2) \leqslant I(a_1, a_3)$, for all $a_1, a_2, a_3 \in [0,1]$,

(I3) $I(0, a_2) = 1$, for all $a_2 \in [0,1]$ (falsity implies anything),

*e-meil: lrutko@kik.pcz.czest.pl

(I4) $I(a_1, 1) = 1$, for all $a_1 \in [0,1]$ (anything implies tautology),
(I5) $(1,0) = 0$ (Booleanity).

Selected fuzzy implications satisfying all or some of the above conditions are listed in Table 1.

Table 1
Fuzzy implications

No	Name	Implication $I(a,b)$
1	Kleene-Dienes (binary)	$\max\{1-a, b\}$
2	Łukasiewicz	$\min\{1, 1-a+b\}$
3	Reichenbach	$1-a+a\cdot b$
4	Fodor	$\begin{cases} 1 & \text{if } a \leqslant b \\ \max\{1-a,b\} & \text{if } a > b \end{cases}$
5	Rescher	$\begin{cases} 1 & \text{if } a \leqslant b \\ 0 & \text{if } a > b \end{cases}$
6	Goguen	$\begin{cases} 1 & \text{if } a = 0 \\ \min\left\{1, \frac{b}{a}\right\} & \text{if } a > 0 \end{cases}$
7	Gödel	$\begin{cases} 1 & \text{if } a \leqslant b \\ b & \text{if } a > b \end{cases}$
8	Yager	$\begin{cases} 1 & \text{if } a = 0 \\ b^a & \text{if } a > 0 \end{cases}$
9	Zadeh	$\max\{\min\{a,b\}, 1-a\}$
10	Willmott	$\min\left\{ \begin{array}{c} \max\{1-a,b\}, \\ \max\{a, 1-b, \min\{1-a,b\}\} \end{array} \right\}$
11	Dubois-Prade	$\begin{cases} 1-a & \text{if } b = 0 \\ b & \text{if } a = 1 \\ 1 & \text{if } otherwise \end{cases}$

In this table, implications 1–4 are examples of an S-implication associated with a t-conorm

$$I(a,b) = S\{1-a, b\}. \tag{6}$$

Neuro-fuzzy systems based on fuzzy implications given in Table 1 are called logical systems. Implications 6 and 7 belong to a group of R-implications associated with the t-norm T and given by

$$I(a,b) = \sup_z \{z | T\{a,z\} \leqslant b\}, \quad a,b \in [0,1]. \tag{7}$$

The Zadeh implication belongs to a group of Q-implications given by

$$I(a,b) = S\{N(a), T\{a,b\}\}, \quad a,b \in [0,1]. \tag{8}$$

It is easy to verify that S-implications and R-implications satisfy all the conditions of Definition 1. However, the Zadeh implication violates conditions I1 and I4, whereas the Willmott implication violates conditions I1, I2, I3 and I4. In practice we frequently use Mamdani-type operators given by

$$I(a,b) = \min\{a,b\}, \quad a,b \in [0,1] \tag{9}$$
$$I(a,b) = a \cdot b, \quad a,b \in [0,1] \tag{10}$$

or generally

$$I(a,b) = T\{a,b\}, \quad a,b \in [0,1]. \tag{11}$$

It should be noted that operators (9)–(11) do not satisfy conditions of Definition 1. Operators (9)–(11) are called "engineering implications" (see [6]).

3. Description of fuzzy inference systems

In this paper, we consider multi-input-single-output fuzzy system mapping $\mathbb{X} \to \mathbb{Y}$, where $\mathbb{X} \subset \mathbb{R}^n$ and $\mathbb{Y} \subset \mathbb{R}$. The system (see Fig. 1) is composed of a fuzzifier, a fuzzy rule base, a fuzzy inference engine and a defuzzifier. The fuzzifier performs a mapping from the observed crisp input space $\mathbb{X} \subset \mathbb{R}^n$ to a fuzzy set defined in $\mathbb{X}$. The most commonly used fuzzifier is the singleton fuzzifier which maps $\bar{\mathbf{x}} = [\bar{x}_1, \ldots, \bar{x}_n] \in \mathbb{X}$ into a fuzzy set $A' \subseteq \mathbb{X}$ characterized by the membership function

$$\mu_{A'}(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{x} = \bar{\mathbf{x}} \\ 0 & \text{if } \mathbf{x} \neq \bar{\mathbf{x}} \end{cases}. \tag{12}$$

The fuzzy rule base consists of a collection of N fuzzy IF-THEN rules, aggregated by the disjunction or the conjunction, in the form

$$R^{(k)} : \left\{ \begin{array}{llll} \text{IF} & x_1 & is & A_1^k \quad AND \\ & x_2 & is & A_2^k \quad AND \quad \ldots \\ & x_n & is & A_n^k \\ THEN & y & is & B^k \end{array} \right. \tag{13}$$

or

$$R^{(k)} : \ \text{IF} \ \ \mathbf{x} \ \ \text{is} \ \ A^k \ \ \text{THEN} \ \ y \ \ \text{is} \ \ B^k \tag{14}$$

where $\mathbf{x} = [x_1, \ldots, x_n] \in \mathbb{X}$, $y \in \mathbb{Y}$, $A^k = A_1^k \times A_2^k \times \ldots \times A_n^k$, $A_1^k, A_2^k, \ldots, A_n^k$ are fuzzy sets characterized by membership functions $\mu_{A_i^k}(x_i)$, $i = 1, \ldots, n$, $k = 1, \ldots, N$, whereas B^k are fuzzy sets characterized by membership functions $\mu_{B^k}(y)$, $k = 1, \ldots, N$. The firing strength of the k-th rule, $k = 1, \ldots, N$, is defined by

$$\tau_k(\bar{\mathbf{x}}) = \mathop{T}_{i=1}^{n} \left\{ \mu_{A_i^k}(\bar{x}_i) \right\} = \mu_{A^k}(\bar{\mathbf{x}}). \tag{15}$$

In the paper notations τ_k and $\mu_{A^k}(\bar{\mathbf{x}})$ will be used interchangeably.

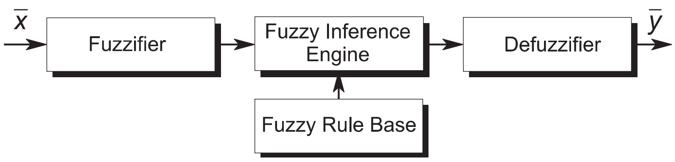

Fig. 1. Fuzzy inference system

The fuzzy inference engine determines the mapping from the fuzzy sets in the input space $\mathbb{X}$ to the fuzzy sets in the output space $\mathbb{Y}$. Each of N rules (14) determines a fuzzy set $\bar{B}^k \subseteq \mathbb{Y}$ given by the compositional rule of inference

$$\bar{B}^k = A' \circ \left(A^k \to B^k\right) \tag{16}$$

where $A^k = A_1^k \times A_2^k \times \ldots \times A_n^k$. Fuzzy sets $\bar{B}^k$ are characterized by membership functions expressed by the sup-star composition

$$\mu_{\bar{B}^k}(y) = \sup_{\mathbf{x} \in \mathbf{X}} \left\{ \mu_{A'}(\mathbf{x}) * \mu_{A_1^k \times \ldots \times A_n^k \to B^k}(\mathbf{x}, y) \right\} \tag{17}$$

where $*$ can be any operator in the class of t-norms. It is easily seen that for a crisp input $\mathbf{x} = [x_1, \ldots, x_n] \in \mathbf{X}$, i.e., the singleton fuzzifier (12), formula (17) becomes

$$\begin{aligned}\mu_{\bar{B}^k}(y) &= \mu_{A_1^k\times\ldots\times A_n^k\to B^k}(\bar{\mathbf{x}},y)\\ &= \mu_{A^k\to B^k}(\bar{\mathbf{x}},y)\\ &= I\left(\mu_{A^k}(\bar{\mathbf{x}}),\mu_{B^k}(y)\right)\end{aligned} \tag{18}$$

where $I(\cdot)$ is an "engineering implication" given by (11) or fuzzy implication given in Table 1. More precisely,

$$I(\mu_{A^k}(\bar{\mathbf{x}}),\mu_{B^k}(y)) = \begin{cases} I_{eng}(\mu_{A^k}(\bar{\mathbf{x}}),\mu_{B^k}(y)) \\ \text{ for the } \textit{Mamdani approach} \\ I_{fuzzy}(\mu_{A^k}(\bar{\mathbf{x}}),\mu_{B^k}(y)) \\ \text{ for the } \textit{logical approach}\end{cases}. \tag{19}$$

As we mentioned in Section 2 (see formula (11)), in the Mamdani approach

$$I_{eng}\left(\mu_{A^k}(\bar{\mathbf{x}}),\mu_{B^k}(y)\right) = T\left\{\mu_{A^k}(\bar{\mathbf{x}}),\mu_{B^k}(y)\right\}. \tag{20}$$

In the logical approach we apply fuzzy implications listed in Table 1. The Kleene-Dienes, Łukasiewicz, Reichenbach and Fodor implications are examples of the S-implication given by

$$I_{fuzzy}\left(\mu_{A^k}(\bar{\mathbf{x}}),\mu_{B^k}(y)\right) = S\left\{1-\mu_{A^k}(\bar{\mathbf{x}}),\mu_{B^k}(y)\right\}. \tag{21}$$

Obviously, an S-implication can be generated by various t-conorms. The aggregation operator, applied in order to obtain the fuzzy set B' based on fuzzy sets $\bar{B}^k$, is the t-norm or t-conorm operator, depending on the type of fuzzy implication. In Table 2 we describe connectives in the Mamdani approach and logical approach. In case of the Mamdani approach, the aggregation is carried out by

$$B' = \bigcup_{k=1}^{N} \bar{B}^k. \tag{22}$$

The membership function of B' is computed by the use of a t-conorm, that is

$$\mu_{B'}(y) = \mathop{S}_{k=1}^{N} \mu_{\bar{B}^k}(y). \tag{23}$$

When we use the logical model, the aggregation is carried out by

$$\bar{B} = \bigcap_{k=1}^{N} \bar{B}^k. \tag{24}$$

The membership function of B' is determined by the use of a t-norm, i.e.

$$\mu_{B'}(y) = \mathop{T}_{k=1}^{N} \left\{\mu_{\bar{B}^k}(y)\right\}. \tag{25}$$

As a result of the fuzzy reasoning we obtain the fuzzy set B'.

