Supervised IAFC Neural Network Based on the Fuzzification of Learning Vector Quantization

Yong Soo Kim[1], Sang Wan Lee[2], Sukhoon Kang[1], Yong Sun Baek[3], Suntae Hwang[4], and Zeungnam Bien[2]

[1] Department of Computer Engineering, Daejeon University, Daejeon, 300-716, Korea
kystj@dju.ac.kr, shkang@dju.ac.kr
[2] Department of Electrical Engineering and Computer Science, KAIST, Daejeon, 305-701, Korea
bigbean@ctrsys.kaist.ac.kr, zbien@ee.kaist.ac.kr
http://ctrgate.kaist.ac.kr/~bigbean
[3] Department of Computer Web Information, Daeduk College, Daejeon, 305-715, Korea
dodo029@ ddc.ac.kr
[4] Department of Information and Communications Engineering, Daejeon University, Daejeon, 300-716, Korea
hwang@dju.ac.kr

Abstract. In this paper, a fuzzy LVQ(Learning Vector Quantization) is proposed which is based on the fuzzification of LVQ. The proposed FLVQ(Fuzzy Learning Vector Quantization) uses the different learning rate depending on the correctness of classification. When the classification is correct, the amount of update is determined by consideration of location of the input vector relative to the decision boundary. When the classification is not correct, the amount of update is determined by the degree of belongingness of the input vector to the winning class. The supervised IAFC(Integrated Adaptive Fuzzy Clustering) neural network 3, which uses FLVQ, is introduced in this paper. The supervised IAFC neural network 3 is both stable and plastic because it uses the control structure which is similar to that of Adaptive Resonance Theory(ART)-1 neural network. We used iris data set to compare the performance of the supervised IAFC neural network 3 with those of LVQ algorithm and backpropagation neural network. The supervised IAFC neural network 3 yielded fewer misclassifications than LVQ algorithm and backpropagation neural network.

1 Introduction

A neural network is a network of interconnected neurons. These neurons are interconnected via weights. These weights are adjusted to improve the performance of neural network. Therefore, a learning rule which controls the adjustment of weights plays an important role on the performance of neural network.

LVQ is one of supervised learning rules. LVQ moves the weight of a winner toward the input vector if the classification is correct[1,2]. On the other hand, LVQ

B. Gabrys, R.J. Howlett, and L.C. Jain (Eds.): KES 2006, Part III, LNAI 4253, pp. 248 – 254, 2006.

moves the weight of a winner away from the input vector if the classification is incorrect. Chung and Lee proposed FLVQ which incorporates fuzzy membership value with LVQ[3,4]. They derived FLVQ by optimizing an appropriate fuzzy objective function which takes into accounts of two goals. The first goal is minimizing the network output error which is the class membership differences between target and actual membership values. The second goal is minimizing the distances between the input patterns and the prototypes of classes. They solved the underutilization problems of LVQ and got better result than that of LVQ. This FLVQ updates the prototypes of classes regardless of winning or losing. The amount of update depends on the difference between the target membership value and the actual membership value in addition to the difference between the input pattern and the prototype of class. The problem of this FLVQ is that it requires target membership value. But, it is not easy to get target membership value in the real situations. Karayiannis also fuzzified LVQ[5,6]. He derived FLVQ by minimizing the average generalized mean between the input vectors and the prototypes using gradient descent. The prototypes are updated through an unsupervised learning process. All prototypes are updated and the amount of update depends on the difference between the input vector and the prototype, the fuzzy membership value, and a learning rate for each prototype. Because it uses batch learning, each prototype is updated with respect to all input vectors. But, it uses a large amount of memory. Karayiannis proposed weighted FLVQ[5]. It is derived by minimizing the weighted generalized mean of the squared distance between the input vector and the prototypes. It is similar to FLVQ by Karayiannis. Tsao et al. also proposed FLVQ[7]. It is similar to FLVQ by Karayiannis.

