신용평가를 위한 신경망 Clustering기법의 적용

최홍윤, 정윤(외국어대), 김정원, 최종욱(상명여대)

1. Introduction

Vector quantization(VQ) is a technique developed for solving data encoding problem leading to minimization of reconstruction error in data compression and decompression[Ritter, Martinnetz, and Schulten, 1992]. In searching for an efficient data decompression skill, one seeks to describe as faithfully as possible "the distribution of data points in a high-dimensional space, using only a space of lower dimension" [Ritter, Martinnetz, and Schulten, 1992, p.238]. That is, the most efficient projection of original data onto lower dimensional planes can yield the smallest project error. The technique has been modified and successfully applied and to various fields; image classification [Cannon, Dave and Bezdek, 1986], phoneme signal processing[Kong and Kosko, 1992], and travelling sales person(TSP) problem [Rose, Gurewitz, and Geoffrey, 1993].

In this research, the generalized learning vector quantization (GLVQ) algorithm which modified Kohonens' learning vector quantization (LVQ) algorithm [Kohonen, 1991] and was suggested by Pal, Bezdek, and Tsao[1992] is applied to credit evaluation problem. In fact, this research was initiated to develop an automated credit evaluation system based on neural network training technique. The first experiment with a small set of training data was successful, showing almost 80 percent classification accuracy. However, as the size of training data and testing data increases, the system showed very low classification accuracy, even though 100 percent accuracy on training data set was achieved. To investigate relationship between data characteristics and classification accuracy of [1)]

This research was supported by NON DIRECTED RESEARCH FUND, Korea research foundation

the neural systems, the GLVQ algorithm was applied to two set of credit evaluation data. Even though many research results conducted in neural network community claimed classification accuracy much higher than or at[2) least equal to statistical methods, we found that such a high classification accuracy cannot be achieved on the data set we have. The successful results of previous researches were obtained only when the network was trained with sound data set, having little number of noisy data or being large enough to control noisy data. As a conclusion, the neural system might not be the best solution, or a panacea to business classification.

In the following section, credit evaluation system is introduced to help understand the domain problem we are trying to solve, and then in section three the GLVQ algorithm is discussed. Experimental results and comparative analysis of GLVQ algorithm to two different data set are n section four. Section five is reserved for the final conclusions and detailed discussions on future research.

2. A Neural-Based Credit Evaluation System

Credit evaluation is one of the most important and difficult tasks usually assigned to experienced officers in credit card companies, mortgage companies, banks, consumer goods companies and other financial institutes. Traditionally, credit scoring has been the most widely used method in which applicant's credit is evaluated by picking up appropriate score corresponding to categories of evaluation value, than by summing up into total credit for thresholding. Recently, various method have been introduced to replace the credit scoring system and to provide more objective and convenient tools : statistical method [Majone, 1968; Apilado, Warner, and Dauten, 1974; Edelstein, 1975; Muchinsky, 1975; Beranek, Taylor, 1976; Borzar, 1978; Capon, 1982]. Induction trees(ID3, C3) [Carter and Catlett, 1987], expert system approach [Dungan, 1982; Dungan and Chandler, 1985; Kastner, Apte, Griesmer, Hong, Karnaugh, Mays, and Tozawa, 1986; Messier and Hansen, 1987; O'Leary, 1987] and the neural network researchers have shown that prediction accuracies of the neural network system, that is, the degrees of generalization, are better than or at least equal to those of the statistical methods [Kim, 1992; Odem and Sharda,

---

본 논문은 1994년 학술진흥재단 연구비 지원에 의해 이루어진 연구논문임

1990; Schumann and Lohrbach, 1992]. Therefore, many researchers have devoted their research efforts to enhance the degree of generalization to achieve higher level of prediction accuracy. In determining the degree of generalization, involved are many internal and external factors : the network architecture(number of hidden nodes, input nodes, hidden layers, initial weights, learning rate, momentum, etc.), training algorithm(back propagation, self-organization map, quickprop algorithm, activation function, etc.) and composition of training data set and test data set. Researchers experimented with various architectures by modifying factor values and learning algorithm.

There are some experimental reports on the relationship between training data set and the degree of generalization, with recommendation of the ways to achieve higher generalization capability. Whitley and Karunanithi [29] proposed a partitional learning strategy in which the training space is divided into a set of subspace according to the data characteristics and then each subspace is trained using a separate network. In the data selection step, emphasized is decision boundaries and the central tendencies of decision regions. In the test of 'two-spiral' problems [9], they achieved almost 100 percent correctness ratio, using the border patterns. Fu and Chen [1993] investigated the sensitivity of input vectors on generalization capability, and found that the norm of Jacobian matrix measures the sensitivity of the network performance with respect to its vector and that good generalization must imply insensitivity to changes in the input vectors.

This research was initiated to develop an integrated on-line credit evaluation system which would monitor system performance and enhance prediction accuracy through constant feedbacking customer's credit data. Especially the neural network mechanism was adopted as a credit evaluating processor in this research. Since the neural network could predict the output values by nonlinear mapping through the hidden layer, even though it didn't know the direct relations between the input values and the output values. With the rapid increase of sales volume and credit market in Korea, many business companies have not imposed any restriction on credit card applicants. This is because, different from American companies with hundreds of years of experienced in financial market, Korean Companies pursue the goal of market penetration and market expansion through granting credit cards to any applicant without any scanning efforts.

E LTD. is one of leading companies in the Korea fashion business. A credit card system of this company is adopted to achieve 'Big Share' in fashion market. Recently, the number of card holders of this company has reached 180,000 and every month the number of overdue or delinquent credits reached 3,500 cases. Such delinquent customers inflict a serious loss to the company and thus the company had to devise a measure to solve the financial problem caused by continuously accumulated bad debts. One of the ideas popped up was to develop an automated credit evaluation system which continuously monitors evaluation system's performance and then can enhance the prediction accuracy through learning from customer's credit data.

S LTD is an another leading credit card company which holds a big share in Korea market. The customer's behaviour of E LTD and S LTD is almost the same, in terms of credit standing. Even though the two companies employ different application variables, accordingly different application forms, the contents were almost the same.

