

PATTERN RECOGNITION BY NEW TYPE OF FUZZY NEURAL NETWORKS WITH LEARNING ON THE BASIS OF GENETIC ALGORITHMS AND EVOLUTION STRATEGIES

Vladimír OLEJ, Milan UNGER
 Department of Informatics and Computers
 Technical University, 031 19 Liptovský Mikuláš
 Slovak Republic

Abstract: The paper presents a possibility of the pattern recognition by a new type of fuzzy neural networks with learning on the basis of genetic algorithms and evolution strategies.

Keywords: Fuzzy neural networks, pattern recognition, genetic algorithms and evolution strategies.

1. Introduction

Fuzzy neural networks (FNNs) can be used (in addition to other fields) for analysis of decision processes of discrete systems [1,2] and for pattern recognition [3]. The important part for using FNNs are the learning algorithms. Genetic algorithms (GAs) and evolution strategies (ESs) represent an efficient technique of learning based on a dynamics of the evolution [3,4,5,6,7]. Fuzzy neural networks employing basic fuzzy logic operations are introduced in [3,4]. In the learning process (by GAs and ESs), they enable to shorten the time of method convergence towards the global maximum. Input and output values of FNNs are expressed by crisp numbers. Input values in the new FNNs formulation can be realized as fuzzy sets [8], outputs as membership function values into the given fuzzy set.

2. The Structure of Fuzzy Neural Network

Let consider the FNN shown on the Fig.1. It is composed from the input layer $IN = \{IN_1, IN_2, \dots, IN_m\}$, the hidden layer $H = \{H_1, H_2, \dots, H_n\}$ and the output layer $OUT = \{OUT_1, OUT_2, \dots, OUT_k\}$. Then in the above defined FNN it is necessary to determine the synapse weights in order to recognition the input patterns P, B, R and damaged input patterns P', B', R' , which are shown on the Fig.2. The input patterns are represented by fuzzy sets P, B, R, P', B', R' .

