Design of Neural Networks for Classification of Remotely Sensed Imagery

Samir R. Chettri
Hughes-STX at NASA/Goddard Space Flight Center
Greenbelt, MD 20771

Robert F. Cromp
Code 934
NASA/Goddard Space Flight Center
Greenbelt, MD 20771

Mark Birmingham
Princeton University, New Jersey

Abstract

As currently planned, future Earth remote sensing platforms (i.e., Earth Observing System [EOS]) will be capable of generating data at a rate of over fifty Megabits per second. To address this issue the Intelligent Data Management (IDM) project at NASA/GSFC has prototyped an Intelligent Information Fusion System (IIFS) that uses backpropagation neural networks for the classification of remotely sensed imagery. This is part of the IDM strategy of providing archived data to a researcher through a variety of discipline–specific indices.

In this paper we discuss classification accuracies of a backpropagation neural network and compare it with a maximum likelihood classifier (MLC) with multivariate normal class models. We have found that, because of its nonparametric nature, the neural network outperforms the MLC in this area. In addition, we discuss techniques for constructing optimal neural nets on parallel hardware like the MasPar MP-1 currently at NASA/GSFC. Other important discussions are centered around training and classification times of the two methods, and sensitivity to the training data. Finally we discuss future work in the area of classification and neural nets.

1 Introduction

With the expected explosive growth of data generated by Earth orbiting platforms such as the Earth Observing System (EOS), it is imperative that the data be rapidly archived and made available to the researcher through a variety of discipline–specific indices. To address this issue, the Intelligent Data Management (IDM) project at NASA/GSFC has prototyped an Intelligent Information Fusion System (IIFS) that classifies satellite data from a number of spectral bands into a number of land use/land cover categories [Anderson 76] and provides rapid access to the classified data as well as the raw data. The choice of land use/land cover categories is part of a larger plan to classify images based on a scientist's specific research interests.

Management of the EOS data can be considered as two overlapping problems: characterization of the data content and subsequent archiving of images; and efficient querying of the resulting voluminous database. The first problem can be solved independently of the choice of database technology, and is elaborated upon in this paper.

In this paper we discuss the use of neural nets for the classification of remotely sensed imagery. In particular we compare backpropagation neural nets (BPNN) with a Gaussian maximum likelihood classifier (GMLC). Some of the items we compare include training time, classification time, accuracy of classification, and sensitivity of classification accuracy to the training set. This study will thus help researchers decide on what classification method to apply, given the constraints of their problem.

The body of the paper is divided into three sections. First, we briefly discuss the algorithms for the neural net and the MLC methods. Next, we compare and contrast the two methods. Finally, we discuss selection criteria for each of the two methods and conclude with our future research directions.

2 Neural net and maximum likelihood classification algorithms

In this section we discuss the basic algorithms for training and classification for the neural net (NN) and Gaussian maximum likelihood classifiers (GMLC). For more details on both topics, refer to [Andrews 72] and [Hertz 91].

2.1 Maximum likelihood classification

The job of designing the pattern classifier consists of first dividing the feature space into decision regions and then constructing a classifier so that it will identify any measurement vector $\mathbf{X}$ as belonging to the class corresponding to the decision region in which it falls.

The maximum likelihood decision rule allows us to construct discriminant functions for the purposes of pattern classification [Andrews 72]. Given K classes, let $f(\mathbf{X} \mid S_k)$ be the probability density function (pdf) associated with the measurement vector $\mathbf{X}$, given that $\mathbf{X}$ is from class k. Let $P(S_k)$ be the *a priori* probability of class k. We can use the **maximum likelihood decision rule** to identify the class to which $\mathbf{X}$ belongs. It can be stated as follows:

$$\textit{Decide } \mathbf{X} \in S_k \textit{ iff } f(\mathbf{X} \mid S_k)P(S_k) \geq f(\mathbf{X} \mid S_j)P(S_j), j = 1, 2, \cdots K \; .$$

The products $f(\mathbf{X} \mid S_k)P(S_k)$, where $k = 1, 2, \cdots K$ correspond to discriminant functions $g_1(\mathbf{X})$, $g_2(\mathbf{X}), \cdots g_K(\mathbf{X})$. Thus these functions are evaluated at $\mathbf{X} = \mathbf{X}_i$ where $\mathbf{X}_i$ is the unknown vector; next, the maximum of these functions $g_k(\mathbf{X}_i)$ is determined and the unknown vector is assigned to the class k.

