Iranian Journal of Fuzzy Systems **Vol. 2, No. 2 ,** (2005) pp. 1-13

INTEGRATED ADAPTIVE FUZZY CLUSTERING (IAFC) NEURAL NETWORKS USING FUZZY LEARNING RULES

Y. S. KIM AND Z. ZENN BIEN

ABSTRACT. The proposed IAFC neural networks have both stability and plasticity because they use a control structure similar to that of the ART-1(Adaptive Resonance Theory) neural network. The unsupervised IAFC neural network is the unsupervised neural network which uses the fuzzy leaky learning rule. This fuzzy leaky learning rule controls the updating amounts by fuzzy membership values. The supervised IAFC neural networks are the supervised neural networks which use the fuzzified versions of Learning Vector Quantization (LVQ). In this paper, several important adaptive learning algorithms are compared from the viewpoint of structure and learning rule. The performances of several adaptive learning algorithms are compared using Iris data set.

1. **Introduction**

A neural network is a network of interconnected neurons. These neurons are connected via weights which are adapted during learning to improve performance. Neural networks are powerful tools for processing numerical data because of their computational capability. On the other hand, fuzzy logic represents uncertainty which we encounter in human thought.

Self-organizing neural networks adjust their internal weights to find inherent structures in a data set. Among self-organizing neural networks, the Kohonen Self-Organizing Feature Map [10] and the ART-1 neural network [3] have been used frequently. The Kohonen Self-Organizing Feature Map needs to initialize weights using small random numbers. However it can cause underutilization problems depending on the structure of a data set. The ART-1 neural network has an emphasis on resolving the stability-plasticity dilemma. This neural network is plastic to learn new input patterns but stable enough to maintain previous learning. Also, it can prevent outliers from deteriorating weights due to its use of the vigilance parameter. It does not have the underutilization problem but it is sensitive to noise.

Fuzzy logic was introduced by Zadeh [17]. Fuzzy logic represents uncertainty which we encounter in human thoughts. Bezdek fuzzified the c-mean algorithm by using the fuzzy membership value [2]. This Fuzzy c-Means algorithm is based on minimization of the fuzzified objective function. It may converge to a local minimum and has the initialization problem: different initial conditions can yield different results. However it

Invited paper: Received in May 2005

Key words and phrases: Neural networks, Fuzzy logic, Fuzzy neural networks, Learning rule, Fuzzification.

is less sensitive than the Kohonen Self-Organizing Feature Map because it uses parallel learning which requires large amount of memory because it needs to store the fuzzy membership values of the input patterns in the clusters. There are several variations of the Fuzzy c-Means algorithm.

A lot of interest has been shown in interfacing between fuzzy logic and neural network. Huntsberger and Ajjimarangsee incorporated fuzzy logic into the Kohonen Self-Organizing Feature Map by using fuzzy membership values instead of a learning rate [7]. This algorithm is forced to terminate by just reducing the size of the neighborhood to zero, but such a modification does not guarantee those weights to converge. Bezdek et al. fuzzified the Kohonen Clustering Network by integrating the Fuzzy c-Means model into the learning rule of the Kohonen Clustering Network [1]. Carpenter et al. fuzzified the ART-1 neural network [4]. In this fuzzy neural network, the input pattern should be normalized to the range [0,1]. Such a normalization can yield poor performance in real application. This fuzzy neural network uses a fuzzy min-max operator in the learning rule. Simpson also fuzzified the ART-1 neural network [16].

The IAFC neural networks are the fuzzy neural networks which have both the stability and the plasticity of the ART-1 neural network because the IAFC neural networks use the control structure which is similar to that of the ART-1 neural network [8]. They are stable enough to preserve significant past learning but plastic enough to incorporate new input pattern whenever it might appear. They control the number of clusters and the size of clusters by the vigilance parameter. However the meaning of the vigilance parameter of the IAFC neural networks is different from that of the ART-1 neural network because they use a reinterpretation of the vigilance parameter in the Euclidean domain. In the IAFC neural networks, the vigilance parameter is related to a distance threshold or cluster diameter [13]. They do not need to initialize weights by small random numbers and therefore they solve the underutilization problem. The unsupervised IAFC neural network is the improved version of the IAFC neural network [9]. It simplifies the method of selecting the winner and uses a learning rule which is a fuzzified version of the leaky learning rule.

