Optimized Fuzzy Artificial Neural Network through Genetic Algorithms in Recognizing Mixture Odors using Higher Number of Sensors

Benyamin Kusumoputro, Teguh P. Arsyad

Faculty of Computer Science, University of Indonesia
Depok Campus, Jakarta Indonesia, PO.Box 3443 Jakarta 10002
Phone:+62-21-7863419, Fax:+62-22-7863415,
Email: kusumo@cs.ui.ac.id

***Abstract*—Recognizing odor mixture is rather difficult problem for the artificial odor recognition system, especially when the system used a limited number of sensors. The classification processes are more hampered when the number of the unlearned mixture odor classes is increased. In this paper, authors developed a Fuzzy-Neuro MLP as a pattern classifier, and compare its recognition capability with that of Probabilistic Neural Network and Backpropagation neural system. To enhance the recognition capability of the system, then we developed an optimized Fuzzy-Neuro MLP, by deleting the weak weight connections through the used of Genetic Algorithms. Experimental results show that the optimized Fuzzy-Neuro MLP has the highest recognition rate in recognizing 18 classes of two mixture odors with almost 98.2% by using hardware system with 16 sensors compare with only 83.3% when using 8 sensors.**

Keywords: Odor Recognition System, Fuzzy-Neuro System, Genetic Algorithms, Neural Structure Optimization Method, MultilayerPerceptron,

I. Introduction

Fuzzy systems and neural networks, as a separate diagnosis techniques, have received considerable attention for decades and obtained many successful application. Fuzzy systems have demonstrated to be well suited for dealing with ill-defined and uncertain systems, while neural networks are well known for its learning capability. Neural networks have known for its high performance on fault tolerance, and together with its non-linier processing and the flexibility for adjusting the networks topology, neural systems can be conveniently used as powerful diagnosis techniques. Incorporated the fuzzy systems into artificial neural networks is then able to enhance the capability of the intelligent systems to learn from experience and adapt to changes in an environment with qualitative, imprecise, uncertain or incomplete information.

Using antecedents and its corresponding consequents as the training pairs of the input-target vectors, a conventional neural network can learn the relationship between the antecedents and the consequents for the problem domain. However, neural network is not able to perform logic-like rules because the distribution of connection weights in the network is almost impossible to be interpreted in terms of if-then rules. Furthermore, it is difficult to map the known domain knowledge onto the structure of the networks to enhance the learning process. Combining the fuzzy logic element into the neural networks, hence, produces a neural topology that can perform fuzzy inference rules through analyzing the values of the weight connection [1-4].

The developed fuzzy neural networks are already widely used [5-7] and some work has been done on obtaining the proper network structure and the initial weights to reduce its training time [8]. However, this type of neural system, as same with that of multilayer perceptron (MLP) neural networks, has a drawback due to its huge neural connections. It is well recognized that the performance of backpropagation-trained MLP neural networks depends highly on their topology and the values of a number of training parameters. Unfortunately, there are no comprehensive analytical methods of determining the optimal topology and training parameters for a particular task, which often hampers the wider application of neural networks to the real world application. Traditionally, the design of a neural network is achieved by trial and error and requires the involvement of an expert. As systems become more costly in its computational performance, no one could guarantee that a near optimal design has been chosen.

Design of the neural topology models typically can be classified into two groups of network topology. The first is a non-adaptive network topology, such as static multilayer perceptron (MLP), where neurons are fully connected layer by layer, however, this simplified neuron topology creates a dilemma from the fact that both large and small networks exhibit a number of disadvantages. If the network size is too small, the error rate tends to increase due to the network might not be able to approximate good enough the functional relationship between the input and the target output. While if the size is too large, the network would not be able to generalize well on the input data that never been learned before. As an additional problem, the larger the neural size, the longer of the learning process that should be done.

The second group of neural network topology allows an adaptive, dynamic network design by optimizing the network architecture and its weight connections. Optimization of this network could be done through two approaches. One starts with a larger network structure that is supposed to be sufficiently complex to model the relation between input and output variables. The training procedure is then implemented and the network tries to reduce the number of neurons as possible, until an acceptable solution is found. These algorithms [9-10], however, have several weaknesses.