The neural network training system, as usual of the back propagation systems, consists of three layers : input layer, hidden layer and output layer. From the customer's records, the eight variables which were believed to have a strong relationship with customer's credit were derived as 'credit factors' : age, sex, marital status, occupation, organization, job position, residential condition, residential area. Selection of input variables for the system's training, that is, selection of critical factors significantly influencing on customer's creditability should be determined by consideration of customer's behaviour, social custom, and statistics. In this sense, the factors included in the current system reflect many features of Korean customers and social practices, and thus factors included in the current system might be much different from factors included in the system developed in other countries. For example, in the study of American loan application, occupation, length of employment, marital status, race and income level are important consideration[Capon, 1982], but work place dose not have a significant impact on credit evaluation. In contrast, work place might be the most important factor in determining an individual's credit status. Also, residential area might be very important factor, which was proved to be unimportant at all.

According to the number of overdue payment, 'credit status' was divided into two status such as 'good' and 'bad'. When the customer's payment is not overdue or the number of overdue payment is less than 3 months, he or she was classified into 'good' credit status. When the number of overdue payment exceeded 3 months, the customer was classified into 'bad' credit status.

In the beginning stage of this research, back propagation algorithm was employed and tested for training sample data. Many experiments with the back propagation algorithm showed that the system with more than 40 training data would not which convergence state within a reasonable time and thus another efficient algorithm should be devised. Later, employed was the quickprop algorithm, an advanced form of back propagation, as a learning mechanism. 'Quickprop' algorithm suggested by Fahlman[1988] is important and well-known for speeding up convergence by jumping out directly the parabolic error space to the minimum point of the parabola. In this algorithm, the error defined as $\partial E/\partial w(t-1)$ is kept and then, for each weight, the weight change measured by the difference between current weight slope and previous weight slope is used for determining a parabola.

As predicted and assured by researchers [Fahlman, 1988], the quickprop algorithm effectively and quickly reduced total error, and thereby enabled the system to reach convergence in a reasonable time limit. As shown in Table-1, ordinary back propagation algorithm required more than 230,000 epochs to reach convergence state in which the degree of generalization was 65%. In contrast, the quickprop algorithm needed only 160 epochs to reach the convergence state in which the degree of generalization was measured around 60%, a slightly lower value than ordinary back propagation algorithm. When the number of training data set increases to 100(50 bad creditors and another 50 good creditors), the back propagation system did not stop running. That is why the research, in back propagation learning system, could not extend testing the generalization capability beyond 80 cases. To the contrary, the quickprop algorithm with 100 training data easily reached the convergence state at the epoch of 267 and showed a little enhanced prediction capability, 67%. The same test results were obtained when the test of classification accuracy were conducted on S LTD data set. As shown in Table-2, the classification accuracy was less than 60%, which is too low for field application.

3. Generalized Learning Vector Quantization (GLVQ)

Vector quantization(VQ) is defined as a technique which "searches for small but representative set of prototypes, which we can then use to match sample patterns with nearest neighbor techniques," [Kong and Kosko, 1992, p.1]. Clustering through VQ is accomplished by partitioning the patterns $x \in R^n$ into k decision classes $\{D_j\} \in R^m$, the prototypes or reference vectors:

$$R^n = \cup D_j, \text{ and } D_i \cap D_j = \varnothing \text{ for } i \neq j.$$
$$x \in D_1 \quad \text{if } d(x, s1) < d(x, s2)$$
$$x \in D_2 \quad \text{if } d(x, s1) > d(x, s2),$$

where $d(x_i, s_i)$ is defined as the distance measure between the pattern x_i and prototype s_j. S1 and s2 are prototypes belonging to the decision classes D_1 and D_2, respectively.

The VQ system attempts to find appropriate decision class (D1, D2, ..., D_k) and centroids ($\overline{S1}$, $\overline{S2}$, .. $\overline{Sk}$), from the patterns (X_1, X_2, ..., X_p). the pattern belongs to. In the view of data compression, the patterns x_i will not completely vary, but rather will be correlated to next patterns. Thus, the essential problem of VQ technique is to find mapping functions from the patterns to hidden variables r_1, r_2, ..., r_m, for M < P, with minimal variances. The variables r_i provide a more economical description of the observed phenomenon. In the linear discriminant functions [Kong and Kosko, 1992; Kosko, 1991], the function behaves as a separating hyperplane in the pattern space R^n, that is, setting up K-dimensional hyperplane lying in the N-dimensional data space. The variables r_i can account for the total data variation. However, if the actual distribution of data points is deviated from the hyperplane, the description resulting from a projection on the principal axes of the distribution will be worse [Ritter, Martinetz, and Schulten, 1992; Cichocki and Unbehauen, 1993]. To overcome this problem, the linear principal axes or hyperplanes are replaced by curved surfaces, which may provide a better description of nonlinear data distributions. This can be interpreted geometrically as "a minimization of the mean-squared perpendicular distance $d(x, \bar{s}_i)^2$ between the data points and the hyperplane"[Ritter, Martinetz, and Schulten, 1993, p.247].

Learning vector quantization (LVQ), suggested by Kohonen, is considered as an approximation procedure for the computation of principal curves, surfaces, or higher-dimensional principal manifolds [Ritter, Martinetz, and Schulten, 1993]. The LVQ system tries to discover cluster substructure hidden in unlabeled N-dimensional data and extract M-dimensional features. The prototypes S={S1, S2, ..., Sk} are a array of unknown cluster centers $s_i \in R^m$ for $1 \leq i \leq k$. In LVQ, learning refers to finding values for the $\{S_{ij}\}$ [Pal, Bezdek, and Tsao, 1993]. When an input vector x_i is submitted to the system, the distance between the input vector and prototypes $d(x, s_i)$ is calculated and then the prototype with the shortest distance becomes a winner. The next step is to update the centroid of the prototype using update rules. The typical LVQ rule of finding the winner node and update is as following:

$$\| x_k - s_{i,t-i} \| = \min_{1 \leq i \leq k} \{ \| x_k - s_{j,t-1} \| \} \text{ for finding}$$

$$s_{i,t} = s_{j,t-1} + \alpha(x_k - s_{j,t-1}) \text{ for updating.}$$

Although the LVQ algorithm has some nice theoretical foundation, it suffers from a serious problem: initialization problem. As the initial position of centroid $s_{i,0}$ have too strong influence on subsequential position updates, especially when they are outside the convex hull of the input data [Kang, Hwang and Yoo, 1994], it may not produce any meaningful clusters[Pal, Bezdek, and Tsao, 1993]. Also, as the winner node only update its position, the result of clustering might be biased by the gravitational force of winners. To overcome these problems, Pal, Bezdek and Tsao [1993] suggested the GLVQ algorithm which updates either all the centroids of prototypes or none, for each new input vector. When there is a perfect match to the winner node, no node is updated.