Table 2
Operations in fuzzy inference

System type	Aggregation of antecedents	Implication	Aggregation of rules
Mamdani	t-norm	"engineering" – t-norm	t-conorm
Logical	t-norm	logical	t-norm

The defuzzifier performs a mapping from the fuzzy set B' to a crisp point $\bar{y}$ in $\mathbf{Y}\subset\mathbf{R}$. The COA (center of area) method is defined by the following formula

$$\bar{y} = \frac{\int_{\mathbb{Y}} y\cdot\mu_{B'}(y)dy}{\int_{\mathbb{Y}} \mu_{B'}(y)dy} \tag{26}$$

or by

$$\bar{y} = \frac{\sum_{r=1}^{N} \bar{y}^r\cdot\mu_{B'}(\bar{y}^r)}{\sum_{r=1}^{N}\mu_{B'}(\bar{y}^r)} \tag{27}$$

in the discrete form, where $\bar{y}^r$ are centers of the membership functions $\mu_{B^r}(y)$, i.e., for $r=1,\ldots,N$

$$\mu_{B^r}(\bar{y}^r) = \max_{y\in\mathbb{Y}}\left\{\mu_{B^r}(y)\right\} \tag{28}$$

4. Neuro-fuzzy structures for system modelling

In this section we generalize the Mamdani-type and the logical-type approach, described in Section 3, and derive neuro-fuzzy structures for system modelling.

4.1. Mamdani-type neuro-fuzzy systems. In this approach, function $I(\cdot)$ given by (19) is a t-norm (e.g. minimum or algebraic), i.e.

$$I\left(\mu_{A^k}(\bar{\mathbf{x}}),\mu_{B^k}(\bar{y}^r)\right) = T\left\{\mu_{A^k}(\bar{\mathbf{x}}),\mu_{B^k}(\bar{y}^r)\right\}. \tag{29}$$

The aggregated output fuzzy set $B'\subseteq\mathbb{Y}$ is given by

$$\begin{aligned}\mu_{B'}(\bar{y}^r) &= \mathop{S}_{k=1}^{N}\left\{\mu_{\bar{B}^k}(\bar{y}^r)\right\}\\ &= \mathop{S}_{k=1}^{N}\left\{T\left\{\mu_{A^k}(\bar{\mathbf{x}}),\mu_{B^k}(\bar{y}^r)\right\}\right\}.\end{aligned} \tag{30}$$

Consequently, formula (27) takes the form

$$\bar{y} = \frac{\sum_{r=1}^{N}\bar{y}^r\cdot\mathop{S}_{k=1}^{N}\left\{T\left\{\mathop{T}_{i=1}^{n}\left\{\mu_{A_i^k}(\bar{x}_i)\right\},\mu_{B^k}(\bar{y}^r)\right\}\right\}}{\sum_{r=1}^{N}\mathop{S}_{k=1}^{N}\left\{T\left\{\mathop{T}_{i=1}^{n}\left\{\mu_{A_i^k}(\bar{x}_i)\right\},\mu_{B^k}(\bar{y}^r)\right\}\right\}}. \tag{31}$$

Obviously, the t-norms used to connect the antecedents in the rules and in the "engineering implication" do not have to be the same. Besides, they can be chosen as differentiable functions as e.g. Yager families (see Section 6.2).

4.2. Logical-type neuro-fuzzy systems. In this approach, function $I(\cdot)$ given by (19) is a fuzzy implication (see Table 1), i.e.

$$I\left(\mu_{A^k}(\bar{\mathbf{x}}),\mu_{B^k}(\bar{y}^r)\right) = I_{fuzzy}\left(\mu_{A^k}(\bar{\mathbf{x}}),\mu_{B^k}(\bar{y}^r)\right). \tag{32}$$

The aggregated output fuzzy set $B'\subseteq\mathbb{Y}$ is given by

$$\begin{aligned}\mu_{B'}(\bar{y}^r) &= \mathop{T}_{k=1}^{N}\left\{\mu_{\bar{B}^k}(\bar{y}^r)\right\}\\ &= \mathop{T}_{k=1}^{N}\left\{I_{fuzzy}\left(\mu_{A^k}(\bar{\mathbf{x}}),\mu_{B^k}(\bar{y}^r)\right)\right\}\end{aligned} \tag{33}$$

and formula (27) becomes

$$\bar{y} = \frac{\sum_{r=1}^{N} \bar{y}^r \cdot \mathop{T}_{k=1}^{N} \left\{ I_{fuzzy} \left(\mathop{T}_{i=1}^{n} \left\{ \mu_{A_i^k}(\bar{x}_i) \right\}, \mu_{B^k}(\bar{y}^r) \right) \right\}}{\sum_{r=1}^{N} \mathop{T}_{k=1}^{N} \left\{ I_{fuzzy} \left(\mathop{T}_{i=1}^{n} \left\{ \mu_{A_i^k}(\bar{x}_i) \right\}, \mu_{B^k}(\bar{y}^r) \right) \right\}}. \tag{34}$$

Now, we generalize both approaches described in points a) and b) and propose a general architecture of fuzzy systems. It is easily seen that systems (31) and (34) can be presented in the form

$$\bar{y} = f(\bar{\mathbf{x}}) = \frac{\sum_{r=1}^{N} \bar{y}^r \cdot agr_r(\bar{\mathbf{x}}, \bar{y}^r)}{\sum_{r=1}^{N} agr_r(\bar{\mathbf{x}}, \bar{y}^r)} \tag{35}$$

where

$$\mathrm{agr}_r(\bar{\mathbf{x}}, \bar{y}^r) = \begin{cases} \mathop{S}_{k=1}^{N} \{ I_{k,r}(\bar{\mathbf{x}}, \bar{y}^r) \} & \text{for } \textit{the Mamdani approach} \\ \mathop{T}_{k=1}^{N} \{ I_{k,r}(\bar{\mathbf{x}}, \bar{y}^r) \} & \text{for } \textit{the logical approach} \end{cases} \tag{36}$$

$$I_{k,r}(\bar{\mathbf{x}}, \bar{y}^r) = \begin{cases} T\{\tau_k(\bar{\mathbf{x}}), \mu_{B^k}(\bar{y}^r)\} \\ \quad \text{for } \textit{the Mamdani approach} \\ I_{fuzzy}(\tau_k(\bar{\mathbf{x}}), \mu_{B^k}(\bar{y}^r)) \\ \quad \text{for } \textit{the logical approach} \end{cases}. \tag{37}$$

Moreover, the firing strength of rules has already been defined by

$$\tau_k(\bar{\mathbf{x}}) = \mathop{T}_{i=1}^{n} \left\{ \mu_{A_i^k}(\bar{x}_i) \right\}.$$

The general architecture of system (35) is depicted in Fig. 2.

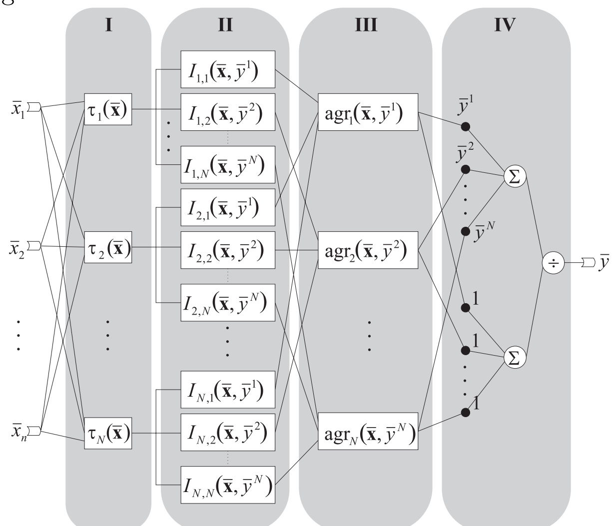

Fig. 2. General architecture of fuzzy systems studied in the paper (flexible and nonflexible)

Remark 1. If an S-implication is used, i.e.

$$I_{k,r}(\bar{\mathbf{x}}, \bar{y}^r) = S\{N(\mu_{A^k}(\bar{\mathbf{x}})), \mu_{B^k}(\bar{y}^r)\} \tag{38}$$

then the aggregated output fuzzy set $B' \subseteq \mathbf{Y}$ is given by

$$\begin{aligned} \mu_{B'}(\bar{y}^r) &= \mathop{T}_{k=1}^{N} \{\mu_{\bar{B}^k}(\bar{y}^r)\} \\ &= \mathop{T}_{k=1}^{N} \{S\{N(\mu_{A^k}(\bar{\mathbf{x}})), \mu_{B^k}(\bar{y}^r)\}\}. \end{aligned} \tag{39}$$

Consequently, formula (35) becomes

$$\bar{y} = \frac{\sum_{r=1}^{N} \bar{y}^r \cdot \mathop{T}_{k=1}^{N} \left\{ S\left\{ N\left(\mathop{T}_{i=1}^{n} \{\mu_{A_i^k}(\bar{x}_i)\} \right), \mu_{B^k}(\bar{y}^r) \right\} \right\}}{\sum_{r=1}^{N} \mathop{T}_{k=1}^{N} \left\{ S\left\{ N\left(\mathop{T}_{i=1}^{n} \{\mu_{A_i^k}(\bar{x}_i)\} \right), \mu_{B^k}(\bar{y}^r) \right\} \right\}}. \tag{40}$$

Remark 2. It should be emphasized that formula (35) and the scheme depicted in Fig. 2 are applicable to all the systems, flexible and nonflexible, studied in this paper with different definitions of $\mathrm{agr}_r(\bar{\mathbf{x}}, \bar{y}^r)$ and $I_{k,r}(\bar{\mathbf{x}}, \bar{y}^r)$. The nonflexible systems are described by (35), (36), (37) and (15), whereas the flexible systems by (35) and $\mathrm{agr}_r(\bar{\mathbf{x}}, \bar{y}^r)$, $I_{k,r}(\bar{\mathbf{x}}, \bar{y}^r)$, $\tau_k(\bar{\mathbf{x}})$ defined in Section 7.

