A NEURO-FUZZY SYSTEM BASED ON LOGICAL INTERPRETATION OF IF-THEN RULES

JACEK ŁĘSKI*, NORBERT HENZEL*

Several important fuzzy implications and their properties are described on the basis of an axiomatic approach to the definition of the fuzzy implications. Then the idea of approximate reasoning using the generalized modus ponens and fuzzy implications is considered. The elimination of the non-informative part of the final fuzzy set before defuzzification plays the key role in this paper. After reviewing well-known fuzzy systems, a new artificial neural network based on logical interpretation of if-then rules (ANBLIR) is introduced. Moreover, this system automatically generates rules from numerical data. Applications of ANBLIR to pattern recognition on numerical examples using benchmark databases are indicated.

Keywords: neuro-fuzzy systems, soft computing, fuzzy implications, approximate reasoning

1. Introduction

Investigation of inference processes when premises and/or conclusions in if-then rules are fuzzy is still a subject of many papers (Cao and Kandel, 1989; Czogała and Kowalczyk, 1996; Cordon *et al.*, 1997; Fodor, 1991; Fodor and Roubens, 1994; Kerre, 1992; Maeda, 1996; Mizumoto and Zimmermann, 1982; Trillas and Valverde, 1985; Weber, 1983). In such processes, a sound and proper choice of logical operators plays an essential role. The theoretical (mathematical) and practical (computational) behavior of logical operators in inference processes has to be known before such a choice is made. Both types of the above mentioned knowledge related to well-known families of triangular norms and implications can also be found in the literature (Fodor, 1991; Fodor and Roubens, 1994; Weber, 1983).

Some selected logical operators and fuzzy implications were also investigated with respect to their behavior in the inference processes. The fuzzy if-then rules have on the one hand a conjunction interpretation and, on the other hand the interpretation in terms of classical logical implications. The inference algorithms based on the conjunctive implication interpretation of if-then rules were simpler and faster with relation to algorithms used for the logical interpretation of such rules. Additionally, applying the conjunctive implication interpretation of if-then rules leads to intuitively better

* Technical University of Silesia, Institute of Electronics, Akademicka 16, 44–100 Gliwice, Poland, e-mail: `jl@biomed.iele.polsl.gliwice.pl`, `nhenzel@zeus.polsl.gliwice.pl`

inference results. In the paper, we present an inference with specific defuzzification that leads to simpler, faster and intuitively acceptable results. An artificial neural network that automatically generates this kind of fuzzy if-then rules is also described.

In the literature, several methods of automatic fuzzy rule generation from given numerical data have been described (Cho and Wang, 1996; Horikawa *et al.*, 1992; Jang and Sun, 1995; Kosko, 1987; Mitra and Pal, 1995; Wang and Mendel, 1992). The fact that there is a functional equivalence between radial basis function networks (RBFNs) and fuzzy systems was used by Jang and Sun (1993) to construct a Sugeno type of adaptive network based fuzzy inference system (ANFIS) which is trained by the back propagation algorithm. Another type of fuzzy system with moving fuzzy sets in the consequents of if-then rules was proposed in (Łęski and Czogała, 1997).

The aim of this paper is a theoretical description and presentation of a new artificial neural network structure based on logical interpretation of if-then rules (ANBLIR). The novelty of the system lies in the introduction of a logical interpretation of fuzzy if-then rules with moving fuzzy sets in the rules' consequents. The described system is applied to benchmark pattern recognition problems.

2. An Approach to Axiomatic Definition of Fuzzy Implications

We start our considerations applying an axiomatic approach (formulated by Fodor (Fodor, 1991; 1995; Fodor and Roubens, 1994)) to the definition of a fuzzy implication, which considers the implication as connective and seems to possess its most general and characteristic properties.

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

I1. If $x \le z$ then $I(x,y) \ge I(z,y)$ for all $x, y, z \in [0,1]$,

I2. If $y \le z$ then $I(x,y) \le I(x,z)$ for all $x, y, z \in [0,1]$,

I3. $I(0,y) = 1$ (falsiness implies anything) for all $y \in [0,1]$,

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

I5. $I(1,0) = 0$ (Booleanity).

Assuming that a strong negation $N : [0,1] \to [0,1]$ is a strictly decreasing continuous function, $N(0) = 1$, $N(1) = 0$, $N(N(x)) = x$ for all $x \in [0,1]$, the N – reciprocal of I defined by

$$\underset{x,y\in[0,1]}{\forall} I_N(x,y) = I\big(N(y), N(x)\big) \tag{1}$$

is also considered to be a fuzzy implication.

Now let us recall further properties, in terms of the function I, which could also be important in some applications:

I6. $I(1,x) = x$ (tautology cannot justify anything) for all $x \in [0,1]$,

I7. $I(x, I(y,z)) = I(y, I(x,z))$ (exchange principle) for all $x,y,z \in [0,1]$,

I8. $x \leq y$ iff $I(x,y) = 1$ (implication defines ordering) for all $x,y \in [0,1]$,

I9. $I(x,0) = N(x)$ for all $x \in [0,1]$ is a strong negation,

I10. $I(x,y) \geq y$ for all $x,y \in [0,1]$,

I11. $I(x,x) = 1$ (identity principle) for all $x \in [0,1]$,

I12. $I(x,y) = I(N(y), N(x))$ with a strong negation N for all $x,y \in [0,1]$,

I13. I is a continuous function.

The most important fuzzy implications representing the classes of fuzzy implications discussed above are summarized in Table 1.

Table 1. Selected fuzzy implications.

Implication Name	Implication Form
Łukasiewicz	$\min(1-x+y, 1)$
Fodor	$\begin{cases} 1, & x \leq y \\ \max(1-x, y), & x > y \end{cases}$
Reichenbach	$1 - x + xy$
Kleene-Dienes	$\max(1-x, y)$
Zadeh	$\max[1-x, \min(x,y)]$

3. Approximate Reasoning Using Fuzzy Implications and Generalized Modus Ponens

Fuzzy implications are mostly used as a way of interpretation of the if-then rules with fuzzy antecedents and/or fuzzy consequents. Such rules constitute a convenient form of expressing pieces of knowledge and a set of if-then rules forms a fuzzy rule base. Let us consider the canonical form of a fuzzy if-then rule $R^{(k)}$, which includes other types of fuzzy rules and fuzzy propositions as special cases, in the (MISO) form:

$$R^{(k)}: \text{ if } X_1 \text{ is } A_1^{(k)} \text{ and } \dots \text{ and } X_n \text{ is } A_n^{(k)} \text{ then } Y \text{ is } B^{(k)}, \quad (2)$$

where X_i and Y stand for the linguistic variables of the antecedent and the consequent and $A_i^{(k)}, B^{(k)}$ are fuzzy sets in the universes of discourse $\mathbb{X}_i \subset \mathbb{R}$, $\mathbb{Y} \subset \mathbb{R}$, respectively.