Learning rules are categorized into two groups. The first group is the unsupervised learning rule. The second group is the supervised learning rule. The Kohonen learning rule is famous among unsupervised learning rules. It updates the weight of a selected class in proportion to the difference between the input pattern and the weight of a selected class [1-4]-[7-10]-[12-13]-[16-17]. This difference is multiplied by the learning rate which is used to guarantee the weights to converge. Huntsberger and Ajjimarangsee fuzzified the Kohonen learning rule by using fuzzy membership values instead of a learning rate [7]. However their learning rule does not guarantee those weights to converge. Kim and Mitra also fuzzified the Kohonen learning rule [8]. They used the combination of a function of iterations, the π membership function [14], and the fuzzy membership value instead of a learning rate. Chung and Lee fuzzified the Kohonen learning rule by using the combination of a learning rate and fuzzy membership value instead of a learning rate [6]. Bezdek et al. fuzzified the Kohonen learning rule by integrating the Fuzzy c-Means model into it [1].

LVQ is a supervised learning rule. LVQ moves the winner's weight toward the input pattern if the classification is correct [11] and away from the input pattern if the classification is incorrect. Chung and Lee proposed Fuzzy Learning Vector Quantization (FLVQ) which incorporates fuzzy membership value with LVQ [5]. They used the combination of a learning rate and the difference between the target membership value and actual membership value. But, it is not easy to get the target membership value in the real situation.

This paper proposes two fuzzy learning rules which are the fuzzified versions of LVQ. These fuzzy learning rules use a combination of a function of iterations, the π membership function, and the fuzzy membership value instead of the learning rate of LVQ. We considered the degree of belongingness of the current input pattern in the winning cluster by using the fuzzy membership value when we update the weight of a winning cluster. The second fuzzy learning rule uses the difference between one and the fuzzy membership value instead of the fuzzy membership value of the first fuzzy learning rule. This replacement comes from the reinterpretation of the meaning of the fuzzy membership value. In general, we use the fuzzy membership value as a measure of belongingness of the input pattern in the specific cluster. Moreover, the fuzzy membership value can be used to measure the relative location of the input pattern to the centroids of existing clusters.

We incorporate the proposed fuzzy learning rules into the structure of the IAFC neural network. This incorporation of the proposed fuzzy learning rules converts the IAFC neural network into one of two supervised neural networks which we will call the supervised IAFC neural network 1 and the supervised IAFC neural network 2. These supervised neural networks control the number of classes and the size of classes by the vigilance parameter. They do not need to initialize the weights using small random numbers. Therefore, they solve the initialization problem.

We compare the performance of several adaptive learning algorithms using Iris data set which is a benchmark data set for comparing the performances of clustering algorithms.

2. IAFC Neural Networks

The IAFC neural networks have both stability and plasticity because they use a control structure similar to that of the ART-1 neural network (Figure 1). The IAFC neural networks control the size of classes and the number of classes by the vigilance parameter. The unsupervised IAFC neural network uses the unsupervised learning rule which is based on the fuzzy leaky learning rule. The supervised IAFC neural networks use the supervised learning rules which are based on fuzzified versions of LVQ.