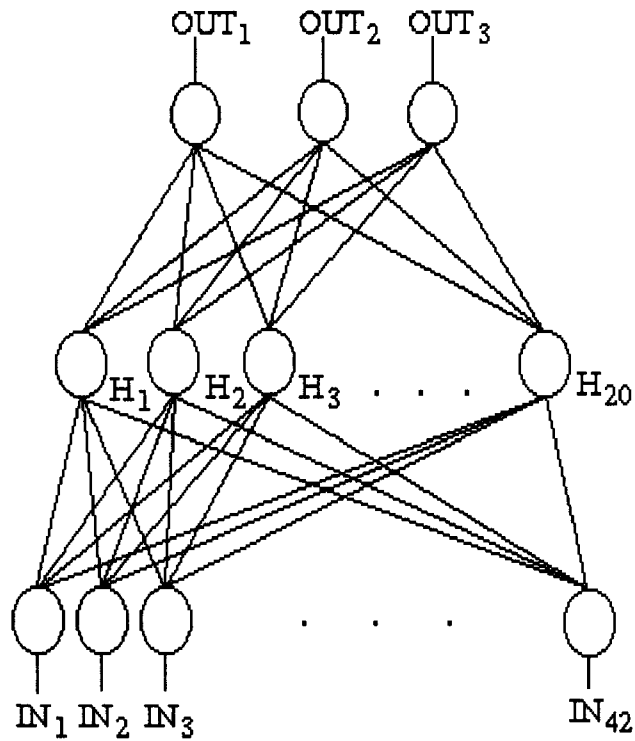


Fig.1 Fuzzy neural network

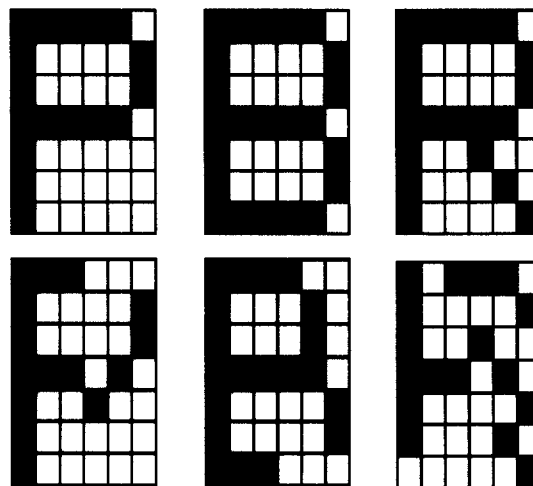


Fig.2 Input patterns for fuzzy neural network

The input neuron IN_i (Fig.3.a) is connected to a neuron of the hidden layer (summation fuzzy neuron) H_j (Fig.3.b). Synapse weights between the input layer neuron IN_i and the hidden layer neuron H_j are v_{ij} . These weights obtained in the learning process by GAs and ESs acquire values from the set of real numbers \mathcal{R} . Synapse weights between the hidden layer neuron H_i and the output layer (logical fuzzy neuron) OUT_j (Fig.3.c) are w_{ij} . These weights obtained in the learning process by GAs and ESs acquire values

from the interval $w_{ij} \in [0,1]$. The input into the neuron IN_i is a_i , the output of the summation fuzzy neuron H_i is b_i and the output of the logical fuzzy neuron OUT_i is c_i .

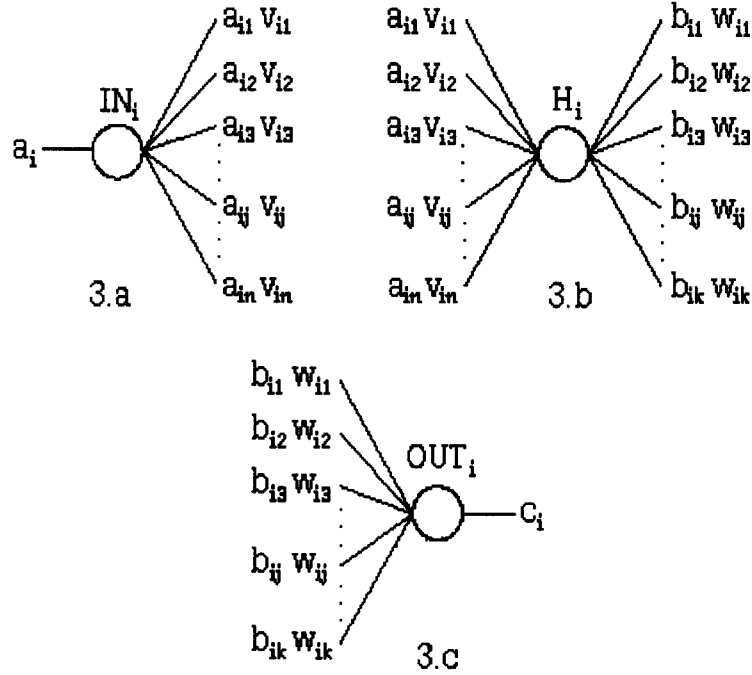


Fig. 3.a Input neuron, 3.b summation fuzzy neuron
3.c logical fuzzy neuron

The output b_i of the summation fuzzy neuron can be expressed as follows

$$b_i = \begin{cases} 1 & \text{if } b_i' > p_{\max} \\ (b_i' - p_{\min}) / (p_{\max} - p_{\min}), & \text{if } p_{\min} < b_i' < p_{\max}, \text{ where} \\ 0 & \text{if } b_i' < p_{\min} \end{cases} \quad (1)$$

$$b_i' = \sum_{j=0}^m a_j w_{ij} \quad (2)$$

and p_{\min} , p_{\max} are condition of a normalizing function. The output c_i of a logical fuzzy neuron can be expressed as follows

$$c_i = \text{MAX}_j(\text{MIN}(b_i, w_{ij})), \quad (3)$$

where the operation MIN (MAX) [8] is a logical operation of conjunction (disjunction).

3. Genetic Algorithms and Evolution Strategies

Every living organism is described by its genes. Genes can stay in several forms - alleles. The set of alleles al_i living organism is called a genotype. A fitness η evaluates genotypes with regard to their adaptation to the environment in which they live. Influence of one allele can depend on others. It causes that almost identical genotypes have a different fitness η . Some living organisms have features that make them more adaptable to their environment. These organisms will have a better fitness η and more chances to stay alive and reproduce. In general, genotypes are represented as allele strings. The first element of the string is an allele of the first gene, the second element of the string is an allele of the second gene, etc. Length l of a genotype expresses the number of alleles in it and is constant.