Such a linguistic form of fuzzy if-then rule can also be expressed as a fuzzy relation

$$R^{(k)} = \left(A_1^{(k)} \times \cdots \times A_n^{(k)} \Longrightarrow B^{(k)}\right) = \left(\underline{A}^{(k)} \Longrightarrow B^{(k)}\right), \tag{3}$$

where $\underline{A}^{(k)} = A_1^{(k)} \times \cdots \times A_n^{(k)}$ is the fuzzy relation in $\mathbb{X} = \mathbb{X}_1 \times \cdots \times \mathbb{X}_n$ defined by

$$\left(A_1^{(k)} \times \cdots \times A_n^{(k)}\right)(x_1, \ldots, x_n) = A_1^{(k)}(x_1) *_T \cdots *_T A_n^{(k)}(x_n) = \underline{A}^{(k)}(\underline{x}), \tag{4}$$

and $*_T$ denotes the respective t-norm T (Weber, 1983).

Fuzzy if-then rules may be interpreted in two ways: as a conjunction of the antecedent and the consequent (Mamdani combination) or as a fuzzy implication (Czogała and Kowalczyk, 1996; Dubois and Prade, 1991; 1996; Weber, 1983; Yager, 1996). In this paper, we mainly exploit the second interpretation.

Approximate reasoning is usually executed in a fuzzy inference system which performs a mapping from an input fuzzy set A' in $\underline{X}$ to a fuzzy set B' in Y via a fuzzy rule base. Two methods of approximate reasoning can be used: a composition based inference (first aggregate then inference—FATI) and individual rule based inference (first inference then aggregate—FITA).

In the composition based inference, a finite number of rules $k = 1, \ldots, K$ are aggregated via intersection or average operations, i.e.:

$$R = \bigcap_{k=1}^{K}{}_{T,\Sigma} R^{(k)}, \tag{5}$$

where $\bigcap_{T,\Sigma}$ denotes the symbol of the aggregation operation using t-norm T or averages (e.g., a normalized arithmetic sum) for aggregation of the respective membership functions:

$$R(\underline{x}, y) = R^{(1)}(\underline{x}, y) \begin{bmatrix} *_T \\ + \end{bmatrix} \cdots \begin{bmatrix} *_T \\ + \end{bmatrix} R^{(K)}(\underline{x}, y). \tag{6}$$

Taking into account an arbitrary input fuzzy set $\underline{A}'$ in $\underline{\mathbb{X}}$ and using the generalized modus ponens we obtain the output of the fuzzy inference (FATI):

$$B' = \underline{A}' \circ R = \underline{A}' \circ \bigcap_{k=1}^{K}{}_{T,\Sigma} R^{(k)} = \underline{A}' \circ \bigcap_{k=1}^{K}{}_{T,\Sigma} \left(\underline{A}^{(k)} \Longrightarrow B^{(k)}\right) \tag{7}$$

or, in terms of membership functions,

$$B'(y) = \sup_{\underline{x} \in \underline{X}} \left[\underline{A}'(\underline{x}) *_{T'} R(\underline{x}, y)\right] = \sup_{\underline{x} \in \underline{X}} \left[\underline{A}'(\underline{x}) *_{T'} \begin{bmatrix} \bigwedge_{k=1}^{K}{}_T \\ \sum_{k=1}^{K} \end{bmatrix} R^{(k)}(\underline{x}, y)\right], \tag{8}$$

where $\wedge_T, *_{T'}$ denote t-norms T, T' for aggregation and composition, respectively. The symbol Σ denotes the normalized arithmetic sum as aggregation.

In an individual rule based inference (FITA) each rule in the fuzzy rule base determines an output fuzzy set and after that an aggregation via intersection or an average operation is performed. So the output fuzzy set is expressed by means of the formulas:

$$B'' = \bigcap_{k=1}^{K}{}_{T,\Sigma} \left\{ \underline{A}' \circ \left(\underline{A}^{(k)} \Longrightarrow B^{(k)} \right) \right\} \tag{9}$$

or

$$B''(y) = \begin{bmatrix} \bigwedge_{k=1}^{K}{}_T \\ \sum_{k=1}^{K} \end{bmatrix} \sup_{\underline{x} \in \underline{X}} \left[\underline{A}'(\underline{x}) *_{T'} R^{(k)}(\underline{x}, y) \right]. \tag{10}$$

It can be proved that B' is more specified than B'', i.e.

$$B' \subseteq B'' \quad \text{or} \quad \forall_{y \in Y} B'(y) \le B''(y). \tag{11}$$

This means that the consequent B' is equal to or contained in the intersection of fuzzy inference results—B''. For simplicity of calculation, the consequent B' is replaced by B'' while supposing that the differences are not too excessive.

If the input fuzzy sets $A'_1, \ldots, A'_n$ or $(\underline{A}')$ are singletons in $x_{10}, \ldots, x_{n0}$ or $(\underline{x}_0)$, the consequent B' is equal to B'' $(B'(y) = B''(y))$. In this case we obtain

$$B'(y) = B''(y) = \begin{bmatrix} \bigvee_{k=1}^{K}{}_S \\ \sum_{k=1}^{K} \end{bmatrix} \left[\underline{A}^{(k)}(\underline{x}_0) *_{T'} B^{(k)} \right] \tag{12}$$

or, with a logical interpretation of the fuzzy implication,

$$B'(y) = B''(y) = \begin{bmatrix} \bigwedge_{k=1}^{K}{}_T \\ \sum_{k=1}^{K} \end{bmatrix} I\left(\underline{A}^{(k)}(\underline{x}_0), B^{(k)} \right). \tag{13}$$

4. Fundamentals of Fuzzy Systems

Assume that m numbers of n-input and one-output (MISO) fuzzy if-then rules are given. The k-th rule in which the consequent is represented by a linguistic variable Y may be written in the following form:

$$R^{(k)}: \text{ if } X_1 \text{ is } A_1^{(k)} \text{ and } \ldots \text{ and } X_n \text{ is } A_n^{(k)} \text{ then } Y \text{ is } B^{(k)} \tag{14}$$

or, in a pseudo-vector notation,

$$R^{(k)}: \text{ if } \underline{X} \text{ is } \underline{A}^{(k)} \text{ then } Y = B^{(k)}, \tag{15}$$

where

$$\underline{X} = [X_1\ X_2\ \dots\ X_n]. \tag{16}$$

$X_1, X_2, \dots, X_n$ and Y are linguistic variables which may be interpreted as the inputs of a fuzzy system and the output of that system. $A_1^{(k)}, \dots, A_n^{(k)}$ represent the linguistic values of the linguistic variables $X_1, X_2, \dots, X_n$ and $B^{(k)}$ is the linguistic value of the linguistic variable Y.

