♣ 특별 기고 ♣

Fuzzy Neural Network Pattern Classifier

Dae Su Kim*and Terrance L. Huntsberger †

September 24, 1991

Abstract

In this paper, we propose a fuzzy neural network pattern classifier utilizing fuzzy information. This system works without any *a priori* information about the number of clusters or cluster centers. It classifies each input according to the distance between the weights and the normalized input using Bezdek's[1] fuzzy membership value equation. This model returns the correct membership value for each input vector and can find several cluster centers. Some experimental studies of comparison with other algorithms will be presented for sample data sets.

1 Introduction

Clustering or pattern classification divides a collection of data sets into several classes based on some degree of similarity. Statistical pattern recognition view methods can be found in [9]. Several cluster seeking methods are known so far including the *Isodata* algorithm, which is an iterative self-organizing data analysis technique[29, 14]. The *Isodata* method is a heuristic approach by using statistical analysis. General theory for pattern recognition can be found in [28, 29, 30, 31, 33]. Topics of image processing for pattern recognition can be found in [15, 26, 11]. Hartigan [16] discussed several clustering algorithms. The nearest neighbor (1-NN)

*Artificial Intelligence Section, Electronics and Telecommunications Research Institute, P. O. Box 8, DaeDuk Science Town, Daejeon, Korea 305-606

†Intelligent Systems Laboratory, Department of Computer Science, University of South Carolina, Columbia, SC. 29208.

classification rule assigns a pattern x of unknown classification to the class of its nearest neighbor for any distance measure defined over the pattern space. The k-nearest neighbor (k-NN) strategy works by assigning a pattern x to the class which has the most neighbors of equal classification among k-nearest neighbors [1].

The K-means algorithm is based on the minimization of a performance index which is defined as the sum of the squared distances from all points in a cluster domain to the cluster center [29]. A more detailed explanation and algorithm will appear in the later section.

The fuzzy c-means algorithm is broadly applied to optimal fuzzy partition [2], pattern classification [4] and image segmentation [18, 5] etc. The fuzzy c-means algorithm uses iterative optimization of an objective function based on a weighted similarity measure between each data point and each of cluster centers [1]. The convergence theorem for the fuzzy c-means clustering algorithm is in [1, 3] and an improved proof was done in 1987 [17]. The fuzzy c-means algorithm is widely studied and applied to several topics [4, 5, 10, 32, 27, 23]. The K-means algorithm is a special case of fuzzy c-means algorithm.

Neural network models have now been applied to visual pattern recognition[13], phonetic typewriter[25] and robotics. Recently, the combination of neural network and fuzzy concept is proposed[19, 20, 21, 22]. A neural network is defined as a system of many processing units operating in parallel whose specification is determined by the network, processing units and weights between units. Hopefully artificial neural network working in parallel can obtain real-time response.

This paper presents a fuzzy neural network(SONN) algorithm for pattern classification utilizing fuzzy concept. Experimental results show clear and correct classification for the sample data. We compared our results with those of the k-means algorithm and fuzz-c means algorithm those are most frequently used for cluster classification.

2 Cluster finding methods

2.1 Partitioning

Two ways of partitioning a data set S can be considered [1]. One method is *hard partitioning*. This method has membership values of either 0 or 1. Only one winning class has 1 and all others have 0. A hard partition of S is a set of subsets of S, $\{S_0, S_1, S_2, ..., S_{c-1}\}$ that satisfies those conditions:

$$S_i \neq \phi,\ 0 \leq i \leq c-1,\ S_i \cap S_j = \phi,\ i \neq j$$

and

$$\bigcup_{i=0}^{c-1} S_i = S.$$

where c is the total number of clusters. In the neural network method, only the winning neuron has value 1 and all other neurons have value 0. This is the "winner take all" strategy. If some competing neurons have similar possibilities to win, only one winner gets all even if the value difference is very small. Therefore this hard partitioning method is considered to be too rigid and a more reasonable partitioning method to assign its membership value proportional to the degree of similarity or closeness is desired. For this purpose fuzzy membership partitioning is proposed. Let u_{ji} be a membership value in the closed interval [0, 1] such that input ξ_i belongs to cluster S_j. The following are the fuzzy membership conditions:

$$\sum_{j=0}^{c-1} u_{ji} = 1,\ 0 < \sum_{i=0}^{n-1} u_{ji} < n, \qquad (1)$$

and

$$u_{ji} \in [0, 1].$$

That means the sum of fuzzy memberships for any input ξ_i should be 1 and no subset is empty and no subset is all of the whole set S.