The updates rule of GLVQ is

$$s_{i,t} = s_{i,t-i} + \alpha(x_k - s_{i,t-i}) \frac{D^2 - D + \| x_k - s_{i,t-i} \|^2}{D^2}$$

$$s_{i,t} = s_{i,t-i} + \alpha(x_k - s_{i,t-i}) \frac{\| x_k - s_{i,t-i} \|^2}{D^2} \quad (r \neq i)$$

where i is the best matching node, $D = \sum_{r=1}^{c} \| x_k - v_r \|^2$, $k=1, 2, \ldots n$; $r=1, 2,$

...k and t is time.

The GLVQ algorithm is very insensitive to initial positions of centroids, and the choice of learning coefficients [Kang, Yoo, and Kang, 1994; Kang, Hwang, and Yoo, 1994].

4. Experiments with GLVQ

In the experiments with GLVQ, two sets of credit evaluation data were employed for performance comparison. In the first experiment, as shown in the figures below, the system showed very low classification accuracy. As the system was developed based on GLVQ algorithm, the system was not sensitive to the modification of learning parameter, alpha and number of iterations. As shown in Figure-1 and Figure-2 of E LTD case, the system partitioned 120 data consisting of 60 'bad' customer and another 60 'good' customer data into two clusters: cluster-1 and cluster-2. The data numbered 0 to 59 should be group-1, while the data numbered 60 to 119 should be in group-2. In other words, the data numbered 0 to 59 and the data numbered 60 to 119 should be not be in the same group to be cohesive. But, the clustering result is that the cluster-1 has 74 units of data including 33 data units from one group and other 41 data units from another group.
This means that clustering the credit data is not so meaningful for real world application, indicating that the data included in clustering does not have any meaningful relationship with each other in the same group.
In other words, neural classification cannot impose any meaningful decision rules on clustered data. The same thing was observed in S LTD case. Figure-3 and Figure-4 showed that inconsistencies between natural clustering of GLVQ algorithm and original credit clustering exist in case of S LTD data set.

This might be attributable to the fact that the training data includes too much noisy data in it, mainly conflicting cases. For example, department head of a business company usually earns better salary than other employees in the department and thus, the head is supposed to be much better in credit standing than others in the department. However, in the review of raw data it was found that the credit standing of employees is not correlated with income level of the employee. Even CEOs of business companies, though CEOs of large group companies are exceptional, were as bad as young undergraduates with less than one year's job experience.

5. Conclusion and Future Research Direction

In this research, GLVQ algorithm was applied to two different credit evaluation data, E LTD and S LTD. In the analysis, it was found that the learning capability of neural classification system strongly depends on data characteristics. That is, when the training data does not truly represent underlying data set, the neural classification system cannot provide highly predictable results. This might be attributable to the fact that the non-linear discriminant functions representing underlying parameters of training data set, even though they accurately classify well-formed data set as found in the previous researches[Surkan and Singleton, 1990; Surkan and Ying, 1991; Kim, 1992], cannot appropriately project th input data from their original N-dimensional space onto the L-dimensional output space performing "a dimensionality reduction which retains most of the intrinsic information in the input data vector" [Cichocki and Unbehauen, 1993, p.339].

As shown above, the extremely low classification accuracy obtained in Quick-Prop tests is interrelated with the extraordinary phenomenon that the groups clustered by GLVQ do not agree with the group classified based on credit standing, at all. In other words, the weight of factors involved in GLVQ clustering is quite different from the weight of factors earned in quick-prop training; GLVQ unsupervised clustering produces much different mapping functions from the Quick-Prop mapping functions. If there exists a strong tendency of correlation between output values and input vectors, GLVQ clustering should have produced output space which is quite consistent with the actual output vectors in data set of E LTD and S LTD.

This research is part of the efforts to develop an automated credit evaluation system which can learn itself and accumulate its knowledge by itself and, eventually provide highly accurate classification data. The automated credit evaluation system will benefit various professional service industries(gas stations, department stores, restaurants, entertainment places, sports complexes, ski resorts), household appliances companies(refrigerators, washers and cleaners, room cleaners, air conditioning system, audio distributors), automobile sales companies, etc.

Reference

Beranek, W. and W. Taylor, "Credit - Scoring Models and the Cut-off Point A Simplification", *Decision Sciences*, vol. 7, 1976 .

Cannon, R.L., J.V. Dave, and J.E. Bezdek, "Efficient Implementation of the Fuzzy c-Means Clustering Algorithms," *IEEE Trans. on Pattern Analysis and Machine Intelligence* (Vol. 8, No.2), March, 1986.

Carter, C., and J. Catlett, "Assessing Credit Card Applications using Machine Learning", *IEEE EXPERT*, pp.71-79, Fall 1987.

Cichocki, A., and R. Unbehauen, *Neural Networks for Optimization and Signal Processing*, Wiley (West Sussez, England), 1993.

Choi, H.Y., Y. Chung, J.W. Kim, and J.U. Choi, "The Bayesian Statistical Properties of Neural Net Clustering," (forthcoming) *한국 경영과학회 추계학술 대회 논문집, 1994.*

Douglas, R. L., E. Collins, C. Schfield, and S. Ghosh, "Risk Assessment of Mortgage Applications with Neural Network Systems: An Update as the Test Portfolio Ages," *Proceedings of IJCNN 1990*, Vol. II, pp. II-479-482, 1990.

Fahlman, S. E., "Faster-Learning Variations on Back-Propagation: An Empirical Study," *in Touietzky D., T. Sejnowski, and G.Hinton (eds.) Proceedings of the '88 Connectionist Models Summer School*, CMU, 1988.

Fu, L., and T. Chen, "Sensitivity Analysis for Input Vector in Multilayer Feedforward Neural Networks," *Proceedings of IJCNN*, pp.215-218, 1993.