A collection of the above rules for $k = 1, 2, \dots K$ forms a rule base which may be activated (fired) under the singleton inputs:

$$X_1 \text{ is } x_{10} \text{ and } \dots \text{ and } X_n \text{ is } x_{n0} \tag{17}$$

or

$$\underline{X} \text{ is } \underline{x}. \tag{18}$$

It can easily be concluded from (13) that for such a type of reasoning the inferred value of the k-th rule output for crisp inputs (singletons) may be written for the logical implication interpretation in the form

$$B'(y) = R_k(\underline{x}_0) \Longrightarrow B^{(k)}(y) = I\big(R_k(\underline{x}_0), B^{(k)}(y)\big) \tag{19}$$

and for the conjunctive implication interpretation as

$$B'(y) = R_k(\underline{x}_0) *_T B^{(k)}(y) = *_T\big(R_k(\underline{x}_0), B^{(k)}(y)\big), \tag{20}$$

where '$\Longrightarrow$' stands for the fuzzy implication and

$$R_k(\underline{x}_0) = A_1^{(k)}(x_{10}) \text{ and } \dots \text{ and } A_n^{(k)}(x_{n0}) = \underline{A}^{(k)}(\underline{x}_0) \tag{21}$$

denotes the degree of activation (the firing strength) of the k-th rule with respect to the minimum ($\wedge$) or the product ($\cdot$). The latter represents an explicit connective (AND) of the predicates X_i is $A_i^{(k)}$; $k = 1, 2, \dots, n$ in the antecedent of an if-then rule.

A crisp value of the output can be obtained using the Modified Center of Gravity (MCOG) as defuzzification (Czogała and Łęski, 1998):

$$\mathrm{MCOG}\big[B(x)\big] = \frac{\int x\,[B(x) - \alpha]\,\mathrm{d}x}{\int [B(x) - \alpha]\,\mathrm{d}x}, \tag{22}$$

where α is constant. The subtraction of $\alpha \in [0, \min_x B(x)]$ eliminates the non-informative part of the membership function $B(x)$ and leads to better inference results in the case of a logical interpretation of the implication. For $\alpha = 0$ we get the well-known COG defuzzification. A final crisp value of the system output for the sum as aggregation and MCOG defuzzification can be evaluated from

$$y_0 = \frac{\int y \sum_{k=1}^{K} \left\{ \Psi\left[R_k(\underline{x}_0), B^{(k)}(y)\right] - \alpha_k \right\} \mathrm{d}y}{\int \sum_{k=1}^{K} \left\{ \Psi\left[R_k(\underline{x}_0), B^{(k)}(y)\right] - \alpha_k \right\} \mathrm{d}y}$$

$$= \frac{\int y \sum_{k=1}^{K} \left[B'^{(k)}(y) - \alpha_k\right] \mathrm{d}y}{\int \sum_{k=1}^{K} \left[B'^{(k)}(y) - \alpha_k\right] \mathrm{d}y}, \tag{23}$$

where Ψ stands for the fuzzy implication I or t-norm T for the logical or conjunctive implication interpretations, respectively. A method of determining the values α will be described later. Now we set $B^{*(k)} := B'^{(k)} - \alpha_k$. The membership functions of the fuzzy sets $B^{*(k)}$ can be represented by the parameterized functions

$$B^{*(k)} \backsim f^{(k)} \left[\mathrm{Area}\left(B^{*(k)}\right), y^{(k)}\right], \tag{24}$$

where

$$y^{(k)} = \frac{\int y B^{*(k)}(y)\, \mathrm{d}y}{\int B^{*(k)}(y)\, \mathrm{d}y} \tag{25}$$

is the center of gravity (COG) of the fuzzy set $B^{*(k)}$.

Consequently, the final output value can be written down in the form

$$y_0 = \frac{\sum_{k=1}^{K} y^{(k)} \mathrm{Area}\left(B^{*(k)}\right)}{\sum_{k=1}^{K} \mathrm{Area}\left(B^{*(k)}\right)}, \tag{26}$$

where $B^{*(k)}$ is the resulting conclusion for the k-th rule before aggregation.

Note that fuzzy systems with Larsen's product as the conjunctive 'fuzzy implication' of the if-then rules and symmetric triangle (isosceles triangle) membership functions for consequents $B^{*(k)}$ can be calculated using the well-known formula

$$y_0 = \frac{\sum_{k=1}^{K} \frac{w^{(k)}}{2} R_k\left(\underline{x}_0\right)\, y^{(k)}}{\sum_{k=1}^{K} \frac{w^{(k)}}{2} R_k\left(\underline{x}_0\right)}, \tag{27}$$

where $w^{(k)}$ is the width of the triangle base for the k-th rule. It should be noted that the factor $w^{(k)}/2$ may be interpreted as a respective weight of the k-th rule or its certainty factor.