This paper proposes a fuzzy LVQ which fuzzified LVQ. The proposed FLVQ uses a function of iterations, Π membership function, and the fuzzy membership value instead of the learning rate of LVQ. The Π membership function reduces the effect of outliers to the prototype of class. LVQ uses the same learning rate regardless of the classification is correct or not. However, the proposed FLVQ uses the different learning rates depending on the correctness of classification. The proposed FLVQ uses the difference between one and the fuzzy membership function if the classification is correct. But it uses the fuzzy membership value when the classification is not correct. When the classification is correct, the weighting factor of the data point, which locates near the decision boundary, for updating amount of the weight is larger than the weighting factor of the data point, which locates far from the decision boundary, for updating amount of the weight. This reduces the effect of outliers to the decision boundary. The outliers deteriorate the decision boundary, because the outliers tend to move away the prototypes of the classes from the proper locations for the decision boundary. The proposed FLVQ prevents the outliers from deteriorating the proper decision boundary, because it uses the difference between one and the fuzzy membership value. The fuzzy membership value of an outlier in the class, where it belongs to, is larger than the fuzzy membership value of the data point, which locates near the decision boundary, in the class where it belong to. The larger the fuzzy membership value, the smaller the difference between one and the fuzzy membership value. Because the FLVQ uses the difference between one and the fuzzy membership value, it considers the data point, which locates near the decision boundary, more important when it updates the prototype of class. When the classification is not correct, the proposed FLVQ uses the fuzzy membership value to update the prototype of the selected class. The updating amount of the prototype of the selected class is proportional to the amount of belongingness of the misclassified data point in the selected class.

The proposed FLVQ is integrated into the supervised IAFC neural network 3. The supervised IAFC neural network 3 has both the stability and the plasticity as the ART-1 neural network because it uses the control structure which is similar to that of the ART-1 neural network[8](Fig.1). It is stable to preserve significant past learning but plastic to incorporate new input point whenever it might appear. It controls the number of clusters and the size of clusters by the vigilance parameter. In the supervised IAFC neural network 3, the vigilance parameter is related to a distance threshold or cluster diameter[9,10]. Even though the ART-1 neural network processes binary data, the supervised IAFC neural network 3 processes continuous-valued data. The supervised IAFC neural network 3 uses the Euclidean distance to choose the nearest prototype and calculate thresholds[9,10].

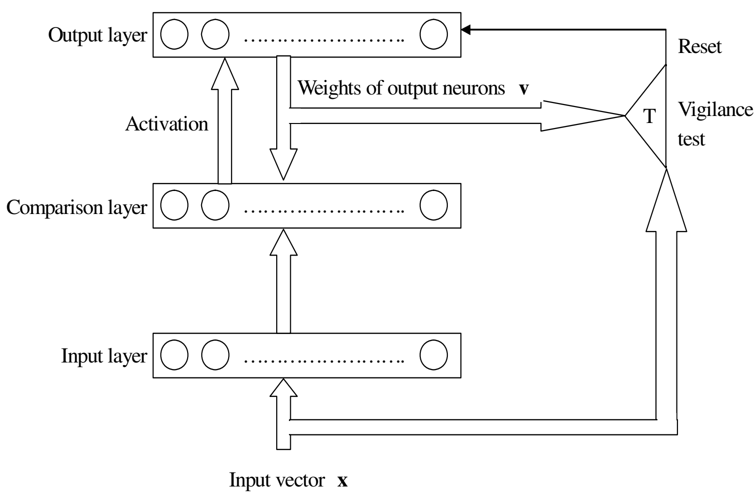


Fig. 1. The structure of the supervised IAFC neural network 3

We compared the performance of the supervised IAFC neural network 3 with those of LVQ algorithm and backpropagation neural network using the iris data set, which is a benchmark data set for comparing the performance of clustering algorithms.