Gerasoulis, A., and Y. Tao, "On the Granularity and Clustering of Directed Acyclic Task Graphs," *IEEE Trans. on Parallel and Distributed*

Systems (Vol. 4, No. 6), pp.686-701, June 1993.

Hampshire, J. B., and B.A. Pearlmutter, *Equivalence Proofs for MultiLayer Perceptron Classifiers and the Bayesian Discriminant Function,* Technical Report, Department of Electrical and Computer Engineering, CMU, 1990.

Hancock, E. R., and J. Kittler, "A Bayesian Interpretation for the Hopfield Networks," *Proceedings of ICNN '93,* pp.241-346, 1993.

Hathaway, R.J., and J.C. Bezdek, "Switching Regression Models and Fuzzy Clustering, "*IEEE Trans. on Fuzzy Systems* (Vol.1, No.3), pp.195-204, August 1993.

Hathaway, R.J., and J.C. Bezdek, "Local Convergence of the Fuzzy C-means Algorithms", *Pattern Recognition* (Vol.19, No.6), pp.477-480, 1986.

Hornik, K., M. Stinchcombe, and H. White, "Multilayer Feedforward Networks are Universal Approximators," *Neural Networks*(Vol.2), pp.359-366, 1989.

Heywood, M. and P. Noakes, "Simple Addition to Back Propagation Learning for Dynamic Weight Pruning, Sparse Network Extraction and Faster Learning," *Proceedings of 1993 IJCNN,* pp.620-625, 1993.

Kang, B.H., J.H. Yoo, and S.J. Kang, "Speeding up Generalized Learning Vector Quantization," will be in the *Proceedings of ICONIP(International Conference on Neural Information Processing)(forthcoming),* Seoul, October, 1994.

Kang, B.H., D.S. Hwang, and J.H. Yoo, "Square Error Clustering Scheme and Clustering Networks," in the *Proceedings of Izuka-94 International Conference on Neural Networks, Fuzzy and Soft Computing,* Izuka, Japan, 1994.

Kavuri, S.N., and V.Venkatasubramanian, "Using Fuzzy Clustering with Ellipsoidal Units in Neural Networks for Robust Fault Classification,"*Computers Chemical Engineering* (Vol.17, No.8), pp.765-784, 1993.

Kim, J.W., *A comparative analysis of rule based, neural networkn and statistical classification for the bond rating problem,* Ph.d, Dissertation, Dept. of Information Systems, Virginia Commonwealth

University Richmond, Virginia (April, 1992).

Kim, J.W., J.U. Choi, H.U. Choi, Y.Chung, and B.H. Kang, "Developing a Neural Net-Based Credit Evaluation System with Noisy Data," (forthcoming) *Proceedings of the Third International Conference on Automation, Robotics, and Computer Vision,* Singapore, November 1994.

Kim, J.W., J.U. Choi, H.U. Choi, Y.Chung, and B.H. Kang, "A Neural Net-Based Credit Evaluation System with Noisy Data," (forthcoming) *Proceedings of the International Conference on Neural Information Processing,* Seoul, Oct. 1994.

Kohonen, T., "The Self-Organizing Map," in Lau, C(eds), *IEEE Neural Networks,* IEEE (New York), pp.74-90, 1992.

Kong, S., and B. Kosko, "Differential Competitive Learning for Phoneme Recognition," in Kosk, B. (eds) *Neural Networks for Signal Processing,* Prentice-Hall (New Jersey), pp. 36, 1992.

Kononenko, I., "Feedforward Bayesian Neural Networks and Continuous Attributes," *Proceedings of ICNN '91,* pp.146-151, 1991.

Lam, W., and F. Bacchus, "Learning Bayesian Belief Networks: An Approach Based On the MDL Principle," *Proceedings of The Second Pacific RIM International Conference on AI,* pp.1237-1243, 1992.

Lansner, A., and O. Ekeberg, "A One-Layer Feedback Artificial Neural Network with a Basyesian Learning Rule," *International Journal of Neural Systems*(Vol.1, No.1), pp.77-87, 1989.

Lee, S., and S. Shimoji, "BAYESNET: Bayesian Classification Network Based on Biased Random Competition Using Gaussian Kernets," *Proceedings of ICNN '93,* pp.1354-1359, 1993.

Leinweber, D., "Knowledge-Based Systems for Financial Applications," *IEEE EXPERT,* pp.18-31, Fall 1988.

MacKay, D.J.C., *A Practical Bayesian Framework for Back Propagation Networks,* Technical Report, Department of Computation and Neural Systems at CalTech, May 1991a.

MacKay, D.J.C., *Bayesian Interpolation,* Technical Report, Department of Computation and Neural Systems at CalTech, May 1991b.

Messier, and J. V. Hansen, "Expert Systems in Auditing : The

State of the Art", *Auditing : A Journal of Practice & Theory* (Fall, 1987, Vol.7, No.1), pp.94-105.

Neal, R. M., *Bayesian Mixture Modeling by Monte Carlo Simulation*, Technical Report, CRG-TR-91-2, Department of Computer Science, University of Toronto, June 1991.

Neal, R.M., *Bayesian Training of Backpropagation Networks by the Hybrid Monte Carlo Method*, Technical Report, CRG-TR-91-1, Department of Computer Science, University of Toronto, April, 1992.

Noel, C., "Credit Scoring Systems: A Critical Analysis", *Journal of Marketing*, (Vol. 46), Spring 1982, pp.82-91.

Odom, M.D., and R. Sharda, "A Neural Network Model for Bankruptcy Prediction," *Proceedigns of IJCNN 1990*, Vol.II, pp.II-163-168, 1990.

Pal, N.R., J.C. Bezdek, E.C, Tsao, "Generalized Clustering Networks and Kohonen's Self-Organizing Scheme," *IEEE Trans. on Neural Networks* (Vol.4, No.4), pp.549-557, July 1993.

Pao, Y., *Adaptive Pattern Recognition and Neural Networks*, Addison Wesley Publishing Company, 1989

Pfister, M. and R. Rojas, "Speeding - up Backpropagation- A Comparison of Orthogonal Technique," *Proceedings of 1993 IJCNN*, pp.517-523, 1993.