Practically, one does not know with where the starting network should be, and since the majority of the training procedure is spent on larger network, high computational cost could not be avoided. And usually, smaller size network with other design may possibly be capable of meeting the same smallest acceptable solution with that of the network derived from pruning algorithm [11].

The other approach on optimization of the dynamic network is based on discrete optimization methods. In these methods, the optimization of the neural in the training stage is discriminated with that of the structure optimization. In this approach, each network structure is assigned an evaluation value (e.g., an estimation of the generalization error), thus formulating a discrete optimization task. A variety of methods can be applied to this class of the problems; however, genetic algorithms [12-13], have gained a strong methodology as the evolutionary approach [14] by either removing the weak connections between neurons or removing the neurons that perform weak activations.

In this paper we developed a GA-optimized fuzzy neural system and applied it as a pattern classifier to discriminate mixture of odors that could not be properly classified with the previous neural system. Analysis and comparison of the proposed neural system and the other neural systems, e.g. Back-propagation, PNN and the un-optimized fuzzy neural will be explained.

II. Fuzzy-Neuro Multilayer Perceptron

Fuzzy–Neuro MLP is a type of multi-layers feed-forward neural network, however, this fuzzy–Neuro MLP has some basic difference compares to the conventional multi-layers feed-forward MLP. Instead of using conventional neuron that use the sum operation ($\sum$) over the multiplication of the input and weights in connection, two types of fuzzy neurons are implemented in this Fuzzy-Neuro MLP. Those fuzzy neurons are fuzzy AND-neuron and fuzzy OR-neuron where *t*-norms are used for AND operation while *t*-conorms for OR operation. The other difference is the neurons in the F-Neuro MLP used fuzzy type of data as their input and the output activation, so they can deal directly with the fuzziness of the data input from experiments.

The F-Neuro MLP will construct an inference system that consists of an antecedent part and a consequent part [15-17]. This system could be expressed in IF-THEN rule in the form of: *If* <u>*P is M*</u>*, then* <u>*Q is N*</u>, where <u>*P is M*</u> stands for the antecedent and <u>*Q is N*</u> denotes the consequent of the IF-THEN rule. The neural networks that provides a logical construction of the IF-THEN rule is developed by using two of Pedrycz's fuzzy set-based neurons [6-7][18] which consists of one input layer, several of AND-neurons as a hidden layer and OR-neurons as an output layer, and its derivation of the min-max operator with respect to its connection weights.

Suppose T is used to denote *t*-norm function and S to denote *t*-conorm function, output activation of the AND-neuron with its weight v and input x can be expressed as:

$$z(t) = \mathop{T}_{i=0}^{n} [v_i(t) \ S \ x_i(t)] \tag{1}$$

while the output activation of the OR-neuron with its weight w and input z can be written as:

$$y(t) = \mathop{S}_{i=0}^{n} [w_i(t) \ T \ z_i(t)] \tag{2}$$

The commonly used operator for T is the min-operator, while for S is the max-operator.

The IF-THEN rule of the FANN is trained using back-propagation learning algorithm, which the weight updating process is done by using formula of:

$$W_{new} = W_{old} + \Delta W(t) \tag{3}$$

$$\Delta W(t) = -\alpha \frac{\partial E}{\partial w}(t) + \beta \Delta w_{(t-1)} \tag{4}$$

As in the backpropagation learning rule, the calculation of the derivative error to its connection weight in this FANN should be done using the derivative of min and max operators with respect to its connection weight. These derivatives, however, are not linear in some conditions, and one approach to solve this problem is by using the Lukasiewicz's linearization formula for those derivatives, such that [6-7]:

$$\frac{\partial}{\partial w_j}\min(w_j, x_j) = \begin{cases} 1 & \text{if } w_j \le x_j \\ 1 - w_j + x_j, & \text{if } w_j > x_j \end{cases}$$

$$\frac{\partial}{\partial f}\max(f(w_j), M_j^*) = \begin{cases} 1, & \text{if } f(w_j) \ge M_j^* \\ 1 + f(w_j) - M_j^*, & \text{if } f(w_j) < M_j^* \end{cases} \tag{5}$$

where $M_j^* = max_i\,(M_i)$, $M_i = min(w_i,x_i))$, $f(w_j) = min(w_j,x_j)$, x_j input signal, and w_j denotes the weight of the neuron.

