Neuro-Fuzzy Structures for Pattern Classification

KRZYSZTOF CPAŁKA[1, 2], LESZEK RUTKOWSKI[1, 2]
(1) Department of Computer Engineering
Technical University of Częstochowa
and
(2) Department of Artificial Intelligence
WSHE University in Łódź
POLAND

Abstract:-In the paper we present a new class of neuro-fuzzy systems for pattern classification. The algorithm is based on the concept of adjustable triangular norms. Various flexibility components are incorporated into construction of neuro-fuzzy systems. A high accuracy of the algorithm is demonstrated.

Key-Words:-Fuzzy systems, Pattern classification, Parameterized triangular norms, Soft fuzzy norms, Neuro-fuzzy inference systems.

1. Introduction

In the last decade various classification methods have been proposed (see e.g. [8]). They are based on soft computing techniques, e.g. neural networks, fuzzy systems and genetic algorithms (see e.g. [1,2, 4-7]). It is well known that neural networks are not able to incorporate a linguistic information coming from human experts. On the other hand traditional fuzzy systems suffer from the lack of learning properties. Therefore, several authors developed neuro-fuzzy systems (see e.g. [5-7, 9-11, 13]). They exhibit advantages of neural networks and fuzzy systems. In this paper we present a new class of neuro-fuzzy systems for pattern classification. Since it is well known that introducing additional parameters to be tuned in neuro-fuzzy systems improves their performance, we incorporate several flexibility concepts in the design of neuro-fuzzy systems. Moreover, we are able to choose a fuzzy inference (Mamdani or logical) in the process of learning. The performance of our approach is illustrated of a typical benchmark.

2. Formal description of the basic neuro-fuzzy classifier

In this paper, we consider multi-input, single-output neuro-fuzzy inference system (NFIS) mapping $\mathbf{X} \to \mathbf{Y}$, where $\mathbf{X} \subset \mathbf{R}^n$ and $\mathbf{Y} \subset \mathbf{R}$.

The fuzzifier performs a mapping from the observed crisp input space $\mathbf{X} \subset \mathbf{R}^n$ to the fuzzy sets defined in $\mathbf{X}$. The most commonly used fuzzifier is the singleton fuzzifier which maps $\bar{\mathbf{x}} = [\bar{x}_1, \ldots, \bar{x}_n] \in \mathbf{X}$ into a fuzzy set $A' \subseteq \mathbf{X}$ characterized by the membership function:

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

The fuzzy rule base consists of a collection of N fuzzy IF-THEN rules in the form:

$$R^{(k)}: \text{IF } \mathbf{x} \text{ is } A^k \text{ THEN } y \text{ is } B^k \tag{2}$$

where $\mathbf{x} = [x_1, \ldots, x_n] \in \mathbf{X}$, $y \in \mathbf{Y}$, $A_1^k, A_2^k, \ldots, A_n^k$ are fuzzy sets characterized by membership functions $\mu_{A_i^k}(x_i)$, whereas B^k are fuzzy sets characterized by membership functions $\mu_{B^k}(y)$, respectively, $k = 1, \ldots, N$.

The fuzzy inference determines a mapping from the fuzzy sets in the input space $\mathbf{X}$ to the fuzzy sets in the output space $\mathbf{Y}$. Each of N rules (2) determines a fuzzy set $\bar{B}^k \subset \mathbf{Y}$ given by the compositional rule of inference:

$$\bar{B}^k = A' \circ \left(A^k \to B^k\right) \tag{3}$$

where $A^k = A_1^k \times A_2^k \times \ldots \times A_n^k$. Fuzzy sets $\bar{B}^k$, according to the formula (3), are characterized by membership functions expressed by the *sup-star* composition:

$$\mu_{\bar{B}^k}(y) = \sup_{\mathbf{x} \in \mathbf{X}} \left\{ \mu_{A'}(\mathbf{x}) \overset{T}{*} \mu_{A_1^k \times \ldots \times A_n^k \to B^k}(\mathbf{x}, y) \right\} \tag{4}$$

where $\overset{T}{*}$ can be any operator in the class of t-norms. It is easily seen that for a crisp input $\bar{\mathbf{x}} \in \mathbf{X}$, i.e. a singleton fuzzifier (1), formula (4) becomes:

$$\mu_{\bar{B}^k}(y) = \mu_{A_1^k \times \ldots \times A_n^k \to B^k}(\bar{\mathbf{x}}, y) = \mu_{A^k \to B^k}(\bar{\mathbf{x}}, y) = I\left(\mu_{A^k}(\bar{\mathbf{x}}), \mu_{B^k}(y)\right) \tag{5}$$

where $I(\cdot)$ is an "engineering implication" (Mamdani approach) or fuzzy implication. 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-konorm operator, depending on the type of fuzzy implication.