Ritter, H., T. Martinetz, and K. Schulten, *Neural Computation and Self-Organizing Maps*, Addison-Wesley(Reading, Massachusetts), 1992.

Rose, K., E. Gurewitz, and G.C. Fox, "Constrained Clustering as an Optimization Method," *IEEE Trans. on Pattern Analysis and Machine Intelligence* (Vol.15, No.8), pp.785-794, August 1993.

Schumann, M. and T. Lohrbach, "Comparing Artificial Neural Networks with Other Methods within the Field of Credit Scoring," *Proceedings of 2nd Pacific Rim International Conference on AI*, Vol.I, pp.72-78, 1992.

Simpson, P.K., "Fuzzy Min-Max Neural Networks-Part 2: Clustering," *IEEE Trans. on Fuzzy Systems* (Vol.1, No.1), pp.32-45, February 1993.

Smith, M., *Neural Networks for Statistical Modeling,* Nostrand Reinhold, New York, 1993.

Surkan, A. J., and J. C. Singleton, "Neural Networks for Bond Rating Improved by Multiple Hidden Layers," *Proceedings of IJCNN* 1990, Vol.II, pp.II-157-162, 1990.

Surkan, A. J., and X.Ying, "Bond Rating Formulas Derived Through Simplifying A Trained Neural Network," *Proceedings of IJCNN 1991,* Vol.II, pp.1566-1570, 1991.

Tam, K.Y., "Applying Conceptual Clustering to Knowledge-bases Construction, " *Decision Support Systems* (Vol.10), pp.173-198. 1993.

Tyagi, A., and M.A. Bayoumi, "Defect Clustering Viewed Through Generalized Poison Distribution," *IEEE Trans. on SemiConductor Manufacturing* (Vol.5, No.3), pp.196-206, August 1992.

Wan, E. A., "Neural Networks Classification: A Bayesian Interpretation," *IEEE Transactions on Neural Networks* (Vol.1, No.4), pp.303-305, Dec. 1990.

Wann, M., T. Hediger, and N. N. Greenbaun, "The Influence of Training Sets on Generalization in Feed-Forward Neural Networks," *Proceedings of 1990 IJCNN,* pp.III-137-142, 1990.

Weigend, A. S., and N.A. Gershenfeld (eds.), *Time Series Prediction: Forecasting the Future and Understanding the Past Eds,* Addison Wesley, 1994.

Whitley, D., and N. Karunanithi, "Generalization in Feed Forward Neural Networks," *Proceedings of IJCNN,* July 8-12, Seattle, Washington, pp. II-77 - II-82, 1991.

Yu, Y. H., and R.F. Simmons, "Descending Epsilon in Back-Propaga tion: A Technique for Better Generalization," *Proceedings of IJCNN 199 0,* June 1990, San Diego, pp.III-167-III-172, 1990.

김정원, 최종욱, 정윤, "Generalization in Developing a Neural Network System for Issuing Credit-Cards," MIS연구(Vol.6), pp.31-47, 1994.

김정원, 최종욱, 최홍윤, 정윤, "Developing a Neural-Based Credit Eval uation System with Noisy Data," 정보처리논문지(Vol.1) (Forthcoming), 1994.

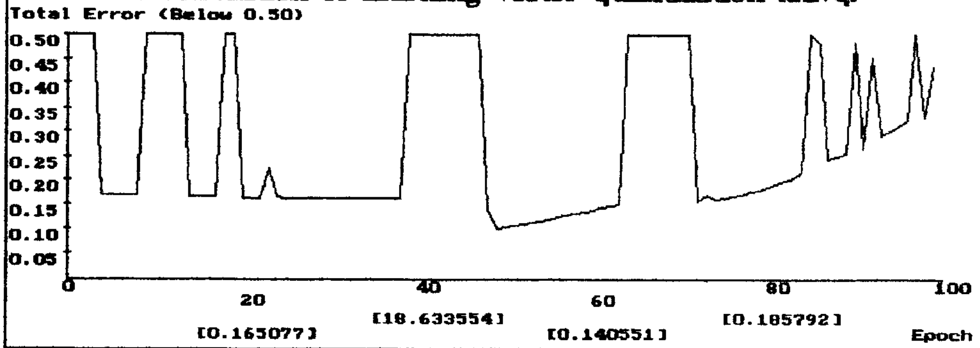
Data Set	Back propagation		Quickprop	
	# of Epoch	Degree of generalization	# of Epoch	Degree of generalization
20 / 20	22,644	19/40(47.5%)	26	23/40(42.5%)
30 / 30	42,212	22/40(55%)	49	24/40(60%)
40 / 40	238,802	26/40(65%)	168	24/40(60%)
50 / 50			267	27/40(67.5%)
60 / 60			338	23/40(57.5%)
70 / 70			467	23/40(57.5%)
80 / 80			488	28/40(70%)

Table-1 Classification Accuracy on E LTD. data (BackPropagation and Quick-Prop)

# of Epoch	Training Data	Test Data	Degree of Generalization
100	60/60	50/50	43%
195	100/600	50/50	46%

Table-2. Classification Accuracy on S LTD. data(Quick-Prop)

Generalization of Learning Vector Quantization [GLVQ]



Input filename : c60.net	Output filename : c60-8.out

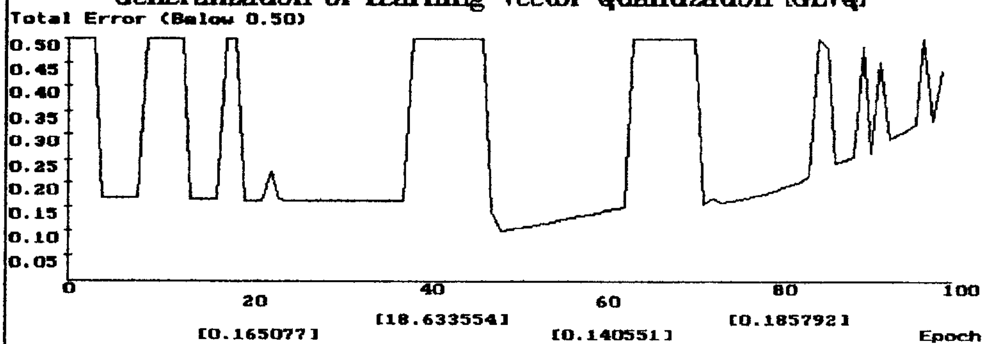
Tolerant Error : 0.01000	Alpha-0 : 0.800000	No. of Clusters : 2
Max. Epoch : 100	Cur. Epoch : 100	Total Error : 0.228735
Pat.: 120 Attr.: 42	Cur. Alpha : 0.000000	Delta Error : -0.203918

