A Comparative Study of Three Neural Approaches in Class Prediction of Cancer

Chenn-Jung Huang*

Department of Computer Science and Information Education, National Taitung Teachers College, Taitung, Taiwan 95004

Wei-Chen Liao

Department of Nature Science Education, National Taitung Teachers College, Taitung, Taiwan 95004

Abstract

Accurate diagnosis and classification is the key issue for the optimal treatment of cancer patients. Several studies demonstrate that cancer classification can be estimated with high accuracy, sensitivity and specificity from microarray-based gene expression profiling using artificial neural networks (ANN). In this paper, a comprehensive study was undertaken to investigate the potential value of other neural networks for the discrimination of acute lymphoblastic leukemia (ALL) and acute myeloid leukemia (AML). Probabilistic neural networks (PNN), multilayer perceptrons (MLP) and the learning vector quantization network (LVQ) were applied for this purpose. The best results were obtained by PNN, followed by MLP networks and LVQ. PNN classifier yields 100% recognition accuracy and is well suited for the AAL/AML classification in cancer treatment. This study presents the capabilities of PNN, and also indicates that PNN should be evaluated in a larger prospective study. Our future work will focus on applying the gene selection method and the PNN network on other dataset to observe the generality of this strategy.

Keywords: Probabilistic neural network, multilayer perceptron, learning vector quantization, feature extraction, gene expression data, class prediction, acute leukemia.

* Contact author. Email: cjh@cc.ntttc.edu.tw

1. Introduction

Successful cancer treatment depends on choosing the right regimen for a given patient. For treatment strategies work differently for different tumors, how to accurately diagnose cancer subtypes becomes one of the biggest challenges in clinical cancer research. A recent study reported by Goulb *et al.* [1], the first microarray-based and bioinformatic-orientated approach for identifying and classifying tumor types, moves cancer diagnosis away from traditional visually based systems to molecular based systems.

The cancer model choused by Golub *et al.* [1] is acute leukemias. According to enzyme-based histochemical analyses, acute leukemias can be classified into those arising from lymphoid precursors (acute lymphoblastic leukemia, ALL) or from myeloid precursors (acute myeloid leukemia, AML). Although several clinical diagnoses have been developed, leukemia classification remains imperfect and errors do occur. Because the acute leukemias are well understood and can generally be predicted correctly, they are a good test case for class prediction methods [1, 2].

Golub *et al.* [1] employed a correlation metric to extract a small set of genes and developed a scheme named weighted voting to distinguish AAL from AML; the recognition rate they obtained was 94.1%. By using the same database, several algorithms have been proposed to deal with class prediction of acute leukemias to improve classification accuracy in the literature [2-10]. Toure *et al.* [3] used the multilayer perceptron network (MLP) to predict the class of cancer and gave 58% accuracy on test data. Ryu *et al.* [4] experimented with the MLP, support vector machine (SVM), and k-nearest neighbor (KNN) as the classifiers, and the best classification rate they achieved was 97.1% if gene is selected via Pearson's correlation analysis and the MLP is used as the classifier. Su *et al.* [5] employed the modular neural networks to classify two types of acute leukemias and the best 75% correct classification was reached. Xu *et al.* [6] adopted the ellipsoid ARTMAP to analyze the AML/AAL data set and the best result was 97.1%.

Although most of the algorithms mentioned above can reach high prediction rate, any misclassification of the disease is still intolerable in acute leukemias treatment. Therefore, the demand of a reliable classifier which gives 100% accuracy in predicting the type of cancer therewith becomes urgent.

To address these challenges, we extract a set of informative patterns from 7129 gene expression data and train three kinds of neural networks [11-16], the MLP, the learning vector quantization network (LVQ) and the probabilistic neural networks (PNN) in turn with 38 leukemia samples, then the classifiers are tested with another 34 samples to inspect the accuracy rate. The experiment results show that PNN can predict the class of cancer correctly when the set of training and test patterns are composed of 50 informative genes.