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

$$\bar{y} = \frac{\int_{\mathbf{Y}} y \mu_{B'}(y) dy}{\int_{\mathbf{Y}} \mu_{B'}(y) dy} \text{ or by } \bar{y} = \frac{\sum_{r=1}^{N} \bar{y}^r \cdot \mu_{B'}(\bar{y}^r)}{\sum_{r=1}^{N} \mu_{B'}(\bar{y}^r)} \tag{6}$$

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

$$\mu_{B^r}(\bar{y}^r) = \max_{y \in \mathbf{Y}} \left\{\mu_{B^r}(y)\right\} \tag{7}$$

Now, we propose a general architecture of NFIS. It includes both the Mamdani and logical type of inference:

$$\bar{y} = f(\bar{\mathbf{x}}) = \frac{\sum_{r=1}^{N} \bar{y}^r \cdot \operatorname{agr}_r(\bar{\mathbf{x}}, \bar{y}^r)}{\sum_{r=1}^{N} \operatorname{agr}_r(\bar{\mathbf{x}}, \bar{y}^r)} \tag{8}$$

where

$$\operatorname{agr}_r(\bar{\mathbf{x}}, \bar{y}^r) = \begin{cases} \overset{N}{\underset{k=1}{S}} \left\{I_{k,r}(\bar{\mathbf{x}}, \bar{y}^r)\right\}, & \text{for Mamdani approach} \\ \overset{N}{\underset{k=1}{T}} \left\{I_{k,r}(\bar{\mathbf{x}}, \bar{y}^r)\right\}, & \text{for logical approach} \end{cases} \tag{9}$$

and

$$I_{k,r}(\bar{\mathbf{x}}, \bar{y}^r) = \begin{cases} T\left\{\tau_k(\bar{\mathbf{x}}), \mu_{B^k}(\bar{y}^r)\right\}, & \text{for Mamdani approach} \\ S\left\{N(\tau_k(\bar{\mathbf{x}})), \mu_{B^k}(\bar{y}^r)\right\}, & \text{for logical approach} \end{cases} \tag{10}$$

Moreover, the firing strength of rules is given by

$$\tau_k(\bar{\mathbf{x}}) = \overset{n}{\underset{i=1}{T}} \left\{\mu_{A_i^k}(\bar{x}_i)\right\} \tag{11}$$

The general architecture of the above system is depicted in Fig. 1.

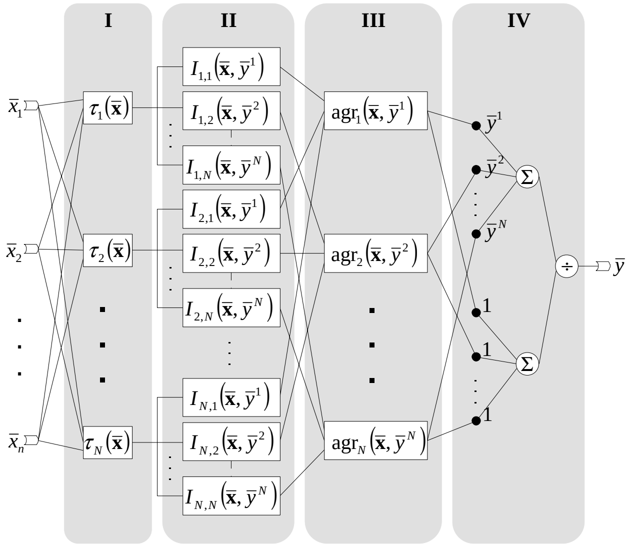

Fig. 1. General architecture of NFIS for pattern classification

3. Flexible neuro-fuzzy classifier

Now we propose new structures of NFIS. The novel systems are characterized by:

- soft strength of firing controlled by parameter α^τ,
- soft implication controlled by parameter α^I,
- soft aggregation of rules controlled by parameter α^{agr},
- weights in antecedents of the rules: $w_{i,k}^\tau \in [0,1]$, $i = 1, \ldots, n$, $k = 1, \ldots, N$,
- weights in aggregation of the rules: $w_k^{\text{agr}} \in [0,1]$, $k = 1, \ldots, N$.