2.2 K-means Algorithm

The K-means algorithm is a sequential cluster center updating method to minimize the sum of the squared distances from all vectors to the cluster centers. The convergence of this algorithm is not guaranteed, but sometimes this algorithm generates reasonable results. The performance is largely dependent upon the number of class centers specified, the order of input vectors, and the way we specify the initial centers. The disadvantage of this algorithm is that results are not stable. The K-means algorithm is given below.

K - means Algorithm

step 1. Choose K arbitrary initial cluster centers $v_1(1), v_2(1), ..., v_K(1)$. We may choose the first K vectors from the vector set. We can also choose them randomly.

step 2. At the kth step,

Distribute each sample vector x among K cluster domain by using the following decision formula.

$$x \,\epsilon S_j(k) \; if \; \|x - v_j(k)\|^2 < \|x - v_i(k)\|^2 \tag{2}$$

for all i = 1, 2, 3, ..., K, $i \neq j$, where $S_j(k)$ denotes the set of vectors whose cluster center is $v_j(k)$

step 3. Compute the new cluster centers $v_j(k+1)$, $j = 1, 2, ..., K$, such that the sum of the squared distances from all points in $S_j(k)$ to the new cluster center is minimized. The new cluster centers are calculated by averaging the vectors that belong to that cluster.

step 4. If new cluster centers are the same as the previous ones for $j = 1, 2, ..., K$, then the algorithm stops. Otherwise go back to step 2.

2.3 Fuzzy c-means Algorithm

Fuzzy theory was first introduced by Zadeh [34] in 1965. Fuzzy concepts can yield more accurate representations of data structures. Whereas hard partitioning has either 0 or 1 membership value, a fuzzy partitioning allows us any value in the closed interval 0 and 1. It makes it possible to have relative values instead of "all or nothing" strategy.

In the case of the hard c-means algorithm, the position of a cluster center is found to be the average of the positions of all the patterns in that cluster. The result is based on minimizing the sum of variances of all variables k for each cluster i. The functional for hard c partition is as follows:

$$J_w(U, v) = \sum_{k=1}^{n} \sum_{i=1}^{c} (u_{ik})(d_{ik})^2, \tag{3}$$

where u_{ik} is the hard membership value of pixel k of cluster i. u_{ik} is 1 if input vector x_k belongs to cluster i and 0 otherwise. d_{ik} is the distance between input vector x_k and cluster center of index i.

Bezdek [1] developed the fuzzy $c - means$ algorithm(FCM) in 1980 based on Dunn's [10] fuzzy $c - means$ clustering algorithm for iterative optimization of the least square error functional J_m. This leads to infinite families of fuzzy clustering algorithms which have been developed and used by a number of investigators. Initial generalization of functional J_w was developed by Dunn [10] and this algorithm became a special case of the first infinite family of fuzzy clustering algorithms based on a least-squared error criterion. The fuzzy c-means functional J_m is as follows:

$$J_m(U, v) = \sum_{k=1}^{n} \sum_{i=1}^{c} (u_{ik})^m (d_{ik})^2, \tag{4}$$

where U is a fuzzy $c - partition$ of X, and u_{ik} is the fuzzy membership value of pixel k of cluster i, $v = \{v_0, v_1, v_2, ..., v_{c-1}\}$ with $v_i \in R^p$ is the cluster center or prototype of u_i, $1 \leq i \leq c$, $(d_{ik})^2 = \|x_k - v_i\|^2$ and $\| * \|$ is any inner product induced norm on R^p and the weighting exponent $m \in [1, \infty)$.