5. Neuro-fuzzy structures for pattern classification

We will explain how to modify formula (35) and structure depicted in Fig. 2 to solve multi-classification problems. Let $[x_1, ..., x_n]$ be the vector of features of an object ν. Let $\Omega = \{\omega_1, ..., \omega_M\}$ be a set of classes. The knowledge is represented by a set of N rules in the form

$$R^{(k)}: \begin{cases} IF & x_1 & is & A_1^k & AND \\ & x_2 & is & A_2^k & AND & \dots \\ & x_n & is & A_n^k \\ THEN & & \nu \in \omega_1 \left(z_1^k\right), \\ & & \nu \in \omega_2 \left(z_2^k\right), & & \dots, \\ & & \nu \in \omega_M \left(z_M^k\right), \end{cases} \tag{41}$$

where z_j^k, $j = 1, ..., M$, $k = 1, ..., N$, are interpreted as "support" for class ω_j given by rule $R^{(k)}$. We will now redefine description (35). Let us introduce vector $\mathbf{z} = [z_1, ..., z_M]$, where z_j, $j = 1, ..., M$, is the "support" for class ω_j given by all M rules. We can scale the support values to the interval $[0, 1]$, so that z_j is the membership degree of an object ν to class ω_j according to all M rules.

The rules are represented by

$$R^{(k)}: \begin{cases} IF & x_1 & is & A_1^k & AND \\ & x_2 & is & A_2^k & AND & \dots \\ & x_n & is & A_n^k \\ THEN & z_1 & is & B_1^k & AND \\ & z_2 & is & B_2^k & AND & \dots \\ & z_M & is & B_M^k \end{cases} \tag{42}$$

and formula (35) adopted for classification takes the form

$$\bar{z}_j = \frac{\sum_{r=1}^{N} \bar{z}_j^r agr_r\left(\bar{\mathbf{x}}, \bar{z}_j^r\right)}{\sum_{r=1}^{N} agr_r\left(\bar{\mathbf{x}}, \bar{z}_j^r\right)} \tag{43}$$

where $\bar{z}_j^r$ are centers of fuzzy sets B_j^r, $j = 1, ..., M$, $r = 1, .., N$.

6. Flexibility in fuzzy systems

In the previous works on neuro-fuzzy systems it was assumed that fuzzy inference (Mamdani or logical) was fixed in advance and during the design process only the parameters of the membership functions were optimized to meet the design requirements. On the other hand it is well known that introducing additional parameters to be tuned in the system usually improves its performance. The system is able to better represent the patterns encoded in the data. In this section we present various concepts leading to the designing flexible neuro-fuzzy systems, characterized by many parameters determined in the process of learning.

6.1 Weighted triangular norms. In this paper we propose the weighted t-norm

$$T^* \{a_1, \ldots, a_n; w_1^\tau, \ldots, w_n^\tau\} = \mathop{T}_{i=1}^{n} \{1 - w_i^\tau (1 - a_i)\} \quad (44)$$

to connect the antecedents in each rule, $k = 1, \ldots, N$, and the weighted t-norm and t-conorm:

$$T^* \{a_1, \ldots, a_N; w_1^{\text{agr}}, \ldots, w_N^{\text{agr}}\} = \mathop{T}_{k=1}^{N} \{1 - w_k^{\text{agr}} (1 - a_k)\} \quad (45)$$

$$S^* \{a_1, \ldots, a_N; w_1^{\text{agr}}, \ldots, w_N^{\text{agr}}\} = \mathop{S}_{k=1}^{N} \{w_k^{\text{agr}} a_k\} \quad (46)$$

to aggregate the individual rules in the logical and Mamdani models, respectively. It is easily seen that formula (44) can be applied to the evaluation of an importance of input linguistic values, and the weighted t-norm (45) or t-conorm (46) to a selection of important rules. The results will be depicted in the form of diagrams.

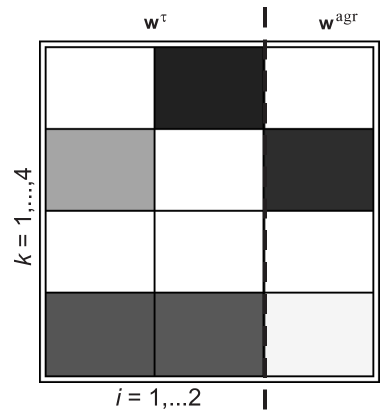

Fig. 3. Exemplary weights representation in a fuzzy system with four rules and two inputs (dark areas correspond to low values of weights and vice versa)

In Fig. 3 we show an example of a diagram for a fuzzy system having four rules ($N = 4$) and two inputs ($n = 2$) described by:

$$R^1 : \left[\text{IF} x_1 \text{is} A_1^1 \left(w_{1,1}^\tau\right) \text{AND} x_2 \text{is} A_2^1 \left(w_{2,1}^\tau\right) \text{THEN} y \text{is} B^1\right] w_1^{\text{agr}}$$

$$R^2 : \left[\text{IF} x_1 \text{is} A_1^2 \left(w_{1,2}^\tau\right) \text{AND} x_2 \text{is} A_2^2 \left(w_{2,2}^\tau\right) \text{THEN} y \text{is} B^2\right] w_2^{\text{agr}}$$

$$R^3 : \left[\text{IF} x_1 \text{is} A_1^3 \left(w_{1,3}^\tau\right) \text{AND} x_2 \text{is} A_2^3 \left(w_{2,3}^\tau\right) \text{THEN} y \text{is} B^3\right] w_3^{\text{agr}}$$

$$R^4 : \left[\text{IF} x_1 \text{is} A_1^4 \left(w_{1,4}^\tau\right) \text{AND} x_2 \text{is} A_2^4 \left(w_{2,4}^\tau\right) \text{THEN} y \text{is} B^4\right] w_4^{\text{agr}}.$$

Observe that the third rule is "weaker" than the others and the linguistic value A_2^4 corresponds to a low value of $w_{2,4}^\tau$. The designing of neuro-fuzzy systems should be a compromise between the accuracy of the model and its transparency.

Example 1. (An example of algebraic triangular norms with weighted arguments) The algebraic triangular norms with weighted arguments are based on classical algebraic triangular norms (see e.g. [6]). The algebraic t-norm with weighted arguments is described as follows

$$T^* \{a_1, a_2; w_1, w_2\} = (1 - w_1 (1 - a_1)) (1 - w_2 (1 - a_2)) . \quad (47)$$

The 3D plots of function (47) are depicted in Fig. 4. The algebraic t-conorm with weighted arguments is given by

$$S^* \{a_1, a_2; w_1, w_2\} = w_1 a_1 + w_2 a_2 - w_1 a_1 w_2 a_2. \quad (48)$$

The 3D plots of function (48) are presented in Fig. 5.

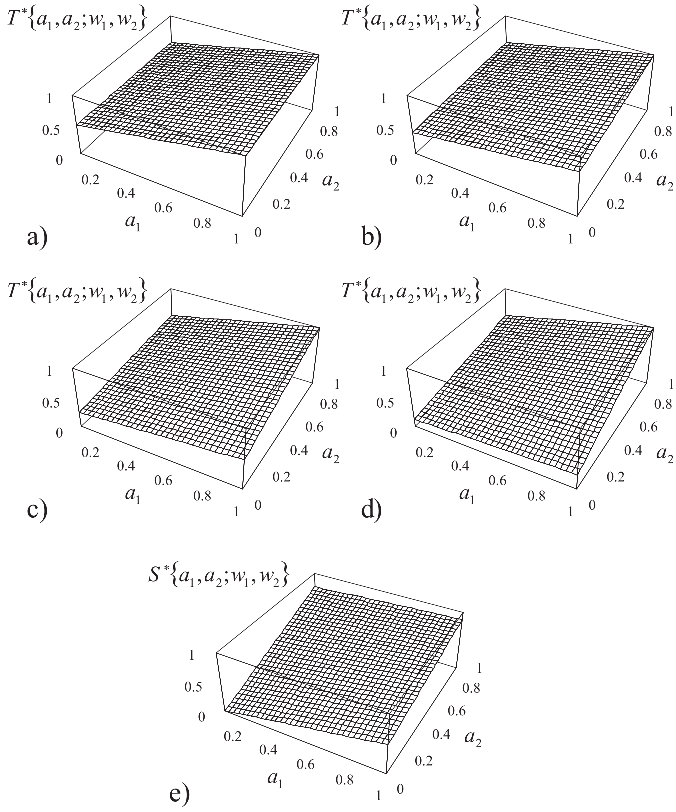

Fig. 4. 3D plots of function (47) for $w_1 = 0.50$ and a) $w_2 = 0.00$, b) $w_2 = 0.25$, c) $w_2 = 0.50$, d) $w_2 = 0.75$, e) $w_2 = 1.00$