Another important fuzzy system is the so called Takagi-Sugeno-Kang system. Assume that m numbers of n-input and one-output (MISO) fuzzy implicative rules or fuzzy conditional statements are given. The k-th rule can be written in the form

$$R^{(k)}: \text{ if } X_1 \text{ is } A_1^{(k)} \text{ and } \dots \text{ and } X_n \text{ is } A_n^{(k)} \text{ then } Y = f^{(k)}(X_1, \dots, X_n) \tag{28}$$

or, in a pseudo-vector notation,

$$R^{(k)}: \text{ if } \underline{X} \text{ is } \underline{A}^{(k)} \text{ then } Y = f^{(k)}(\underline{X}). \tag{29}$$

A crisp value of the output for Larsen's fuzzy relation (product) and aggregation (normalized sum) can be evaluated from (Cho and Wang, 1996)

$$y_0 = \frac{\sum_{k=1}^{K} A_k(\underline{x}_0)\, f^{(k)}(\underline{x}_0)}{\sum_{k=1}^{K} A_k(\underline{x}_0)}. \tag{30}$$

Taking into account that

$$f^{(k)}(\underline{x}_0) = p_0^{(k)}, \tag{31}$$

where $p_0^{(k)}$ is a crisply defined constant in the consequent of the k-th rule. Such a model is called the zeroth-order Sugeno fuzzy model. A more general first-order Sugeno fuzzy model is of the form

$$f^{(k)}(\underline{x}_0) = p_0^{(k)} + p_1^{(k)} x_{10} + \dots + p_n^{(k)} x_{n0}, \tag{32}$$

where $p_0^{(k)}, p_1^{(k)}, \dots, p_n^{(k)}$ are all constant.

In vector notation it takes the form

$$f^{(k)}(\underline{x}_0) = \underline{p}^{(k)T} \underline{x}_0', \tag{33}$$

where

$$\underline{x}_0' = \begin{bmatrix} 1 \\ \underline{x}_0 \end{bmatrix} \tag{34}$$

denotes the extended input vector. Notice that in both the models the consequent is crisp.

In (26) the value describing the location of COGs for consequent fuzzy sets in if-then rules is constant and equals $y^{(k)}$ for the k-th rule. A natural extension of the situation described above is the assumption that the location of the consequent fuzzy set is a linear combination of all inputs for the k-th rule:

$$y^{(k)}(\underline{x}_0) = \underline{p}^{(k)T} \underline{x}_0'. \tag{35}$$

Hence we get the final output value in the form

$$y_0 = \frac{\sum_{k=1}^{K} \text{Area}\left(B^{*(k)}\right) \underline{p}^{(k)T} \underline{x}_0'}{\sum_{k=1}^{K} \text{Area}\left(B^{*(k)}\right)}, \tag{36}$$

where $B^{*(k)}$ is the conclusion for the k-th rule before aggregation.

5. Fuzzy System with Logical Interpretation of If-Then Rules (ANBLIR)

We assume that the premises of the if-then rules $A_1^{(k)}, \ldots, A_n^{(k)}$ have Gaussian membership functions:

$$A_j^{(k)}(x_{j0}) = \exp\left[-\frac{\left(x_{j0} - c_j^{(k)}\right)^2}{2\left(s_j^{(k)}\right)^2}\right], \tag{37}$$

where $c_j^{(k)}$ and $s_j^{(k)}$ for $j = 1, 2, \ldots, n$ and $k = 1, 2, \ldots, K$ are parameters. On the basis of (21) and for the explicit connective AND taken as the product we get

$$\underline{A}^{(k)}(\underline{x}_0) = \prod_{j=1}^{n} A_j^{(k)}(x_{j0}). \tag{38}$$

On the basis of (37), we get

$$R_k(\underline{x}_0) = \exp\left[-\sum_{j=1}^{n} \frac{\left(x_{j0} - c_j^{(k)}\right)^2}{2\left(s_j^{(k)}\right)^2}\right]. \tag{39}$$

Additionally, we assume that the consequents $B^{(k)}$ of the k-th if-then rule have symmetric triangle (isosceles triangle) membership functions with the width of the triangle base $w^{(k)}$. For computing the system output we must calculate $\text{Area}\,(B^{*(k)})$. From (19) and (23), we have

$$B^{*(k)}(y) = I\left[R_k(\underline{x}_0), B^{(k)}\right] - \alpha_k. \tag{40}$$

For an implication satisfying I9 we assume that

$$\alpha_k = 1 - R_k(\underline{x}_0). \tag{41}$$

For example, if we use the Reichenbach implication, we get

$$\begin{aligned}
\text{Area}\left(B^{*(k)}\right) &= 2 \int_{y^{(k)} - w^{(k)}/2}^{y^{(k)}} \left\{I\left[R_k(\underline{x}_0), B^{(k)}\right] - \alpha_k\right\} \mathrm{d}y \\
&= 2 \int_{y^{(k)} - w^{(k)}/2}^{y^{(k)}} \left[1 - R_k(\underline{x}_0) + R_k(\underline{x}_0) B^{(k)} - 1 + R_k(\underline{x}_0)\right] \mathrm{d}y \\
&= 2R_k(\underline{x}_0) \int_{y^{(k)} - w^{(k)}/2}^{y^{(k)}} \left[\frac{2\left(y - y^{(k)}\right)}{w^{(k)}} - 1\right] \mathrm{d}y \\
&= \frac{w^{(k)}}{2} R_k(\underline{x}_0) := g\left[R_k(\underline{x}_0), w^{(k)}\right].
\end{aligned} \tag{42}$$

This situation is graphically illustrated in Fig. 1.

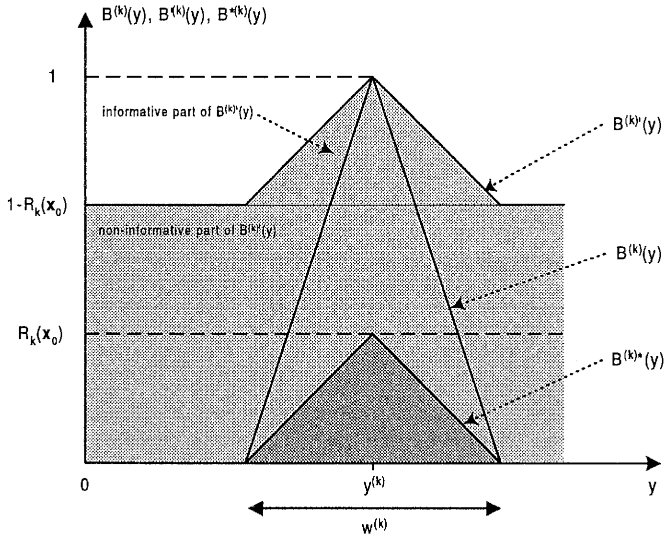

Fig. 1. Informative and non-informative parts for the resulting conclusion before the aggregation using the Reichenbach fuzzy implication.