Input to the fuzzy c-means algorithm consists of a data set of n units and the algorithmic parameters c, m, n, and the convergence threshold (ϵ_L). The algorithm from Bezdek [1] is given below.

Fuzzy c - means Algorithm

step 1. Fix c, $2 \leq c < n$, where c is the number of clusters and n is the number of data items. Choose any inner product norm metric for R^p. Fix $m, 1 \leq m < \infty$. Initialize $U^{(0)}$ in fuzzy c-partition space. Then at step $s, s = 0, 1, 2, ...$:

step 2. Calculate the c cluster centers $\{v_i^{(s)}\}$ with the following formula and $U^{(s)}$.

$$v_i = \frac{\sum_{k=1}^{n} (u_{ik})^m x_k}{\sum_{k=1}^{n} (u_{ik})^m}. \tag{5}$$

where v_i is the center for cluster i.

step 3. Update $U^{(s)}$: calculate the memberships in $U^{(s+1)}$ as follows:

(1) Define T_k and I_k:

$$T_k = 1, 2, ..., c - I_k$$

$$I_k = \{i | 1 \le i \le c; d_{ik} = \|x_k - v_i\| = 0\}$$

(2) For data item k, compute new membership values

(a) $I_k = \emptyset, \Longrightarrow$

$$u_{ik} = \frac{1}{[\sum_{j=1}^{c}(\frac{d_{ik}}{d_{jk}})^{\frac{2}{(m-1)}}]} \tag{6}$$

(b) $I_k \neq \emptyset \Longrightarrow u_{ik} = 0 \ \forall \ i \ \epsilon \ T_k$ and

$$\sum_{i \epsilon I_k} u_{ik} = 1 \tag{7}$$

step 4. Compare $U^{(s)}$ and $U^{(s+1)}$ in a convenient matrix norm: if $U^{(s)} - U^{(s+1)} \le \epsilon_L$, stop; otherwise, increment s by 1 and return to step 2.

By using formulas (2.6) and (2.7), we can determine fuzzy membership values after the distance is determined using SONN algorithm. For example, let's assume that we have four neurons and the square distance between each neuron and the input value is 1, 2, 4, 8 respectively. The relative value u_{ji} is $\frac{8}{15}$, $\frac{4}{15}$, $\frac{2}{15}$, $\frac{1}{15}$ according to the formula in (2.6). If there are some neurons that have distance value 0, then the total fuzzy values of that particular neuron will be 1 by the formula in (2.7).

The limit value of ϵ_L should be sufficiently small. If this value is large, the algorithm will stop without having sufficient stabilization steps. No definite criteria for this limit value is specified. This fuzzy c-means clustering algorithm is proved to have good convergence properties [1, 3]. In spite of its early convergence proof, several counter examples are found [23]. A new improved convergence theorem for the fuzzy c-means clustering algorithm was proved [17]. Windham [32] also discussed cluster validity for the fuzzy c-means clustering algorithm.

3 Fuzzy membership value assignment

We utilize feedback information in the SONN algorithm. Another layer is added to the linear Kohonen network to map the distance d_{ji} into membership values. d_{ji} is square root of the vector product of ξ_i and W_j. One adjustable layer neural network model appears in Figure 1 and fuzzy model for membership value assignment appears in Figure 2. In case of the hard c membership, only the winning node that has the

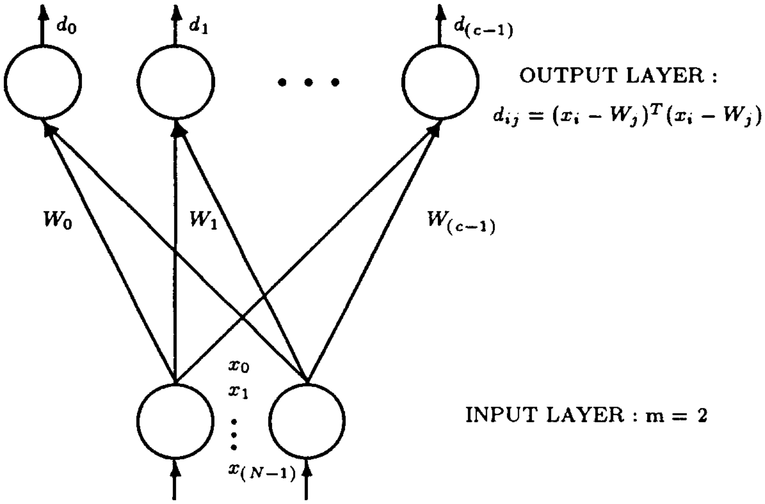

Figure 1: One adjustable layer neural network model.