The discriminant function for the multivariate normal density can be written as

$$g_k(\mathbf{X}) = \ln[P(S_k)] - \frac{1}{2} \ln |\Sigma_k| - \frac{1}{2}(\mathbf{X} - \mathbf{U}_k)^T \; \Sigma_k^{-1} \; (\mathbf{X} - \mathbf{U}_k). \tag{1}$$

In the above equation, both $\mathbf{\Sigma}_k$ (the variance–covariance matrix) and $\mathbf{U}_k$ (mean vector) are provided by the user. In practice, training samples are used to obtain *estimates* of $\mathbf{\Sigma}_k$ and $\mathbf{U}_k$. Also from equation (1) we see that once the training statistics are generated, only the quadratic kernel varies with each input vector $\mathbf{X}$. Such a classifier is called a Gaussian Maximum Likelihood Classifier (GMLC) and is used in our classification experiments. It is a parametric supervised technique for estimation of *a posteriori* probabilities.

2.2 Backpropagation

The backpropagation algorithm is the backbone of much of the current resurgence of research into neural nets [Hertz 91]. With respect to pattern recognition, backpropagation can be considered to be a nonparametric technique for estimation of *a posteriori* probabilities [Wan 90].

The backpropagation network consists of a series of layers: an input layer, an output layer and one or more hidden layers. Each layer has a number of nodes or processing elements (PE). Each node in a layer is connected to every node in the next layer and the propagation of information is unidirectional. Also, in our simulations, connections are only permitted between nodes belonging to adjacent layers. Each connection has a value associated with it called its weight, and each PE has a value associated with it called a threshold value.

To find the output of any node we first sum the products of the output of all the nodes before it with the weights associated with each connection. Next we subtract the threshold value of the node from this sum, and finally we pass this value through an activation function that determines the output of the current node. The activation function used in this study is the sigmoid function defined as

$$f(h) = \frac{1}{1 + \exp^{-kh}} \tag{2}$$

where $h = \sum w_i \xi_i - \Theta$, w_i are the weights, ξ_i are the inputs to the current node (or output of nodes in the previous layer), and Θ is the threshold value in the the current PE.

The training phase of backpropagation gives a method of changing the weights in a network such that it learns a series of input/output pairs (ξ_k^μ, ζ_i^μ) where ξ_k^μ is the k^{th} input for the μ^{th} pattern, and ζ_i^μ is the correct output for the i^{th} output unit for the μ^{th} pattern. The basis for the weight change is gradient descent, thus the weights w_{jk}^r are changed by an amount Δw_{jk}^r that is proportional to the derivative error function E with respect to the weights, where w_{jk}^r is the weight that lies on a connection between the j^{th} PE of one layer with the k^{th} PE of the previous layer, and r indicates the number of the current layer. The most commonly used error function is the quadratic error function [Hertz 91] defined as

$$E = \frac{1}{2} \sum_{\mu i} [\zeta_i^\mu - O_i^\mu]^2 . \tag{3}$$

Here, the summation is over *all* training samples, and O_i^μ is the network output for a given input pattern for which the expected output is ζ_i^μ.

Training proceeds by randomly selecting the weights in the net, passing the input pattern through the network, getting the resulting output, obtaining $\Delta w_{jk}^r = -\eta \frac{\partial E}{\partial w_{jk}^r}$, and finally updating

Table 1: Distribution of data, Blackhills and DC data sets

Class	# of Pixels Blackhills		# of Pixels DC		Class name
	Training	Entire image	Training	Entire image	
0	453	6676	73	2668	Urban
1	478	42432	74	776	Agricultural
2	464	16727	75	3733	Rangeland
3	482	194868	75	13826	Forested Land
4	0	0	0	0	Water bodies
5	0	0	0	0	Wetland
6	368	1441	74	936	Barren
7	0	0	0	0	Tundra
8	0	0	0	0	Perennial snow and ice

the weights by using $w^r_{jk,\text{new}} = w^r_{jk,\text{old}} + \Delta w^r_{jk}$. Training is done either for a maximum number of iterations or until the error E goes below a pre-defined threshold level. At this point the trained network can be used on data in feed-forward mode for the purposes of classification.

3 Experimental Method

In this section we describe the data that we used to compare our neural net (NN) classifier and Gaussian Maximum Likelihood classifier (GMLC). In addition we discuss the selection process that we employed for the training and testing data. Finally, we describe the training and testing methodology used.