Let α be a set of all possible genotypes (APGs)

$$\alpha = \{ G_i = \{ al_1 al_2 \dots al_i \dots al_n \}, \text{ where } al_i \text{ is an allele of the } i\text{-th gene} \}. \quad (4)$$

Then a fitness $\eta: G_i \rightarrow \mathbb{R}^+$ assign a real number to each genotype from APGs α . There are M individuals available and everyone has an unambiguously specified genotype from APGs α . These M individuals create a population. Genetic algorithms and ESs modify a population by a time step t . The set of individuals of the population at the time t is expressed as $\beta(t)$. Further, an individual with a genotype $G_i \in \alpha$ will be identified with this genotype. The size of $\beta(t)$ is constant and equal to M .

The genetic algorithm GA1 [5,6,9] can be expressed as follows:

1. **Initialization:** Let $t = 0$. Randomly choose M individuals from APG α and create a population $\beta(0)$ of them.
2. **Selection:** To each genotype $G_i \in \beta(t)$ assign a fitness η . Randomly choose genotype $G_i \in \beta(t)$ and place n of its copies into population $\beta(t+1)$. The number of n copies is given by the following relation

$$n = \left(\eta / \sum_{i=1}^M \eta_i \right) M. \quad (5)$$

Repeat this step until the number of genotypes G_i in population $\beta(t+1)$ is equal M . Then $t = t + 1$.

3. **Crossing:** Randomly choose two genotypes $G_i, G_j \in \beta(t)$. With probability $p = 0.5$ interchange all alleles between genotypes G_i and G_j . Repeat the foregoing steps to that every genotype is crossed with probability p_c .

4. **Mutation:** With the probability p_m alter the value of all alleles a_i by a random value from the interval $\langle +\delta, -\delta \rangle$. The value δ depend on the type of an analyzed problem.
5. **Stop condition:** If $t \geq t_{\max}$ stop GA1, otherwise continue from the second step. Genotype $G_j \in \beta(t)$ with the highest fitness η consider as the most suitable.

The GA1 changes and modified the sets of alleles of mutual various genotypes. These algorithms alternates between selection and variation (crossing and mutation)-the two fundamental processes of the evolution. The evolution is the process changing the generations of the organisms in such way that they are adapting to the environment. The GA1 finds a suboptimal solution of this problem, but the experiments show that the convergence is too slow and the synapse weights can be far away from the optimum depending on the initial values. Therefore we propose a modification of GA1.

The genetic algorithm GA2 [3,4] can be expressed as follows:

1. **Initialization:** It is realized as in GA1.
2. **Selection:** To each genotype $G_j \in \beta(t)$ assign a fitness η . From the population $\beta(t)$ choose M genotypes with the highest fitness η into the population $\beta(t+1)$. Then $t = t + 1$.
3. **Crossing:** Take out the first genotype $G_j \in \beta(t)$. Create copies of four randomly choosen genotypes G_1 to $G_4 \in \beta(t)$. With the probability $p = 0.5$ change alleles genotype copies G_1 to G_4 for the allele of genotype G_j . Add the genotype copies G_1 to G_4 into the population $\beta(t)$. Repeat the foregoing steps for all genotypes from the population $\beta(t)$.
4. **Mutation:** It is realized as in GA1.
5. **Stop condition:** It is realized as in GA1.

The GA2 also starts with random initial values of alleles. The selection step is replaced by sorting and choosing the best organisms for the following evolution depending on their fitness. The organisms are then randomly crossed four times in one population.

The evolution strategy [7,9] has the following form:

1. **Initialization:** It is realized as in GA1.
2. **Selection:** It is realized as in GA2.
3. **Crossing:** Create copies of two randomly choosen genotypes $G_i, G_j \in \beta(t)$. With the probabillity $p = 0.5$ change alleles between copies of the genotypes G_i and G_j . Add copies of the genotypes G_i and G_j into the population $\beta(t)$. Repeat the foregoing steps so that the number of genotypes in the population $\beta(t)$ meets the condition $N < M$, where N is a maximal number of individuals in the population.