The remainder of the paper is organized as follows. The feature extraction method for choosing effective predictive genes in our work is introduced in Section 2. Then Sections 3, 4, and 5 give a brief introduction for three types of neural networks for class prediction of cancer, respectively. Section 6 compares the simulation results of three classifiers. Conclusions are made in Section 7.

2. Gene Selection

We used the dataset collected in [1-2] for training and testing of our classifiers. The dataset consists of 72 leukemia samples, and each sample contains 7129 gene expression numbers. As a result, extracting informative genes from of the dataset before classification is essential since the data set is highly dimensional and many genes in the data set are irrelevant to distinction of the cancer class.

Slonim *et al*. [2] tested with several gene selection methods and reported that the best performance was obtained with the relative class separation metric defined by:

$$P = \frac{\mu_1 - \mu_2}{\sigma_1 + \sigma_2}, \tag{1}$$

where μ_1 denotes the mean expression level and σ_1 represents the standard deviation of expression for the samples in class 1, respectively. μ_2 and σ_2 are defined similarly for the samples in class 2. Apparently Eq. (1) tries to pick up the genes with the feature of wider class separation and the smaller spread around class means. We also adopted this gene selection method in our work to find out the set of the most informative genes for training and testing of our classifiers.

3. Multilayer Perceptrons

Although the MLP was already experimented as the classifier for the type of cancer in [3-4], we still hope to know what the outcome is when the MLP is trained with the data selected by the method introduced in the preceding section. Fig. 1 shows the architectural graph of a multilayer perceptron built with one hidden layer. Note that the number of the hidden layers can surpass one if necessary. The incoming signals are fed into the input layer, and are then propagated from left to right. The net input of the hidden layer, $n_j^{(H)}$, is computed as the weighted sum of the incoming signals:

$$n_j^{(H)} = \sum_i w_{ij}^{(H)} \cdot x_i\ , \tag{2}$$

where $w_{ij}^{(H)}$ denotes the weight associated with the link connecting node i in the input layer and node j in the hidden layer, and x_i is the input signal handed over from node i in the input layer.

The output of the node j in the hidden layer can be calculated as

$$H_j = f\left(n_j^{(H)}\right) = f\left(\sum_i w_{ij}^{(H)} \cdot x_i\right), \tag{3}$$

where $f(\cdot)$ is a differentiable nonlinear activation function, such as a logistic function

$$f(X) = \frac{1}{1+e^{-X}}. \tag{4}$$

Fig. 2 shows a logistic function, which can squash the inputs to the range of 0 and 1, and the first derivative of the logistic function can be shown as

$$f'(x) = \frac{e^{-x}}{\left(1+e^{-x}\right)^2} = \frac{1}{1+e^{-x}}\left(1-\frac{1}{1+e^{-x}}\right) = f(x)(1-f(x)). \tag{5}$$