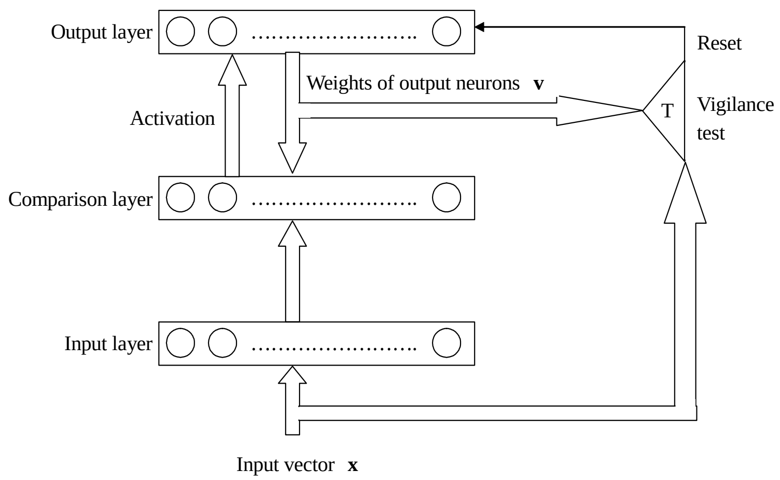

FIGURE 1. The Structure of IAFC Neural Networks

2. 1 Unsupervised IAFC Neural Network

In this neural network, a winner is determined by the Euclidean distance. An output neuron whose weight has the smallest Euclidean distance to the input pattern becomes a winner. The Ith output neuron,

$$I = \min_i \left\| \mathbf{x} - \mathbf{v_i}(t) \right\|,$$

where $\mathbf{x}$ is the input pattern and $\mathbf{v_i}(t)$ is the weight of the ith output neuron, wins the competition.

After selecting a winning output neuron, the unsupervised IAFC neural network performs the vigilance test according to the vigilance criterion:

$$e^{-\mu_I} \left\| \mathbf{x} - \mathbf{v}_I(t) \right\| \le T,$$

where $\mathbf{x}$ is the input pattern, $\mathbf{v}_I(t)$ is the weight of the Ith output neuron, and μ_I is the fuzzy membership value of the input pattern $\mathbf{x}$ in the Ith cluster. μ_I is defined as

$$\mu_I = \frac{\left[\dfrac{1}{\left\| \mathbf{x} - \mathbf{v}_I(t) \right\|^2} \right]^{\frac{1}{m-1}}}{\sum_{j=1}^{n} \left[\dfrac{1}{\left\| \mathbf{x} - \mathbf{v}_j(t) \right\|^2} \right]^{\frac{1}{m-1}}},$$

where m is a weight exponent and is experimentally set to 2. But, when the number of existing clusters is one, the vigilance criterion is $\left\| \mathbf{x} - \mathbf{v}_1(t) \right\| \le T$.

If the winning output neuron satisfies the vigilance test, the weights of all output neurons are updated regardless of winning or losing using the following fuzzy learning rule:

$$\mathbf{v}_i(t+1) = \mathbf{v}_i(t) + \lambda_{fuzzy}(t)\left[\mathbf{x} - \mathbf{v}_i(t)\right] \qquad \text{for all i,}$$

where $\lambda_{fuzzy}(t)$ is the fuzzy learning rate at time t. The $\lambda_{fuzzy}(t)$ is $f(t)\cdot \mu_i \cdot \pi\left(x, \mathbf{v}_i(t), T\right)$, where $f(t)$ is a function of the number of iterations, μ_i is the fuzzy membership value of the input pattern in the ith cluster, and $\pi\left(x, \mathbf{v}_i(t), T\right)$ is the fuzzy within-cluster membership value of the input pattern in the ith cluster. $f(t)$ is $\frac{1}{k(t-1)+1}$, where k is the constant and t is the number of iterations. $\pi\left(\mathbf{x}, \mathbf{v}_i(t), T\right)$ is defined as

$$\pi\left(\mathbf{x}, \mathbf{v}_i(t), T\right) = \begin{cases} 1 - 2\left(\frac{\left\|x - \mathbf{v}_i(t)\right\|}{T}\right)^2, & \text{when } 0 \le \left\|x - \mathbf{v}_i(t)\right\| \le \frac{T}{2} \\ 2\left(1 - \frac{\left\|x - \mathbf{v}_i(t)\right\|}{T}\right)^2, & \text{when } \frac{T}{2} \le \left\|x - \mathbf{v}_i(t)\right\| \le T \\ 0, & \text{when } \left\|x - \mathbf{v}_i(t)\right\| \ge T \end{cases}$$