4. **Mutation:** With the probability p_m change the value of alleles of all genotypes created by crossing with a random value from the interval $\langle +\delta, -\delta \rangle$.
5. **Stop condition:** It is realized as in GA1.

In addition to GA1 [5,6,9], GA2 [3,4] and ES [7,9] there exist other algorithms. The number of states in which alleles of a genotype can stay divides the genotypes into two-value genotypes (TVG) and multiple-value genotypes (MVG).

The simplest genotypes are TVG representing features of an analyzed task by binary numbers. One allele is assigned to each feature and represents a presence of a feature. A two-value genotypes can represent multiple-value feature of the analyzed task. Number of states is specified for each feature in this case. A binary number is assigned to each state and the sequence of these numbers creates TVG. In the case of mutation of allele of TVG, logical negation of allele's value is made. A multiple-value genotypes represents multiple-value features of the analyzed task. The set of states is represented in the set of real numbers assigned to alleles. A mutation operator used over MVG adds a random number from the interval $\langle +\delta, -\delta \rangle$ to the value of mutated allele.

Genetic algorithms and ESs representing their various classes, are analyzed in [9]. From this analysis it follows, that GAs and ESs with multiple-value genotype is suitable for realizing of the pattern recognition.

4. Learning Processes by of Genetic Algorithms and Evolution Strategies

For the purpose of setting synapse weights of the defined FNNs, GA1s, GA2s and ESs with multiple-value genotype can be used. In the learning process, the alleles represent the weights of corresponding synapses on FNNs. Learning results (FNN outputs) obtained by means GAs and ESs for recognition the input patterns P, B, R and damaged input patterns P', B', R' are shown in Fig.4, Fig.5 and Fig.6. The given outputs represent values of membership functions μ , obtained by means of the introduced algorithms which expresses memberships into the input fuzzy sets P, B, R, P', B', R' .

From the values of a membership functions μ of outputs $OUT = \{OUT_1, OUT_2, OUT_3\}$ (Fig.4) it follows that, if after synapse weights setting in FNN there is at its input the pattern $P (P')$, then output OUT_1 represents membership to fuzzy set $P (P')$ with membership function value $\mu_P=1 (\mu_{P'}=1)$, OUT_2 refers to fuzzy set $B (B')$ with $\mu_B=0.13 (\mu_{B'}=0)$ and OUT_3 belongs to fuzzy set $R (R')$ with $\mu_R=0 (\mu_{R'}=0.5)$. Similarly, one can interpret the results in the case when pattern $B(B')$, or $R(R')$ are at the input of FNN.