Derivation of the error function for the OR-neuron, then, can be written as:

$$\frac{\partial E}{\partial w_{jk}} = (y_k - t_k)\frac{\partial}{\partial f}\{\max[f(w_{jk}), M_k^*]\}.\frac{\partial}{\partial w_{jk}}(\min(w_{jk}, z_{jk})) \tag{6}$$

while for the AND-neuron:

$$\frac{\partial E}{\partial v_{ij}} = \{\sum_{k=1}^{m}(y_k - t_k).\frac{\partial}{\partial f}\max[f(z_j), M_j^*].\frac{\partial}{\partial z_j}\min[z_j, w_{jk}]\} * \frac{\partial}{\partial g}\min[g(v_{ij}), M_i^{\#}].\frac{\partial}{\partial v_{ij}}\max(v_{ij}, x_i) \tag{7}$$

where $f(z_j) = min(w_{jk},z_j)$, $g(v_{ij}) = max(v_{ij},x_i)$, $M_i^{\#} = min[g(v_{ij})]$, and $M_j^*=max[f(z_j)]$.

Result of the F-Neuro MLP learning process also determines the neuron with its weak connections, which can be removed to find the ideal topology of network. In this paper, the removing its weak connection for optimizing the network structure is done by evolving the process through Genetic Algorithms that will be explained in the next section.

III. Optimization of F-Neuro MLP Through Evolutionary Genetic Techniques

Like the artificial neural networks, genetic algorithms (GAs) are also one of the most popular techniques among numerous branch of computational intelligence. GA can also be feasibly and powerfully used to solve optimization problems in diverse fields, because GAs can immediately provide a critical value for a variable at a certain function during repeatedly searching processes. GAs is a searching algorithm that developed based on natural selection of genetics and evolution. The underlying principles of GAs were first developed by Holland and its mathematical framework was developed and presented in Holland's pioneering book [12], which is intensively observed and implemented by Goldberg [13].

The basic element processed by a GA is the string formed by concenating substrings, each of which is a binary coding of a parameter of the search space. Thus, each spring represents a point in the search space and hence a possible solution to the problem. Each string is decoded by an evaluator to obtain its objective function value. This value, which should be minimized by the GAs, is converted to a fitness value which determines the probability of the individual undergoing genetic operators. The population then evolves from generation to generation through the application of the genetic operators. The total number of strings included in a population is kept unchanged trough generations. A simple genetic algorithm that yields good results in many practical problems is composed of these operators: *reproduction*, *crossover* and *mutation*.

As previously described, the size of an ANN determines its performance. It is obviously known that if the size of the network is too small then the model will not be capable to represents the desired function. However, if the size is too big, the network will memorize all the examples by forming a large lookup table, but not be able to generalize well to the inputs that have not been learning before. As like other statistical models, neural networks are subject to over-fitting when there are too many parameters (i.e., weights) in the model. Other additional problem is occurred, because the network size also determines the length of learning process. If there are m examples, and $|W|$ weights, each epoch takes $O(m|W|)$ calculation time.

The optimization of the F-Neuro MLP through GAs is then initially done by making a network with a rather big and complex structure. In their optimization process, GAs will search the most optimal subset of the initial basic structure. Each subset structure will become an individual in the population to be processed, which is represented by an individual string. As the knowledge representation formed in F-Neuro MLP is kept in its connections, the GAs optimization is directed in its F-Neuro MLP connection weights. Preliminary experimental result showed that optimization over networks connection weights work more efficient and accurate compare with that of networks hidden neurons. The optimization procedure of GAs is then implemented by initially encoding the problem and defining the objective function.

The process of the problem encoding can be summarized as follows. As the problem parameter that should be optimized in the network structure lies in its connection weights, all these connections are encoded to a chromosome chain. Each chromosome forms a binary string *<100….1001>* that represents the network structure, and the chromosome length equivalents to the number of weights in the network, that is

*number_of_input_neurons *number_of_hidden_neurons + number_of_hidden_neurons *number_of_output_neurons*.