The respective formulas for $g\left[R_k(\underline{x}_0), w^{(k)}\right]$ for other implications are presented in Table 2 (for simplicity, the abbreviated forms are used: $R \triangleq R_k(\underline{x}_0)$, $w \triangleq w^{(k)}$). If we use the symbols from the table, then (36) takes the form

$$y_0 = \frac{\sum_{k=1}^{K} g\left[R_k(\underline{x}_0), w^{(k)}\right]\ \underline{p}^{(k)T}\underline{x}_0'}{\sum_{k=1}^{K} g\left[R_k(\underline{x}_0), w^{(k)}\right]}. \tag{43}$$

For n inputs and K if-then rules we have to determine the following unknown parameters:

- $c_j^{(k)}, s_j^{(k)}$, $j = 1, 2, \ldots, n$; $k = 1, 2, \ldots, K$, i.e., the parameters of the membership functions of the input sets,
- $p_j^{(k)}$, $j = 0, 1, \ldots, n$; $k = 1, 2, \ldots, K$, i.e., the parameters determining the location of the output sets,
- $w^{(k)}$, $k = 1, 2, \ldots, K$, i.e., the parameters of the output sets.

Obviously, in real problems, the number of if-then rules is unknown. Let us observe that (39) and (43) describe a radial-like neural network. The unknown parameters (except the number of rules K) are estimated by means of a gradient method performing the steepest descent on a surface in the parameter space. Therefore the

Table 2. Some derivatives required for selected implications.

Implication	$g(R,w)$
Łukasiewicz	$\frac{w}{2}\left(2R-R^2\right)$
Fodor	$\begin{cases} \frac{w}{2}\left(1-2R+R^2\right) & \text{if } R>\frac{1}{2} \\ \frac{w}{2}\left(2R-R^2\right) & \text{if } R\le\frac{1}{2} \end{cases}$
Reichenbach	$\frac{w}{2}R$
Kleene-Dienes	$\frac{w}{2}R^2$
Zadeh	$\begin{cases} \frac{w}{2}(2R-1) & \text{if } R\ge\frac{1}{2} \\ 0 & \text{if } R<\frac{1}{2} \end{cases}$

so-called learning set is necessary, i.e., a set of inputs for which the output values are known $\{\underline{x}_0(i), t_0(i)\}$, $i = 1, 2, \ldots, N$. The measure of the output error may be defined for a single pair from the training set:

$$E = \frac{1}{2}\left(t_0 - y_0\right)^2, \tag{44}$$

where t_0 stands for the desired (target) output value.

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

$$(\alpha)_{\text{new}} = (\alpha)_{\text{old}} - \eta \left.\frac{\partial E}{\partial \alpha}\right|_{\alpha=(\alpha)_{\text{old}}}, \tag{45}$$

where η signifies the learning rate.

The partial derivatives of E with respect to the unknown parameters are of the form

$$\frac{\partial E}{\partial c_j^{(k)}} = (y_0 - t_0)\frac{[y^{(k)}(\underline{x}_0) - y_0]R_k(\underline{x}_0)}{\sum\limits_{i=1}^{K} g\left[R_i(\underline{x}_0), w^{(i)}\right]}\frac{\partial g\left[R_k(\underline{x}_0), w^{(k)}\right]}{\partial R_k(\underline{x}_0)}\frac{x_{j0} - c_j^{(k)}}{2\left(s_j^{(k)}\right)^2}, \tag{46}$$

$$\frac{\partial E}{\partial s_j^{(k)}} = (y_0 - t_0)\frac{[y^{(k)}(\underline{x}_0) - y_0]R_k(\underline{x}_0)}{\sum\limits_{i=1}^{K} g\left[R_i(\underline{x}_0), w^{(i)}\right]}\frac{\partial g\left[R_k(\underline{x}_0), w^{(k)}\right]}{\partial R_k(\underline{x}_0)}\frac{(x_{j0} - c_j^{(k)})^2}{2\left(s_j^{(k)}\right)^3}, \tag{47}$$

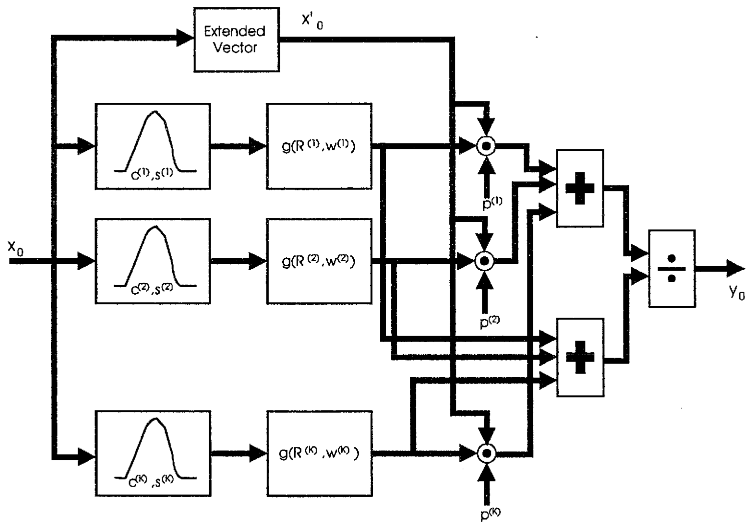

Fig. 2. Graphical illustration of an artificial neural network based on the logical interpretation of if-then rules (ANBLIR).

$$\underset{j\neq 0}{\forall}\ \frac{\partial E}{\partial p_j^{(k)}} = (y_0 - t_0)\,\frac{g\left[R_k(\underline{x}_0), w^{(k)}\right]}{\sum\limits_{i=1}^{K} g\left[R_i(\underline{x}_0), w^{(i)}\right]}\,x_{j0}, \tag{48}$$

$$\frac{\partial E}{\partial p_0^{(k)}} = (y_0 - t_0)\,\frac{g\left[R_k(\underline{x}_0), w^{(k)}\right]}{\sum\limits_{i=1}^{K} g\left[R_i(\underline{x}_0), w^{(i)}\right]}, \tag{49}$$

$$\frac{\partial E}{\partial w^{(k)}} = (y_0 - t_0)\,\frac{y^{(k)}\left(\underline{x}_0\right) - y_0}{\sum\limits_{i=1}^{K} g\left[R_i(\underline{x}_0), w^{(i)}\right]}\,\frac{\partial g\left[R_k(\underline{x}_0), w^{(k)}\right]}{\partial w^{(k)}}. \tag{50}$$

The unknown parameters can be modified on the basis of (45) after collecting each input-output pair or after collecting all such pairs (cumulative method). Additionally, the following heuristic rules for changes in η may be applied (Jang *et al.*, 1997). If the mean square error in four consecutive iterations has decreased for the whole learning set, then the learning parameter is increased (multiplied by n_I). If the error in four consecutive iterations has increased and decreased, then the learning parameter is decreased (multiplied by n_D).