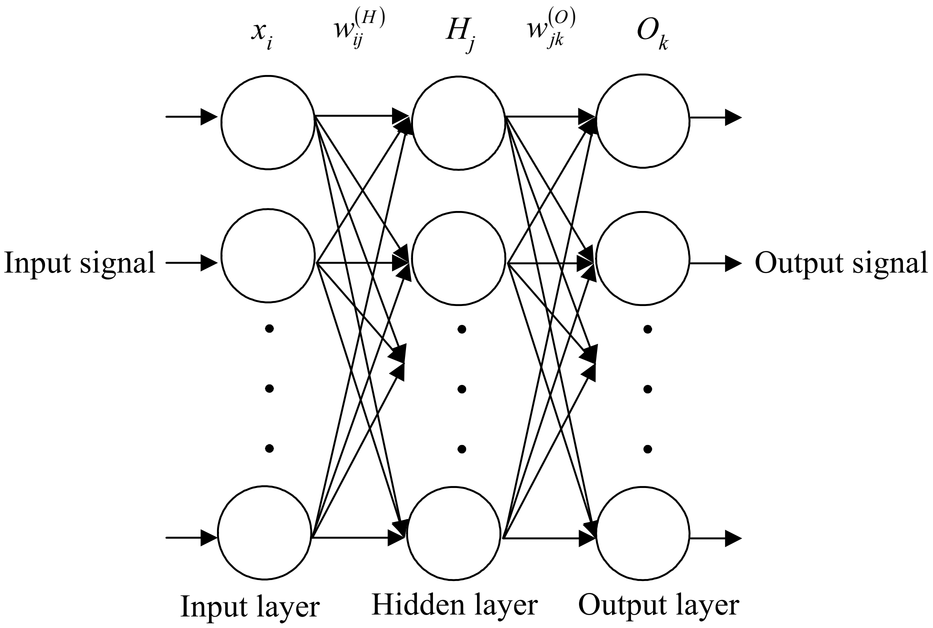

Fig. 1. Architecture of a multilayer perceptron with one hidden layer.

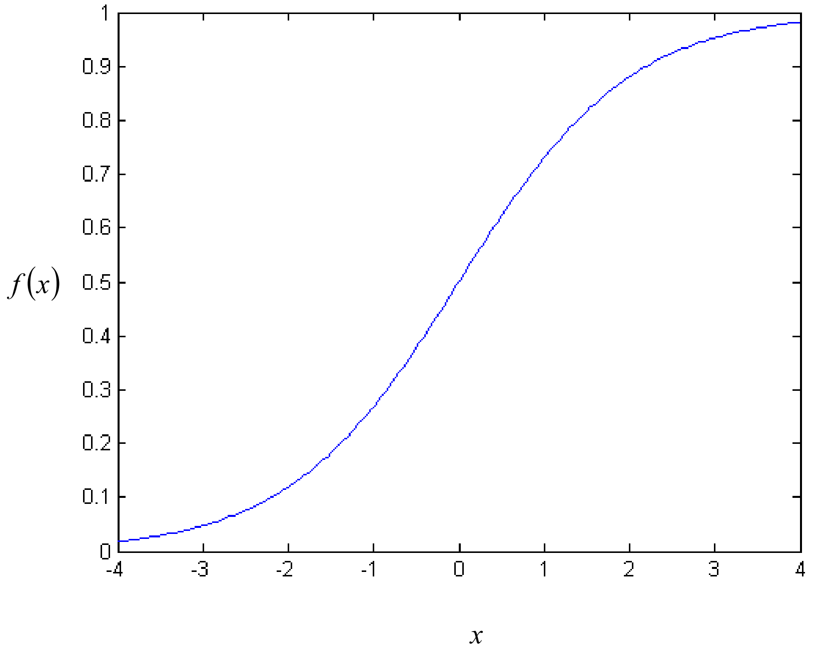

Fig. 2. A logistic activation function.

The net input of the output layer is then computed as the weighted sum of the output of the hidden layer:

$$n_k^{(O)} = \sum_j w_{jk}^{(O)} \cdot H_j = \sum_j w_{jk}^{(O)} \cdot f\left(\sum_i w_{ij}^{(H)} \cdot x_i\right), \tag{6}$$

where $w_{ij}^{(O)}$ represents the weight associated with the link connecting node j in the hidden layer and node k in the output layer.

After applying the activation function to the net input of the output layer, the output of the node k in the output layer becomes

$$O_k = f\left(n_k^{(O)}\right) = f\left(\sum_i w_{jk}^{(O)} \cdot H_j\right) = f\left(\sum_j w_{jk}^{(O)} \cdot f\left(\sum_i w_{ij}^{(H)} \cdot x_i\right)\right). \tag{7}$$

This completes a feedforward process of the multilayer perceptron. Next we shall describe the operation of the back-propagation learning algorithm, which alters the behavior of each node through adjusting the weights used for the forward propagation process and in turn modifies the output of the multilayer perceptron.

3.1 Backpropagation learning rule

The error signal at the output of node *k* in the output layer is defined as

$$\varepsilon_k = T_k - O_k\,, \tag{8}$$

where T_k denotes the expected output of node *k*.