Suppose v_{ij} represents the connection weight between neuron in the input layer and neurons in hidden layer, w_{jk} represents the connection weight between neurons in hidden layer and neurons in output layer, with I the number of input neuron, J the number of hidden neuron, K the number of output neurons, then the chromosome chain will represents all connection in the network in a sequence of :

$$v_{00},v_{01}\ldots v_{0(J-1)},v_{10}\ \ldots\ v_{1(J-1)},\ \ldots\ ,v_{(I-1)0}\ \ldots\ v_{(I-1)(J-1)},$$
$$w_{00},w_{01}\ldots w_{0(K-1)},w_{10}\ldots w_{1(K-1)},\ \ldots\ \ w_{(J-1)0}\ \ldots\ w_{(J-1)(K-1)}.$$

Each gene/bit in the chromosome/string is mapped one by one to each connection weight of the FANN. The gene value represents the activation state of connection weight, with a value of *1* means the connection is activated while value of *0* means that the connection is not activated. The non-activated connection will not be involved in both forward and backward phase of the learning process.

The objective function of the system to be optimized is done through its fitness value. To calculate the fitness value, each individual chromosome is decoded back to a F-Neuro MLP structure and be trained using backpropagation learning algorithm. Weight initialization of each structure is performed using Nguyen-Widrow method [19], and the network is trained until small error tolerance is accomplished or a maximum epoch is reached. After the learning process is completed, the fitness value is calculated by:

*number_of_non_activated_connections / (error_rate * number_of_epochs).*

By using fitness value evaluated by the objective function, GAs searches an individual best network topology with large number of non-activated connection weights, small error rate, and small number of epochs. Schematic diagram of the methodology for the optimization of F-Neuro MLP topology through GAs is depicted in Fig.1

An example of a schematic diagram of the F-Neuro MLP system is depicted in Figure 2. This system consists of a fuzzifier that fuzzifing the crips data from each sensor, a fuzzy-neural network for recognizing the unknown-odors, and the genetic algorithms that optimizing the F-Neuro MLP topology for higher recognition capability.

The most appropriate Genetic Algorithms parameter that used for the optimization of the F-Neuro MLP topology is as follows. The number of generations: 10; number of population: 60; crossover probability: 0.6; mutation probability: 0.1; learning rate: 0.2, momentum: 0.2; error rate: 0.01 and maximum epoch: 200. An example of the optimized F-Neuro MLP as a result from the use of genetic algorithms is depicted in Figure 3. It is shown that the number of neuron is exactly the same with that of the un-optimized neural networks, however, the number of neural connection weights is reduced significantly.

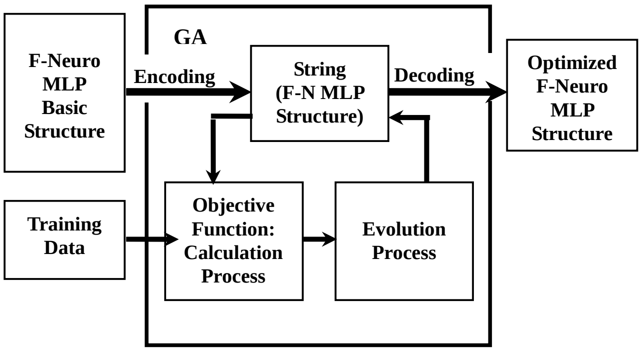


Figure 1. Schematic diagram of the optimization method of F-Neuro MLP through genetic algorithms.

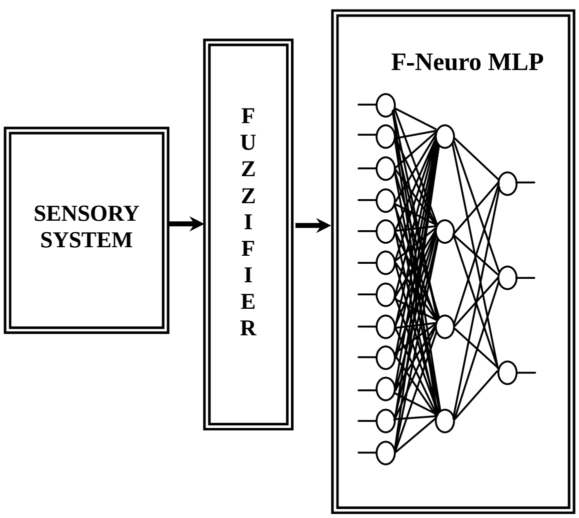


Figure 2. The architectural structure of the developed fuzzy-neural networks that is used as pattern classifier in the artificial odor recognition system.