where T is the vigilance parameter [14]. The above fuzzy learning rule is based on fuzzification of leaky learning rule and fuzzification of the theory of conditional probability. The leaky learning rule updates weights of both winning and losing output neurons but the updating amount of the weight of an output neuron depends on whether or not it wins [12]. The updating amount of the weight of the winning output neuron is larger than that of losing output neuron. The fuzzy leaky learning rule controls the updating amounts by fuzzy membership values. The fuzzy membership value in the winning class is larger than that of the losing class. The conditional probability is the multiplication of the probability of selecting one class among existing classes with the probability of the existence of the input pattern in the selected class. The probability of selecting one class among existing classes is replaced by the fuzzy membership value. And the probability of the existence of the input pattern in the selected class is replaced by the π function.

This fuzzy neural network does not need to initialize weights by small random numbers because it uses the control structure which is similar to that of the ART-1 neural network. The size and number of classes are controlled by the vigilance parameter.

The summary of the algorithm is as follows:

(1) Initialize parameters (T, k), input neurons, and output neurons.

(2) Apply the input pattern.

(3) If the input pattern is the first input pattern, go to Step (4); Otherwise, go to Step (5).

(4) Assign the input pattern as the weight of the first output neuron. Go to Step (2).

(5) Find a winning output neuron.

(6) If the winning output neuron satisfies the vigilance test, go to Step (7); Otherwise, go to Step (8).

(7) Update the weight of the winning output neuron. Go to Step (2).

(8) Activate the first uncommitted output neuron. The input pattern is set to the weight of the first uncommitted output neuron. Go to Step (2).

2. 2 Supervised IAFC Neural Networks

Supervised IAFC neural networks are the supervised versions of IAFC neural network where supervised learning rules are incorporated with the structure of the IAFC neural network. These neural networks control the size of classes and the number of classes by the vigilance parameter. And they do not need to initialize weights by small random numbers.

After the input pattern is applied to the supervised IAFC neural networks, competition among output neurons occurs in a winner-take-all fashion. The output neuron, of which the weight has the minimum Euclidean distance to the input pattern, wins the competition. The Ith output neuron,

$$I = \min_i \left\| \mathbf{x} - \mathbf{v}_i(t) \right\|,$$

wins the competition.

After selecting a winning output neuron, the supervised IAFC neural networks performs the vigilance test according to the vigilance criterion:

$$e^{-\mu_I} \left\| \mathbf{x} - \mathbf{v}_I(t) \right\| \le T,$$

where T is the vigilance parameter. But, when the number of existing classes is one, the vigilance criterion is $\left\| \mathbf{x} - \mathbf{v}_1(t) \right\| \le T$.

If the winning output neuron satisfies the vigilance test, the supervised IAFC neural network 1 updates the weight of the winning output neuron as follows:

$$\mathbf{v}_I(t+1) = \mathbf{v}_I(t) + f(t)\cdot\pi\left[\mathbf{x}, \mathbf{v}_I(t), T\right]\cdot\mu_I\cdot\left[\mathbf{x} - \mathbf{v}_I(t)\right] \qquad \text{if x is classified correctly,}$$
$$\mathbf{v}_I(t+1) = \mathbf{v}_I(t) - f(t)\cdot\pi\left[\mathbf{x}, \mathbf{v}_I(t), T\right]\cdot\mu_I\cdot\left[\mathbf{x} - \mathbf{v}_I(t)\right] \qquad \text{if x is classified incorrectly,}$$
$$\mathbf{v}_i(t+1) = \mathbf{v}_i(t) \qquad \text{for all } i \neq I,$$

The above learning rule is based on the fuzzification of LVQ. It uses a function of the number of iterations, the π function, and the fuzzy membership value instead of the learning rate of LVQ.

