Fast Training Algorithms for Large Data Sets with Application to Classification of Multispectral Images

Qi Li, Donald W. Tufts, Roland J. Duhaime*, and Peter V. August*

Dept. of Electrical Engineering *Dept. of Natural Resources Science
University of Rhode Island
Kingston, RI, 02881

Abstract

Two methods of classification and related fast training algorithms are compared with each other and with backpropagation in this paper. The first method is the Discriminant Neural Network (DNN) [1, 2]. One hidden node is added at each design stage until the DNN meets the design requirements. The second method uses the Radial Basis Function Network (RBF) [3]. We modify the RBF by solving a succession of binary classification problems in order to provide fast training. These two classification methods are applied to automatically classify 14 categories of land cover using multispectral aerial images. We find that the training times for the DNN and the modified RBF (MRBF) are much less than the training times for backpropagation or RBF. The performances of DNN (72%) and MRBF (60%) are better than obtained by linear discriminant analysis (LDA) (55%) [4]. The resulting structure and computations are simpler for the DNN than for the other methods.

1 Introduction

For real world applications of classification and pattern recognition, such as remote sensor images, radar, sonar and speech data, the data sets are often very large. Even through multilayer perceptron (MLP) neural networks with backpropagation algorithms can be applied to these problems, it is well known that the training phase normally takes a tremendous amount of time. To overcome the problem, two neural network architectures and associated training algorithms are proposed here.

The first one, called the *Discriminant Neural Network* (DNN) is motivated by optimal multivariate Gaussian classification. When two training data populations C_1 and C_2 are described as multivariate Gaussian distributions with sample mean vectors and covariance matrices μ_1, Σ_1 and μ_2, Σ_2 respectively, the optimal classification rule to minimize the expected cost of misclassification is given by [5]:

$$C_1 : -\frac{1}{2}\mathbf{x}^t(\Sigma_1^{-1} - \Sigma_2^{-1})\mathbf{x} + (\mu_1^t\Sigma_1^{-1} - \mu_2^t\Sigma_2^{-1})\mathbf{x} \geq \theta, \quad (1)$$

$$C_2 : -\frac{1}{2}\mathbf{x}^t(\Sigma_1^{-1} - \Sigma_2^{-1})\mathbf{x} + (\mu_1^t\Sigma_1^{-1} - \mu_2^t\Sigma_2^{-1})\mathbf{x} < \theta. \quad (2)$$

where $\mathbf{x}$ is an observed data vector or feature vector and θ is a threshold determined by the cost ratio, the prior probability ratio, and the determinants of the covariance matrices. When the covariance matrices are the same, the first quadratic term is zero, and the above classifier computes Fisher's linear discriminant. When the second term can be ignored, the above formulas only have the first quadratic term. These two properties helped us to define two kinds of hidden nodes in DNN training, a Fisher's node and a principal component node. The latter is an approximation to the quadratic term.

The second one, modified from the *Radial Basis Function Network* (RBF) [3], is also motivated by minimization of the Probability of Misclassification of Gaussian Populations [5]. The rule is to allocate a vector $\mathbf{x}$ to class k if the quadratic score

$$d_k(\mathbf{x}) = \textit{largest of } d_1(\mathbf{x}),\ d_2(\mathbf{x}),\ ...,\ d_n(\mathbf{x}). \quad (3)$$

where d_i is *quadratic discrimination score* for the ith class,

$$d_i(\mathbf{x}) = -\frac{1}{2}\ln|\Sigma_i| - \frac{1}{2}(\mathbf{x}-\mu_i)^t\Sigma_i^{-1}(\mathbf{x}-\mu_i) + \ln p_i, \quad (4)$$

where Σ_i and μ_i are as defined above, and the p_i is the prior probability for class i.

1058-6393/95 $4.00

2 Fast Training and Design of the Discriminant Neural Network

In [1, 2], Li and Tufts presented the DNN and the associated fast training algorithm.

As shown in Figure 1, the DNN has two layers, one hidden layer and one output layer. The hidden layer consists of hidden nodes. Each hidden node forms one or several parallel hyperplanes to partition the input pattern space into regions. Each region is represented by a binary word which is the output of the hidden node. The output nodes in the output layer are logic functions of the binary outputs of the hidden nodes.