minimum distance between the input value and the existing weights will have value 1 and all other neurons have membership value 0. This method is the so called "winner take all" strategy.

In our SONN model, fuzzy feedback membership values in the range of [0, 1] will be used. The fuzzy membership equation from Bezdek [1] is as follows.

Fuzzy membership value assignment equation

Fuzzy membership value in [0,1] will be decided by the following equations.

$$u_{ji} = \begin{cases} 1 & \text{if } d_{ji} = 0 \\ 0 & \text{if } d_{ki} = 0, \ (for\ some\ k \neq j,\ 0 \leq k, j \leq c-1) \end{cases} \tag{8}$$

otherwise

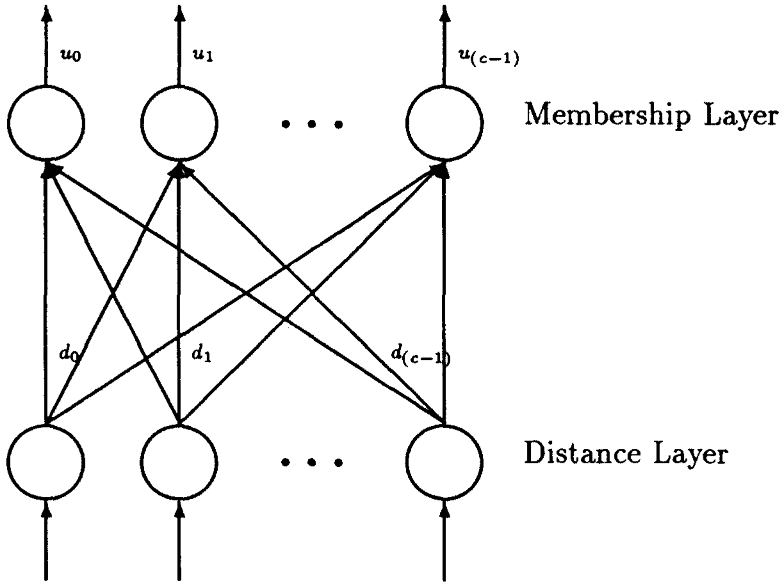

Figure 2: Fuzzy model for membership value assignment.

$$u_{ji} = \frac{1}{[\sum_{l=0}^{c-1}(\frac{d_{ji}}{d_{li}})^2]} \tag{9}$$

where c is the number of cluster. u_{ji} is the fuzzy membership value of input i to the class j. The total sum of u_{ji} is always 1.

For a hard c membership, the output value in {0,1}

$$u_{ji} = \begin{cases} 1 & \text{if } d_{ji} = \min(d_{li}, 0 \leq l \leq c-1) \\ 0 & \text{otherwise} \end{cases}$$

The corresponding membership layer appears at the top level of Figure 3 of our SONN model.

4 SONN Algorithm

We introduce our SONN algorithm.

SONN Algorithm

step 1. Initialize weights W_j for all j randomly between 0 and 1 and set the ring-structure neighborhood radius (NB) to be $\lfloor c/2 \rfloor$ where c is the number of output nodes. Also set the learning rate (LR) between 0 and 1 and set the limit value to be sufficiently small, e.g, 0.0001.

step 2. For each input ξ_i, $i = 1,\ 2, ...,\ n$ where n is the number of input.