IV. Optimized F-Neuro as a Pattern Classifier in Odor Recognition System

The experiments are designed to elaborate the capability of the developed odor recognition system to recognize and determine the-unlearn of mixture odors. The odor recognition system is consists of a quartz crystal microbalance as a sensor, and a frequency counter for measuring the shifted frequency of the sensor as it absorbed the odorant molecule, and a computer to perform neural network analysis of the data and determined the odorant category.

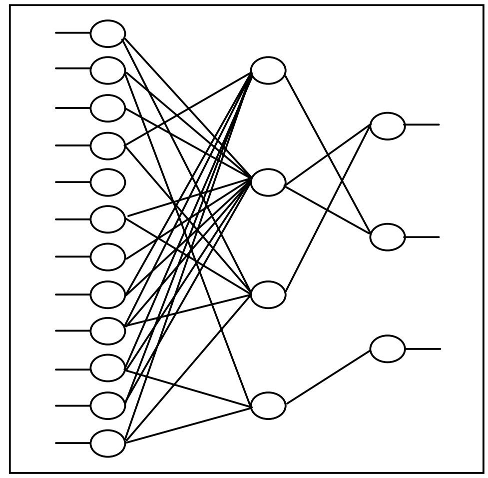


Figure 3. The optimized structure of F-Neuro MLP system using genetic algorithms.

In this experiment, two categories of sensory system are used, with 8 sensors of 20 MHz quartz resonator sensors as the first category of sensory systems and another 16 sensors of the same frequency base resonator as the second sensory system. For both of the sensory systems, each sensor is constructed by applying a sensitive membrane on the two surfaces of those quartz resonator sensors. A sample of aroma is injected and evaporated in the chamber, and the frequency shift is measured at the equilibrium point, before the next sample is repeatedly injected through the same procedure. The characteristic-frequency of the sensor reduces when the odorant molecules are adsorbed onto the membrane, and recover to its properties after de-adsorbtion process using a fresh air that purges the sensory systems. This phenomenon is called the mass-loading effect [20].

Since the shifted frequency is proportional to the total mass of the adsorbed odorant molecules, it is possible to use this mechanism as the fingerprint of the odor concern. To increase the accuracy of the recognition system, various types of membrane-coated sensors are necessary, which is arranged as an arrayed sensor.

The two-mixture of odors are prepared by mixing 50% of aroma-based odor (citrus, canangga and rose) and 50% of alcohol with various gradient concentrations, ranging from 0% to 70%. The data used for the learning stage and its recognition tests are obtained from 10 experiments of each two-mixture of odors, where 100 data are taken from each sensor for each experiment, and the training/testing paradigm is determined to be 70%: 30%. Table 1 shows in details the used sample odors including with percentage of the alcohol concentration.

Based on the number of classes in each experiment, three groups of experiments are designed, simulating the degree of difficulties on recognizing the unlearn odors, i.e. 6 classes of two-mixture odor, 12 classes of two-mixture odor and 18 classes of the two-mixture odor, respectively. Table 2 shows that 12 classes experiments are conducted by combining every two of 6 classes experiments, while 18 classes experiments are conducted by combining all of the 6 classes experiment into only one experiment.

Table 1. Two-mixture odor with various gradient alcohol concentrations

No	Type of odor-mixture	Odor-mixture with various gradient alcohol concentration
1	**CnAlch**	CnA0%, CnA15%, CnA25%,
	Cannagga based odors	CnA35%, CnA45%, CnA70
2	**RoAlch**	CiA0%, CiA15%, CiA25%,
	Rose based odors	CiA35%, CiA45%, CiA70
3	**CiAlch**	CiA0%, CiA15%, CiA25%,
	Citrus based odors	CiA35%, CiA45%, CiA70

Table 2. Experimental design for recognizing the unlearn two-mixture of odor with different number of classes

Number of Classes	Data composition of the Two-Mixture odors		
6 classes	RnAlch	CiAlch	CnAlch
12 classes	RnAlch + CiAlch	CiAlch + CnAlch	CnAlch + RnAlch
18 classes	RnAlch + CiAlch + CnAlch		

Results of experiment on recognizing the unlearn odor within 6 classes of two-mixture odors using 8 sensors are depicted in Table 3, while for 12 classes of two-mixture odors are depicted in Table 4, respectively. It is shown in the Table 3 that the average recognition rate of the Back-Propagation neural is about 83.9%, PNN is about 96.6%, while fuzzy neural is 99.1%, respectively. The highest recognition rate, 100% could be achieved when the GA-optimized fuzzy neural is used.