2 Supervised IAFC Neural Network 3

After the input vector is applied to the supervised IAFC neural network 3, competition among output neurons occurs in a winner-take-all fashion. The output neuron, of which weight has the minimum Euclidean distance to the input vector, wins the competition. The I-th output neuron,

$$I = \min_i \left\| \mathbf{x} - \mathbf{v}_\mathbf{i}(t) \right\|, \tag{1}$$

where $\mathbf{x}$ is the input vector and $\mathbf{v}_\mathbf{i}(t)$ is the weight of the i-th output neuron, wins the competition.

After selecting a winning output neuron, the supervised IAFC neural network 3 performs the vigilance test according to the following vigilance criterion :

$$e^{-\mu_I} \left\| \mathbf{x} - \mathbf{v}_I(t) \right\| \le T, \tag{2}$$

where T is the vigilance parameter. The fuzzy membership value μ_I is defined as follows :

$$\mu_I = \frac{\left[\dfrac{1}{\left\| \mathbf{x} - \mathbf{v}_I(t) \right\|^2} \right]^{\frac{1}{m-1}}}{\sum_{j=1}^{n} \left[\dfrac{1}{\left\| \mathbf{x} - \mathbf{v}_j(t) \right\|^2} \right]^{\frac{1}{m-1}}} \tag{3}$$

where n is the number of committed output neurons, and $m \in [1, \infty]$ is a weight exponent. m is experimentally set to 2. However, when the number of committed output neurons is one, the vigilance criterion is $\left\| \mathbf{x} - \mathbf{v}_\mathbf{1}(t) \right\| \le T$.

The dissimilarity measure in Eq. (2) is the relative distance which considers both the Euclidean distance and the relative location of the input vector to the prototypes of the existing classes[9]. The weighting factor for the input vector, which locates far from the decision boundary, is smaller than the weighting factor for the input vector, which locates near the decision boundary. This weighting factor is multiplied by the Euclidean distance between the input vector and the prototype of the winning class. We can compare this relative distance with Mahalanobis distance which considers statistical properties of data[11]. In the case of the Mahalanobis distance, weighting factor is large when the covariance is small. On the other hand, weighting factor is small when the covariance is large.

If the winning output neuron satisfies the vigilance test, the supervised IAFC neural network 3 updates the weight of the winning output neuron as follows:

$$\mathbf{v}_I(t+1) = \mathbf{v}_I(t) + f(t) \cdot \pi\left[\mathbf{x}, \mathbf{v}_I(t), T\right] \cdot (1 - \mu_I) \cdot \left[\mathbf{x} - \mathbf{v}_I(t)\right] \tag{4}$$

if **x** is classified correctly,

$$\mathbf{v}_I(t+1) = \mathbf{v}_I(t) - f(t) \cdot \pi\left[\mathbf{x}, \mathbf{v}_I(t), T\right] \cdot \mu_I \cdot \left[\mathbf{x} - \mathbf{v}_I(t)\right] \tag{5}$$

if x is classified incorrectly,

$$\mathbf{v}_i(t+1) = \mathbf{v}_i(t) \qquad \text{for } i \ne I, \tag{6}$$

where $f(t)$ is the function of iterations. $f(t)$ is defined as $\frac{1}{k(t-1)+1}$, where k is the constant which controls convergent speed. $\pi(\mathbf{x}, \mathbf{v}_i(t), T)$ is defined as

$$\pi(\mathbf{x}, \mathbf{v}_i(t), T) = \begin{cases} 1-2\left(\frac{\|x-\mathbf{v}_i(t)\|}{T}\right)^2, & \text{when } 0 \le \|x-\mathbf{v}_i(t)\| \le \frac{T}{2} \\ 2\left(1-\frac{\|x-\mathbf{v}_i(t)\|}{T}\right)^2, & \text{when } \frac{T}{2} \le \|x-\mathbf{v}_i(t)\| \le T \\ 0, & \text{when } \|x-\mathbf{v}_i(t)\| \ge T. \end{cases} \tag{7}$$