3.1 Description of data set

Two data sets were used for the purpose of comparing the the GMLC and the NN approach. The first is the Blackhills data set, taken from the Landsat 2 multispectral scanner (MSS) (see Figure 1). The spectral bands are 0.5 - $0.6\mu m$ (green), 0.6 - $0.7\mu m$ (red), 0.7 - $0.8\mu m$ (near-infrared) and 0.8 - $1.1\mu m$ (near-infrared). These bands correspond to channels 4 through 7 of the Landsat sensors. There are 262, 144 pixels corresponding to a 512 × 512 image size, and each pixel represents $79m \times 79m$ on the ground. The image region covers a range of latitudes from $44°15'$ to $44°30'$ and longitudes from $103°30'$ to $103°45'$; the images were obtained in September 1973. The ground truth was also provided in the form of United States Geological Survey level II land use/land cover data [Anderson 76]. Since we were only interested in level I classification, the different classes were conglomerated into the various higher level classes in the hierarchy; the distribution of pixels is shown in column three of Table 1.

The second data set has been used previously in [Campbell 89], which, to our knowledge, was the first open-literature publication of the use of the backpropagation network to do classification of remotely sensed imagery. In contrast to that publication, we will only be using the USGS level I land use/land cover scheme for classification. The first four spectral bands from a LANDSAT-4

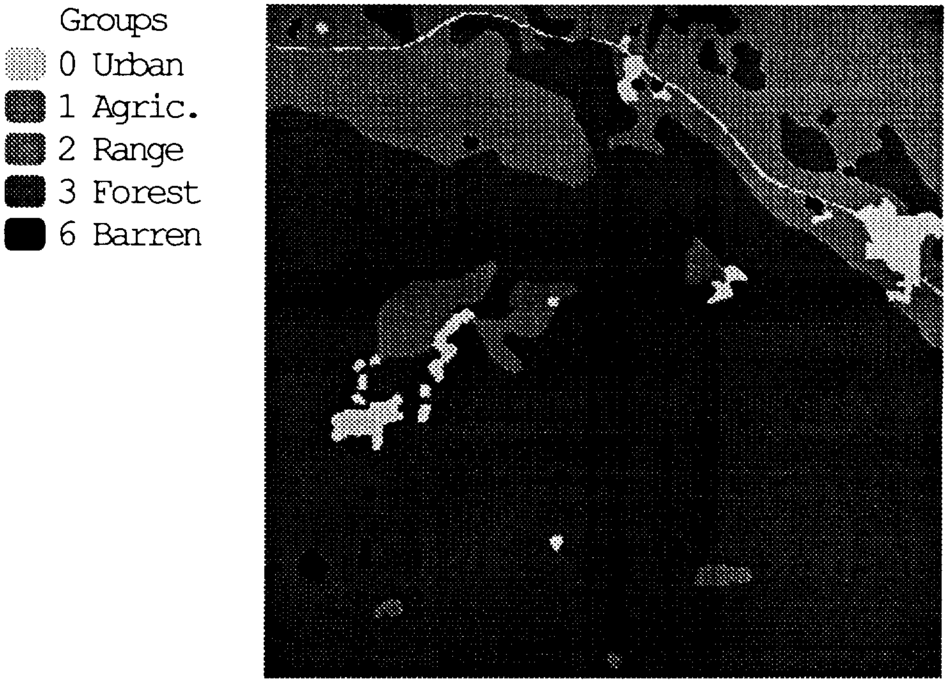

Figure 1: Ground truth for the Blackhills image

thematic mapper (TM) image were used, and the corresponding spectral bands were 0.45 - 0.52μm, 0.52 - 0.60μm, 0.60 - 0.69μm and 0.76 - 0.90μm respectively. There are a total of 22,801 pixels in a 151 $\times$ 151 grid, and each pixel is representative of a $30m \times 30m$ area on the ground. Only 21,939 pixels of valid ground truth were available, and these are tabulated in column five of Table 1. Figure 2 shows a gray-level thematic map, with the various classes labeled. The area covered is about 25 miles SSE of Washington, DC, and is called the DC data set.

3.2 Selection of training and test data

The most important point to note is that identical data were presented to the NN and GML classifiers for training and testing. The ground truth was viewed on a display device to get an idea of the spatial distribution of the ground truth pixels. According to [Richards 86], a minimum sample size of 60 pixels is necessary for accurate classification. Also, according to [Campbell 87], a large number of smaller training sites should be used rather than a few large ones. Following these recommendations, we formed training sets from both the TM (DC data set) and MSS (Blackhills data set) scenes. The results are summarized in columns two and four of Table 1.