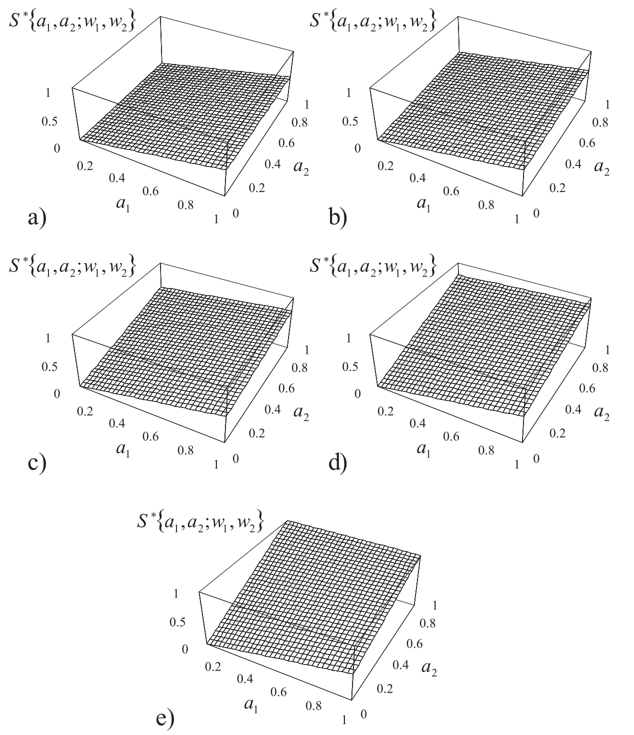

Fig. 5. 3D plots of function (48) for $w_1 = 0.50$ and a) $w_2 = 0.00$, b) $w_2 = 0.25$, c) $w_2 = 0.50$, d) $w_2 = 0.75$, e) $w_2 = 1.00$

6.2. Parameterized triangular norms. It is well known that any construction of fuzzy systems relies on triangular norms. Most fuzzy inference structures studied in literature employ the standard triangular norms as min/max or product. There is only a little knowledge within the engineering community about the so-called parameterized families of t-norm and t-conorms. They include the Dombi, Hamacher, Yager, Frank, Weber I, Weber II, Dubois-Prade and other families [34]. We use notation $\overleftrightarrow{T}\{a_1, a_2, \ldots, a_n; p\}$ and $\overleftrightarrow{S}\{a_1, a_2, \ldots, a_n; p\}$ for parameterized triangular norms. The hyperplanes corresponding to them can be adjusted in the process of learning of parameter p. As an example we present the Yager family of parameterized triangular norms. The t-norm and t-conorm are given as follows:

The Yager t-norm

$$\overleftrightarrow{T}\{\mathbf{a}; p\} = \begin{cases} drastic\ t-norm & \text{for} \quad p = 0 \\ \max\left\{0, 1 - \left(\sum\limits_{i=1}^{n} (1-a_i)^p\right)^{\frac{1}{p}}\right\} & \text{for} \quad p \in (0,\infty) \\ Zadeh\ t-norm & \text{for} \quad p = \infty \end{cases} \tag{49}$$

where $\overleftrightarrow{T}$ stands for a t-norm of the Yager family parameterized by p.

The Yager t-conorm

$$\overleftrightarrow{S}\{\mathbf{a}; p\} = \begin{cases} drastic\ t-conorm & \text{for} \quad p = 0 \\ \min\left\{1, \left(\sum\limits_{i=1}^{n} a_i^p\right)^{\frac{1}{p}}\right\} & \text{for} \quad p \in (0,\infty) \\ Zadeh\ t-conorm & \text{for} \quad p = \infty \end{cases} \tag{50}$$

where $\overleftrightarrow{S}$ stands for a t-conorm of the Yager family parameterized by p.

Obviously formula (49) defines the "engineering implication" for $n = 2$. Combining the S-implication and (50) we get the fuzzy S-implication generated by the Yager family

$$\overleftrightarrow{I}(a, b; p) = \min\left\{1, ((1-a)^p + b^p)^{\frac{1}{p}}\right\}. \tag{51}$$

6.3. Soft fuzzy norms. In this sections we recall a concept of soft fuzzy norms proposed by Yager and Filev [31]. Let $a_1, \ldots, a_n$ be numbers in the unit interval that are to be aggregated. The soft version of triangular norms suggested by Yager and Filev is defined by

$$\tilde{T}\{\mathbf{a}; \alpha\} = (1-\alpha)\frac{1}{n}\sum_{i=1}^{n} a_i + \alpha \mathop{T}_{i=1}^{n}\{a_i\} \tag{52}$$

and

$$\tilde{S}\{\mathbf{a}; \alpha\} = (1-\alpha)\frac{1}{n}\sum_{i=1}^{n} a_i + \alpha \mathop{S}_{i=1}^{n}\{a_i\} \tag{53}$$

where $\alpha \in [0, 1]$. They allow to balance between the arithmetic average aggregator and the triangular norm aggregator depending on parameter α.

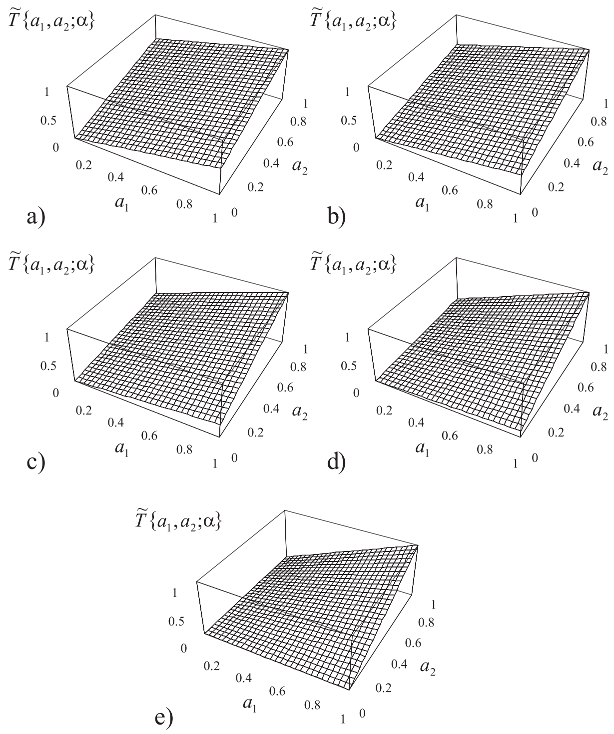

Fig. 6. 3D plots of function (54) for a) $\alpha = 0.00$, b) $\alpha = 0.25$, c) $\alpha = 0.50$, d) $\alpha = 0.75$, e) $\alpha = 1.00$

Example 2. (An example of soft algebraic triangular norms) The soft algebraic triangular norms are based on classical algebraic triangular norms (see e.g. [6]). The soft algebraic t-norm is described as follows

$$\tilde{T}\{a_1, a_2; \alpha\} = (1-\alpha)\,\frac{1}{2}\,(a_1 + a_2) + \alpha a_1 a_2. \tag{54}$$

The 3D plots of function (54) are depicted in Fig. 6.

The soft algebraic t-conorm is given by

$$\tilde{S}\{a_1, a_2; \alpha\} = (1-\alpha)\,\frac{1}{2}\,(a_1 + a_2) + \alpha\,(a_1 + a_2 - a_1 a_2)\,. \tag{55}$$

The 3D plots of function (55) are presented in Fig. 7.

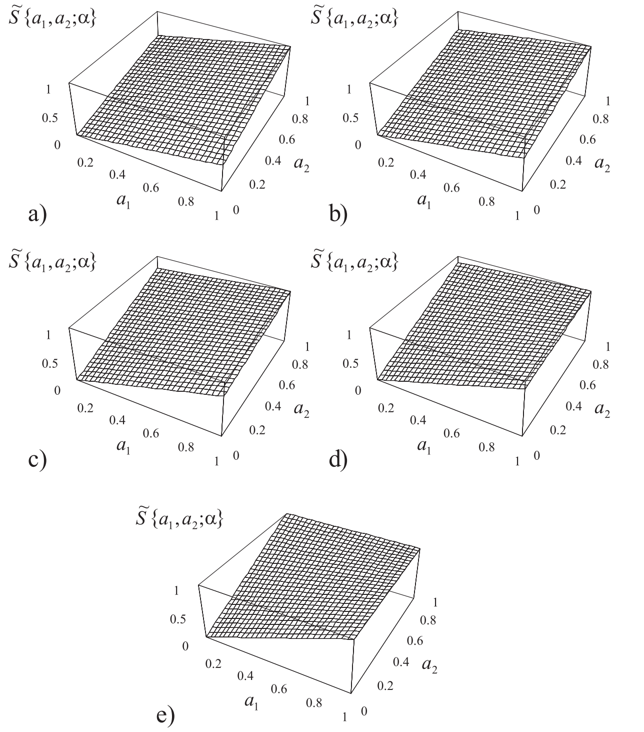

Fig. 7. 3D plots of function (55) for a) $\alpha = 0.00$, b) $\alpha = 0.25$, c) $\alpha = 0.50$, d) $\alpha = 0.75$, e) $\alpha = 1.00$

6.4. Design of flexible neuro fuzzy systems. In Sections 6.1, 6.2 and 6.3 we introduced the following flexibility concepts in the design of neuro-fuzzy systems:

— softness to fuzzy implication operators, to the aggregation of rules and to the connectives of antecedents,

— certainty weights to the aggregation of rules and to the connectives of antecedents,

— parameterized families of t-norms and t-conorms to fuzzy implication operators, to the aggregation of rules and to the connectives of antecedents.

In Fig. 8 we show a design process of flexible neuro-fuzzy systems developed in this paper. The design process includes the automatic determination of fuzzy inference which will be explained in Sections 7 and 8 and simulated in Section 9.