1. Find the feedback distance u_{ji} for each input vector.

$$u_{ji} = \frac{1}{[\sum_{l=0}^{c-1} (\frac{d_{ji}}{d_{li}})^{\frac{2}{(m-1)}}]} \tag{10}$$

2. Select the output node j^*, $(0 \leq j^* \leq c-1)$, such that the sum of total distances between ξ_i and node j, d_{j^*i} is minimum.
3. Update the weight change factor dw_i such that $dw_i = (\xi_i - W_i)$. If node j is within the boundary of NB node j^*, then let $dw[j]$ be the difference between input ξ_i and existing weight w. $dw[j]$ will be 0 otherwise.
4. Save the current weight W_j to W_old_j before updating weight.
5. Update weight W_j using the rule:

$$W_j = W_j + LR * dw * u_{ji} * N$$

6. Return the value diff, where diff $= \sum (W_j - W\_old_j)^2$. Add the value of diff to the total_diff using total_diff = total_diff + diff.

step 3. If total_diff > limit then go to step 4 else go to step 5.

step 4. Reduce the learning rate (LR), reset total_diff to 0 and go to step 2.

step 5. If $NB = 0$ then go to step 6. Otherwise reduce NB by 1 and go to step 2.

step 6. Determine fuzzy membership value u_{ji} by using the formulas (3.1) and (3.2).

When we update W, we multiply by some factor N, e.g 2 or 5, to the changing factors if j is the minimum distance neuron. The value of N will be just 1 in other