The squared error measure is obtained by summing ε_k over all *k*:

$$E = \sum_k \varepsilon_k = \sum_k \left(T_k - O_k\right)^2 . \tag{9}$$

Then based on the gradient-descent method [11], the correction of the synaptic weight $w_{jk}^{(O)}$ in the hidden-to-output connections is updated by

$$\Delta w_{jk}^{(O)} = -\eta \frac{\partial E}{\partial w_{jk}^{(O)}} = -\eta \frac{\partial E}{\partial O_k} \frac{\partial O_k}{\partial w_{jk}^{(O)}} = -\eta \frac{\partial E}{\partial O_k} f'\left(n_k^{(O)}\right) H_j\,, \tag{10}$$

where η denotes the learning-rate parameter of the backpropagation algorithm.

For the weight update in the input-to-hidden connections, the chain rule is used along with the gradient-descent method as shown by

$$\Delta w_{ij}^{(H)} = -\eta \frac{\partial E}{\partial w_{ij}^{(H)}} = -\eta \frac{\partial E}{\partial H_j} \frac{\partial H_j}{\partial w_{ij}^{(H)}} = -\eta \sum_k \frac{\partial E}{\partial O_k} \frac{\partial O_k}{\partial H_j} \frac{\partial H_j}{\partial w_{ij}^{(H)}}$$

$$= -\eta \sum_k \left(\frac{\partial E}{\partial O_k} f'\left(n_k^{(O)}\right) w_{jk}^{(O)} \right) f'\left(n_j^{(H)}\right) x_i\,. \tag{11}$$

With a choice of the logistic function as the activation function, we may rewrite Eqs. (10) and (11) by using Eq. (5) as follows:

$$\Delta w_{jk}^{(O)} = -\eta \frac{\partial E}{\partial O_k} O_k \left(1 - O_k\right) H_j\,, \tag{12}$$

$$\Delta w_{ij}^{(H)} = -\eta \sum_k \left(\frac{\partial E}{\partial O_k} O_k (1 - O_k) w_{jk}^{(O)} \right) H_j (1 - H_j) x_i \ . \tag{13}$$

The supervised training cycles will proceed until some stopping criterion is met, such as the squared error measure is smaller than a predefined threshold, or the computation epochs reaches some maximum limit.

4. Learning Vector Quantization

The second type of neural network we investigate in this work is the learning vector quantization network (LVQ) which also consists of three layers as shown in Fig.1 [14-15]. In the LVQ network, the net input of the hidden layer, $n_j^{(H)}$, is computed as the Euclidean distance between the input vector and its corresponding weight vector:

$$n_j^{(H)} = \sqrt{\sum_i \left(w_{ij}^{(H)} - x_i \right)^2} \ , \tag{14}$$

and the output of the node in the hidden layer will be 1 if the Euclidean distance computed by Eq. (2) is shortest, and the other nodes will output 0.

Similar to Eq. (6), the net input of the output layer is computed as the weighted sum of the hidden layer output:

$$n_k^{(O)} = \sum_j w_{jk}^{(O)} \cdot H_j \ . \tag{15}$$

Note that $w_{jk}^{(O)}$ can be expressed as

$$w_{jk}^{(O)} = \begin{cases} 1 & \text{if } (k-1) \cdot m \le j \le k \cdot m \\ 0 & \text{else} \end{cases} , \tag{16}$$

where *m* denotes the ratio of the node count in the hidden layer to that in the output layer.

The output of the node *k* in the output layer is

$$O_k = \begin{cases} 1 & \text{if } n_k^{(O)} > 0 \\ 0 & \text{else} \end{cases}. \tag{17}$$

The correction of the synaptic weight $w_{ij}^{(H)}$ in the input-to-hidden connections is updated by

$$\Delta w_{ij}^{(H)} = \alpha\left(x_i - w_{ij}\right), \tag{18}$$

where α is positive if the input pattern is classified correctly by Eq. (17), and negative otherwise. The supervised training cycles will proceed until some stopping criterion is met.