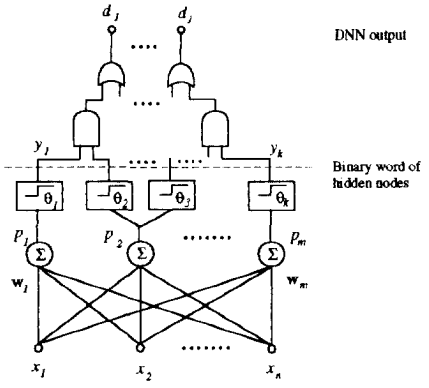

Figure 1: **A recommended implementation for a Discriminant Neural Network (DNN).**

The output of one of the biased hardlimiters of one hidden node in response to a given data vector $\mathbf{x}$ is

$$y = f(\mathbf{x}^t\mathbf{w}), \tag{5}$$

in which $\mathbf{x}^t\mathbf{w}$ is the inner product of the input vector $\mathbf{x}$ and weight vector $\mathbf{w}$. The biased hardlimiter function, $f(.)$, is specified by the formula,

$$f(\mathbf{x}^t\mathbf{w}) = \begin{cases} 1, & \mathbf{x}^t\mathbf{w} > \theta; \\ 0, & \mathbf{x}^t\mathbf{w} \leq \theta, \end{cases} \tag{6}$$

in which θ is a threshold value which can also be considered as a bias for the argument of the nonlinearity. Several hardlimiters are allowed for any one of the hidden nodes.

The design algorithm [1, 2] to train the input layer is to choose the weight vector $\mathbf{w}$ for each hidden node by applying Fisher's linear discriminant analysis or principal component discriminant analysis recursively. Depending on the residual data distribution at each training stage, one of the analysis methods will be selected to determine a weight vector $\mathbf{w}$ for the new node and to prune the already classified training vectors from the training set. That is, the training data vectors in the subsets which already satisfy a prescribed misclassification rate are deleted from the training data set. The remaining training data vectors are carried over to the training of the next hidden node. We sequentially add and train new, needed hidden nodes one by one pruning training data vectors at each steps until the network performance meets the design specification.

If we only use the concept of Fisher's linear discriminant analysis, then the corresponding weight vector provides little classification ability when the mean vectors of the training classes are too close. To avoid this problem, we use a *Principal Component Discriminant* to design one or more weight vectors $\mathbf{w}$ of principal component nodes. The output values of a principal component node provides an approximation to the quadratic term in the Gaussian classifier.

$$\mathbf{x}^t(\mathbf{\Sigma}_1^{-1} - \mathbf{\Sigma}_2^{-1})\mathbf{x} \approx \lambda_1(\mathbf{x}^t\mathbf{e}_1)^2, \tag{7}$$

where λ_1 and $\mathbf{e}_1$ are the largest eigenvalue and the corresponding principal eigenvector of $(\mathbf{\Sigma}_1^{-1} - \mathbf{\Sigma}_2^{-1})$. Because the squared quantity on the right will be thresholded, the square does not need to be computed. We simply use two hardlimited values of $\mathbf{x}^t\mathbf{e}_1$, one with positive bias and one with negative bias. For the case of multiple classes, $\mathbf{\Sigma}_1$ is the covariance matrix of one class and $\mathbf{\Sigma}_2$ is the covariance matrix of the combined other classes.

2.1 A Design Example

We use an example in Figure 2 to illustrate the above training methods. The design starts from the training of the first hidden node and the associated two hyperplanes with all the training data using Fisher's method. Then, the classified data is pruned off and only the unclassified data in between of the two hyperplanes are used to train the second hidden node. The residual data set from Figure 2 is shown in Figure 3. Since the mean vectors of the two classes are close, Fisher's method does not give the best classification result. We find this out by designing a alternate second hidden node and determining its two associated hyperplanes. The residual data set was totally separated by the second node. The classification improvement showed that the principal component node is a better choice then the linear discriminant node for the second hidden node. The design procedure was then stopped. The partitioned input space is shown in Figure 4.

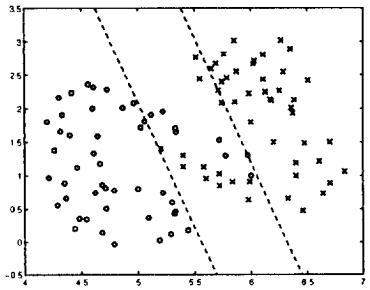

Figure 2: **The original data and the hyperplanes of the first hidden node (Fisher's node).**

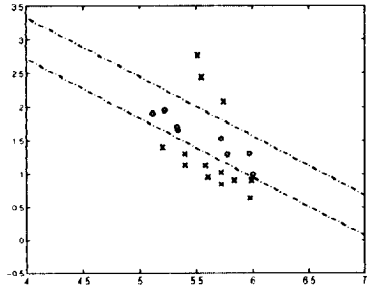

Figure 3: **The residual data set and the hyperplanes of the second hidden node (Principal Component Discriminant node).**

If we used Fisher's method to design the second node, the residual data set would not be totally partitioned by the second Fisher's node, and one more node would be needed to totally partition the input space. Even though that inefficient extra node can be pruned by Boolean minization [1, 2] at the end of the design, it will slow down the training procudere.