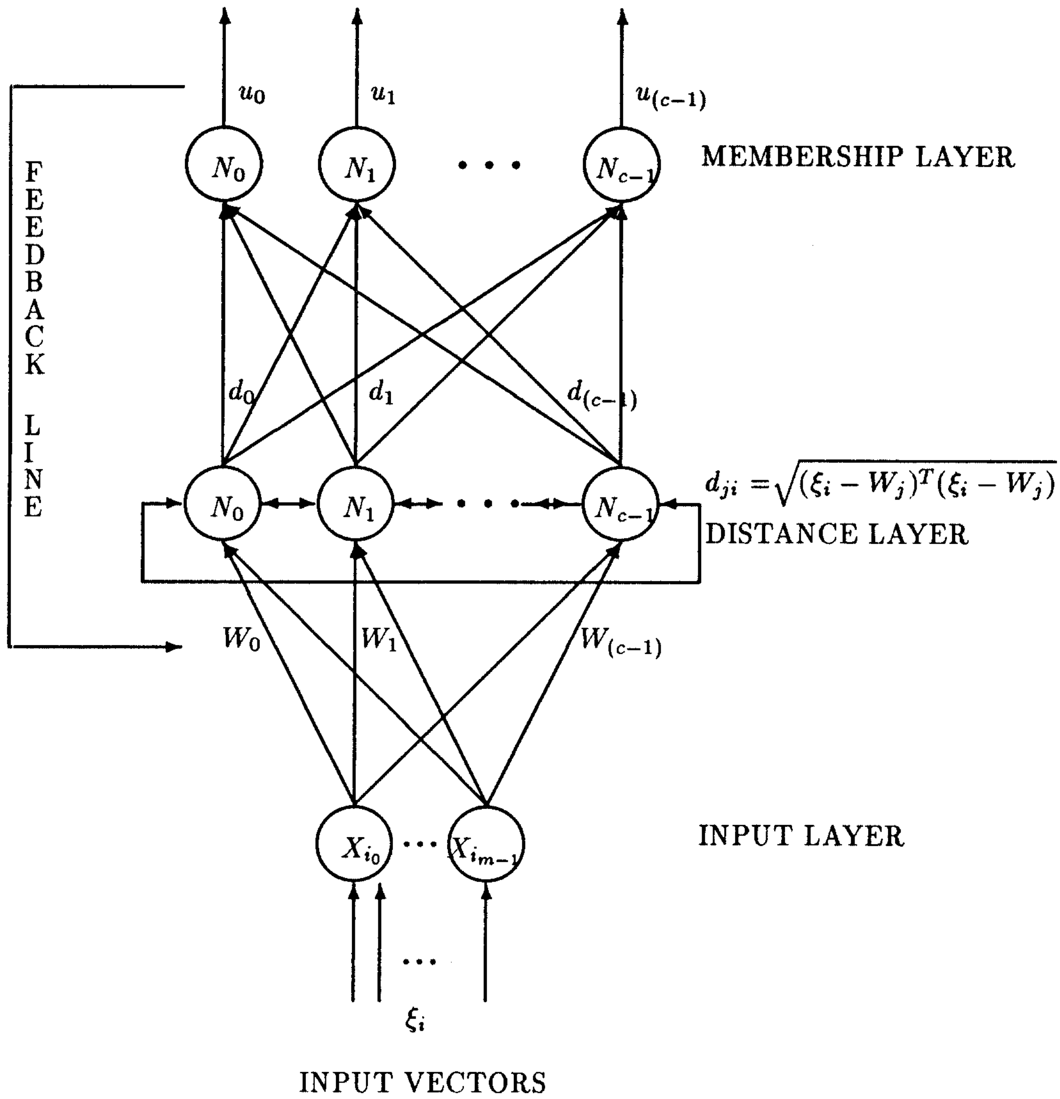

Figure 3: Self-Organizing Neural Network (SONN) model.

cases. If the total changing factor $LR * dw * u_{ji} * N$ is greater than 1, the changing factor is set to 1.

We emphasize the winning node that is the closest to the input. The background for this lateral inhibitory technique can be found in [24]. We can also find a similar technique in Grossberg's $ART1$ and $ART2$ models [6, 7]. A Mexican hat function serves as lateral inhibition based on a function of distance from the winning node. In our work the winning node is emphasized more in that we multiply by some positive integer factor greater than 1. Neurons in the winner's neighborhood boundary perform a similar adjustment. They are multiplied by a factor of 1. Neurons that are out of the neighborhood will not adjust weights at all. This algorithm shows a perfectly correct and clear classification for the testing data. The convergence properties of this algorithm are more difficult to prove and may not be represented by any known closed form, but it is guaranteed to stabilize after some amount of time by using a reduced learning rate [24]. Comparisons with other clustering methods are discussed in the next section.

5 Comparisons with other algorithms

We are going to compare our results with those of k-means algorithm and fuzzy-c means algorithm those are most frequently used for clustering classification.

5.1 Cubic data

Cubic data consists of 1280 three dimensional vectors. Cluster 1 has values (95-100, 0-5, 5-10) and Cluster 2 has values (0-5, 0-5, 95-100) and Cluster 3 has values (96-100, 97-100,88-90). The values between ranges, e.g 95-100, are determined by the random number generator. The order of clusters are also determined by random number. We set limit as 0.005 and consider the learning rate for 10 cases, 0.1, 0.2,..., 1.0. If the percentage difference between the largest and the second largest values is not sufficiently enough, that classification is considered ambiguous and it is included in the error classification.

In SONN Algorithm, the percentage of misclassification error for our testing data is all zero independent of the number of output and the learning rate. That means this algorithm classifies every input data correctly. When the number of output is

3 and its learning rate is 0.1, its three cluster centers are (97.44, 2.60, 7.52), (2.43, 2.55, 97.50) and (98.02, 98.56, 89.00). Those cluster centers are quite reasonable considering the data set. One of the advantage of SONN algorithm is that this algorithm is independent of the number of the ouput nodes. It classifies perfectly for the number of output is 10 or 15. Since the values between ranges, e.g 95-100, are determined by a random number generator, the expected cluster center value is in the middle of the range.

The results are similar for all three cases. SONN algorithm and FCM algorithm are not so sensitive to the limit value, but K-means algorithm sometimes gives unreasonable results. Experimental comparison of Cubic data with several other algorithms appears in Table 1.

Cluster Algorithm	**Cluster**		
	1	2	3
SONN	(97.44, 2.60, 7.52)	(98.02, 98.56, 89.00)	(2.43, 2.55, 97.50)
FCM	(95.53, 3.42, 7.55)	(95.92, 98.41, 88.25)	(2.41, 4.37, 95.74)
K-Means	(97.44, 2.60, 7.50)	(98.01, 98.57, 89.00)	(2.43, 2.54, 97.48)

Table 1: Comparison of cluster centers of *Cubic data* for different algorithms.