3.3 Training and testing

The training set and the test set are disjoint. The classifiers were derived from the training group and the error estimate obtained from the test group. This method is known as the "holdout" or H method of estimating errors. The training data itself consists of a series of sites from each class in the image. For the GMLC we can compute the mean vectors and covariance matrices for each site separately and combine them to form the class mean vectors and covariance matrices. For the NN

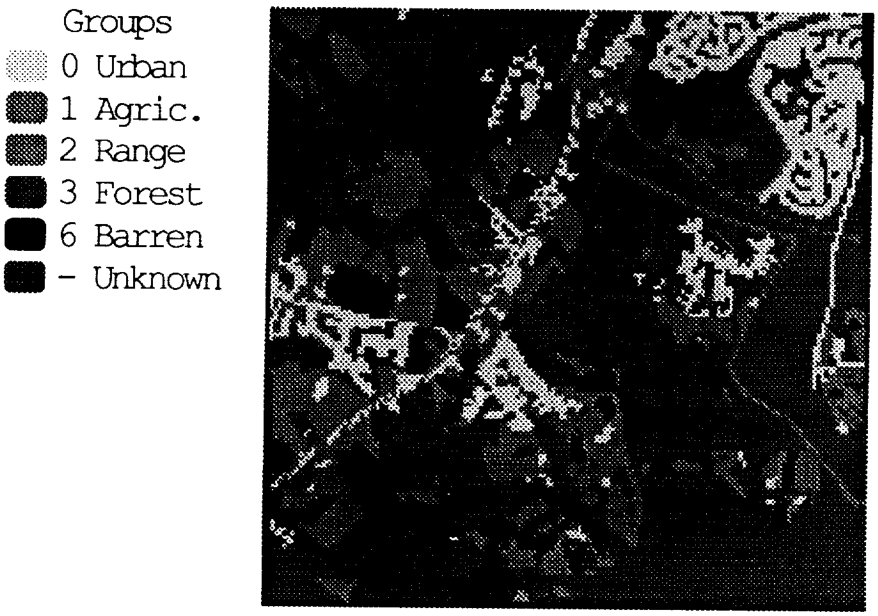

Figure 2: Ground truth for the DC image

approach, the site information is not important. However, the channel information, which is an integer in the range [0 255], is scaled to [0.1 0.9] for training. The training of the NN is achieved by repeatedly presenting the data to the net and performing the backpropagation algorithm as described in section 2.2. Training in the NN is completed when either the error as described in equation (3) goes below a threshold level, or a maximum number of iterations of the BP algorithm is reached. It is important to avoid overtraining the net as it would classify the training data perfectly but would not perform as well on the testing data.

An alternative method for training and testing data is recommended in [Weiss 89] and is called *leaving-one-out*. The principle is very simple and involves taking $n - 1$ points from the sample and training the classifier on that information. The n^{th} point is then classified and this training and testing procedure is repeated for all n points in the set. Quite clearly, for our data it is entirely infeasible to use the leaving-one-out process since we will have to design $n = 262,144$ and $n = 22,801$ classifiers for the Blackhills and DC data sets respectively. However, with large sample sizes (which is the case for our data) the accuracy in estimating error is adequate by the H method [Kanal 68], hence our selection of that procedure.

The training of the NN proceeds on the basis that it is a function optimization procedure. Remembering that the function optimization process is sensitive to initial conditions, we:

1. randomize the initial selection of weights and thresholds;

2. randomize the order of the training data.

The effect of this is to produce different neural nets for each set of initial conditions. Thus, when we compare the NN and GMLC accuracies, we will be referring to the *average* correctly classified by the NN, while there will be only one value for the GMLC. Training of these multiple neural nets can be achieved on the MasPar MP-1, with each processing element generating an independent NN. The best net is one that obtains the lowest error on the test set, and it is selected for general use.

4 Comparing backpropagation (BP) with Gaussian maximum likelihood classification (GMLC)

In this section we compare BP with GMLC under different sets of categories. These categories include time for training, time for classification, memory requirements, and classification accuracy. Instead of exact calculations, we give order–of–magnitude estimates for these quantities. It is important to note that we assume that our neural net is restricted to one hidden layer, because according to Kolmogorov's theorem [Hecht-Nielsen 90], a three layer neural net can be constructed that performs any continuous mapping with $(2N + 1)$ PE's in the hidden layer, where N is the number of elements in the input layer. Another assumption is that the output layer has m nodes where m is proportional to N. Note that N is also the dimension of the unknown vector whose class we are trying to determine. This notation will be used in the subsequent subsections.

4.1 Training time

Training time in the neural net can be shown to be $\mathcal{O}(N^6)$ based on arguments in [Muhlenbein 90]. The training phase of GMLC has a worst case complexity of $\mathcal{O}(N^3)$.