The classification of input patterns is launched by

$$n_j^{(H)} = \sqrt{\sum_i \left(w_{ij}^{(H)} - x_i\right)^2}\ . \tag{19}$$

The node with the minimal Euclidean distance will output 1, and the other nodes will output 0. Then the inferred output is 1 if the value of $n_k^{(O)} = \sum_j w_{jk}^{(O)} \cdot H_j$ is positive, or else the output is 0.

5. Probabilistic Neural Network

The final network we will look into is the probabilistic neural network (PNN) [16-19]. The inspiration of using this type of neural network comes from the gene selection method used before training stage in this work. As pointed out in Section 2, the gene selection strategy given in Eq. (1) attempts to pick up the gene expressions with the characteristic of evident class distinction and better correlation, while the PNN models the Bayesian classifier [20-21] and tries to minimize the expected risk of classifying patterns in the wrong class. Thus, the PNN will perform well if the training and test data hold the feature mentioned above.

Fig. 3 illustrates the architecture of a PNN network. Note that the number of the nodes in the pattern units as shown in Fig. 3 is identical to the counts of the training samples, and the synaptic weight $w_{ij}^{(P)}$ in the input-to-pattern connections is

$$w_{ij}^{(P)} = x_i^{(j)}, \tag{20}$$

where $x_i^{(j)}$ denotes the *i*th node input of the *j*th sample at the input layer.

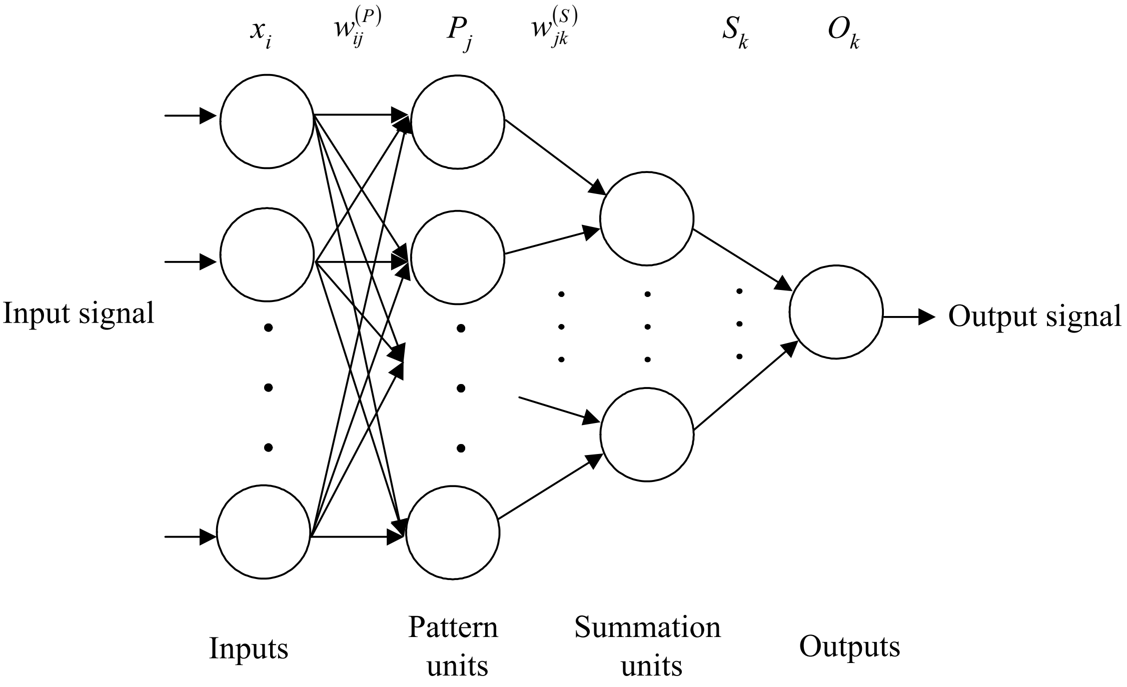

Fig. 3. Architecture of a probabilistic neural network.