5.2 Iris data

The iris data[1] has three clusters. According to the result of our experiment, SONN algorithm and FCM algorithm show similar result for the cluster center. K-means algorithm returns somewhat different result. The 4th vector component in cluster 1 is around 1, SONN algorithm returns around 1, but K-means algorithm returns around 5. The results of SONN and FCM are very similar. Experimental comparison of results for Iris data with the three algorithms appears in Table 2.

Cluster Algorithm	Cluster		
	1	2	3
SONN	(5.54, 2.90, 3.55, 1.08)	(5.03, 3.38, 1.59, 0.30)	(6.37, 2.93, 4.88, 1.64)
FCM	(5.86, 2.75, 4.34, 1.38)	(5.00, 3.41, 1.48, 0.25)	(6.75, 3.05, 5.62, 2.04)
K-Means	(5.11, 3.39, 1.53, 5.07)	(5.30, 3.50, 1.40, 0.20)	(6.34, 2.88, 4.94, 6.24)

Table 2: Comparison of cluster centers of *IRIS* data for different algorithms.

5.3 Chiou's data

Chiou's data set [8] consists of 40 two dimensional vectors. It has two cluster centers. The results of running the three algorithms and Chiou's report are quite similar. Experimental comparison of Cubic data with several other algorithms appears in Table 3.

Cluster Algorithm	Cluster	
	1	2
SONN	(27.57, 18.63)	(7.46, 18.72)
FCM	(27.02, 18.67)	(7.51, 18.55)
K-Means	(26.25, 18.70)	(7.05, 18.54)
Chiou's Report	(27.00, 18.70)	(7.40, 18.00)

Table 3: Comparison of cluster centers of *Chiou's data* for different algorithms.

6 Conclusion

We presented a fuzzy neural network classifier that can find several cluster centers and determine fuzzy clustering membership for any given data sets without *a priori* information utilizing fuzzy information. This model returns the correct membership value for each input vector. The fuzzy membership value is determined by Bezdek's fuzzy membership value equation.

Some experimental comparisons show that our fuzzy neural algorithm is suitable for pattern classification. Further research is on progress concentrating on the correctness and speed up.

References

[1] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum Press, New York, 1981.

[2] J. C. Bezdek and J. C. Dunn, Optimal Fuzzy Partitions: A Heuristic for Estimating the Parameters in a Mixture of Normal Distributions. *IEEE Trans. on Computers*, Aug. 1975.

[3] J. C. Bezdek, A Convergence Theorem for the Fuzzy ISODATA Clustering Algorithms. *IEEE Trans. on Pattern Analysis and Machine Intelligence.* Vol. PAMI-2, NO. 1, January 1980. Morgan Kaufmann Publishers, 1989. Recognition by a Self-organizing Neural Network. *IEEE,Computer*, March 1988.

[4] R. L. Cannon, J. V. Dave, and J. C. Bezdek, Efficient Implementation of the Fuzzy c-Means Clustering Algorithms. *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. PAMI-8, No. 2, pp. 248-255, 1986.

[5] R. L. Cannon, J. V. Dave, J. C. Bezdek, and M. M. Trivedi, Segmentation of a Thematic Mapper Image Using the Fuzzy c-Means Clustering Algorithm. IEEE Transactions on Geoscience and Remote Sensing. Vol. GE-24, NO. 3, May 1986.

[6] G. A. Carpenter and S. Grossberg, The art of Adaptive Pattern Recognition by a Self-organizing Neural Network. *IEEE*, *Computer*, March 1988.

[7] G. A. Carpenter and S. Grossberg, A Massively Parallel Architecture for a Self-Organizing Neural Pattern Recognition Machine. *Computer Vision*, *Graphics*, *and Image Processing*, 37, pp. 54-115, 1987.

[8] E. W. Chiou, Core Zone Scatterplots: A New Approach to Feature Extraction for Visual Displays. Master's thesis, University of South Carolina, 1986.

[9] P. A. Devijver and J. Kittler, *Pattern Recognition as statistical Approach*, Prentice Hall, London, 1982.