On the other hand, the supervised IAFC neural network 2 updates the weight of the winning output neuron as follows:

$$\mathbf{v}_I(t+1) = \mathbf{v}_I(t) + f(t)\cdot\pi\left[\mathbf{x}, \mathbf{v}_I(t), T\right]\cdot\left[1-\mu_I\right]\cdot\left[\mathbf{x} - \mathbf{v}_I(t)\right] \qquad \text{if x is classified correctly,}$$
$$\mathbf{v}_I(t+1) = \mathbf{v}_I(t) - f(t)\cdot\pi\left[\mathbf{x}, \mathbf{v}_I(t), T\right]\cdot\left[1-\mu_I\right]\cdot\left[\mathbf{x} - \mathbf{v}_I(t)\right] \qquad \text{if x is classified incorrectly,}$$
$$\mathbf{v}_i(t+1) = \mathbf{v}_i(t) \qquad \text{for all } i \neq I.$$

The above learning rule is based on considering the relative location of the input pattern to the decision boundary. The input pattern, which locates near the decision boundary, has more information about the optimal decision boundary. Therefore, the above learning rule updates more for the input pattern near decision boundary than the input pattern far from decision boundary using $1-\mu_I$.

The summary of the algorithm is as follows:

(1) Initialize parameters (T, k), input neurons, and output neurons.

(2) Apply the input pattern.

(3) If the input pattern is the first input pattern, go to Step (4); Otherwise, go to Step (5).

(4) Assign the input pattern as the weight of the first output neuron. Go to Step (2).

(5) Find a winning output neuron.

(6) If the winning output neuron satisfies the vigilance test, go to Step (7); Otherwise, go to Step (8).

(7) Update the weight of the winning output neuron. Go to Step (2)

(8) Activate the first uncommitted output neuron. The input pattern is set to the weight of the first uncommitted output neuron. Go to Step (2)

3. **Comparison of Adaptive Learning Algorithms**

Three adaptive learning algorithms are compared in the viewpoint of structure and learning rule. The Kohonen Self-Organizing Map has an underutilization problem depending on the structure of a data set. ART-1 neural network is sensitive to noise. Fuzzy c-Means algorithm requires large amount of memory.

3. 1 Kohonen Self-Organizing Feature Map [10]

In this neural network a winner is determined using Euclidean distance. An output neuron whose weight has the smallest Euclidean distance to an input pattern becomes a winner. The weights of a winner and its neighborhood are updated using the following learning rule:

$$\mathbf{v}_i(t+1) = \mathbf{v}_i(t) + \alpha(t)\left[\mathbf{x} - \mathbf{v}_i(t)\right] \qquad \text{for } i \in N_I(t),$$

where $\mathbf{v}_i(t)$ is the current weight of ith output neuron, $\mathbf{x}$ is the input pattern, $\alpha(t)$ is the learning rate at time t, and $N_I(t)$ is the neighborhood of the Ith output neuron at time *t*. In the above learning rule, $\alpha(t)$ is $0<\alpha(t)<1$ and decreases when t goes to ∞.

This neural network fixes the number of clusters at the initial stage. The size and the number of clusters are controlled by the initialized number of clusters. Also, this neural network requires the initialization of weights, but it can cause the underutilization problem depending on the structure of a data set.

3. 2 ART-1 Neural network [3]

In this neural network an output neuron which receives the largest input activation wins the competition. Then the bottom-up and top-down weight of a winning output neuron are updated if a winning output neuron satisfies the vigilance test. The top-down weight of a winning output neuron is updated using the following learning rule:

$$t_{Ji}(t+1) = t_{Ji}(t) \cdot x_i ,$$

where $t_{Ji}(t)$ is the top-down weight from the Jth output neuron to the ith input neuron and x_i is the ith element of the input pattern. Since the above learning rule is the logical "AND", the component of a top-down weight which becomes 0 cannot be changed back to 1. Therefore, ART-1 neural network is sensitive to noise which changes 1 to 0. But it does not have the underutilization problem because it does not need to initialize weights randomly. It controls the size and the number of clusters by the vigilance parameter.