Another solution accelerating the convergence of the method is the estimation of parameters $\underline{p}^{(k)}$, $k = 1, \ldots, K$ by means of the least-squares method. The output value y_0 of the system in (43) may be considered to be a linear combination of the unknown parameters $\underline{p}^{(k)}$. If we introduce the notation

$$S^{(k)}\left(\underline{x}_0\right) = \frac{g\left[R_k(\underline{x}_0), w^{(k)}\right]}{\sum_{i=1}^{K} g\left[R_i(\underline{x}_0), w^{(i)}\right]}, \tag{51}$$

$$\underline{D}\left(\underline{x}_0\right) = \left[S^{(1)} x_0'^T \;\vdots\; S^{(2)} x_0'^T \;\vdots\; \cdots \;\vdots\; S^{(K)} x_0'^T\right]^T, \tag{52}$$

$$\underline{P} = \left[\underline{p}^{(1)T} \;\vdots\; \underline{p}^{(2)T} \;\vdots\; \cdots \;\vdots\; \underline{p}^{(K)T}\right]^T, \tag{53}$$

eqn. (43) can be written in the form:

$$y_0 = \underline{D}\left(\underline{x}_0\right)^T \underline{P}. \tag{54}$$

Hence the parameters $\underline{P}$ may be estimated via the least-squares method. To eliminate matrix inversion, we use a recurrent method. For the k-th step (k-th element from the learning set), we get (de Larminat and Thomas, 1977):

$$\widehat{\underline{P}}(k) = \widehat{\underline{P}}(k-1) + \underline{G}(k-1)\underline{D}\left[\underline{x}_0(k)\right]\left\{y_0(k) - \underline{D}\left[\underline{x}_0(k)\right]^T \widehat{\underline{P}}(k-1)\right\}, \tag{55}$$

$$\underline{G}(k) = \underline{G}(k-1) - \underline{G}(k-1)\underline{D}\left[\underline{x}_0(k)\right] \times \left\{\underline{D}\left[\underline{x}_0(k)\right]^T \underline{G}(k-1)\underline{D}\left[\underline{x}_0(k)\right] + 1\right\}^{-1} \underline{D}\left[\underline{x}_0(k)\right]^T \underline{G}(k-1). \tag{56}$$

To initialize the computations, we take

$$\begin{cases} \widehat{\underline{P}}(0) = \underline{0}, \\ \underline{G}(0) = \beta \mathbb{I}, \end{cases} \tag{57}$$

where $\mathbb{I}$ is the identity matrix and β denotes a large positive constant (e.g., 10^6). Finally, in each iteration the parameters $\underline{p}^{(k)}$ are estimated on the basis of (55) and (56), whereas the other parameters by means of the gradient method (45)–(47), (50).

Another problem is the estimation of the number m of if-then rules and initial values of the membership functions for the premise part. This task is solved by means of preliminary clustering of the input part of the training data using a fuzzy c-means method (Bezdek, 1981; Pal and Bezdek, 1995). This method assigns each input vector $\underline{x}_0(k)$, $k = 1, 2, \ldots, N$ to clusters represented by prototypes $\underline{v}_i$, $i = 1, \ldots c$ measured by grades of membership $u_{ik} \in [0, 1]$. The $c \times n$ partition matrix satisfies the following assumptions:

$$\begin{cases} \forall_k \; \sum_{i=1}^{c} u_{ik} = 1, \\ \forall_j \; \sum_{k=1}^{N} u_{ik} \in (0, N). \end{cases} \tag{58}$$

The c-means method minimizes the scalar index

$$J_r = \sum_{k=1}^{N} \sum_{i=1}^{c} u_{ik}^{r} \left\| \underline{x}_0 (k) - \underline{v}_i \right\|^2 \tag{59}$$

with respect to $r > 1$.

Defining $D_{ik} = \|\underline{x}_0 (k) - \underline{v}_i\|$, where $\|\cdot\|$ is a vector norm, we get an iterative method of the consecutive modification of the partition matrix and prototypes (Bezdek, 1981):

$$\underset{i}{\forall} \quad \underline{v}_i = \frac{\sum\limits_{k=1}^{N} u_{ik}^{r} \, \underline{x}_0 (k)}{\sum\limits_{k=1}^{N} u_{ik}^{r}}, \tag{60}$$

$$\underset{i,k}{\forall} \quad u_{ik} = \left[\sum_{j=1}^{c} \left(\frac{D_{ik}}{D_{jk}} \right)^{\frac{2}{r-1}} \right]^{-1}. \tag{61}$$

Accordingly, the obtained calculations are initialized using a random partition matrix $\underline{U}$ which satisfies (58). Such a method leads to a local minimum for (59). Therefore, the most frequently used solution is baseed on multiple repeated calculations in accordance with (60), (61) for various random realizations of partition matrix initializations. The computation is stopped when a predefined number of iterations (in our case 500) are executed or when in two consecutive iterations the change in J_r is less than an imposed value (in our case 0.001).

As a result of preliminary clustering, the following assumption on the ANBLIR initialization can be made: $c^{(j)} = v_j$ for $j = 1, 2, \ldots, K$ and

$$s^{(j)} = \frac{\sum\limits_{k=1}^{N} u_{ik}^{r} [\, \underline{x}_0 (k) - \underline{v}_i]^2}{\sum\limits_{k=1}^{N} u_{ik}^{r}}. \tag{62}$$

For the calculations presented in the next section, the Reichenbach fuzzy implication due to the simplicity of the $g\,(R, w)$ function was applied. The ANBLIR parameters were set to the values $\eta = 0.01$, $n_I = 1.1$, $n_D = 0.9$, $\beta = 10^6$, $r = 2$.