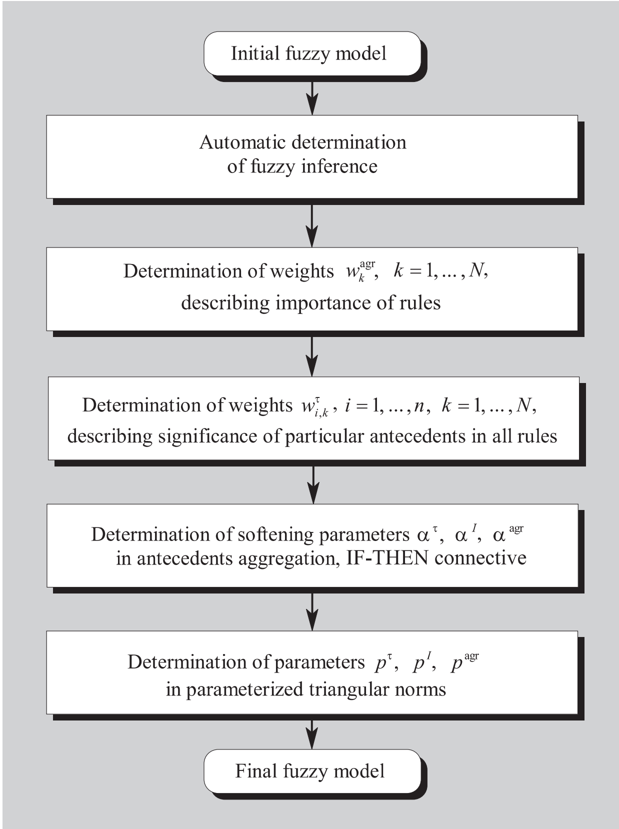

Fig. 8. Design of flexible fuzzy system

7. Generalized triangular norms

We start with a definition which is a generalization of a strong negation (see [34]).

DEFINITION 2. (Compromise operator) Function

$$\tilde{N}_\nu : [0,1] \to [0,1] \tag{56}$$

given by

$$\begin{aligned}\tilde{N}_\nu(a) &= (1-\nu)N(a) + \nu N(N(a)) \\ &= (1-\nu)N(a) + \nu a\end{aligned} \tag{57}$$

is called a compromise operator where $\nu \in [0,1]$ and $N(a) = \tilde{N}_0(a) = 1 - a$.

Observe that

$$\tilde{N}_\nu(a) = \begin{cases} N(a) & \text{for} \quad \nu = 0 \\ \frac{1}{2} & \text{for} \quad \nu = \frac{1}{2} \\ a & \text{for} \quad \nu = 1 \end{cases} \tag{58}$$

Obviously function $\tilde{N}_\nu$ is a strong negation for $\nu = 0$. The 3D plot of function (57) is depicted in Fig. 9.

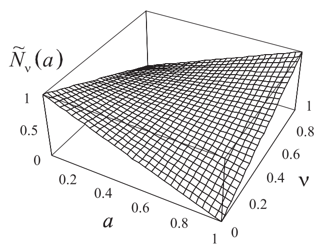

Fig. 9. 3D plot of function (57)

Definition 3. (H-function) Function

$$H : [0,1]^n \to [0,1] \qquad (59)$$

given by

$$H(\mathbf{a};\nu) = \tilde{N}_\nu \left(\overset{n}{\underset{i=1}{S}} \left\{ \tilde{N}_\nu(a_i) \right\} \right) = \tilde{N}_{1-\nu} \left(\overset{n}{\underset{i=1}{T}} \left\{ \tilde{N}_{1-\nu}(a_i) \right\} \right) \qquad (60)$$

is called an H-function where $\nu \in [0,1]$.

Theorem 1. Let T and S be dual triangular norms. Function H defined by (60) varies between a t-norm and a t-conorm as ν goes from 0 to 1.

Proof. From the assumption it follows that

$$T\{\mathbf{a}\} = N(S\{N(a_1), N(a_2), \ldots, N(a_n)\}) \qquad (61)$$

For $\nu = 0$ formula (61) can be rewritten with the notation of the compromise operator (57)

$$T\{\mathbf{a}\} = \tilde{N}_0 \left(S\left\{ \tilde{N}_0(a_1), \tilde{N}_0(a_2), \ldots, \tilde{N}_0(a_n) \right\} \right). \qquad (62)$$

Apparently

$$S\{\mathbf{a}\} = \tilde{N}_1 \left(S\left\{ \tilde{N}_1(a_1), \tilde{N}_1(a_2), \ldots, \tilde{N}_1(a_n) \right\} \right) \qquad (63)$$

for $\nu = 1$.

The right-hand sides of (62) and (63) can be written as follows

$$H(\mathbf{a};\nu) = \tilde{N}_\nu \left(\overset{n}{\underset{i=1}{S}} \left\{ \tilde{N}_\nu(a_i) \right\} \right) \qquad (64)$$

for $\nu = 0$ and $\nu = 1$, respectively. If parameter ν changes from 0 to 1, then the result is established.

Remark 3. Observe that

$$H(\mathbf{a};\nu) = \begin{cases} T\{\mathbf{a}\} & \text{for } \nu = 0 \\ \frac{1}{2} & \text{for } \nu = \frac{1}{2} \\ S\{\mathbf{a}\} & \text{for } \nu = 1. \end{cases} \qquad (65)$$

It is easily seen, that for $0 < \nu < 0.5$ the H-function resembles a t-norm and for $0.5 < \nu < 1$ the H-function resembles a t-conorm.

Example 3. (An example of the H-function generated by the algebraic triangular norms) We will apply Theorem 1 to illustrate (for $n = 2$) how to switch between the algebraic t-norm

$$T\{a_1, a_2\} = H(a_1, a_2; 0) = a_1 a_2 \qquad (66)$$

and the algebraic t-conorm

$$S\{a_1, a_2\} = H(a_1, a_2; 1) = a_1 + a_2 - a_1 a_2. \qquad (67)$$

The H-function generated by formulas (66) or (67) takes the form

$$H(a_1, a_2; \nu) = \tilde{N}_{1-\nu} \left(\tilde{N}_{1-\nu}(a_1) \tilde{N}_{1-\nu}(a_1) \right) = \tilde{N}_\nu \left(1 - \left(1 - \tilde{N}_\nu(a_1)\right) \left(1 - \tilde{N}_\nu(a_1)\right) \right) \qquad (68)$$

and varies from (66) to (67) as ν goes from zero to one. In Fig. 10, we illustrate function (68) for $\nu = 0.00$, $\nu = 0.15$, $\nu = 0.50$, $\nu = 0.85$, $\nu = 1.00$.

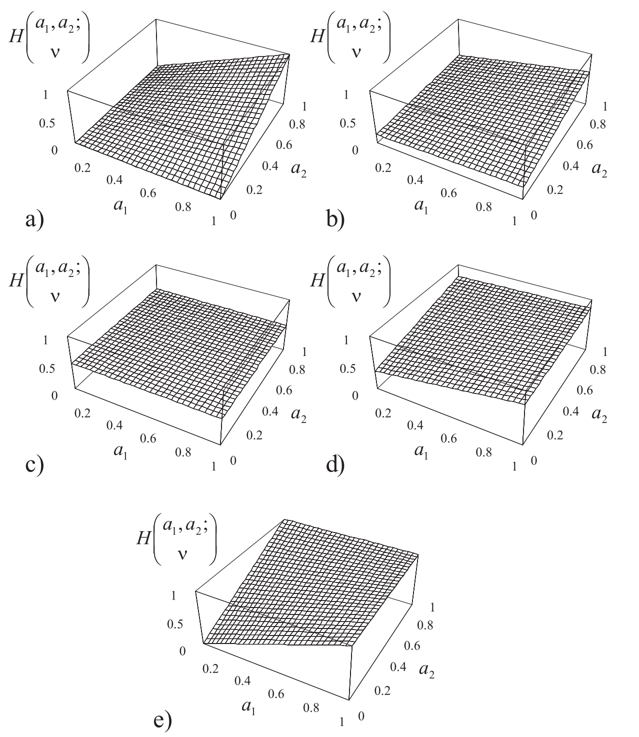

Fig. 10. 3D plots of function (68) for a) $\nu = 0.00$, b) $\nu = 0.15$, c) $\nu = 0.50$, d) $\nu = 0.85$, e) $\nu = 1.00$

Theorem 2. Let T and S be dual triangular norms. Then

$$I(a, b; \nu) = H\left(\tilde{N}_{1-\nu}(a), b; \nu \right) \qquad (69)$$

switches between an "engineering implication"

$$I_{\text{eng}}(a, b) = I(a, b; 0) = T\{a, b\} \qquad (70)$$

and an S-implication

$$I_{\text{fuzzy}}(a, b) = I(a, b; 1) = S\{1 - a, b\} \qquad (71)$$

Proof. Theorem 2 is a straightforward consequence of Theorem 1.

Example 4. (An example of the H-implication generated by the algebraic triangular norms) We will define the H-implication generated by the algebraic triangular norms and based on formula (24). Let

$$I_{\text{eng}}(a, b) = H(a, b; 0) = T\{a, b\} = ab \qquad (72)$$

and

$$I_{\text{fuzzy}}(a, b) = H\left(\tilde{N}_0(a), b; 1 \right) = S\{N(a), b\} = 1 - a + ab. \qquad (73)$$

Then

$$I(a,b;\nu) = H\left(\tilde{N}_{1-\nu}(a), b; \nu\right) \tag{74}$$

goes from (72) to (73) as ν varies from 0 to 1. The 3D plots of function (74) are depicted in Fig. 11.