Moreover, we assume that fuzzy norms (and H-function) in connection of antecedents, implication and aggregation of rules are parameterised by parameters p^τ, p^I, p^{agr}, respectively.

The concept of adjustable triangular norms is based on the following definition:

Definition 1. (Compromise operator)

Function

$$\tilde{N}_\nu : [0,1] \to [0,1] \tag{12}$$

given by

$$\tilde{N}_\nu(a) = (1-\nu)N(a) + \nu N(N(a)) = (1-\nu)N(a) + \nu a \tag{13}$$

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

Observe that

$$\tilde{N}_\nu(a) = \begin{cases} N(a) & \text{for} \quad \nu = 0 \\ \frac{1}{2} & \text{for} \quad \nu = \frac{1}{2} \\ a & \text{for} \quad \nu = 1 \end{cases} \tag{14}$$

Definition 2. (H-function)

Function

$$H : [0,1]^n \to [0,1] \tag{15}$$

given by

$$H(\mathbf{a};\nu) = \tilde{N}_\nu\left(\overset{n}{\underset{i=1}{S}}\{\tilde{N}_\nu(a_i)\}\right) = \tilde{N}_{1-\nu}\left(\overset{n}{\underset{i=1}{T}}\{\tilde{N}_{1-\nu}(a_i)\}\right) \tag{16}$$

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

Observe that

$$H(\mathbf{a};\nu) = \begin{cases} T\{\mathbf{a}\} & \text{for} \quad \nu = 0 \\ \frac{1}{2} & \text{for} \quad \nu = \frac{1}{2} \\ S\{\mathbf{a}\} & \text{for} \quad \nu = 1 \end{cases} \tag{17}$$

The NFIS for pattern classification is presented below:

$$\tau_k(\bar{\mathbf{x}}) = \begin{pmatrix} (1-\alpha^\tau)\operatorname{avg}\left(\mu_{A_1^k}(\bar{x}_1), \ldots, \mu_{A_n^k}(\bar{x}_n)\right) + \\ +\alpha^\tau \overset{\leftrightarrow}{H}{}^*\begin{pmatrix} \mu_{A_1^k}(\bar{x}_1), \ldots, \mu_{A_n^k}(\bar{x}_n) \ ; \\ w_{1,k}^\tau, \ldots, w_{n,k}^\tau, p^\tau, 0 \end{pmatrix} \end{pmatrix} \tag{18}$$

$$I_{k,r}(\bar{\mathbf{x}}, \bar{y}^r) = \begin{pmatrix} (1-\alpha^I)\operatorname{avg}\left(\tilde{N}_{1-\nu}(\tau_k(\bar{\mathbf{x}})), \mu_{B^k}(\bar{y}^r)\right) + \\ +\alpha^I \overset{\leftrightarrow}{H}\begin{pmatrix} \tilde{N}_{1-\nu}(\tau_k(\bar{\mathbf{x}})), \mu_{B^k}(\bar{y}^r) \ ; \\ p^I, \nu \end{pmatrix} \end{pmatrix} \tag{19}$$

$$\operatorname{agr}_r(\bar{\mathbf{x}}, \bar{y}^r) = \begin{pmatrix} (1-\alpha^{\text{agr}})\operatorname{avg}\left(I_{1,r}(\bar{\mathbf{x}}, \bar{y}^r), \ldots, I_{N,r}(\bar{\mathbf{x}}, \bar{y}^r)\right) + \\ +\alpha^{\text{agr}} \overset{\leftrightarrow}{H}{}^*\begin{pmatrix} I_{1,r}(\bar{\mathbf{x}}, \bar{y}^r), \ldots, I_{N,r}(\bar{\mathbf{x}}, \bar{y}^r) \ ; \\ w_1^{\text{agr}}, \ldots, w_N^{\text{agr}}, p^{\text{agr}}, 1-\nu \end{pmatrix} \end{pmatrix} \tag{20}$$

$$\bar{y} = \frac{\sum_{r=1}^{N} \bar{y}^r \cdot \operatorname{agr}_r(\bar{\mathbf{x}}, \bar{y}^r)}{\sum_{r=1}^{N} \operatorname{agr}_r(\bar{\mathbf{x}}, \bar{y}^r)} \tag{21}$$