Final Centers of Each Cluster

Cluster: 0		0.33743	0.57279	0.05689	0.03369	0.00006	0.00000	0.10006	0.89994
0.09368	0.30591	0.24255	0.02896	0.18810	0.09495	0.04824	0.05511	0.33011	0.06932
0.18120	0.00000	0.01752	0.34851	0.02124	0.00000	0.01705	0.01575	0.94598	0.03200
0.85539	0.02234	0.78042	0.01338	0.15880	0.00003	0.02543	0.03137	0.41581	0.00967
0.23742	0.01050	0.27657	0.02017						
Cluster: 1		0.05866	0.25797	0.52023	0.11971	0.04604	0.00000	0.90804	0.09196
0.36529	0.14372	0.25519	0.00018	0.19692	0.00054	0.04196	0.06056	0.32612	0.10187
0.23534	0.00000	0.02668	0.25284	0.07241	0.00000	0.09102	0.06550	0.77148	0.42977
0.51291	0.23289	0.37164	0.00005	0.33133	0.02826	0.03805	0.05666	0.50804	0.07727
0.08820	0.12511	0.13715	0.02567						

press any key to see the final patterns of each cluster

Generalization of Learning Vector Quantization [GLVQ]



Input filename : c60.net	Output filename : c60-8.out

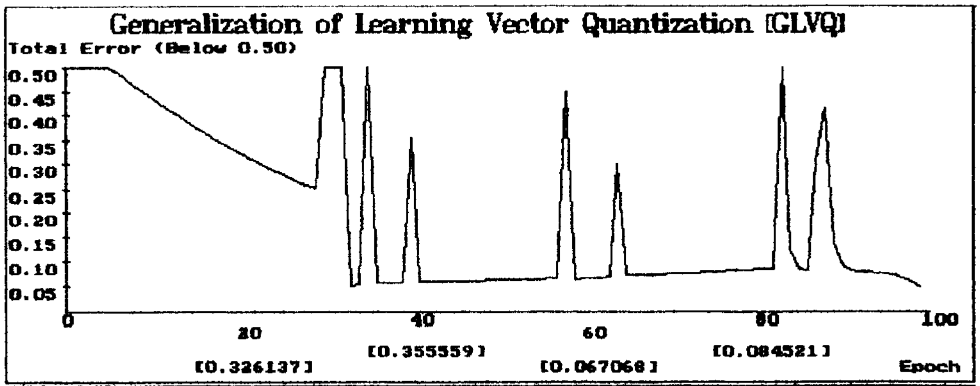
Tolerant Error : 0.01000	Alpha-0 : 0.800000	No. of Clusters : 2
Max. Epoch : 100	Cur. Epoch : 100	Total Error : 0.228735
Pat.: 120 Attr.: 42	Cur. Alpha : 0.000000	Delta Error : -0.203918

Final Patterns of Each Cluster

Cluster: 0		0	1	7	8	9	10	12	16
17	20	23	24	29	31	32	33	35	36
37	40	42	44	46	49	50	51	52	53
54	55	56	58	59	60	61	62	63	66
67	70	72	74	76	78	80	81	83	85
89	92	93	94	95	96	97	98	99	100
101	103	104	105	106	107	109	110	111	112
113	114	115	116	118	119				
Cluster: 1		2	3	4	5	6	11	13	14
15	18	19	21	22	25	26	27	28	30
34	38	39	41	43	45	47	48	57	64
65	68	69	71	73	75	77	79	82	84
86	87	88	90	91	102	108	117		

press any key to return

Figure-1 Clustered Data of E LTD with α=0.8

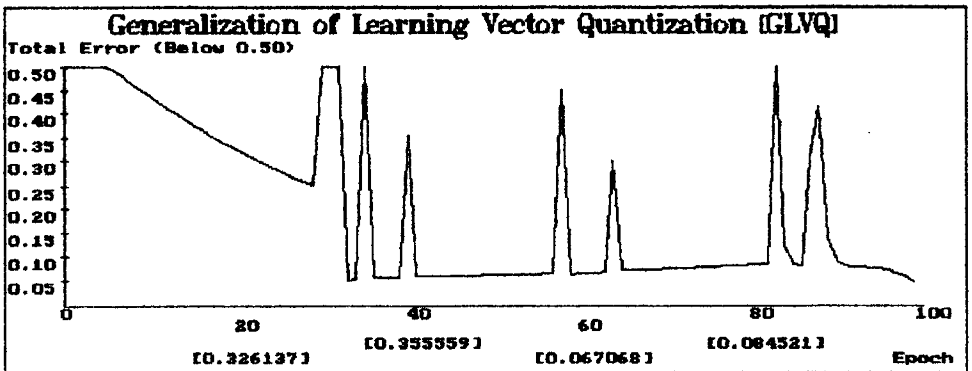


Input filename : c60.net		Output filename : c60-2.out
Tolerant Error : 0.01000	Alpha-0 : 0.200000	No. of Clusters : 2
Max. Epoch : 100	Cur. Epoch : 100	Total Error : 0.030444
Pat.: 120 / Attr.: 42	Cur. Alpha : 0.000000	Delta Error : -0.018258