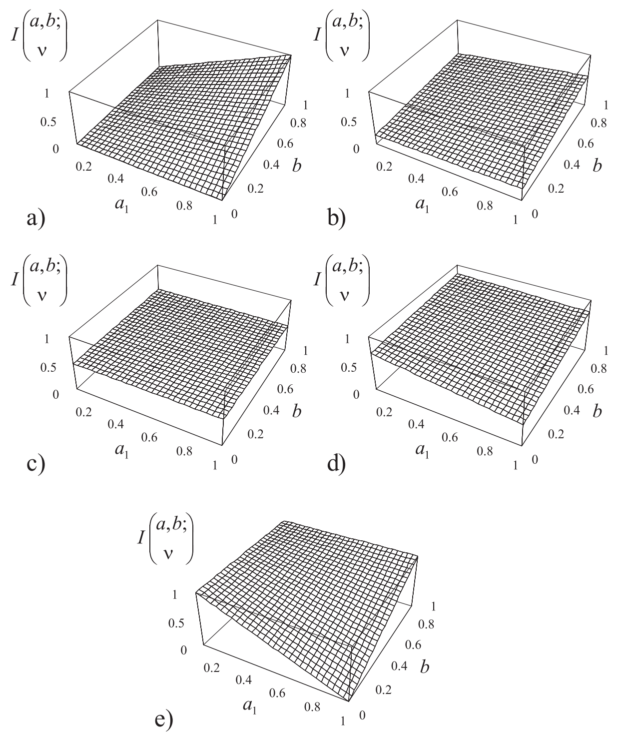

Fig. 11. 3D plots of function (74) for a) $\nu = 0.00$, b) $\nu = 0.15$, c) $\nu = 0.50$, d) $\nu = 0.85$, e) $\nu = 1.00$

8. Flexible neuro-fuzzy systems

In this section we incorporate flexibility parameters given in Section 6 to design neuro-fuzzy systems defined by formula (35). By using the concept of generalized triangular norms introduced in Section 7, we get the flexible neuro-fuzzy systems given by:

$$\tau_k(\bar{\mathbf{x}}) = \begin{pmatrix} (1-\alpha^\tau)\operatorname{avg}\left(\mu_{A_1^k}(\bar{x}_1), \ldots, \mu_{A_n^k}(\bar{x}_n)\right) + \\ +\alpha^\tau \overset{\leftrightarrow}{H}^* \begin{pmatrix} \mu_{A_1^k}(\bar{x}_1), \ldots, \mu_{A_n^k}(\bar{x}_n); \\ w_{1,k}^\tau, \ldots, w_{n,k}^\tau, p^\tau, 0 \end{pmatrix} \end{pmatrix} \tag{75}$$

$$I_{k,r}(\bar{\mathbf{x}}, \bar{y}^r) = \begin{pmatrix} (1-\alpha^I)\operatorname{avg}\left(\tilde{N}_{1-\nu}(\tau_k(\bar{\mathbf{x}})), \mu_{B^k}(\bar{y}^r)\right) + \\ +\alpha^I \overset{\leftrightarrow}{H} \begin{pmatrix} \tilde{N}_{1-\nu}(\tau_k(\bar{\mathbf{x}})), \mu_{B^k}(\bar{y}^r); \\ p^I, \nu \end{pmatrix} \end{pmatrix} \tag{76}$$

$$\operatorname{agr}_r(\bar{\mathbf{x}}, \bar{y}^r) = \begin{pmatrix} (1-\alpha^{\text{agr}})\operatorname{avg}\left(I_{1,r}(\bar{\mathbf{x}}, \bar{y}^r), \ldots, I_{N,r}(\bar{\mathbf{x}}, \bar{y}^r)\right) + \\ +\alpha^{\text{agr}} \overset{\leftrightarrow}{H}^* \begin{pmatrix} I_{1,r}(\bar{\mathbf{x}}, \bar{y}^r), \ldots, I_{N,r}(\bar{\mathbf{x}}, \bar{y}^r); \\ w_1^{\text{agr}}, \ldots, w_N^{\text{agr}}, p^{\text{agr}}, 1-\nu \end{pmatrix} \end{pmatrix}. \tag{77}$$

In the above system we use parameterised families $\overset{\leftrightarrow}{H}(\cdot)$ and parameterised families with weights $\overset{\leftrightarrow}{H}^*(\cdot)$ analogously to formula (44) and (49). More specifically, in (75) and (77) we use the following definition

$$\overset{\leftrightarrow}{H}^* \begin{pmatrix} a_1, \ldots, a_n; \\ w_1, \ldots, w_n, p, \nu \end{pmatrix} = \overset{\leftrightarrow}{H} \begin{pmatrix} \arg_1(a_1, w_1, \nu), \ldots, \arg_n(a_n, w_n, \nu); \\ p, \nu \end{pmatrix} \tag{78}$$

where

$$\arg_i(a_i, w_i, \nu) = (1-\nu)(1 - w_i(1-a_i)) + \nu w_i a_i. \tag{79}$$

It is well known that the basic concept of the backpropagation algorithm, commonly used to train neural networks, can also be applied to any feedforward network. Let $\bar{\mathbf{x}}(t) \in \mathbb{R}^n$ and $d(t) \in \mathbb{R}$ be a sequence of inputs and desirable output signals, respectively.

Based on the learning sequence $(\bar{\mathbf{x}}(1), d(1)), (\bar{\mathbf{x}}(2), d(2)), \ldots$ we wish to determine all parameters (including the system's type ν) and weights of neuro-fuzzy systems such that

$$e(t) = \frac{1}{2}\left[f(\bar{\mathbf{x}}(t)) - d(t)\right]^2 \tag{80}$$

is minimized, where $f(\cdot)$ is given (35). The steepest descent optimization algorithm can be applied to solve this problem. For instance, parameters $\bar{y}^r$, $r = 1, \ldots, N$, are trained by the iterative procedure

$$\bar{y}^r(t+1) = \bar{y}^r(t) - \eta \frac{\partial e(t)}{\partial \bar{y}^r(t)}. \tag{81}$$

Directly calculating partial derivatives in recursion (81) is rather complicated. Therefore, we recall that our system has a layered architecture (see Fig. 2) and apply the idea of the backpropagation method to train the system. The exact recursions are not shown here, however they can be derived analogously to the method given in [24]. We can apply the gradient optimization with constraints in order to optimize:

— $\nu \in [0,1]$,
— $\alpha^\tau \in [0,1]$, $\alpha^I \in [0,1]$, $\alpha^{\text{agr}} \in [0,1]$,
— $p^\tau \in [0,\infty)$, $p^I \in [0,\infty)$, $p^{\text{agr}} \in [0,\infty)$,
— $w_{i,k}^\tau \in [0,1]$, $i = 1, \ldots, n$, $k = 1, \ldots, N$,
— $w_k^{\text{agr}} \in [0,1]$, $k = 1, \ldots, N$.

The same technique can be used in order to find in the process of learning parameters of the membership functions $\mu_{A_i^k}(x_i)$ and $\mu_{B^k}(y)$, $i = 1, \ldots, n$, $k = 1, \ldots, N$:

— $p_{u,i,k}^A$, $u = 1, \ldots, P^A$, $i = 1, \ldots, n$, $k = 1, \ldots, N$,
— $p_{u,k}^B$, $u = 2, \ldots, P^B$, $k = 1, \ldots, N$,
— $p_{1,k}^B = \bar{y}^k$, $k = 1, \ldots, N$.

9. Simulation results

In this section we present simulations of neuro-fuzzy systems derived in this paper. All the simulations are designed in the same fashion. We will gradually incorporate flexibility parameters in experiments (i)–(iv):

Table 3
Experimental results

Fuzzy systems with non-parametrised H-functions (Glass Classification Problem)								
Experiment number	Name of flexibility parameter	Initial values	Final values after learning		Mistakes [%] (learning sequence)		Mistakes [%] (testing sequence)	
			Zadeh H-function	Algebraic H-function	Zadeh H-function	Algebraic H-function	Zadeh H-function	Algebraic H-function
i	ν	0.5	1.0000	1.0000	3.33	3.33	3.13	3.13
ii	ν	0	—	—	4.00	4.67	3.13	3.13
iii	ν	0.5	1.0000	1.0000	2.67	2.67	3.13	3.13
	α^{τ}	1	0.0163	0.0019				
	α^{I}	1	0.9939	0.9970				
	α^{agr}	1	0.9554	0.9912				
iv	ν	0.5	1.0000	1.0000	2.67	2.67	1.56	1.56
	α^{τ}	1	0.0038	0.0137				
	α^{I}	1	0.9858	0.9937				
	α^{agr}	1	0.8674	0.9693				
	$\mathbf{w}^{\tau}$	**1**	Fig. 12-a	Fig. 12-b				
	$\mathbf{w}^{\mathrm{agr}}$	**1**	Fig. 12-a	Fig. 12-b				

Table 4
Experimental results

Fuzzy systems with parametrised H-functions (Glass Classification Problem)								
Experiment number	Name of flexibility parameter	Initial values	Final values after learning		Mistakes [%] (learning sequence)		Mistakes [%] (testing sequence)	
			Dombi H-function	Yager H-function	Dombi H-function	Yager H-function	Dombi H-function	Yager H-function
i	ν	0.5	1.0000	1.0000	3.33	3.33	3.13	3.13
ii	ν	0	-	-	4.00	4.67	3.13	3.13
iii	ν	0.5	1.0000	1.0000	2.67	2.67	1.56	1.56
	p^{τ}	10	9.7496	12.5239				
	p^{I}	10	10.0006	9.9965				
	p^{agr}	10	9.9999	9.9920				
	α^{τ}	1	0.0302	0.1122				
	α^{I}	1	0.9173	0.9413				
	α^{agr}	1	0.9934	0.9973				
iv	ν	0.5	1.0000	1.0000	2.00	2.00	1.56	1.56
	p^{τ}	10	9.1328	12.1261				
	p^{I}	10	10.0601	9.8597				
	p^{agr}	10	10.3097	9.9544				
	α^{τ}	1	0.0948	0.1280				
	α^{I}	1	0.8896	0.9349				
	α^{agr}	1	0.9600	0.9695				
	$\mathbf{w}^{\tau}$	**1**	Fig. 13-a	Fig. 13-b				
	$\mathbf{w}^{\mathrm{agr}}$	**1**	Fig. 13-a	Fig. 13-b				

— In the first experiment (i), based on the input-output data, we learn the parameters of the membership functions and a system type $\nu \in [0,1]$ assuming that there are no other flexibility parameters in the system description. It will be seen that the optimal values of ν, determined by a gradient procedure, are either zero or one.