Table 4 showed that the recognition rate on recognizing unlearn odor within 12 classes of two-mixture odors. As the number of classes increases, the recognition rates of all of the neural systems are decreased. Same with that of 6 classes of two-mixture odors, recognition rate of Back-Propagation neural system is the lowest, with only 53.1% in average. This results show that the conventional Back-Propagation neural system could not be utilized when it is used to discriminate two-mixture of odors with higher number of classes. As also shown in this table, the un-optimized fuzzy neural system still shows higher recognition rate with the average of 81.7%, which is nearly equal to the PNN system of 81.0%; however, the optimized fuzzy neural shows an average of 91.9%.

In order to more deeply seen the ability of various neural systems used as a classifier, an odor recognition system using 16 sensors is used. Results of experiment on recognizing the unlearn odor within 6 classes of two-mixture odors using the system with 16 sensors are depicted in Table 5, while for 12 classes of two-mixture odors are depicted in Table 6, respectively.

Comparison of the overall neural systems performances are depicted in Figure 4 and Figure 5 respectively, which is done based on the average recognition rate as shown in the respective tables. As clearly seen in the Figure 4, increasing the number of sensors increases the recognition capability of the neural system, especially for BPNN, while for the other neural system, the increment are not significantly changes.

However, when the neural system is considered to distinguish 12 classes of the mixture odors, as it is clearly seen in Figure 5, increasing the number of sensors has higher impact on its neural performance. Increment of almost 35% is achieved for BP-NN, while for the other three neural systems is about 10%. Again it is shown that PNN and the Fuzzy-Neuro MLP has almost the same capability, while the optimized Fuzzy-Neuro MLP shown higher recognition capability.

Table 3. Comparison of recognition rate of neural systems to discriminate unlearn odor within 6 classes of two-mixture odors using8 sensors

	CiAlch	CnAlch	RoAlch	Average
GAF-NMLP	100%	100%	100%	100%
F-NMLP	99.3%	98.2%	99.9%	99.1%
PNN	93.7%	99.6%	96.6%	96.6%
BP-NN	68.0%	98.2%	85.6%	83.9%

Table 4. Comparison of recognition rate of neural systems to discriminate unlearn odor within 12 classes of two-mixture odors using 8 sensors

	CiAlch	CnAlch	RoAlch	Average
GAF-NMLP	92.3%	95.5%	97.9%	91.9%
F-NMLP	83.3%	79.4%	82.3%	81.7%
PNN	84.2%	78.2%	80.6%	81.0%
BP-NN	50.5%	41.6%	67.3%	53.1%

Table 5. Comparison of recognition rate of neural systems to discriminate unlearn odor within 6 classes of two-mixture odors using 16 sensors

	CiAlch	CnAlch	RoAlch	Average
GAF-NMLP	100%	100%	100%	100%
F-NMLP	100%	100%	100%	100%
PNN	93.7%	99.6%	96.6%	96.6%
BP-NN	88.9%	90.8%	90.0%	89.9%

Table 6. Comparison of recognition rate of neural systems to discriminate unlearn odor within 12 classes of two-mixture odors using 16 sensors

	CiAlch	CnAlch	RoAlch	Average
GAF-NMLP	100%	98.8%	100%	99.6%
F-NMLP	90.7%	92.4%	89.0%	90.7%
PNN	89.4%	88.3%	90.8%	89.5%
BP-NN	83.8%	76.8%	86.6%	81.0%

Table 7 shows the comparison results of the recognition rate of the overall used neural systems as a classifier for odor recognition system using 8 sensors and 16 sensors. When the fuzzy neural systems are applied for recognizing the unlearn odors within 18 classes of two-mixture odors, the recognition capability of the un-optimized fuzzy neural system decreases significantly, to only 61.1%. This recognition rate is not enough to discriminate unlearn mixture odor properly. In contrary, the GA-optimized fuzzy neural system still shows its higher recognition rate of about 83.3%. This results show that the GA-optimized fuzzy neural system is necessary when more difficult task should be encountered in discriminating two-mixture of odors.