Remark 1. We will explain how to modify formula (21) to solve multi-classification problems. Let $[x_1,\ldots,x_n]$ be the vector of features of an object ν . Let $\Omega = \{\omega_1,\ldots,\omega_M\}$ be a set of classes. The knowledge is represented by a set of N rules in the form

$$R^{(k)} : \begin{cases} \text{IF} \quad x_1 \ \text{is} \ A_1^k \ \text{AND} \\ \quad x_2 \ \text{is} \ A_2^k \ \text{AND} \ \ldots \\ \quad x_n \ \text{is} \ A_n^k \\ \text{THEN} \quad \nu \in \omega_1(z_1^k) \ , \\ \quad \nu \in \omega_2(z_2^k) \ , \quad \ldots, \\ \quad \nu \in \omega_M(z_M^k) \ , \end{cases} \tag{22}$$

where z_j^k, $j = 1,\ldots,M$, $k = 1,\ldots,N$, are interpreted as "support" for class ω_j given by rule $R^{(k)}$. We will now redefine description (22). Let us introduce vector $\mathbf{z} = [z_1,\ldots,z_M]$, where z_j, $j = 1,\ldots,M$, 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

$$R^{(k)} : \begin{cases} \text{IF} \quad x_1 \ \text{is} \ A_1^k \ \text{AND} \\ \quad x_2 \ \text{is} \ A_2^k \ \text{AND} \ \ldots \\ \quad x_n \ \text{is} \ A_n^k \\ \text{THEN} \quad z_1 \ \text{is} \ B_1^k \ \text{AND} \\ \quad z_2 \ \text{is} \ B_2^k \ \text{AND} \ \ldots \\ \quad z_M \ \text{is} \ B_M^k \end{cases} \tag{23}$$

and formula (21) adopted for classification takes the form

$$\bar{z}_j = \frac{\sum_{r=1}^{N} \bar{z}_j^r \operatorname{agr}_r(\bar{\mathbf{x}}, \bar{z}_j^r)}{\sum_{r=1}^{N} \operatorname{agr}_r(\bar{\mathbf{x}}, \bar{z}_j^r)} \tag{24}$$

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

4. Simulation results

Neuro-fuzzy classifier, described by formulas (18)-(21), is simulated on *Wine Recognition* problem [12].

The experimental results for the Wine Recognition problem are depicted in Tables 1 and 2 for the not-parameterised (Zadeh and product) and parameterised (Dombi and Yager) H-functions, respectively. For experiment (iv) the final values (after learning) of weights $w_{i,k}^\tau \in [0,1]$ and $w_k^{\text{agr}} \in [0,1]$, $i = 1,\ldots,13$, $k = 1,\ldots,2$, are shown in Fig. 2 (Zadeh and produt H-functions) and Fig. 3 (Dombi and Yager H-functions).

Table 1. Experimental results for a) Zadeh triangular norms, b) product triangular norms

Experiment number	Name of flexibility parameter	Initial values	Final values after learning		Mistakes [%] (learning sequence)		Mistakes [%] (testing sequence)	
			Zadeh H-function	Product H-function	Zadeh H-function	Product H-function	Zadeh H-function	Product H-function
i	ν	0.5	1.0000	1.0000	0.00	0.00	3.77	1.89
ii	ν	0	-	-	0.80	0.80	3.77	3.77
iii	ν	0.5	1.0000	1.0000	0.00	0.00	1.89	1.89
	α^{τ}	1	0.0004	0.0036				
	α^{I}	1	0.9907	0.9986				
	α^{agr}	1	0.9938	0.9908				
iv	ν	0.5	1.0000	1.0000	0.00	0.00	0.00	0.00
	α^{τ}	1	0.0329	0.0180				
	α^{I}	1	0.9987	0.9756				
	α^{agr}	1	0.9896	0.9861				
	$\mathbf{w}^{\tau}$	**1**	Fig.2-a	Fig.2-b				
	$\mathbf{w}^{agr}$	**1**	Fig.2-a	Fig.2-b				