— In the second experiment (ii), we learn the parameters of the membership functions choosing value ν as opposite to that obtained in experiment (i). Obviously, we expect a worse performance of the neuro-fuzzy system comparing with experiment (i).

— In the third experiment (iii), we learn the parameters of the membership functions, system type $\nu \in [0,1]$ and soft parameters $\alpha^{\tau} \in [0,1]$, $\alpha^{I} \in [0,1]$, $\alpha^{\text{agr}} \in [0,1]$ of the flexible system assuming that classical (not-parameterised) triangular norms are applied.

— In the fourth experiment (iv), we learn the same parameters as in the third experiment and, moreover, the weights $w^{\tau}_{i,k} \in [0,1]$, $i = 1,\dots,n$, $k = 1,\dots,N$, in the antecedents of rules and weights $w^{\text{arg}}_{i,k} \in [0,1]$, $k = 1,\dots,N$, of the aggregation operator of rules. In all diagrams (weights representation) we separate $w^{\tau}_{i,k} \in [0,1]$, $i = 1,\dots,n$, $k = 1,\dots,N$, from $w^{\text{arg}}_{i,k} \in [0,1]$, $k = 1,\dots,N$, by a vertical dashed line.

In each of the above simulations we apply the Zadeh H-implication (generated by the min/max triangular norms) and the algebraic H-implication (generated by the algebraic triangular norms). In separate experiments we repeat simulations (i)–(iv) replacing the Zadeh H-implication and the algebraic H-implication by quasi-implications generated by parameterised triangular norms: the Dombi H-implication and the Yager H-implication. In these simulations we additionally incorporate parameters $p^{\tau} \in [0,\infty)$, $p^{I} \in [0,\infty)$, $p^{\text{agr}} \in [0,\infty)$.

9.1. Glass classification problem. The Glass Classification problem contains 214 instances and each instance is described by nine attributes (RI: refractive index, Na: sodium, Mg: magnesium, Al: aluminium, Si: silicon, K: potassium, Ca: calcium, Ba: barium, Fe: iron). All attributes are continuous. There are two classes: the window glass and the non-window glass. In our experiments, all sets are divided into a learning sequence (150 sets) and a testing sequence (64 sets). The study of the classification of the types of glass was motivated by criminological investigation. At the scene of the crime, the glass left can be used as evidence if it is correctly identified. The experimental results for the Glass Classification problem are depicted in Tables 3 and 4 for the not-parameterised (Zadeh and algebraic) and parameterised (Dombi and Yager) H-functions, respectively. For experiment (iv) the final values (after learning) of weights $w^{\tau}_{i,k} \in [0,1]$ and $w^{agr}_{k} \in [0,1]$, $i = 1,\dots,9$, $k = 1,\dots,2$, are shown in Fig. 12 (Zadeh and algebraic H-functions) and Fig. 13 (Dombi and Yager H-functions).

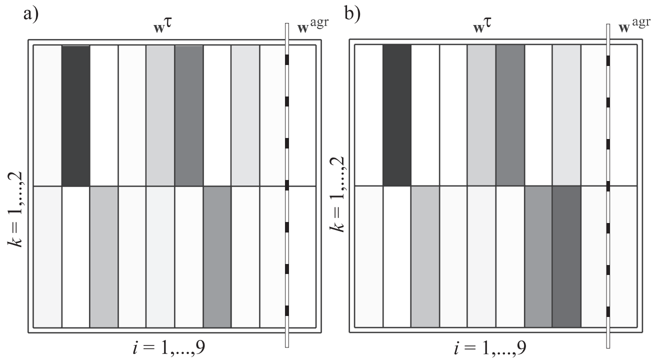

Fig. 12. Weights representation in the Glass Classification Problem a) Zadeh H-function, b) algebraic H-function

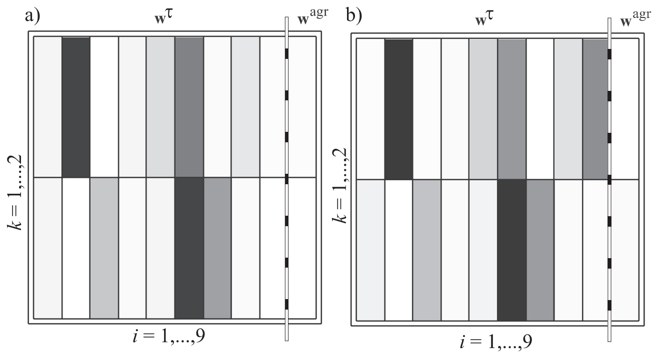

Fig. 13. Weights representation in the Glass Classification Problem a) Dombi H-function, b) Yager H-function

The comparison table for the Glass Classification Problem is shown in Table 5.

Table 5
Comparison table (Glass Classification Problem)

Glass Classification Problem	
Method	Testing Accuracy
Dong and Kothari (IG) [35]	92.86
Dong and Kothari (IG+LA) [35]	93.09
Dong and Kothari (GR) [35]	92.86
Dong and Kothari (GR+LA) [35]	93.10
Rutkowski and Cpałka [27]	93.75
our result (N=2)	98.44

9.2. Nonlinear dynamic plant identification Problem. We consider the second-order nonlinear plant described by

$$y(k) = g\left(y(k-1), y(k-2)\right) + u(k) \tag{82}$$

with

$$g\left(y(k-1), y(k-2)\right) = \frac{y(k-1)y(k-2)\left(y(k-1)-0.5\right)}{1+y^2(k-1)+y^2(k-2)}. \tag{83}$$

The goal is to approximate the nonlinear component $g(y(k-1), y(k-2))$ of the plant (82) with a fuzzy model.

Table 6
Experimental results

Fuzzy systems with non-parametrised H-functions (Nonlinear Dynamic Plant Identification Problem)								
Experiment number	Name of flexibility parameter	Initial values	Final values after learning		RMSE (learning sequence)		RMSE (testing sequence)	
			Zadeh H-function	Algebraic H-function	Zadeh H-function	Algebraic H-function	Zadeh H-function	Algebraic H-function
i	ν	0.5	0.0000	0.0000	0.0445	0.0238	0.0309	0.0133
ii	ν	1	—	—	0.0490	0.0316	0.0313	0.0201
iii	ν	0.5	0.0000	0.0000	0.0341	0.0236	0.0255	0.0123
	α^{τ}	1	0.8080	0.9969				
	α^{I}	1	0.7294	0.9904				
	α^{agr}	1	0.9990	0.9752				
iv	ν	0.5	0.0000	0.0000	0.0305	0.0198	0.0196	0.0107
	α^{τ}	1	0.7626	0.9720				
	α^{I}	1	0.6769	0.9385				
	α^{agr}	1	0.9605	0.9219				
	$\mathbf{w}^{\tau}$	**1**	Fig. 14-a	Fig. 14-b				
	$\mathbf{w}^{agr}$	**1**	Fig. 14-a	Fig. 14-b				

Table 7
Experimental results

Fuzzy systems with parametrised H-functions (Nonlinear Dynamic Plant Identification Problem)								
Experiment number	Name of flexibility parameter	Initial values	Final values after learning		RMSE (learning sequence)		RMSE (testing sequence)	
			Dombi H-function	Yager H-function	Dombi H-function	Yager H-function	Dombi H-function	Yager H-function
i	ν	0.5	0.0000	0.0000	0.0448	0.0259	0.0264	0.0160
ii	ν	1	—	—	0.0531	0.0435	0.0298	0.0259
iii	ν	0.5	0.0000	0.0000	0.0348	0.0249	0.0194	0.0147
	p^{τ}	10	10.0065	6.7442				
	p^{I}	10	9.9802	11.6769				
	p^{agr}	10	10.0823	4.2762				
	α^{τ}	1	0.8634	0.8990				
	α^{I}	1	0.1743	0.9999				
	α^{agr}	1	0.9955	0.9998				
iv	ν	0.5	0.0000	0.0000	0.0291	0.0225	0.0185	0.0129
	p^{τ}	10	9.3823	5.1690				
	p^{I}	10	8.9950	7.9606				
	p^{agr}	10	12.5209	0.2942				
	α^{τ}	1	0.8568	0.8420				
	α^{I}	1	0.1285	0.9974				
	α^{agr}	1	0.9692	0.9767				
	$\mathbf{w}^{\tau}$	**1**	Fig. 15-a	Fig. 15-b				
	$\mathbf{w}^{agr}$	**1**	Fig. 15-a	Fig. 15-b				

In [30] 400 simulated data were generated from the plant model (82). Starting from the equilibrium state (0,0), 200 samples of the identification data were obtained with a random input signal $u(k)$ uniformly distributed in $[-1.5, 1.5]$, followed by 200 samples of evaluation data obtained using a sinusoidal input signal $u(k) = \sin(2\pi k/25)$.

The experimental results for the Nonlinear Dynamic Plant Identification Problem are depicted in Tables 6 and 7 for the not-parameterised (Zadeh and algebraic) and parameterised (Dombi and Yager) H-functions, respectively. For experiment (iv) the final values (after learning) of weights $w^{\tau}_{i,k} \in [0,1]$ and $w^{agr}_{k} \in [0,1]$, $i = 1,\ldots,2$, $k = 1,\ldots,6$, are shown in Fig. 14 (Zadeh and algebraic H-functions) and Fig. 15 (Dombi and Yager H-functions).