For the above classification problem, the Backpropagation (BP) training method takes hundreds of seconds to hours, and one still does not get satisfactory classification. The Radial Basis Network (RBF) can converge to an acceptable performance in 35 seconds, but it needs 56 nodes. On the same problem, DNN design only takes 0.2 seconds and needs only two hidden nodes with a better performance than both BP and RBF.

3 Modified Radial Basis Function Network

The training of RBF networks is faster than multilayer perceptron networks which are trained using backpropagation training algorithm. However, when

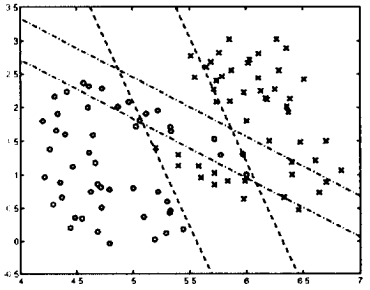

Figure 4: **The partitioned input space, two hidden nodes and four thresholds.**

training data sets are large and with multiple classification categories, it still needs a long training time. To speed up the training process for a large data set with multiple outputs, we break down the training problem into the training of each member of a group of RBF modules, called a *Modified Radial Basis Function Network* (MRBF). A MRBF network consists of a group of single-output RBF modules. The number of modules is equal to the number of classification categories. Thus each single-output RBF module is only responsible for one classification category.

The structure of the MRBF network is shown in Figure 5. It has a group of RBF modules running in parallel and a Winner-Take-All (WTA) unit at the output layer. Since each of the RBF modules is trained for one specified category, when a pattern is applied to the MRBF, the RBF module trained for that category will have the largest response among all RBF module outputs. Then that RBF output is the winner and the category represented by that RBF module is the classified category.

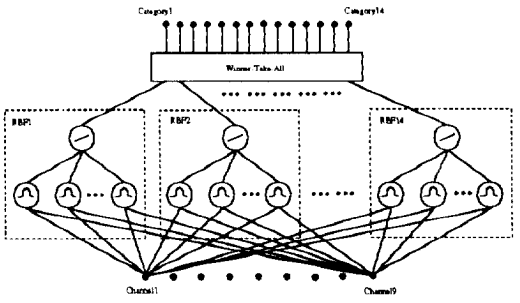

Figure 5: **The structure of a Modified Radial Basis Function Network (MRBF).**

The training procedure is a succession of solutions of binary classification problems. When training one of the RBF modules, the data in the category which corresponds to that RBF module is in one class and

the data in all other categories are grouped into another class. As drawn in the broken-line boxes in Figure 5, a RBF network (a module in a MRBF) is a two-layer network. The output node in each RBF network forms a linear combination of the outputs of the hidden nodes. Each hidden node has a basis function. The most common basis function is a Gaussian kernel function. The basis functions produce a localized response to an input stimulus, i.e. they output a significant nonzero response only when the input vector falls within a small localized region of the input space. The details of the RBF training algorithms can be found in [3] and [6].

4 Land Cover Recognition from Multispectral Images

We applied the DNN and MRBF networks to recognize categories of land cover from three images of Block Island, Rhode Island using three spectral bands - two visible and one infrared. Each complete image has 4591 × 7754 pixels. Each pixel has a resolution of 1.27 m and belongs to one of 14 categories of land cover.

The training data set is a matrix which is formed from a subset of pixels which have been classified and labeled. Each row is one training vector which has 9 feature elements associated with one pixel. In [4], each of these vectors was labeled with one of the 14 land cover categories. The nine feature components of each row vector consist of pixel intensity in the three color bands, three local standard deviations of intensity, one for each color, and three features from the side information of a soil database. The definitions of the features in the entries of the data matrix are listed as follows [4].

Column 1 The number of the category to which the pixel belongs (1-14).

Column 2 Band 1 represents the intensity of the infrared portion of the reflected light received by camera in a pixel designated by the row index(0-255).

Column 3 Band 2 represents the intensity of the visible red portion of the reflected light in a row-designated pixel (0-255).

Column 4 Band 3 represents the intensity of the visible blue portion of the reflected light in a row-designated pixel (0-255).

Column 5 The standard deviation of Band 1 reflectance in a diameter of 10m floating window around the row-designated pixel.

Column 6 The standard deviation of Band 2 reflectance in a diameter of 10m floating window around the row-designated pixel.