3. 3 Fuzzy c-Means Algorithm [2]

In the Fuzzy c-Means algorithm, the winner is decided by two methods. These two methods are called the maximum-membership method and the nearest-center classifier [15]. In the max-membership method, the cluster in which the input pattern has the largest membership becomes the winner. In the nearest-center classifier, the cluster whose centroid is closest to the input pattern becomes the winner.

This algorithm updates all cluster centroids regardless of winning or losing using the following equation:

$$\mathbf{v}_{i,t} = \frac{\sum_{k=1}^{n} \mu_{ik,t}^{m} \mathbf{x_k}}{\sum_{k=1}^{n} \mu_{ik,t}^{m}} \qquad \text{for all i,}$$

where $\mu_{ik,t}$ is the fuzzy membership value of the kth input pattern in the ith cluster at the time t. This is a parallel learning scheme. All input patterns contribute to update the centroids of clusters. Therefore, it requires large amount of memory. This algorithm needs to initialize the value of cluster. This can cause the initialization problem. And this algorithm controls the size and the number of clusters by initializing the number of clusters.

4. Test and Results

We use Iris data set, which is a benchmark data set for comparing the performance of clustering algorithms, to compare the performances of adaptive learning algorithms. Iris data set consists of 150 four-dimensional data and has 3 classes. Each class has 50 data values.

The unsupervised IAFC neural network clustered Iris data set into 3 clusters on the range from 1.2 to 1.8 for the vigilance parameter T when K is 0.5. During iterative operations, if the square-root of all squared difference between previous and current weights, $\sqrt{\left\|\mathbf{v}(t)-\mathbf{v}(t-1)\right\|^2}$, is less than 0.001, we considered them to basically equal and stopped iterative operations.

The unsupervised IAFC neural network yielded 11 misclassifications and required 36 iterations when K is 0.5 and T is 1.3 (Figure 2). The Kohonen Self-Organizing Feature Map clustered Iris data set incorrectly into one cluster when weights were initialized by small random numbers in the range 0 to 1. And it clustered Iris data set incorrectly into two clusters when weights were initialized by random numbers in the range 0 to 7. Therefore, we used arbitrarily selected data from Iris data set as initial weights. Fourteen misclassifications occurred and more than 100 iterations were required (Figure 2).

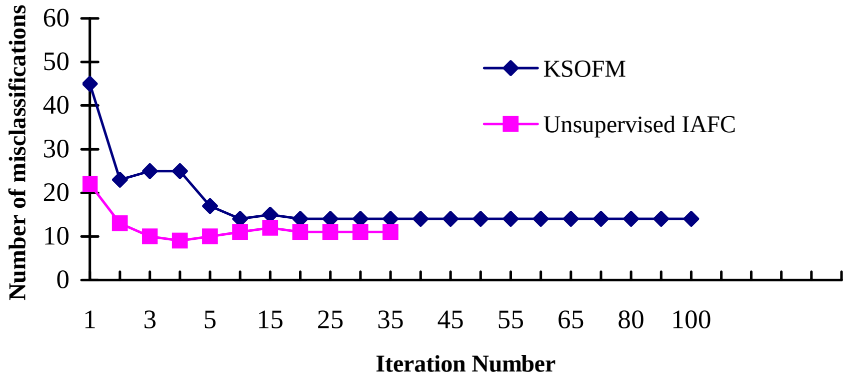

FIGURE 2. Performance Comparison between the Unsupervised IAFC Neural Network And the Kohonen Self-Organizing Feature Map

Fuzzy c-means yielded 16 misclassifications. Figure 3 compares the misclassifications yielded by the unsupervised IAFC neural network with those by two other algorithms in the form of confusion matrices.