[10] J. C. Dunn, A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. J. Cybernet, 31 (3):32-57, 1974.

[11] O. D. Faugeras (Edited by), *Fundamentals in Computer Vision.* Cambridge University Press, 1983.

[12] D. H. Fisher, Knowledge Acquisition via Incremental Conceptual Clustering, *Machine Learning*, 2, pp.139-172, 1987.

[13] K. Fukushima, A Hierarchical Neural Network Capable of Visual Pattern Recognition. *Neural Network.* Vol. 1, 1989.

[14] I. Gath and A. B. Geva, Unsupervised Optimal Fuzzy Clustering. *IEEE Trans. on Pattern analysis and Machine Intelligence.* Vol 1. 11, No. 7, July, 1989.

[15] R. C. Gonzalez and Paul Wintz, *Digital Image Processing.* 2nd Edition, Addison-Wesley, 1987.

[16] J. Hartigan, *Clustering Algorithms*, Wiley, New York, 1975.

[17] R. J. Hathaway, J. C. Bezdek and W. T. Tucker, An improved Convergence Theorem for the Fuzzy c-Means Clustering Algorithms. *The Analysis of Fuzzy Information*(J. C. Bezdek, Ed.), CRC Press, Boca Ralton, 1987. Wiley-Interscience, 1985.

[18] T. L. Huntsberger, C, L. Jacobs and R. L. Cannon, Iterative fuzzy image segmentation. *Pattern Recognition.* 18, No. 2, 1985, pp. 131-138.

[19] Dae Su Kim, Properties and Characteristics of Self-Organizing Neural Networks for Unsupervised Pattern recognition, Ph. D Dissertation, Depart. of Computer Science, University of South Carolina 1990.

[20] Dae Su Kim and T. L. Huntsberger, Self-organizing Neural Networks for Unsupervised Color Image Recognition, 23rd Southeastern Symposium on System Theory, March 1991.

[21] Dae Su Kim and T. L. Huntsberger, Self-organizing Neural Networks for Unsupervised Pattern Recognition, 10th Annual IEEE International Phoenix Conf. on Computers and Communications, March 1991.

[22] Dae Su Kim, Neural Network Models for Unsupervised Pattern Recognition, Korean Information Science Society, Artificial Intelligence group Spring conference, pp. 183-187, 1991.

[23] T. Kim, J. C. Bezdek and R. J. Hathway, Optimality Tests for Fixed Points of the Fuzzy $c - means$ Algorithm. *Pattern Recognition*, Vol. 21, No. 6, pp. 651-663, 1988.

[24] T. K. Kohonen, *Self – Organization and Associative Memory*. Springer-Verlag, Berlin, 1984.

[25] T. K. Kohonen, The Neural Phonetic typewriter. *IEEE Computer*, Vol. 21, No. 3 pp 11-22, March, 1988.

[26] A. Rosenfeld and A. C. Kak, *Digital Image Processing*, 2nd Edition, Vol 1. Academic Press, 1982.

[27] E. Ruspini, Numerical Methods for fuzzy clustering, *Information Science*, Vol. 2, pp. 319-350, 1970.

[28] C. W. Therrien, *Decision Estimation and Classification*. Wiley, 1989.

[29] J. T. Tou and R. C. Gonzalez, *Pattern Recognition Principles*. Addison-Wesley, 1977.

[30] S. Watanabe, *Pattern Recognition : Human and Mechanical*, Wiley-Interscience, 1985.

[31] B. Widrow and R. Winter, Neural nets for Adaptive Filtering and Adaptive Pattern Recognition, *IEEE, Computer*, March, 1988.

[32] M. P. Windham, Cluster Validity for the Fuzzy c-Means Clustering Algorithm. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. PAMI-4, NO. 4, July 1982.

[33] D. D. Wolff and M. L. Parsons, *Pattern Recognition Approach to Data Interpretation*. Plenum Press, 1983.

[34] L. A. Zadeh, Fuzzy Sets. *Information and Control*. 8, pp. 338-353, 1965.