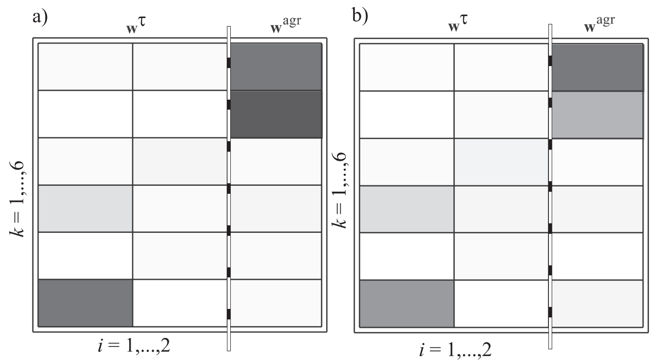

Fig. 14. Weights representation in the Nonlinear Dynamic Plant Identification Problem a) Zadeh H-function, b) algebraic H-function

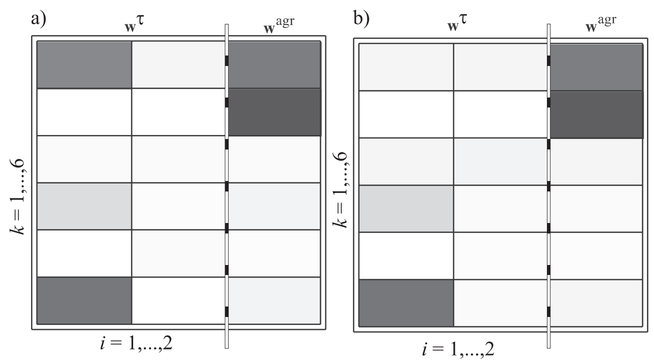

Fig. 15. Weights representation in the Nonlinear Dynamic Plant Identification Problem a) Dombi H-function, b) Yager H-function

The comparison table for the Nonlinear Dynamic Plant Identification Problem is shown Table 8.

Table 8
Comparison table
(Nonlinear Dynamic Plant Identification Problem)

Nonlinear Dynamic Plant Identification Problem			
Method	No of rules	Training RMSE	Testing RMSE
Wang and Yen [30]	40	0.0182	0.0263
Wang and Yen [30]	28	0.0182	0.0245
Wang and Yen [29]	36	0.0053	0.0714
Wang and Yen [29]	23	0.0057	0.0436
Wang and Yen [29]	36	0.0014	0.0539
Wang and Yen [29]	24	0.0014	0.0253
Yen and Wang [32]	25	0.0152	0.0202
Yen and Wang [32]	20	0.0261	0.0155
Setnes and Roubos [28]	7	0.1265	0.0346
Setnes and Roubos [28]	7	0.0548	0.0221
Setnes and Roubos [28]	5	0.0762	0.0500
Setnes and Roubos [28]	5	0.0274	0.0187
Setnes and Roubos [28]	4	0.0346	0.0217
Roubos and Setnes [21]	5	0.0700	0.0539
Roubos and Setnes [21]	5	0.0374	0.0243
Roubos and Setnes [21]	5	0.0288	0.0187
Rutkowski and Cpałka [27]	5	0.0328	0.0211
our result (N=6)	6	0.0196	0.0107

10. Final remarks

In the paper a new method for system modelling and pattern classification has been proposed. The method is based on the concept of flexible parameters incorporated into construction of neuro-fuzzy systems. Obviously, the computational burden of flexible neuro-fuzzy systems is much higher comparing with traditional fuzzy modelling. The main advantage of our approach is remarkable accuracy of new algorithms in various problems of system modelling and classification as shown in Section 9. A similar property is possessed by probabilistic neural networks [36–38], [8, 39, 40] applied to system modelling and classification. In the future research it would be interesting to investigate relations between flexible neuro-fuzzy systems and probabilistic neural networks. Moreover, the evolutionary techniques [16] should be combined with the gradient methods in order to further improve the optimization process.

References

[1] L. A. Zadeh, "Fuzzy sets", *Information and Control* 8(3), 338–353 (1965).

[2] K. Hirota, *Industrial Applications of Fuzzy Technology*, Berlin: Springer Verlag, 1993.

[3] R. Tadeusiewicz, *Neural Networks*, Warsaw: RM Academic Publishing House, 1993 (in Polish).

[4] R. Tadeusiewicz, *Elementary introduction to neural networks with computer programs*, Warsaw: Academic Publishing House, 1998 (in Polish).

[5] J. Kacprzyk, *Multistage Fuzzy Control*, Chichester: John Wiley & Sons, 1997.

[6] L. Rutkowski, *Flexible Neuro-Fuzzy Systems*, Dordrecht: Kluwer Academic Publishers, 2004.

[7] R. L. Eubank, *Nonparametric Regression and Spline Smoothing*, New York: Marcel Dekker, 1999.

[8] L. Rutkowski, *New Soft Computing Techniques for System Modeling, Pattern Classification and Image Processing*, Berlin: Springer-Verlag, 2004.

[9] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*, Berlin: Springer-Verlag, 1992.

[10] Z. Pawlak, "Rough sets", *Intern. Journal of Information and Computer Science* 11, 341 (1982).

[11] Z. Pawlak, *Rough Sets. Theoretical Aspects of Reasoning About Data*, Dordrecht: Kluwer Academic Publishers, 1991.

[12] Z. Bubnicki, "Uncertain variables and their application to decision making", *IEEE Trans. on SMC, Part A: Systems and Humans* 31, 587–596 (2001).

[13] Z. Bubnicki, *Uncertain Logics, Variables and Systems*, Berlin: Springer-Verlag, 2002.

[14] R. A. Aliev and R. R. Aliev, *Soft Computing and its Applications*, Singapore: World Scientific Publishing, 2001.

[15] V. Kecman, *Learning and Soft Computing*, MIT, Cambridge 2001.

[16] M. Y. Chen and D. A. Linkens, "A systematic neuro-fuzzy modeling framework with application to material property prediction", *IEEE Trans. on Fuzzy Systems* 31, 781–790 (2001).

[17] González and R. Pérez, "SLAVE: a genetic learning system based on an iterative approach", *IEEE Trans. on Fuzzy Systems* 7, 176–191 (1999).

[18] J. S. Jang, C. T. Sun and E. Mizutani, *Neuro-Fuzzy and Soft Computing*, London: Prentice Hall, 1997.

[19] N. Kasabov, "DENFIS: dynamic evolving neural-fuzzy inference system and its application for time-series prediction", *IEEE Trans. on Fuzzy Systems* 10, 144–154 (2002).

[20] E. Kim, M. Park, S. Ji and M. Park, "A new approach to fuzzy modeling", *IEEE Trans. on Fuzzy Systems* 5, 328–337 (1997).

[21] H. Roubos H. and M. Setnes, "Compact and transparent fuzzy models and classifiers through iterative complexity reduction", *IEEE Trans. on Fuzzy Systems* 9, 516–524 (2001).

[22] L. Rutkowski and K. Cpałka, "Flexible structures of neuro-fuzzy systems", *Quo Vadis Computational Intelligence, Studies in Fuzziness and Soft Computing*, Berlin: Springer-Verlag, Vol. 54, 479–484 (2000).

[23] L. Rutkowski and K. Cpałka, "A general approach to neuro-fuzzy systems", *The 10th IEEE Intern. Conference on Fuzzy Systems*, Melbourne, 2001.

[24] L. Rutkowski and K. Cpałka, "Designing and learning of adjustable quasi-triangular norms with applications to neuro-fuzzy systems", *Technical Report, Department of Computer Engineering, Technical University of Czestochowa*, Częstochowa 2002.

[25] L. Rutkowski and K. Cpałka, "A neuro-fuzzy controller with a compromise fuzzy reasoning", *Control and Cybernetics* 31(2), 297–308 (2002).

[26] L. Rutkowski and K. Cpałka, "Flexible weighted neuro-fuzzy systems", *9th Intern. Conference on Neural Information Processing (ICONIP'02)*, Orchid Country Club, Singapore, 2002.

[27] L. Rutkowski and K. Cpałka, "Flexible neuro-fuzzy systems", *IEEE Trans. Neural Networks* 14, 554–574 (2003).

[28] M. Setnes and H. Roubos, "GA-fuzzy modeling and classification complexity and performance", *IEEE Trans. on Fuzzy Systems* 8, 509–521 (2000).

[29] L. Wang and J. Yen, "Application of statistical information criteria for optimal fuzzy model construction", *IEEE Trans. on Fuzzy Systems* 6, 362–371 (1998).

[30] L. Wang and J. Yen, "Extracting fuzzy rules for system modeling using a hybrid of genetic algorithms and Kalman filter", *Fuzzy Sets and Systems* 101, 353–362 (1999).

[31] R. R. Yager and D. P. Filev, *Essentials of Fuzzy Modeling and Control*, New York: John Wiley & Sons, 1994.

[32] J. Yen and L. Wang, "Simplifying fuzzy rule based models using orthogonal transformation methods", *IEEE Trans. Syst. Man Cybern.* 29, 13–24 (1999).

[33] J. C. Fodor, "On fuzzy implication operators", *Fuzzy Sets and Systems* 42, 293–300 (1991).

[34] E. P. Klement, R. Mesiar and E. Pap, *Triangular Norms*, Dordrecht: Kluwer Academic Publishers, 2000.

[35] M. Dong and R. Kothari, "Look-ahead based fuzzy decision tree induction", *IEEE Trans. on Fuzzy Systems* 9, 461–468 (2001).

[36] L. Rutkowski and E. Rafajłowicz, "On global rate of convergence of some nonparametric identification procedures", *IEEE Trans. on Automatic Control* AC-34(10), 1089–1091 (1989).

[37] L. Rutkowski, "Identification of MISO nonlinear regressions in the presence of a wide class of disturbances", *IEEE Trans. on Information Theory* IT-37, 214–216 (1991).

[38] L. Rutkowski, "Multiple Fourier series procedures for extraction of nonlinear regressions from noisy data", *IEEE Trans. on Signal Processing* 41, 3062–3065 (1993).

[39] L. Rutkowski, "Adaptive probabilistic neural networks for pattern classification in time-varying environment", *IEEE Trans. on Neural Networks* 15 (2004).

[40] L. Rutkowski, "Generalized regression neural networks in time-varying environment", *IEEE Trans. on Neural Networks* 15, (2004).