Figure 4. Comparison of recognition rate of various neural system to discriminate unlearn odor within 6 classes of two-mixture odors using 8 and 16 sensors

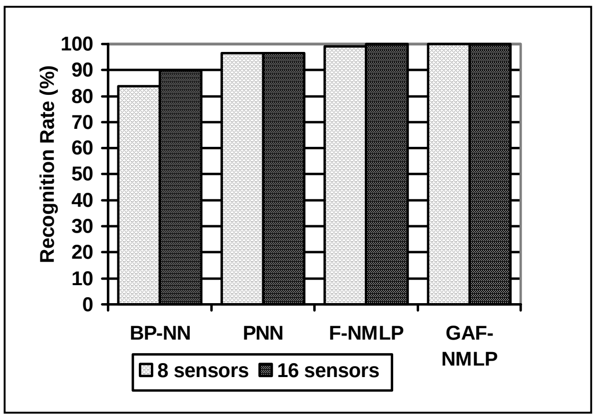
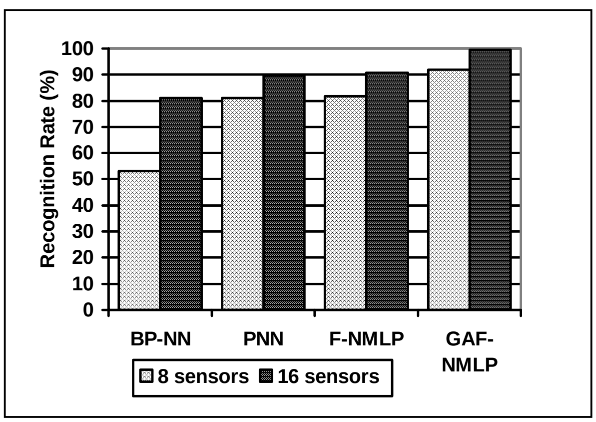


Figure 5. Comparison of recognition rate of various neural system to discriminate unlearn odor within 12 classes of two-mixture odors using 8 and 16 sensors

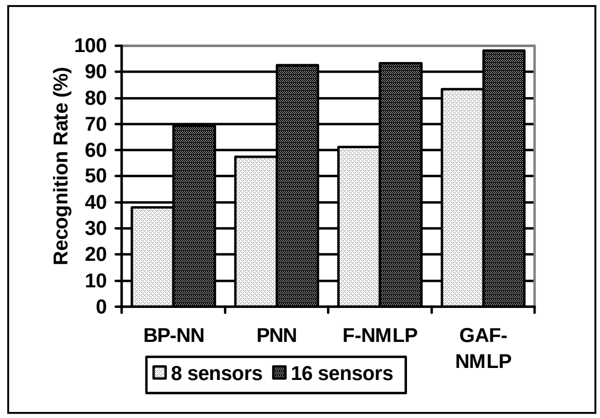


Increasing the number of the used sensors is significantly seen from this experiment. As clearly seen in the Figure 6, the recognition rate of the Fuzzy-Neuro MLP and PNN are shown to be nearly the same, that is about 93%, while only about 60% when using hardware system with only 8 sensors. It can also be seen that the developed GA optimized Fuzzy-Neuro MLP still can increased its recognition rate up to 98.2%.

Table 7. Comparison of recognition rate of neural systems to discriminate unlearn odor using 8 sensors and 16 sensors

	8 Sensors	16 Sensors
GA-Fuzzy NN	83.3%	98.2%
Fuzzy NN	61.1%	93.3%
PNN	57.6%	92.6%
BP	38.1%	69.2%

Figure 6. Comparison of recognition rate of various neural system to discriminate unlearn odor within 18 classes of two-mixture odors using 8 and 16 sensors



V. Conclusion

A method for optimization of the weight connections in Fuzzy-Neuro MLP structure through Genetic Algorithms is developed and used as a pattern classifier subsystem in the artificial odor recognition system. The size and topology of the neural network are optimized in term of its cost function, i.e. its error rate, learning computational cost, and its generalization capability. The performance of this GA-optimized Fuzzy-Neuro MLP is then compared with that of un-optimized Fuzzy-Neuro MLP, PNN and BP neural system. The experiment results showed that the GA-optimized Fuzzy-Neuro MLP method has successfully found the nearly optimized fuzzy-neural structure, by omitting the its weak weight connections, which in return can provide higher recognition rate. It is showed from the experiments that PNN always performed higher recognition capability compare with that of Back-Propagation method, which has nearly the same value with the un-optimized Fuzzy-Neuro MLP. Increasing the number of unlearned classes to be recognized has decreasing the ability of the system to discriminate odors properly. Improvement of the hardware system using higher number of the used sensors, i. e. 16 sensors instead of 8 sensors, has increased the recognition capability of the system, even using a PNN or the un-optimized Fuzzy Neuro MLP. However, the GA optimized Fuzzy-Neuro MLP has the highest recognition capability, for all of the used data sets.