Table 2. Experimental results for a) Dombi H-function, b) Yager H-function

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	0.00	0.00	1.89	1.89
ii	ν	0	-	-	0.00	0.00	3.77	3.77
iii	ν	0.5	1.0000	1.0000	0.00	0.00	1.89	1.89
	p^{τ}	10	9.9999	10.0498				
	p^{I}	10	10.0005	9.9936				
	p^{agr}	10	9.9991	10.0014				
	α^{τ}	1	0.0032	0.0029				
	α^{I}	1	0.9911	0.9917				
	α^{agr}	1	0.9919	0.9920				
iv	ν	0.5	1.0000	1.0000	0.00	0.00	0.00	0.00
	p^{τ}	10	7.8330	6.9528				
	p^{I}	10	11.7084	13.3122				
	p^{agr}	10	14.3699	12.1427				
	α^{τ}	1	0.0028	0.0389				
	α^{I}	1	0.9826	0.9740				
	α^{agr}	1	0.9914	0.9599				
	$\mathbf{w}^{\tau}$	**1**	Fig.3-a	Fig.3-b				
	$\mathbf{w}^{agr}$	**1**	Fig.3-a	Fig.3-b				

Acknowledgement

This work was sponsored by the Polish Ministry of Scientific Research and Information Technology (grant 2004-2007).

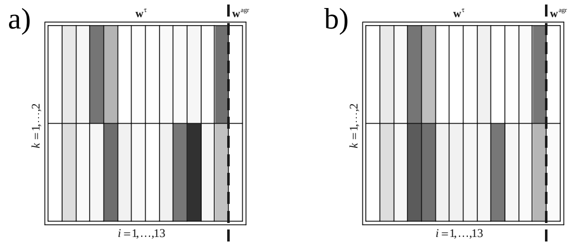

Fig. 2. Weights representation in the Wine Recognition problem for a) Zadeh triangular norms, b) product triangular norms

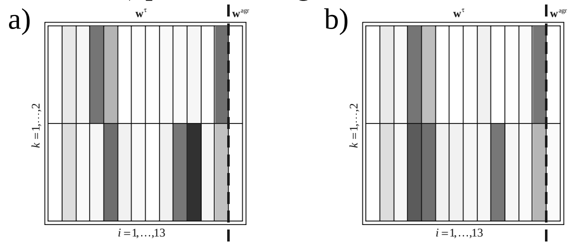

Fig. 3. Weights representation in the Wine Recognition problem for a) Dombi H-function, b) Yager H-function

References:

[1] J. S. Jang, C. T. Sun, E. Mizutani, *Neuro-Fuzzy and Soft Computing*. Prentice Hall, Englewood Cliffs, 1997.

[2] N. Kasabov, *Foundations of Neural Networks, Fuzzy Systems and Knowledge Engineering*. The MIT Press, CA, MA, 550 pages, 1996.

[3] E. P. Klement, R. Mesiar, E. Pap, *Triangular Norms*. Kluwer Academic Publishers, Netherlands 2000.

[4] J. M. Mendel, *Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions*. Prentice Hall PTR, Upper Saddle River, NJ, 2001.

[5] D. Nauck, F. Klawon, R. Kruse, *Foundations of Neuro-Fuzzy Systems*. Chchester, U.K.: Wiley, 1997.

[6] D. Rutkowska, *Neuro-Fuzzy Architectures and Hybrid Learning*. Springer-Verlag 2002.

[7] L. Rutkowski (2004), *Flexible Neuro-Fuzzy Systems: Structures, Learning and Performance Evaluation*, Kluwer Academic Publishers 2004.

[8] L. Rutkowski, New Soft Computing Techniques For System Modelling, Pattern Classification and Image Processing. Springer-Verlag 2004.

[9] L. Rutkowski, K. Cpałka, “A general approach to neuro-fuzzy systems,” *The 10th IEEE International Conference on Fuzzy Systems*, Melbourne 2001

[10] L. Rutkowski and K. Cpałka, *Designing and learning of adjustable quasi-triangular norms with applications to neuro-fuzzy systems*, IEEE Trans. on Fuzzy Systems, vol. 13, pp. 140-151, 2005.

[11] Rutkowski L., Cpałka K., Flexible neuro-fuzzy systems, *IEEE Trans. Neural Networks*, vol. 14, pp. 554-574, May 2003.

[12] UCI respository of machine learning databases, C. J. Mertz, P. M. Murphy. Available online: *http://www.ics.uci.edu/pub/machine-learning-databases*.

[13] R. R. Yager, D. P. Filev, *Essentials of fuzzy modelling and control*. John Wiley & Sons, 1994.