(a) Unsupervised IAFC neural network

Output \ Desired	1	2	3
1	45	6	
2	5	44	
3			50

(b) Kohonen Self-Organizing Feature Map

Output \ Desired	1	2	3
1	37	1	
2	13	49	
3			

(c) Fuzzy c-Means

Output \ Desired	1	2	3
1	37	3	
2	13	47	
3			

FIGURE 3. Comparison of Results of Three Algorithms

To compare the performances of the supervised IAFC neural network 1 and the supervised IAFC neural network 2 with that of the back propagation neural network, 75 points of data were used as a training data set. We chose 25 data points from each class and used the remaining 75 data points as a testing data set.

The supervised IAFC neural network 1 took 11 iterations to train as shown in Figure 4 and the supervised IAFC neural network 2 took 9 iterations to train as shown in Figure 5.

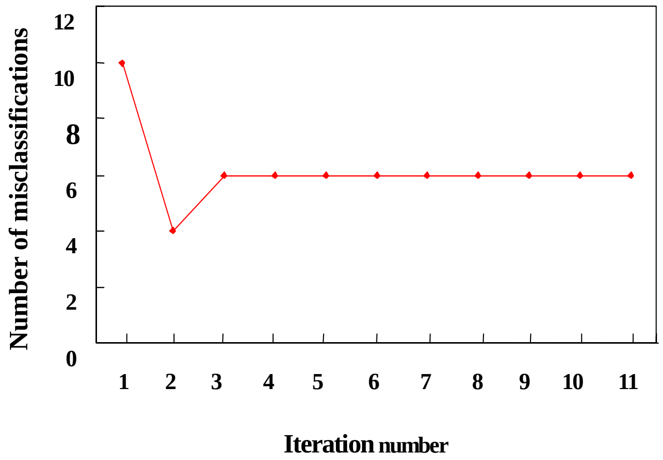

FIGURE 4. The Number of Misclassifications Versus the Iteration Number When the Supervised IAFC Neural Network 1 Was Trained

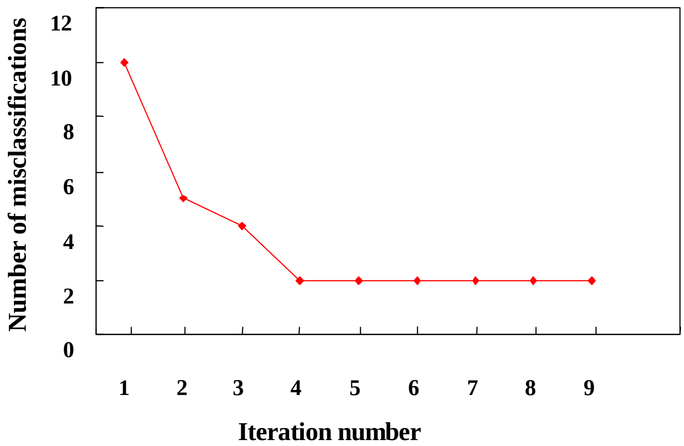

FIGURE 5. The Number of Misclassifications Versus the Iteration Number When the Supervised IAFC Neural Network 2 Was Trained

The supervised IAFC neural network 1 yielded 6 misclassifications when T is 1.55 and K is 0.5. The supervised IAFC neural network 2 yielded 2 misclassifications when T is 1.6 and K is 0.5. The back propagation neural network yielded 5 misclassifications. Figure 6 compares the results of the three neural networks.

Output \ Desired	1	2	3
1	**19**	**0**	
2	**6**	**25**	
3			**25**

(a) Supervised IAFC neural network 1

Output \ Desired	1	2	3
1	**20**	**0**	
2	**5**	**25**	
3			**25**

(b) Back propagation neural network

Output \ Desired	1	2	3
1	**23**	**0**	
2	**2**	**25**	
3			**25**

(c) Supervised IAFC neural network 2

FIGURE 6. Comparison of Results of Three Neural Networks

5. **Conclusion**

Adaptive learning algorithms are compared from the viewpoint of structure and learning rules. The IAFC neural networks have both stability and plasticity. The result shows that the Kohonen Self-Organizing Feature Map has an underutilization problem. The unsupervised IAFC neural network does not need to initialize weights by small random numbers. Therefore, it solves the underutilization problem. The results show that the performance of the unsupervised IAFC neural network is better than performances of the Kohonen Self-Organization feature Map and the Fuzzy c-Means algorithm.