As for the weight between the pattern and summation units $w_{jk}^{(S)}$ can be expressed as

$$w_{jk}^{(S)} = \begin{cases} 1 & \text{if } T_k^{(j)} = 1 \\ 0 & \text{else} \end{cases}, \tag{21}$$

where the value of $T_k^{(j)}$ is 1 only sample *j* is associated with class *k*, and 0's elsewhere.

After the instantaneous training process as shown in Eqs. (20) and (21), the classification of input patterns can be initiated by computing the net input to the pattern units as follows:

$$n_j^{(P)} = \sqrt{\sum_i \left(w_{ij}^{(P)} - x_i\right)^2} \ . \tag{22}$$

Then the output of the pattern units is computed as

$$P_j = \exp\left(-\frac{n_j^{(P)}}{2\sigma^2}\right), \tag{23}$$

where σ is a smoothing parameter corresponding to the standard deviation of the Gaussian distribution. Note that if the input is close to one or several training vectors of a single class, it is represented by one or several outputs at the pattern units that are close to 1.

At the summation units, each node represents an individual class. The output of each node can be expressed as

$$S_k = \frac{1}{\sum_j w_{jk}^{(S)}} \sum_j w_{jk}^{(S)} \cdot P_j \ . \tag{24}$$

Then the output layer classifies the input vector into a specific one of *k* classes if that class had the maximum output value from the corresponding node at the summation units.

6. Simulation Results

We employ the gene selection method described in Section 2 to choose 50 informative genes for the training and test data, respectively, and apply three kinds of neural networks mentioned above on these data in turn. Within the 72 leukemia samples, 38 samples are used for training, and the other 34 are for test of the classification. Table 1 shows the comparison of prediction rate for different classifiers. The test results show that the PNN network, which attains 100% prediction accuracy in both training and test data, indeed achieves the best performance as expected.

Table 1 Comparison of best prediction rate for different classifiers

Type of Classifier	Training Accuracy (%)	Test Accuracy (%)
Weighted Voting [1-2]	97.1	94.1
MLP [3]	100	58
MLP [4]	100	97.1
SVM [4]	100	97.1
KNN [4]	100	94.1
Multi-domain Gating Network [5]	100	75
Ellipsoid ARTMAP [6]	100	97.1
Our MLP	100	94.1
LVQ	100	94.1
PNN	100	100

7. Conclusion

In order to predict the class of leukemia cancer, we have demonstrated the usefulness of three neural networks using an informative genes extraction method based on their correlation with the class distinction. Experimental results show that the PNN network is most effective in classifying the type of leukemia cancer. It yields 100% recognition accuracy and is well suited for the AAL/AML classification in cancer treatment. The happening of the precise prediction is mainly contributed by the characteristic of manifest class distinction and the smaller spread around class means possessed by the test data. Our future work will focus on applying the gene selection method and the PNN network on other datasets to observe the generality of this strategy.

REFERENCES

[1] T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, C. D. Bloomfield, and E. S. Lander, "Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring," *Science*, vol. 286, pp. 531–537, 1999.

[2] D. K. Slonim, P. Tamayo, J. P. Mesirov, T. R. Golub, and E. S. Lander, "Class prediction and discovery using gene expression data," *Proceedings of the Fourth Annual International Conference on Computational Molecular Biology*, pp. 263–272, 2000.

[3] A. Toure, and M. Basu, "Application of Neural Network to Gene Expression Data for Cancer Classification," *Proceedings of the 2001 International Joint Conference on Neural Networks*, vol. 1, pp. 583–587, 2001.

[4] J. Ryu, and S.-B. Cho, "Gene Expression Classification Using Optimal Feature/Classifier Ensemble with Negative Correlation," *Proceedings of the 2002 International Joint Conference on Neural Networks*, vol. 1, pp. 198–203, 2002.

[5] Min Su, M. Basu, and A. Toure, "Multi-Domain Gating Network for Classification of Cancer Cells Using Gene Expression Data," *Proceedings of the 2002 International Joint Conference on Neural Networks*, vol. 1, pp. 286–289, 2002.