6. Application of ANBLIR to Pattern Recognition

In this section, we present an application of the proposed fuzzy system to a pattern recognition problem. If the patterns from a learning set belong to classes ω_1 and ω_2, then we can build a fuzzy system whose output takes positive values for the patterns from ω_1 and negative or zero values for ω_2. If we denote by $y_0 = \Phi(\underline{x}_0)$ the fuzzy system, we get

$$y_0(k) = \Phi_{12}\left[\underline{x}_0(k)\right] \begin{cases} > 0 & \text{if } \underline{x}_0(k) \in \omega_1, \\ \leq 0 & \text{if } \underline{x}_0(k) \in \omega_2. \end{cases} \tag{63}$$

During the learning process of a classifier, we take $t_0(k) = 1$ for each pattern $\underline{x}_0(k)$ from class ω_1 and $t_0(k) = -1$ for each pattern from class ω_2. For a larger number of classes $(\omega_1, \omega_2, \ldots, \omega_p,\ p > 2)$, an extension class-rest or class-class can be used (Ripley, 1996; Tou and Gonzalez, 1974). The latter was used in our method due to the existence of common feature regions for which the classifier class-rest does not give the answer which class the classified pattern belongs to. The disadvantage of such a solution is the necessity of constructing a larger number of classifiers. Let us denote by

$$y_0(k) = \Phi_{ij}\left[\underline{x}_0(k)\right] \begin{cases} > 0 & \text{if } \underline{x}_0(k) \in \omega_i, \\ \leq 0 & \text{if } \underline{x}_0(k) \in \omega_j \end{cases} \tag{64}$$

the classifier making the decision whether a pattern belongs to the i-th or the j-th class.

Obviously, we do not construct the classifier Φ_{ii} and the information about the membership to the i-th and the j-th classes can be obtained on the basis of the Φ_{ij} or Φ_{ji} classifiers. Hence we construct $p(p-1)/2$ classifiers Φ_{ij} for $1 \leq i < p,\ j > i$. The classification condition for the i-th class has the form

$$\underset{j \neq i}{\forall} \quad \Phi_{ij}\left[\underline{x}_0(k)\right] > 0 \Longrightarrow \underline{x}_0(k) \in \omega_i. \tag{65}$$

The learning process runs as follows: for each pair of indices ij $(1 \leq i < p,\ j > i)$ we assume $t_0(k) = 1$ for a pattern $x_0(k)$ belonging to class ω_i and $t_0(k) = -1$ for a pattern $x_0(k)$ belonging to class ω_j (the patterns belonging to other classes are removed from the training set) and we conduct the learning process of the classifier. The final pattern classification is made based on the condition (65).

7. Numerical Examples

All the databases presented in this section were obtained from the UCI machine learning repository (`http://www.ics.uci.edu/~mlearn/MLSummary.html`). These standard databases are commonly used for evaluating the performances of classifiers.

7.1. Application to Forensic Glass Classification

The data from forensic glass tests were collected by B. German on 214 fragments of glass. Each case has a measured refractive index and composition weight percentage of oxides of Na, Al, Mg, Si, K, Ca, Fe and Ba. The fragments were classified into six types: window float (WinF, 70 cases), window non-float (WinNF, 76 cases), vehicle window (Veh, 17 cases), containers (Con, 13 cases), tableware (Tabl, 9 cases) and vehicle headlamps (Head, 29 cases). This database had been tested exhaustively using standard methods of pattern recognition in (Ripley, 1996). The obtained error rates are as follows: a linear classifier—38%, logistic discrimination—26.2%, a neural network (back-propagation with eight hidden units)—24.8%, a nearest neighbor method—23.4%, learning vector quantization—29.9% and a tree-structured classifier—32.2%. The method of classifier construction proposed in this paper was also applied to this database. 500 iterations of learning were executed for each classifier. The number of if-then rules varied from 2 to 5. The error rates equal: 18.22% (two rules), 12.62% (three rules), 10.75% (four rules) and 7.48% (five rules) with the confusion matrix presented in Table 3.

Table 3. Simulation results for classification of the forensic glass.

	WinF	WinNF	Veh	Con	Tabl	Head
WinF	66	3	1	0	0	0
WinNF	8	68	0	0	0	0
Veh	3	1	13	0	0	0
Con	0	0	0	13	0	0
Tabl	0	0	0	0	9	0
Head	0	0	0	0	0	29

7.2. Application to the Famous Iris Problem

The iris database is perhaps the best known database to be found in the pattern recognition literature. The data set contains 3 classes of 50 instances each, where each class refers to a type of the iris plant. The vector of features consists of the following elements: sepal length in cm, sepal width in cm, petal length in cm, and petal width in cm. We consider three classes of patterns: Iris Setosa, Iris Versicolour and Iris Virginica. The confusion matrix for 500 learning iterations and two if-then rules is shown in Table 4. The error rate is equal to 1.33%. The increase in the rule number did not cause any decrease in the error rate.

Table 4. Simulation results for classification of the iris database.

	Iris Setosa	Iris Versicolour	Iris Virginica
Iris Setosa	50	0	0
Iris Versicolour	0	50	0
Iris Virginica	0	2	48

7.3. Application to Wine Recognition Data

These databases are the results of a chemical analysis of wine produced in the same region in Italy but coming from three different producers. The analysis determined the quantities of 13 constituents found in each of the three types of wine. The data were collected by M. Forina and used by many others for comparing various classifiers. The classes are separable, though only a radial discriminant analysis achieved 100% of correct classifications (RDA: 100%, QDA: 99.4%, LDA: 98.9%, 1NN: 96.1%). 500 learning iterations were executed for the classifier described in Section 6. Correct classifications 99.43 and 100% were obtained.

7.4. Application to MONK's Problems

The MONK's problem was the basis of the first international comparison of learning algorithms. The result of this comparison is summarized in (Thrun *et al.*, 1991). One significant characteristic of this comparison is that it was performed by a collection of researchers, each of whom was an advocate of the technique they tested (often they were the authors of various methods).

In this sense, the results are less biased than those obtained by a single person advocating a specific learning method, and more accurately reflect the generalization behavior of the learning techniques as applied by knowledgeable users. There are three MONK's problems. The domains for all of them are the same. One of the MONK's problems has noise added. For each problem, the domain was partitioned into training and testing sets. The vector of features for each pattern consists of 7 features which take the following values: first feature—1,2,3, second—1,2,3, third—1,2, fourth—1,2,3, fifth—1,2,3,4, sixth—1,2. The patterns were classified into two classes. Taken from (Thrun *et al.*, 1991), the results of testing for various methods are collected in Table 5. It should be pointed out that methods which gave the highest percentage of correct classification were selected. The testing results obtained by means of the method described in this paper are presented in Table 5 as well. The number of if-then rules varied from 2 to 4 and the number of executed iterations varied from 25 to 6000 depending on the considered problem.