ACKNOWLEDGEMENT

The authors would like to express their gratitude to Ministry of Research and Technology of Indonesia for their support funding of this research through RUT X under contract number 14.06/SK/RUT/2003

REFERENCES

[1] R.R. Yager, "Using fuzzy logic to build neural networks," *Proc. IFSA'91*, Brussels, Belgium, pp. 210-213, 1991.

[2] Z-Q. Liu and F. Yan, "Fuzzy Neural Network in Case-Based Diagnostic System," *IEEE Trans. on Fuzzy Systems*, 5, 209-222, 1997.

[3] J. R. Yang, "ANFIS: Adaptive-network-based-fuzzy inference system," *IEEE Trans. Syst., Man, Cybern*, 23, pp.665-685, 1993.

[4] M. M. Gupta and D. H. Rao, "On the principles of fuzzy neural networks," *Fuzzy Sets and Systems*, 61, pp.1-18, 1994.

[5] B. Kusumoputro, H. Budiarto and W. Jatmiko, "Fuzzy-neuro LVQ and its comparison with fuzzy-LVQ in odor discrimination system", *Journal of ISA Transactions*, 41, pp.395-407, 2002

[6] W. Pedrycz and A. F. Rocah, Fuzzy – set based model of neurons and knowledge based networks, *IEEE Trans. Fuzzy Syst*., vol. 1, Nov. 1993, 254 – 266.

[7] W. Pedrycs, "Fuzzy neural networks with reference neurons as pattern classifiers," *IEEE Trans. Neural Networks*,3, pp. 770-775, 1992.

[8] Z-Q. Liu and F. Yan, "Fuzzy Neural Network in Case-Based Diagnostic System," *IEEE Trans. on Fuzzy Systems*, 5, 209-222, 1997.

[9] H.R. Reed, " Pruning algorithm- A survey" , *IEEE Transaction on Neural Networks*, 4, pp. 740-747, 1993.

[10] G. Thimm and E. Fiesler, "Neural network pruning and pruning parameters", Presented on the 1st Online Workshop on Soft Computing, Nagoya, Japan, 1996, http://www.bioele.nuee.nagoya-u.ac.jp/wsc1

[11] B. Andree, "Constructive learning algorithms for feed-forward neural networks", Ph.D. dissertation, Univ. Utrecht, The Netherlands, 1995.

[12] J. H. Holand, *Adaptation in Natural and Artificial Systems*, Cambridge, Massachusetts, MIT-Press, 1995.

[13] D. E. Goldberg, *Genetic Algorithm in Search, Optimization, and Machine Learning*, Addison – Wesley, 1989.

[14] L.M. Brasil, F.M. de Azevedo, J.M. Barreto and, M. Noirhomme, A new approach to classical backpropagation algorithm for neuro-fuzzy-ga systems learning, *In Proc. The IASTED International Conference on AI and Soft Computing*, Mexico, 1998.

[15] J-S. R. Jang, *Neuro – Fuzzy and Softcomputing: A Computational Approach to Learning and Machine Intelligence*, (Prentice Hall, 1997).

[16] G. J. Klir, *Fuzzy Sets, Uncertainty and Information*. New York:Prentice Hall, 1988.

[17] Russel and Norvig, *Artificial Intelligence: A Modern Approach*, Prentice Hall, 1995.

[18] K. Hirota and W. Pedrycs, "Fuzzy logic neural networks: Design and computations," Proc. Int. Joint Conf. Neural Networks, Singapore, Nov. 1991, vol. 3, pp.88-21.

[19] L. Fausett, *Fundamentals of Neural Networks*, (Prentice Hall, 1994).

[20] W.H. King, Piezoelectric sorption detector, *Anal. Chem*., 36, 1964, 1735.