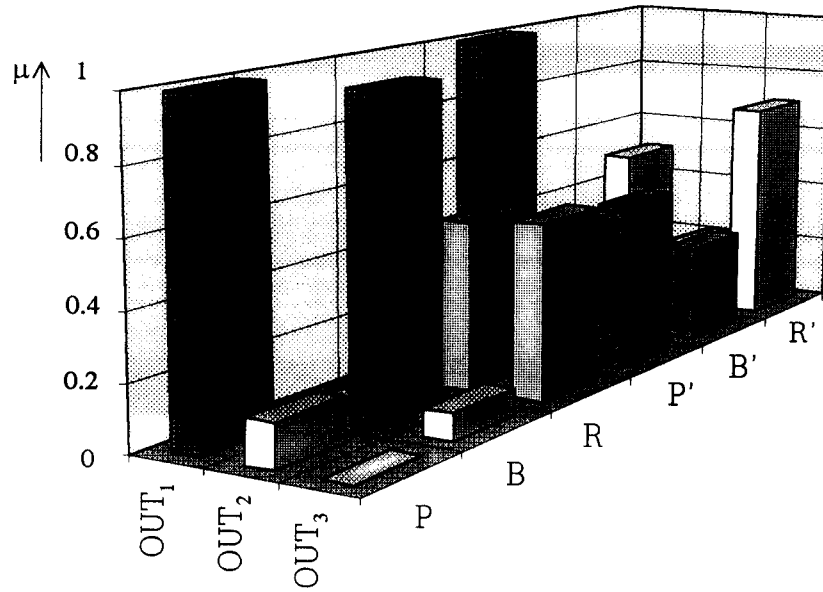


Fig.4 Output values of FNN obtained in the learning process by GA1

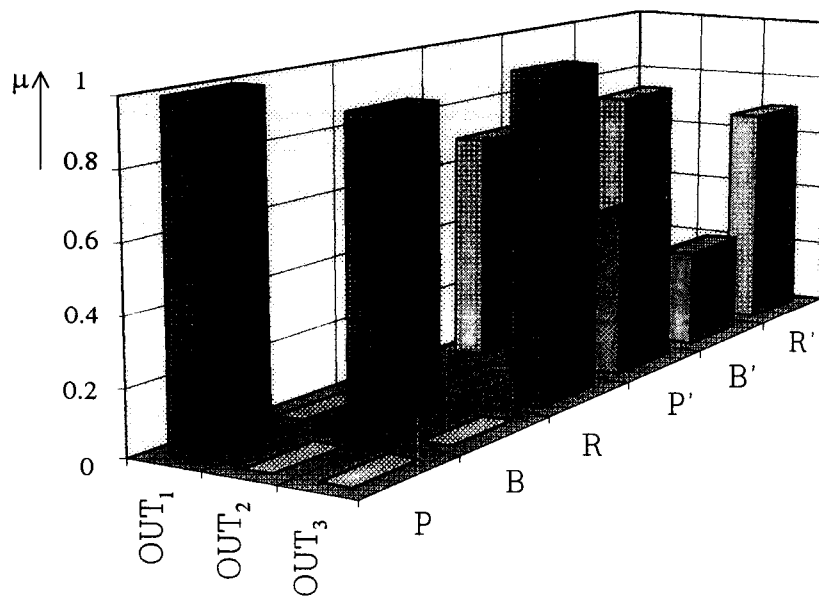


Fig.5 Output values of FNN obtained in the learning process by GA2

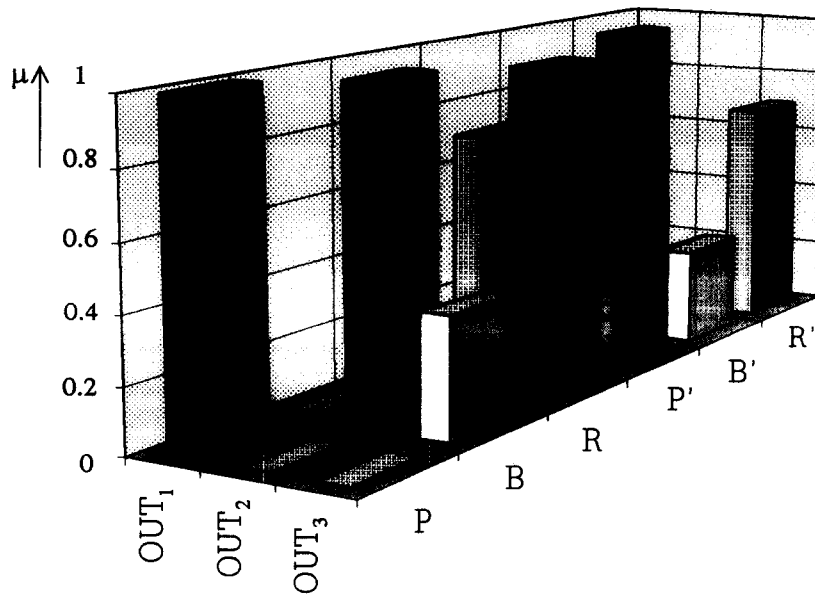


Fig.6 Output values of FNN obtained in the learning process by ES

Table 1 shows parameters and experiments results. It contains the type of algorithms, the crossing probability p_c , the mutation probability p_m , the number M individuals in the population, the maximal number N of individuals in the population, the number of generation, the fitness η and computation time. The value η represents the fitness obtained during the evolution by means of the given GAs and ESs. From Table 1 it also follows that the highest fitness η is obtained by algorithm GA2 with the smallest number of generations. However, its calculation time is the longest. Maximal fitness ($\eta_{\max}=18$) is determined as the product of input patterns number and FNN outputs number.