While it seems quite clear the training time for the GMLC is significantly less, the *current, off-the-shelf* availability of hardware to do backpropagation training reduces the advantage of the GMLC. Since BP is a far more general process, hardware will continue to be supported and developed for it, whereas since the GMLC is a specific method of classification, it is uneconomical to develop custom hardware for this process. In addition, we have only discussed a simple BP scheme. In fact there exist a number of speed–up procedures [Hertz 91], that would make BP competitive with GMLC in software implementations.

Also, while the GMLC is suited for similar types of data (in our case spectral information), it is unsuited for multi–source data, since the underlying distribution may change when one adds (say) elevation data [Benediktsson 90]. The NN handles these problems in an effective manner. In addition, our application permits the training to be performed off–line, thus eliminating the time factor entirely.

4.2 Classification time

Both the NN and GMLC method can be shown to take constant time for the classification of one pixel. Again, the availability of hardware makes the NN method more attractive. In addition, even in software, the time needed for neural network computations can be considerably decreased by using integer calculations for the sigmoid function [Birmingham 91]. In this paper, a Taylor series

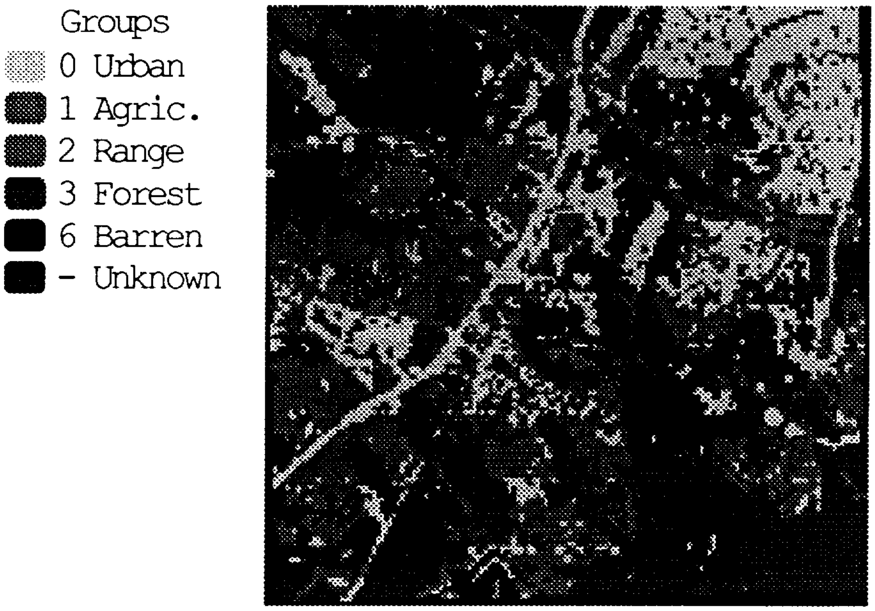

Figure 3: Maximum likelihood classified DC image

approximation to the sigmoid with only integer fractions is used. It was found that an almost six-fold speed-up factor can be obtained. Another advantage of the NN is that the same hardware can be used for training and feedforward classification, which is not the case for the GMLC.

4.3 Accuracy

The accuracy of each method can be summarized by the *contingency table*, which is an $R \times R$ matrix of numbers, where R is the number of classes. Each entry C_{ij} in the matrix represents the number of times a pixel in class i was put into class j. C_{ii} is the number of correct classifications in class i.

For the DC data set we have two sets of contingency tables. In Table 2, we present the GMLC accuracy results for the training and test data respectively. In Table 3, *typical* accuracy results for the NN are presented. In addition, Table 4 presents the average percent correctly classified (PCC), the maximum PCC, and the minimum PCC for all the nets that were trained. We see that even the minimum PCC for the NN exceeds the PCC value obtained by GMLC. The number of nets trained to get these readings was six. For the purposes of visual comparison, the classified images are shown in Figures 3 and 4.

We have two sets of contingency tables for the Blackhills data set. In Table 5, we present the GMLC accuracy results for the training and test data, respectively. In Table 6, *typical* accuracy results for the NN are presented. In addition, Table 7 presents the average PCC, the maximum PCC, the minimum PCC as well as the standard deviation of the PCC for all the nets that were