Column 7 The standard deviation of Band 3 reflectance in a diameter of 10m floating window around the row-designated pixel.

Column 8 Degree of local slope at the designated pixel.

Column 9 Aspect of the slope at the designated pixel. A south facing aspect is a 0 and north facing or no aspect is 1.

Column 10 Drainage class of soil. Range is from 0 to 5 with the meanings of 0 = variable, 1 = well drained and somewhat excessively drained, 2 = moderately well drained, 3 = poorly drained, 4 = very poorly drained, and 5 = open water.

In [4], Duhaime identified 14 categories of land covers for supervised training. These categories are as follows:

Category 1 Freshwater shrub wetlands are comprised of arrowwood and rosaceous shrubs in very poorly drained soils.

Category 2 Morainal grasslands consists of herbaceous fields.

Category 3 Pastures and grass fields.

Category 4 Upland pine shrublands occuring as small patches of coniferous shrubs.

Category 5 Dunes are hills of wind-down sand.

Category 6 Salt marsh is characterized by perennial, salt tolerant graminoids, etc.

Category 7 Hayfields consists of areas that are managed for hay.

Category 8 Upland deciduous shrublands consist of dense stands of shadbush, bayberry, and arrowwood.

Category 9 Old fields represent a successional stage between pastures and shrublands.

Category 10 Sand consists of gravel pits or beaches with little or no vegetation.

Category 11 Freshwater consists of standing water with no rooted vegetation.

Category 12 Ocean

Category 13 Freshwater emergent wetlands are dominated by hydrophytic perennials, graminoids, and tussock sedges.

Category 14 Other is a miscellaneous category comprised of roads, roofs of buildings, paved areas, or rocky shores.

5 Experimental Results

The computer experiments on the multispectral image features started by using backpropagation and RBF algorithms. However, both of them did not get the needed classification results in a reasonable amount of times as estimated from their converage speeds. Then the DNN and the Modified RBF algorithms were applied to solve the problem. The experimental results are listed in Table 1 and compared with one another.

Table. 1. ComparingThree Training Methods

Methods	MFLOPS	CPU Times in Seconds	No. of Nodes	Accuracy on Test Sets
DNN	37.64	58	77	72 %
MRBF	221.93	518	490	60 %
LDA	—	—	—	55 %

The MRBF method used a training data set of 140 sample vectors, 10 from each category, and tested on a test data set of 700 samples, 50 samples from each category. It gets an average accuracy of 60% on the test set for all of the 14 categories defined above. The training took 518 seconds CPU time on a Sun Sparc IPX workstation. The DNN is trained by 700 training samples since the DNN can run much faster than others. It took only 58 seconds CPU time and reached an average performance of 72% on the same test set and on all 14 categories (the performance is 65% if using the 140 sample set for training). The performance of linear discriminant analysis (LDA) was reported in [4]. It is 55% on an average of 11 categories out of the all 14 categories based on different training and test data sets.

6 Conclusions

Two neural network structures, a discriminant neural network and a modified radial basis function network, and associated fast design methods were presented. These two methods are also compared with three other existing training methods, linear discriminant analysis, backpropagation, and radial basis based function, for the design of a classifier for the multispectral images. The computer experiments show that the performance of the Discriminant Neural Network is better than other approaches. It takes much less training time than the others. The performance and the training time of the modified RBF are acceptable but it needs more nodes than the DNN.

Acknowledgements

The authors thank Dr. James Kowalski for technical discussions and for facilitating this project.

References

[1] Q. Li and D.W. Tufts, "Synthesizing neural networks by sequential addition of hidden nodes," Proc. IEEE International Conference on Neural Networks, pp. 708-713, Orlando, Florida, June 1994.

[2] Q. Li and D.W. Tufts, "Discriminant networks: a simple, effective, and rapidly trainable class of neural networks," Submitted to the *IEEE Trans. on Neural Networks*, February 1994.

[3] S. Chen, C.F.N. Cowan, and P.M. Grant, "Orthogonal least squares learning algorithm for radial basis function networks," *IEEE Trans. on Neural Networks*, vol. 2, No. 2, March 1991.

[4] R.J. Duhaime, "The use of color infrared digital orthophotography to map vegetation on Block Island, Rhode Island," M.S. Thesis, Dept. of natural resources science, University of Rhode Island, May 1994.

[5] R.A. Johnson and D.W. Wichern, "Applied multivariate statistical analysis", pp. 470-530, New Jersey: Prentice Hall, 1988.

[6] H. Demuth and M. Beale, "Neural network toolbox user's guide," The MathWorks Inc., 1994.