Table 1 Parameters and results of experiments

Algorithms	p_c	p_m	M	N	η	Number of generation	Calculation time [time unit]
GA1	0.6	0.01	40	-	15.7	260	53
GA2	-	0.01	40	-	17.92	100	48
ES	-	0.01	40	100	17.03	250	30

The evolution of the given GAs and ESs with multiple-value genotype is shown in Fig. 7.

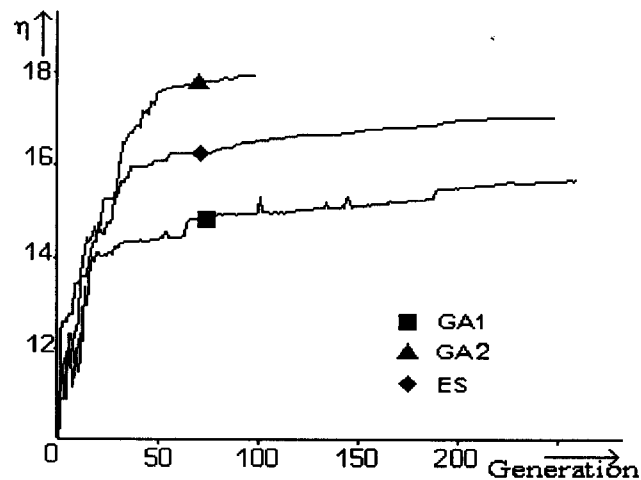


Fig.7 Evolution by GA1, GA2 and ES with multiple-value genotype

5. Conclusion

The paper introduces a new type of FNNs, that are suitable (in addition to other fields) for pattern recognition the input patterns and damaged input patterns. The learning process in FNNs defined in this way is realized on the basis of GAs and ESs with MVG. For the purpose of setting synapse weights in FNNs the program system, that enables to realize various of GAs and ESs with TVG and MVG can be used. This program systems has been realized in the programming language C⁺⁺.

References

- [1] OLEJ,V.-KŘUPKA,J.: Synthesis and Analysis of Decision Processes for the Automation Control System with Uncertainty. Part II. BUSEFAL, Université Paul Sabatier, Toulouse, France, No.62, 1995, pp.41-50.
- [2] OLEJ,V.-KŘUPKA,J.: Analysis of Decision Processes of Automation Control Systems with Uncertainty. [Scientific Monograph], Technical University Press, ELFA, Košice, Slovak Republic, 1996, ISBN 80-88786-29-0.
- [3] LEHOTSKÝ,M.-OLEJ,V.: Pattern Recognition Based on the Fuzzy Neural Networks. BUSEFAL, Université Paul Sabatier, Toulouse, France, No.60, 1994, pp.15-22.
- [4] LEHOTSKÝ,M.-OLEJ,V.-CHMÚRNÝ,J.: Pattern Recognition Based on the Fuzzy Neural Networks and their Learning by Modified Genetic Algorithms. Neural Network World, Czech Republic, Vol.5, No.1, 1995, pp.91-97.
- [5] HOLLAND,J.H.: Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor, 1975.
- [6] SPIESSENS,P.: Genetic Algorithms. AI MEMO, No.88-19, Vrije Universiteit Brussel, 1988.
- [7] HOFFMEISTER,F.-BACK,T.: Genetic Algorithms and Evolution Strategies. Similarities and Differences. Proc. of 1st Workshop on Parallel Problem Solving from Nature, Dortmund, Germany, 1990, pp.455-469.
- [8] ZADEH,L.A.: Fuzzy Sets. Information and Control, Vol.8, 1965, pp.338-353.
- [9] UNGER,M.-OLEJ,V.: Analysis of Genetic Algorithms and Evolution Strategies. Proc. of Technical University, Vol.2, No.4, Liptovský Mikuláš, 1995, pp.69-77. (in Slovak).