Final Centers of Each Cluster

Cluster: 0		0.31882	0.59003	0.05501	0.03696	0.00006	0.00000	0.11887	0.88113
0.09280	0.29808	0.26560	0.02778	0.18131	0.09352	0.04325	0.04700	0.33563	0.06741
0.19030	0.00000	0.01529	0.34609	0.01684	0.00000	0.01524	0.01461	0.95333	0.03624
0.86094	0.02463	0.76445	0.01344	0.16560	0.00003	0.03227	0.04287	0.40439	0.01150
0.23699	0.01202	0.27743	0.01637						
Cluster: 1		0.05261	0.22622	0.55889	0.11819	0.04665	0.00000	0.90569	0.09431
0.38903	0.14140	0.24229	0.00016	0.18600	0.00052	0.04430	0.06531	0.32221	0.09697
0.24511	0.00000	0.02485	0.24896	0.07143	0.00000	0.09209	0.06707	0.76983	0.45715
0.49167	0.23222	0.33976	0.00005	0.36231	0.02562	0.04238	0.02656	0.51961	0.08518
0.09004	0.14324	0.13246	0.02476						

press any key to see the final patterns of each cluster



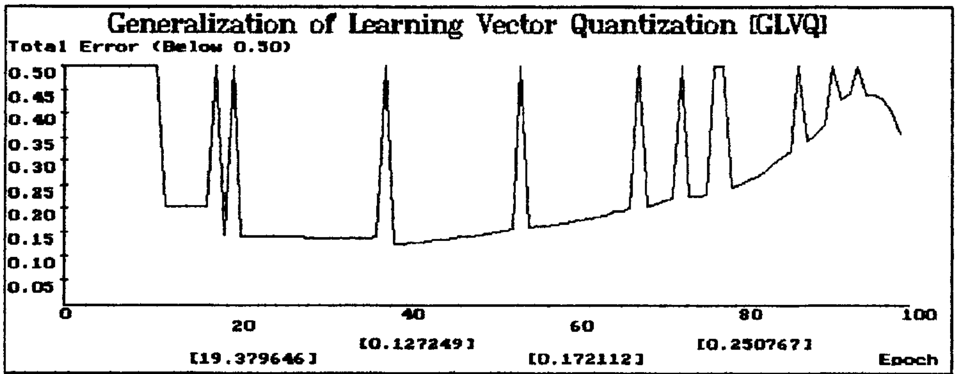
Input filename : c60.net		Output filename : c60-2.out
Tolerant Error : 0.01000	Alpha-0 : 0.200000	No. of Clusters : 2
Max. Epoch : 100	Cur. Epoch : 100	Total Error : 0.030444
Pat.: 120 / Attr.: 42	Cur. Alpha : 0.000000	Delta Error : -0.018258

Final Patterns of Each Cluster

Cluster: 0		0	1	7	8	9	10	12	16
17	20	23	24	29	31	32	33	35	36
37	40	42	44	46	49	50	51	52	53
54	55	56	58	59	60	61	62	63	66
67	70	72	74	76	78	80	81	82	83
85	89	92	93	94	95	96	97	98	99
100	101	103	104	105	106	107	109	110	111
112	113	114	115	116	118	119			
Cluster: 1		2	3	4	5	6	11	13	14
15	18	19	21	22	25	26	27	28	30
34	38	39	41	43	45	47	48	57	64
65	68	69	71	73	75	77	79	84	86
87	88	90	91	102	108	117			

press any key to return

Figure-2 Clustered Data of E LTD with α=0.2

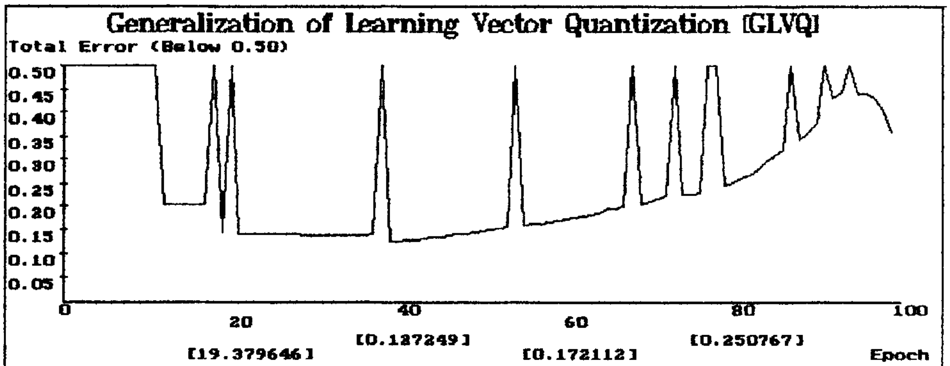


Input filename : cw-100.clu	Output filename : cw-100-8.out	
Tolerant Error : 0.01000	Alpha-0 : 0.800000	No. of Clusters : 2
Max. Epoch : 100	Cur. Epoch : 100	Total Error : 0.285522
Pat.: 120 / Attr.: 62	Cur. Alpha : 0.000000	Delta Error : -0.073124

Final Centers of Each Cluster

Cluster: 0		0.00000	0.00125	0.44732	0.30922	0.20252	0.04110	0.63023	0.36977
0.97685	0.02315	0.08306	0.91694	0.39825	0.60175	0.32770	0.18740	0.32186	0.00007
0.00003	0.10028	0.20428	0.31876	0.28199	0.05923	0.07612	0.02930	0.00000	0.04526
0.00000	0.00000	0.01938	0.00004	0.04223	0.00006	0.02239	0.00007	0.03707	0.02464
0.00595	0.14689	0.00000	0.00000	0.01360	0.00000	0.00003	0.61420	0.36837	0.19354
0.19221	0.09067	0.00000	0.00004	0.15687	0.03372	0.55764	0.10392	0.29971	0.02029
0.24932	0.21923	0.17771	0.26948						
Cluster: 1		0.00000	0.58385	0.36528	0.04522	0.00607	0.00005	0.76336	0.23664
0.07291	0.92709	0.00008	0.99992	0.15933	0.84067	0.04845	0.52139	0.24181	0.03397
0.01579	0.13940	0.22338	0.34014	0.19409	0.06883	0.15896	0.02097	0.00000	0.00005
0.00000	0.00000	0.03165	0.01803	0.00005	0.03104	0.01631	0.03685	0.01348	0.19056
0.03818	0.42377	0.00000	0.00000	0.00001	0.00000	0.01234	0.16803	0.00043	0.36813
0.20697	0.21509	0.00000	0.02169	0.18892	0.01632	0.03068	0.09377	0.79419	0.01389
0.68583	0.25755	0.02420	0.01886						

press any key to see the final patterns of each cluster



Input filename : cw-100.clu	Output filename : cw-100-8.out	
Tolerant Error : 0.01000	Alpha-0 : 0.800000	No. of Clusters : 2
Max. Epoch : 100	Cur. Epoch : 100	Total Error : 0.285522
Pat.: 120 / Attr.: 62	Cur. Alpha : 0.000000	Delta Error : -0.073124