When the classification is correct, the proposed FLVQ moves the weight of the winning class toward the input vector. In Eq. (4), the proposed FLVQ considers the location of the input vector for updating the weight of the winning output neuron using $1-\mu_I$. In Fig. 2, the fuzzy membership value of the input vector B in the class 1 is larger than that of the input vector A in the class 1. The input vector near the decision boundary has more information about the proper decision boundary. The input vector, which locates far from the decision boundary like the input vector B, moves the decision boundary away from the proper position. It deteriorates the decision boundary. Using $1-\mu_I$ can prevent the input vector, which locates far from the decision boundary like the input vector B, from deteriorating the decision boundary. By using $1-\mu_I$ in Eq. (4), the weighting factor for the input vector A is larger than that for the input vector B. On the other hand, when the classification is not correct, the proposed FLVQ moves the weight of the winning class away from the input vector. The proposed FLVQ uses the fuzzy membership value to update the weight of the winning class as in Eq. (5). The updating amount of the weight of the winning output neuron is proportional to the fuzzy membership value.

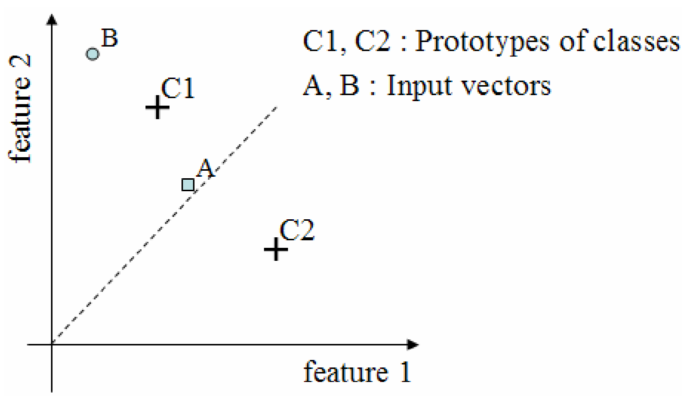
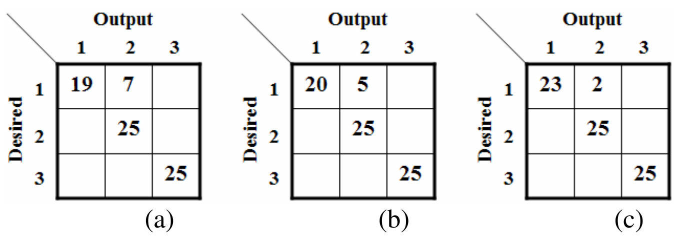


Fig. 2. Consideration of location of the input vector with respect to the decision boundary

3 Test and Result

We used iris data set, which is a benchmark data set for comparing the performance of clustering algorithms, to compare the performance of the supervised IAFC 3 with those of LVQ algorithm and backpropagation neural network. Iris data set consists of 150 four-dimensional data, and 3 subspecies[12]. Each subspecies has 50 data. We chose 75 data arbitrarily from 150 data, and used as a training data set. We chose 25 data from each subspecies. And we used the other 75 data as a testing data set.

During the training, if $\|\mathbf{v}(t)-\mathbf{v}(t-1)\|$ is less than 0.01, we considered the weights to be converged experimentally and stopped the iterative operations of training for the supervised IAFC neural network 3. After that, we tested the supervised IAFC neural network 3. Fig. 3 shows the comparison between the result of the supervised IAFC neural network 3 and the results of LVQ algorithm and backpropagation neural network. The supervised IAFC neural network 3 iterated 7 iterations to train, and yielded 2 misclassifications when T is 1.7 and K is 0.5. We tested LVQ algorithm using MATLAB Toolbox. We tested LVQ algorithm under the condition that the learning rate is 0.01. LVQ algorithm yielded 7 misclassifications. And backpropagation neural network yielded 5 misclassifications. Fig. 4 shows the number of misclassifications versus iteration number when we trained the supervised IAFC neural network 3.