[6] R, Xu, G. Anagnostopoulos, and D. Wunsch, "Tissue Classification Through Analysis of Gene Expression Data Using A New Family of ART Architectures," *Proceedings of the 2002 International Joint Conference on Neural Networks*, vol. 1, pp. 300–304, 2002.

[7] A. Ben-Dor, L. Bruhn, N. Friedman, I. Nachman, M. Schummer, and Z. Yakhini, "Tissue Classification with Gene Expression Profiles," *Proceedings of the Fourth Annual International Conference on Computational Molecular Biology*, pp. 54–64, 2000.

[8] M. Kuramochi, and G. Karypis, "Classification Using Expression Profiles: A Feasibility Study," *Proceedings of the IEEE 2nd International Symposium on Bioinformatics and Bioengineering*, pp. 191–200, 2001.

[9] F. Azuaje, "Gene Expression Patterns and Cancer Classification: A Self-Adaptive and Incremental Neural Approach," *Proceedings of the 2000 IEEE EMBS International Conference on Information Technology Applications in Biomedicine*, pp. 308–313, 2000.

[10] F. Azuaje, "Making Genome Expression Data Meaningful: Prediction and Discovery of Classes of Cancer Through a Connectionist Learning Approach," *Proceedings of IEEE International Symposium on BioInformatics and Biomedical Engineering*, pp. 208–213, 2000.

[11] J. Principe, N. Euliano, and W. Lefebvre, Neural and Adaptive Systems: Fundamentals Through Simulations, John Wiley & Sons, Inc., 2000.

[12] S. Haykin, Neural Networks: A Comprehensive Foundation, Prentice-Hall, Inc., 1999.

[13] C. G. Looney, Pattern Recognition Using Neural Networks, Oxford University Press,

1997.

[14] T. Kosaka, S. Omatu, and T. Fujinaka, “Bill Classification by Using the LVQ Method,” *2001 IEEE International Conference on Systems, Man, and Cybernetics*, vol. 3, pp. 1430-1435, 2001.

[15] T. Kosaka, S. Omatu, T. Fujinaka, H. Yanagimoto, and M. Yoshioka, “Italian Lira Classification by LVQ,” *Proceedings of the 2001 International Joint Conference on Neural Networks*, vol. 4, pp. 2947-2951, 2001.

[16] D. F. Specht, “Probabilistic Neural Networks and the Polynomial Adaline as Complementary Techniques for Classification,” *IEEE Trans. Neural Networks,* vol. 1, no. 1, pp. 111-121, 1990.

[17] P. K. Patra, M. Nayak, S. K. Nayak, and N. K. Gobbak, “Probabilistic Neural Network for Pattern Classification,” *Proceedings of the 2002 International Joint Conference on Neural Networks*, vol. 2, pp. 1200–1205, 2002.

[18] D. S. Huang, and W. Zhao, “A Novel Method for Improving the Classification Capability of Radial Basis Probabilistic Neural Network Classifiers,” *Proceedings of the 2002 International Joint Conference on Neural Networks*, vol. 1, pp. 102–106, 2002.

[19] X. Jin, D. Srinivasan, and L. C. Ruey, “Classification of Freeway Traffic Patterns for Incident Detection Using Constructive Probabilistic Neural Networks,” *IEEE Trans. Neural Networks,* vol. 12. 1, no. 5, pp. 1173-1187, 2001.

[20] M. B. Menhaj, F. Delgosha, “A Soft Probabilistic Neural Network for Implementation of Bayesian Classifiers,” *Proceedings of the 2001 International Joint Conference on Neural Networks*, vol. 1, pp. 454-458, 2001.

[21] Y. Zhu, Y., Zhao, K., Palaniappan, X., Zhou, and X. Zhuang, “Optimal Bayesian Classifier for Land Cover Classification Using Landsat TM Data,” *IEEE 2000 International Geoscience and Remote Sensing Symposium*, vol. 1, pp. 447-450, 2000.