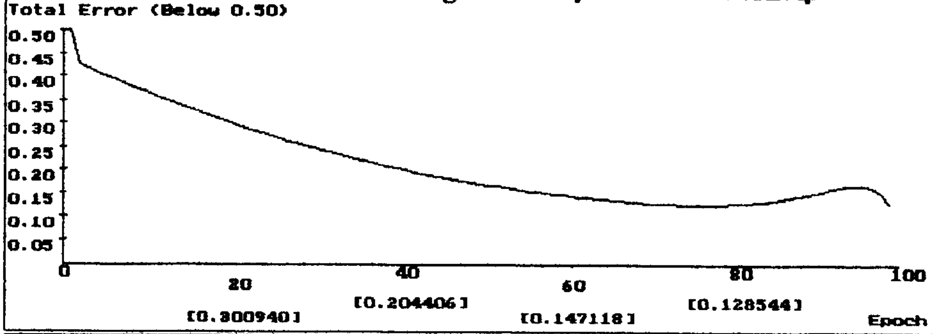
Final Patterns of Each Cluster

Cluster: 0		0	1	2	3	4	5	6	7
8	9	10	12	13	14	15	16	18	19
20	21	22	23	24	26	27	28	31	32
33	34	35	37	38	39	40	41	42	43
45	46	47	48	49	51	52	55	57	62
65	68	72	75	79	85	88	92	93	97
99	103	105	117						
Cluster: 1		11	17	25	29	30	36	44	50
53	54	56	58	59	60	61	63	64	66
67	69	70	71	73	74	76	77	78	80
81	82	83	84	86	87	89	90	91	94
95	96	98	100	101	102	104	106	107	108
109	110	111	112	113	114	115	116	118	119

press any key to return

Figure-3 Clustered Data of S LTD with α=0.8

Generalization of Learning Vector Quantization [GLVQ]



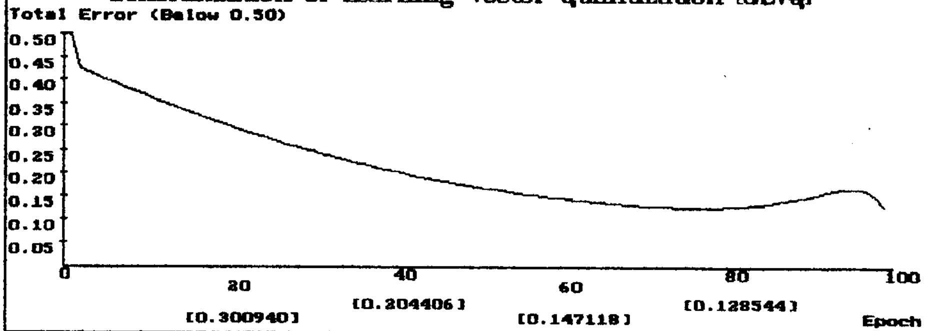
Input filename : cw-100.clu		Output filename : cw-100-2.out	
Tolerant Error : 0.01000		Alpha-0 : 0.200000	No. of Clusters : 2
Max. Epoch : 100		Cur. Epoch : 100	Total Error : 0.087333
Pat.: 120	Attr.: 62	Cur. Alpha : 0.000000	Delta Error : -0.042740

Final Centers of Each Cluster

Cluster: 0		0.00000	0.20145	0.24533	0.58898	0.08364	0.01963	0.80232	0.19768
0.29269	0.70731	0.53939	0.46060	0.17664	0.38633	0.58657	0.65004	0.62786	0.01275
0.53295	0.56857	0.60432	0.21052	0.16058	0.04632	0.57290	0.53906	0.53040	0.01278
0.53040	0.00000	0.53690	0.00908	0.01146	0.53606	0.01282	0.53545	0.53860	0.56600
0.02295	0.19482	0.00000	0.00000	0.58281	0.00000	0.00677	0.24879	0.59274	0.62708
0.59530	0.11195	0.53040	0.53398	0.11775	0.53755	0.19664	0.06617	0.32011	0.01274
0.70862	0.15598	0.07993	0.10055						
Cluster: 1		0.00000	0.80919	0.40475	0.16716	0.09883	0.02035	0.70245	0.29655
0.48798	0.51202	0.03534	0.96466	0.26871	0.73129	0.17412	0.36181	0.28735	0.01867
0.00901	0.11991	0.21436	0.33023	0.23394	0.06055	0.12345	0.02548	0.00000	0.01883
0.00000	0.00000	0.02573	0.00958	0.01861	0.01745	0.01853	0.01947	0.02471	0.11223
0.02518	0.29197	0.00000	0.00000	0.00672	0.00000	0.00795	0.37791	0.17158	0.28439
0.20311	0.15865	0.00000	0.01054	0.17215	0.02395	0.27448	0.09814	0.56677	0.01676
0.48487	0.23996	0.09658	0.13232						

press any key to see the final patterns of each cluster

Generalization of Learning Vector Quantization [GLVQ]



Input filename : cw-100.clu		Output filename : cw-100-2.out	
Tolerant Error : 0.01000		Alpha-0 : 0.200000	No. of Clusters : 2
Max. Epoch : 100		Cur. Epoch : 100	Total Error : 0.087333
Pat.: 120	Attr.: 62	Cur. Alpha : 0.000000	Delta Error : -0.042740

Final Patterns of Each Cluster

Cluster: 0									
Cluster: 1		0	1	2	3	4	5	6	7
8	9	10	11	12	13	14	15	16	17
18	19	20	21	22	23	24	25	26	27
28	29	30	31	32	33	34	35	36	37
38	39	40	41	42	43	44	45	46	47
48	49	50	51	52	53	54	55	56	57
58	59	60	61	62	63	64	65	66	67
68	69	70	71	72	73	74	75	76	77
78	79	80	81	82	83	84	85	86	87
88	89	90	91	92	93	94	95	96	97
98	99	100	101	102	103	104	105	106	107
108	109	110	111	112	113	114	115	116	117

press any key to next

Figure-4 Clustered Data of S LTD with α=0.2