(a) LVQ algorithm (b) Backpropagation neural network (c) Supervised IAFC neural network 3

Fig. 3. Comparison of results using iris data set

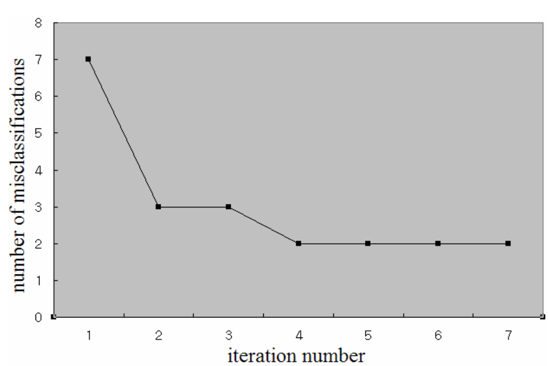


Fig. 4. The number of misclassifications versus the iteration number when the supervised IAFC neural network 3 was trained

4 Conclusion

We proposed FLVQ which is based on the fuzzification of LVQ. The proposed FLVQ uses the different learning rate depending on the correctness of classification. The proposed FLVQ considers the location of input vector relative to the decision boundary. This prevents the outlier from deteriorating the decision boundary.

We used iris data set to compare the performance of the supervised IAFC neural network 3 with those of LVQ algorithm and backpropagation neural network. The supervised IAFC neural network yielded fewer numbers of misclassifications than LVQ algorithm and backpropagation neural network. It required a few iterations to converge experimentally.

References

1. C-T Lin and C. S. G. Lee: Neural Fuzzy Systems – A Neuro-Fuzzy Synergism to Intelligent Systems. New Jersey : Prentice-Hall. (1996)
2. J.C. Bezdek, E. C. Tsao, and N.R. Pal.: Fuzzy Kohonen Clustering Networks. Proceeding of the First IEEE conference on Fuzzy Systems, San Diego. pp.1035-1043. (1992)
3. 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. vol. 1. pp. 426-431. (1994)
4. F. -L. Chung and T. Lee: Fuzzy Learning Vector Quantization, Proceedings of 1993 International Joint Conference on Neural Networks, Nagoya, Vol. 3, pp. 2739-2743. (1993)
5. N. B. Karayiannis: Weighted Fuzzy Learning Vector Quantization and Weighted Fuzzy C-Means Algorithms, IEEE International Conference on Neural Networks, Vol. 2, pp. 1044-1049. (1996)
6. N. B. Karayiannis, and Bezdek, J.C.: An Integrated Approach to Fuzzy Learning Vector Quantization and Fuzzy C-Means Clustering, IEEE Transactions on Fuzzy Systems, Vol. 5, pp. 662-629. (1997)
7. E. C. –K. Tsao, J. C. Bezdek, and N. R. Pal: Fuzzy Kohonen Clustering Networks, Pattern Recognition, Vol 27., No. 5, pp. 757-764. (1994)
8. G. A. Carpenter and S. Grossberg: A Massively Parallel Architecture for A Self-Organizing Neural Pattern Recognition Machine. Computer Vision, Graphics, and Image Processing. vol. 37. pp. 54-115. (1987)
9. Y. S. Kim and S. Mitra: An adaptive integrated fuzzy clustering model for pattern recognition. Fuzzy Sets and Systems. vol. 65. pp. 297-310. (1994)
10. B. Moore: ART1 and Pattern Clustering. Proceedings of the 1988 Connectionist Models Summer School. San Mateo. pp.174-185. (1989)
11. J. T. Tou and R. C. Gonzalez: Pattern Recognition Principles. Massachsetts: Addison-Wesley. (1974)
12. E. Anderson : The IRISes of the Gaspe Penninsula. Bulletin American IRIS Society. Vol. 59. pp. 2-5. (1935)