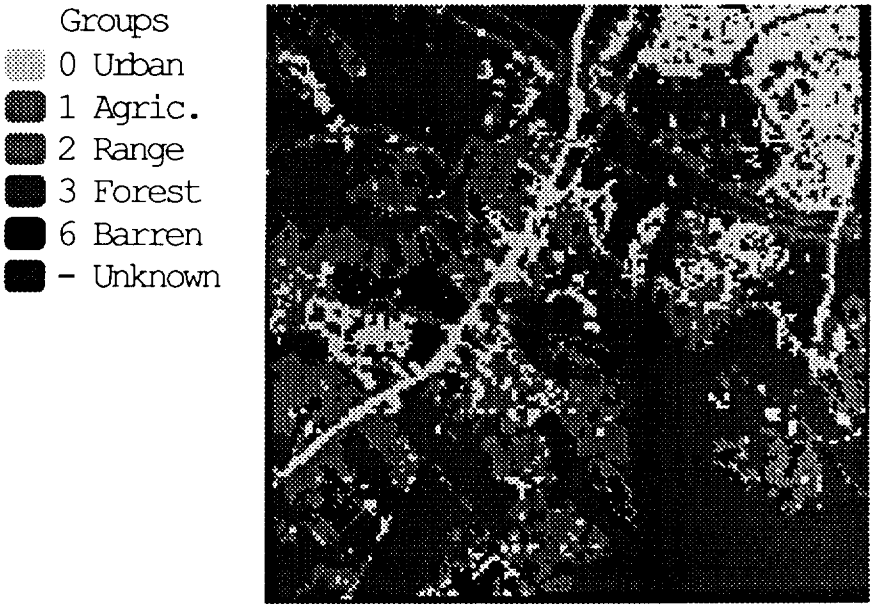

Figure 4: Neural net classified DC image

Table 2: Contingency table for GMLC, DC training data on left (PCC = 0.827), DC test data on right (PCC = 0.623)

	0	**1**	**2**	**3**	**6**	**0**	**1**	**2**	**3**	**6**
0	**68**	0	5	0	0	**1843**	219	505	16	12
1	2	**56**	12	4	0	30	**380**	158	14	120
2	6	5	**64**	0	0	605	1132	**1472**	220	229
3	0	0	1	**72**	2	1661	715	1634	**9408**	333
6	0	27	0	0	**47**	46	380	100	3	**333**

Table 3: Contingency table for NN, DC training data on left (PCC = 0.871), DC test data on right (PCC = 0.677)

	0	**1**	**2**	**3**	**6**	**0**	**1**	**2**	**3**	**6**
0	**68**	0	5	0	0	**1714**	94	751	26	10
1	1	**47**	15	0	11	15	**195**	243	21	228
2	2	1	**72**	0	0	440	404	**2054**	322	438
3	0	0	0	**75**	0	1144	133	1952	**10227**	295
6	0	13	0	0	**61**	33	251	170	4	**404**

Table 4: Statistics for NN performance on DC test data set

#	Av.	Max.	Min.	Std. dev.
6	0.675	0.688	0.665	0.093

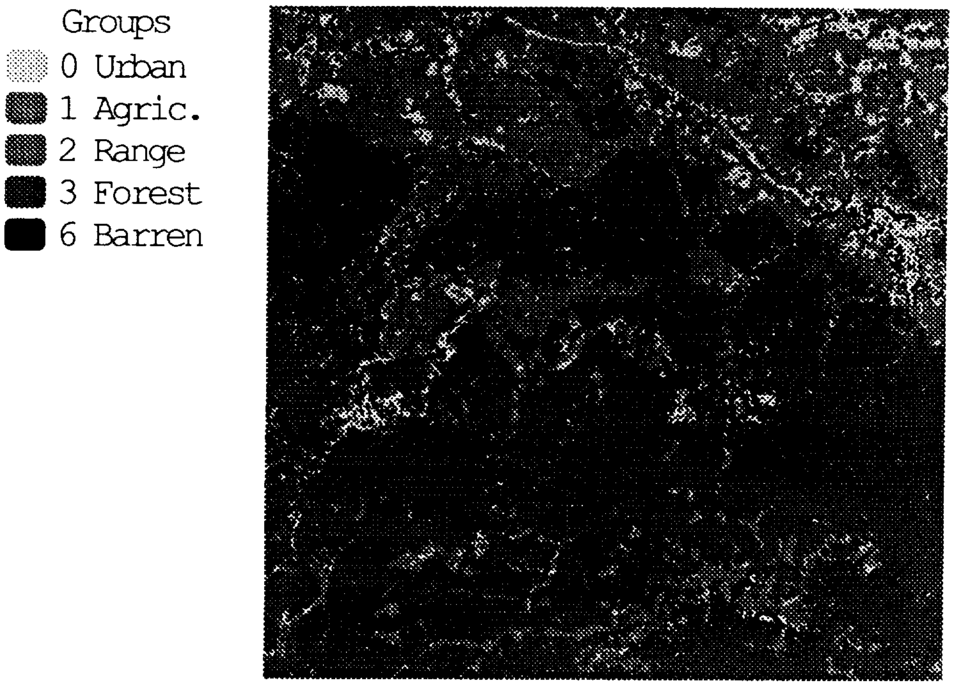

Figure 5: Maximum likelihood classified Blackhills image

trained. We see that even the minimum PCC for the NN exceeds the PCC value obtained by GMLC. To compare the classified images visually, refer to Figures 5 and 6.