We proposed the supervised IAFC neural network 1 and the supervised IAFC neural network 2 which incorporated new fuzzy learning rules into the structure of the IAFC neural network. These new fuzzy learning rules are based on the fuzzification of LVQ. And the results show that the performance of the supervised IAFC neural network 2 is better than the performance of the back propagation neural network.

REFERENCES

[1] J. C. Bezdek, E. C. Tsao and N. R. Pal, *Fuzzy Kohonen clustering networks*, Proceeding of the First IEEE Conference on Fuzzy System, (1992) 1035-1043.

[2] J. C. Bezdek, *Pattern recognition with fuzzy objective function algorithms*, Plenum Press, New York, (1981).

[3] G. A. Carpenter and S. Grossberg, *A massively parallel architecture for a self-organization neural pattern recognition machine,* Computer vision, Graphics, and Image processing, **37** (1987) 54-115.

[4] G. A. Capenter, S. Grossberg and D. B. Rosen, *Fuzzy ART : fast stable learning and categorization of analog pattern by an adaptive resonance systems*, Neural Networks, **4** (1992) 759-772.

[5] F-L Chung and T. Lee, *A fuzzy learning model for membership function estimation and pattern classification,* Proceedings of the third IEEE conference on Fuzzy systems, **1** (1994) 426-431.

[6] F. L. Chung and T. Lee, *Fuzzy competitive learning*, Neural Networks, **7** (1992) 539-551.

[7] T. L. Huntsberger and P. Ajjimarangsee, *Parallel self-oraganizing feature maps for unsupervised pattern recognition*, Int. J. General System, **16** (1990) 357-372.

[8] Y. S. Kim and S. Mitra, *An adaptive integrated fuzzy clustering model for pattern recognition,* Fuzzy

Sets and Systems, **65** (1994) 297-310.

[9] Y. S. Kim, *An unsupervised neural network using a fuzzy learning rule*, Proceedings of 1999 IEEE International Fuzzy Systems, **I** (1999) 349-353.

[10] T. Kohonen, *Self-organization and associative memory*, 3rd ed., Springer-Verlag, Berlin, (1984)

[11] T. Kohonen, *The Self-organizing map*, Proceedings of the IEEE, **78** (1990) 1464-1480.

[12] C-T Lin and C. S. G Lee, *Neural fuzzy systems-a neuro-fuzzy synergism to intelligent systems*, Prentice-Hall, New Jergy, (1996).

[13] B. Moore, *ART-1 and pattern clustering,* Proceedings of the 1988 Connectionist Models Summer School, (1981) 174-185.

[14] S. K. Pal and S. Mitra, *Fuzzy dynamic clustering algorithm*, Pattern Recognition Letters, **11** (1990) 525-535.

[15] T. J. Ross, *Fuzzy logic with engineering applications*, McGraw-Hill, New York, (1997).

[16] P. K. Simpson, *Fuzzy min-max neural network-part 2 : clustering*, IEEE Trans. on Fuzzy Systems, **1** (1993).

[17] L. A. Zadeh, *Fuzzy sets*, Information and Control, **8** (1965) 338-352.

YONG SOO KIM*, DIVISION OF COMPUTER ENGINEERING, DAEJEON UNIVERSITY, DAEJEON, 300-716, KOREA

E-mail address: **kystj@dju.ac.kr**

Z. ZENN BIEN, DEPARTMENT OF ELECRICAL ENGINEERING AND COMPUTER SCIENCE, KAIST, DAEJEON, 305-701, KOREA

E-mail address: **zbien@ee.kaist.ac.kr**

* CORRESPONDING AUTHOR