8. Conclusions

In some cases the inference algorithms based on conjunctive operators seem faster, simpler and more exact than fuzzy implication based inference systems. Moreover, the interpretation of the fuzzy if-then rules based on fuzzy implications is sounder from the logical point of view.

In this paper, a new artificial neural network based on a logical interpretation of if-then rules (ANBLIR) has been described. Such a system can be used for an automatic if-then rule generation. The novelties of this system in comparison with the well-known solutions from the literature are the logical interpretation of fuzzy if-then rules and moving fuzzy sets in consequents. A combination of gradient and

Table 5. Simulation results for classification of the MONKS problems.

Method	MONKS-1	MONKS-2	MONKS-3
ANBLIR, $K = 2$	97.9%	88.8%	92.9%
ANBLIR, $K = 3$	100%	100%	97.6%
ANBLIR, $K = 4$	100%	100%	95.5%
AQ-15 Genetic	100%	86.8%	100%
Assistant Professional	100%	81.3%	100%
NN with weight decay	100%	100%	97.2%
Cascade Correlation	100%	100%	97.2%
CN2	100%	69.0%	89.1%
ECOBWEB	71.8%	67.4%	68.2%
ID5R-hat	90.3%	65.7%	—
mFOIL	100%	69.2%	100%
PRISM	86.3%	72.7%	90.3%

least-squares methods of parameter optimization for ANBLIR was used. For initialization of calculations preliminary fuzzy c-means clustering was applied. A promising application of the system to standard pattern recognition problems was demonstrated.

References

Bezdek J.C. (1981): *Pattern Recognition with Fuzzy Objective Function Algorithms.* — New York: Plenum.

Cao Z. and Kandel A. (1989): *Applicability of some fuzzy implication operators.* — Fuzzy Sets Syst., Vol.31, No.2, pp.151–186.

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.

Cordon O., Herrera F. and Peregrin A. (1997): *Applicability of the fuzzy operators in the design of fuzzy logic controllers.* — Fuzzy Sets Syst., Vol.86, No.1, pp.15–41.

Czogała E. and Kowalczyk R. (1996): *Investigation of selected fuzzy operations and implications in engineering.* — Proc. 5th IEEE Int. Conf. *Fuzzy Systems*, New Orleans, pp.879–885.

Czogała E. and Łęski J. (1998): *An equivalence of approximate reasoning under defuzzification.* — BUSEFAL, Vol.74, pp.83–92.

Dubois D. and Prade H. (1991): *Fuzzy sets in approximate reasoning*, Part 1: *Inference with possibility distributions.* — Fuzzy Sets Syst., Vol.40, No.2, pp.143–202.

Dubois D. and Prade H. (1996): *What are fuzzy rules and how to use them.* — Fuzzy Sets Syst., Vol.84, No.2, pp.169–185.

Fodor J.C. (1991): *On fuzzy implication operators.* — Fuzzy Sets Syst., Vol.42, No.3, pp.293–300.

Fodor J.C. (1995): *Contrapositive symmetry of fuzzy implications.* — Fuzzy Sets Syst., Vol.69, No.2, pp.141–156.

Fodor J.C. and Roubens M. (1994): *Fuzzy Preference Modelling and Multicriteria Decision Support.* — Dordrecht: Kluwer.

Horikawa S., Furuhaski T. and Uchikawa Y. (1992): *On fuzzy modeling using fuzzy neural networks with the back-propagation algorithm.* — IEEE Trans. Neural Netw., Vol.4, No.5, pp.801–806.

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

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

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

Kerre E. (1992): *A comparative study of the behavior of some popular fuzzy implication operators on the generalized modus ponens*, In: Fuzzy Logic for the Management of Uncertainty (L.Zadeh and J.Kacprzyk, Eds.). — New York: Wiley, pp.281–295.

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

de Larminat P. and Thomas Y. (1977): *Automatique des systemes lineaires, 2. Identification.* — Paris: Flammarion Sciences.

Łęski J. and Czogała E. (1997): *A new artifical neural network based fuzzy inference system with moving consequents in if-then rules.* — BUSEFAL, Vol.71, pp.72–81.

Maeda H. (1996): *An investigation on the spread of fuzziness in multi-fold multi-stage approximate reasoning by pictorial representation – under sup-min composition and triangular type membership function.* — Fuzzy Sets Syst., Vol.80, No.2, pp.133–148.

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.

Mizumoto M. and Zimmermann H.J. (1982): *Comparison of fuzzy reasoning methods.* — Fuzzy Sets Syst., Vol.8, No.3, pp.253–283.

Pal N.R. and Bezdek J.C. (1995): *On cluster validity for the fuzzy c-means model.* — IEEE Trans. Fuzzy Syst., Vol.3, No.3, pp.370–379.

Ripley B.R. (1996): *Pattern Recognition and Neural Network.* — Cambridge: Cambridge University Press.

Tou J.T. and Gonzalez R.C. (1974): *Pattern Recognition Principles.* — London: Addison-Wesley.

Thrun S.B. *et al.* (1991): *The MONK's problems. A performance comparision of different learning algorithms.* — Scientific Report, No.CMU-CS-91-197, Carnegie Mellon University.

Trillas E. and Valverde L. (1985): *On implication and indistinguishability in the setting of fuzzy logic*, In: Management Decision Support Systems Using Fuzzy Sets and Possibility Theory (J. Kacprzyk and R. Yager, Eds.). — Köln: Verlag TÜV Rheinland, pp.198–212.

Weber S. (1983): *A general concept of fuzzy connectives, negations and implications based on t-norms and t-conorms.* — Fuzzy Sets Syst., Vol.11, No.2, pp.115–134.

Wang L. and Mendel J.M. (1992): *Generating fuzzy rules by learning from examples.* — IEEE Trans. Syst. Man Cybern., Vol.22, No.6, pp.1414–1427.

Yager R.R. (1996): *On the interpretation of fuzzy if-then rules.* — Appl. Intelligence, Vol.6, No.2, pp.141–151.