It is important to note that the contingency table can be used as an aid to further improving classification accuracy. This is called the conditional probabilities matrix (CPM) technique and is described in detail in [Cromp 91]. Using this technique, a distance measure representing the error was reduced by approximately 50%. Of course, the method applies to the contingency tables produced by both the NN and GMLC.

4.4 Memory requirements

Both the NN and the GMLC can be shown to require $\mathcal{O}(N^2)$ memory elements.

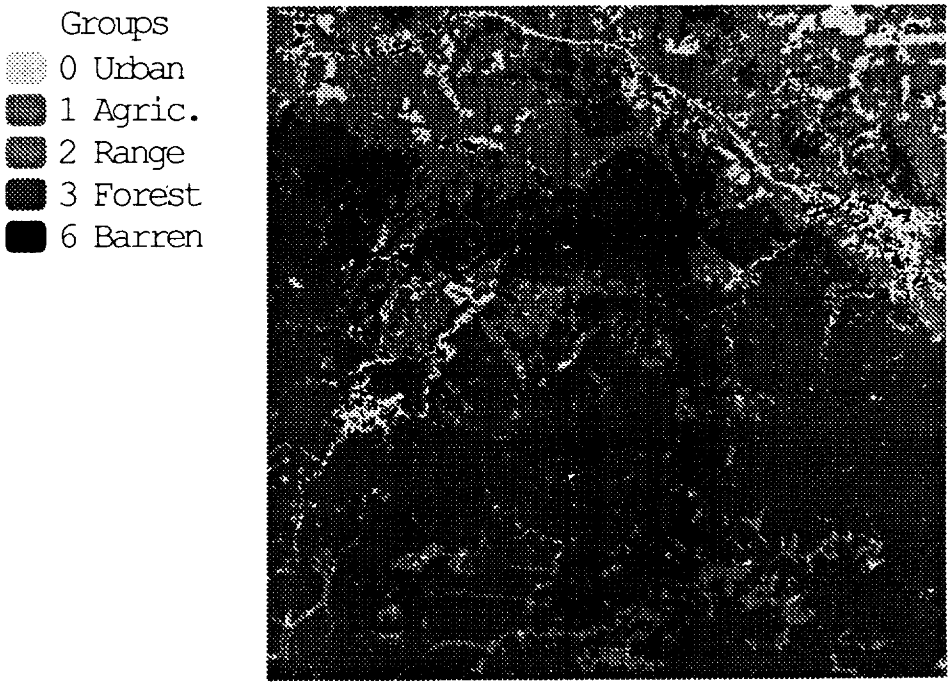

Figure 6: Neural net classified Blackhills image

Table 5: Contingency table for GMLC, Blackhills training data on left (PCC = 0.571); Blackhills test data on right (PCC = 0.653)

	0	1	2	3	6	0	1	2	3	6
0	**236**	72	91	18	36	**2425**	731	1307	876	884
1	26	**316**	135	0	1	6631	**16140**	15741	1463	1979
2	16	119	**279**	43	7	1840	3450	**9333**	1165	475
3	1	4	77	**385**	15	4077	8761	25804	**141644**	14100
6	61	28	78	136	**65**	157	116	147	442	**211**

Table 6: Contingency table for NN, Blackhills training data on left (PCC = 0.578); Blackhills test data on right (PCC = 0.727)

	0	**1**	**2**	**3**	**6**	**0**	**1**	**2**	**3**	**6**
0	**274**	74	86	16	3	**3021**	709	1413	892	188
1	26	**323**	124	5	0	7689	**16700**	15285	1998	282
2	21	115	**284**	44	0	2134	3610	**9360**	1128	31
3	5	4	91	**381**	1	2183	11572	20749	**159832**	50
6	87	31	75	139	**36**	228	125	155	473	**92**

Table 7: Statistics for NN performance on Hills test data set

#	Av.	Max.	Min.	Std. dev.
6	0.736	0.754	0.706	0.019

5 Concluding remarks and future work

In this research we have compared the backpropagation neural network (BPNN) with Gaussian maximum likelihood classification (GMLC). The accuracy level of BPNN (i.e., the number of correctly classified pixels in a test set) is better than the accuracy obtainable by GMLC. This is because the BPNN makes no *a priori* assumptions about the underlying densities of the data. The memory requirements and classification time were shown to be equivalent for both methods. Finally, the time for training was discussed. In this case, the GMLC takes less time than the BPNN; however, this is not considered to be a disadvantage because: the training can be performed off–line in our application; special purpose BPNN hardware exists for training and testing; and a variety of speed–up techniques are available for BP in software. From these results we feel that the BPNN is a better candidate for doing supervised characterization of remotely sensed data.

Recently, a new type of neural network called the probabilistic neural network (PNN) has been developed [Specht 90]. It uses the technique of Parzen windows for nonparametric density estimation and uses the technique of maximum likelihood estimation for classification. It offers the twin advantages of being available in hardware [Washburne 91] as well as being considerably quicker to train than BP. We will investigate the application of such classifiers to our problem. In addition we will research the use of ancillary data such as texture and spatial information to improve our classification accuracy.

We have mentioned the MasPar MP-1 as a parallel computer alternative in previous sections. It will be the focus of IDM to implement parallel code for the BPNN as well as the GMLC, thus providing fast alternatives to the remote sensing researcher.

6 Acknowledgements

A number of people within and without the IDM group have contributed to this work. The authors would like to thank William Campbell, Erik Dorfman, George Fekete, Co Horgan and Nicholas Short, Jr. for their input into the content of this paper.

References

[Anderson 76] J. R. Anderson, E. E. Hardy, J. T. Roach, and R. E. Witmer. A land use and land cover classification system for use with remote sensor data. Geological Survey Professional Paper 964, United States Government Printing Office, Washington, D.C., 1976.

[Andrews 72] H. C. Andrews. *Introduction to mathematical techniques in pattern recognition.* Wiley–Interscience, New York, 1972.

[Birmingham 91] M. Birmingham. Acceleration of neural networking through the use of integer approximations for floating point operations. IDM memo 13, NASA, Intelligent Data Management, Code 934, Greenbelt, Maryland 20771, 1991.

[Benediktsson 90] J. A. Benediktsson, P. H. Swain, and O. K. Ersoy. Neural network approaches versus statistical methods in classification of multisource remote sensing data. *IEEE Trans. on Geoscience and Remote Sensing*, 28(4):540–551, 1990.

[Campbell 87] J. B. Campbell. *Introduction to remote sensing.* Guilford Press, New York, 1987.

[Campbell 89] W. J. Campbell, R. F. Cromp, and S. E. Hill. Automatic labeling and characterization of objects using artificial neural networks. *Telematics and Informatics*, 6(3–4):259–271, 1989.

[Cromp 91] R. F. Cromp. Automated extraction of metadata from remotely sensed satellite imagery. In *Technical Papers, 1991 ACSM-ASPRS Annual Convention, Volume 3*, pages 91–101. ASCM/ASPRS, 1991.

[Hertz 91] J. Hertz, A. Krogh, and Palmer R. *Introduction to the theory of neural computation.* Addison–Wesley, Redwood City, California, 1991.

[Hecht-Nielsen 90] R. Hecht-Nielsen. *Neurocomputing.* Addison-Wesley, Reading, Massachusetts, 1990.

[Kanal 68] L. Kanal and B. Chandrasekaran. On dimensionality and sample size in statistical pattern recognition. In *Proc. Nat. Electron. Conf.*, pages 2–7, 1968.

[Muhlenbein 90] H. Muhlenbein. Limitations of multi–layer perceptron networks – steps towards genetic neural networks. *Parallel Computing*, 14:249–260, 1990.

[Richards 86] J. A. Richards. *Remote sensing digital image analysis, an introduction.* Springer–Verlag, Berlin, 1986.

[Specht 90] D. Specht. Probabilistic neural networks. *Neural Networks*, 3:109–118, 1990.

[Wan 90] E. A. Wan. Neural network classification: A bayesian interpretation. *IEEE Trans. on Neural Networks*, 1(4):303–305, 1990.

[Weiss 89] S. M. Weiss and I. Kapouleas. An empirical comparison of pattern recognition, neural nets, and machine learning classification methods. In *Eleventh Int. Joint Conference on Artificial Intelligence*, pages 781–787. American Association of Artificial Intelligence, 1989.

[Washburne 91] T.P. Washburne, M. M. Okamura, D. F. Specht, and W. A. Fisher. The Lockheed probabilistic neural network processor. In *International joint conference on neural networks, volume I*, pages 513–518. Institute of Electrical and Electronics Engineers, 1991.