A Detailed Comparison of Backpropagation Neural Network and Maximum-Likelihood Classifiers for Urban Land Use Classification

Justin D. Paola and Robert A. Schowengerdt

***Abstract*—A detailed comparison of the backpropagation neural network and maximum-likelihood classifiers for urban land use classification is presented in this paper. Landsat Thematic Mapper images of Tucson, Arizona, and Oakland, California, were used for this comparison. For the Tucson image, the percentage of matching pixels in the two classification maps was only 64.5%, while for the Oakland image it was 83.3%. Although the test site accuracies of the two Tucson maps were similar, the map produced by the neural network was visually more accurate; this difference is explained by examining class regions and density plots in the decision space and the continuous likelihood values produced by both classifiers. For the Oakland scene, the two maps were visually and numerically similar, although the neural network was superior in suppression of mixed pixel classification errors. From this analysis, we conclude that the neural network is more robust to training site heterogeneity and the use of class labels for land use that are mixtures of land cover spectral signatures. The differences between the two algorithms may be viewed, in part, as the differences between nonparametric (neural network) and parametric (maximum-likelihood) classifiers. Computationally, the backpropagation neural network is at a serious disadvantage to maximum-likelihood, taking nearly an order of magnitude more computing time when implemented on a serial workstation.**

I. Introduction

The remote sensing literature on backpropagation neural network applications to *multispectral image classification* is relatively new, dating back only about six years. The first studies established the feasibility of the method [2], [17], [23], [29]. Subsequent studies examined the classifier in more detail and compared it to standard techniques such as maximum-likelihood. Some researchers found the statistical classifiers to be superior [3], [6], while a majority found that the network produced similar or superior classifications [4], [13], [16], [19], [20], [22], [31].

Few studies, however, have looked at the finer details of class decision regions and classifier–produced probability estimates in order to more fully understand *how* and *why* the two algorithms perform differently on a particular image. This is exactly what we address in this paper. Hopefully this analysis, although limited to two scenes, will provide some insight on applications of neural networks in multispectral image classification. To address our goal of a consistent comparison, we have not included any of the numerous network variations intended to improve training speed or classifier accuracy [3], [8], [9], [15], [22]. We have used a single basic network configuration that is often the starting point for variation.

A. Backpropagation Neural Networks

Many variants of neural network algorithms derive from the *three layer backpropagation* neural network. For multispectral image classification, the most widely used input/output configuration is one input node for each input channel (typically each band of a multispectral image) and one output node for each desired class label. The size of the intermediate, or *hidden* layer, is not determinate, and few guidelines exist to help the user. Every input and output node is connected to all of the hidden layer nodes. Each interconnection has an associated weight and as a whole contain (after training) the distributed, learned information about the classes.

The basic element of a neural network is the *processing node*, which performs two functions. First it sums the values of its inputs. This sum is then passed through an arbitrary *activation function* to produce the node's output value. For the backpropagation training algorithm, the activation function must be differentiable. The most common form is the sigmoid function, defined as

$$f(\mathrm{NET}) = \frac{1}{1 - e^{-\mathrm{NET}}} \tag{1}$$

where NET is the sum of weighted inputs to the processing node.

For the training stage of supervised pattern recognition, the network weights are adjusted in an iterative, gradient descent training procedure called *backpropagation* [12], [14], [26], [30]. The training data consists of a pair of data vectors. The input data vector is the pattern to be learned and the output vector is the desired set of output values to be produced by the network upon recall of that training pattern. The goal of the training is to minimize the overall error between the desired

Manuscript received July 22, 1994; revised January 26, 1995. This research was supported in part by NASA under contract NAS 2-13721 to the Universities Space Research Association and in part under High Performance Computing and Communications grant NAG 5-2198 to the University of Arizona.

The authors are with the Department of Electrical and Computer Engineering, University of Arizona, Tucson, AZ 85721 USA.

IEEE Log Number 9412590.

and actual outputs of the network. In order to guarantee a decrease in error, the incremental adjustments in the weights at each iteration must be infinitesimal. In order to achieve a realistic training time, a *learning rate* parameter, which represents the percentage of the step taken towards minimum error must be specified. If this quantity is too small, training will take too long, and if it is too large, the gradient descent will degenerate and the error will increase.

Backpropagation, like all gradient descent algorithms, is not guaranteed to find the global minimum error. During the training phase, the network takes the steepest descent from the current position to one of lower error [5]. If the network encounters a valley, or local minimum, in the error space, it can become stuck and the error will not decrease to the global minimum value. It is also possible for the system to oscillate between two points. One way to alleviate these problems is to add some fraction of the weight change calculated in the previous iteration to the weight update formula [26]. The added push from this term can keep the network from becoming stuck in local minima during training. The *momentum parameter*, like the learning rate, is set at the beginning of the training and must be determined experimentally.

Training of the network results in the formation of decision boundaries in the feature space. Reference [21] provides an excellent discussion of the dependence of the decision regions on the number of network layers and nodes per layer. Reference [21] shows that for a network with threshold activation functions, a three layer configuration can produce any convex region in the feature space. Another way to view the operation of the network is as a nonlinear transformation of the feature space into a new space in which the data is linearly separable [26]. The nodes of the output layer perform a simple hyperplane resolution on the output space of the hidden layer.

Once training is complete, the network is used in a feed-forward mode like a hard-wired circuit to produce the classification. The entire image is fed into the net pixel-by-pixel, and a simple metric (such as the maximum) is used to process the network output to make a class selection for each pixel. (For a discussion of the use of neural networks for multispectral image classification and a survey of previous work see [27]).

B. Maximum-Likelihood Classification

The maximum-likelihood classifier is a parametric classifier that relies on the second-order statistics of a Gaussian probability density function (pdf) model for each class. It is often used as a reference for classifier comparison because, if the class pdf's are indeed Gaussian, it is the optimal classifier. The basic *discriminant function* for each class is

$$\begin{aligned} g_i(X) &= p(X|w_i)p(w_i) \\ &= \frac{p(w_i)}{(2\pi)^{n/2}|\Sigma_i|^{1/2}} \cdot e^{-(1/2)(X-U_i)^T\Sigma_i^{-1}(X-U_i)} \end{aligned} \tag{2}$$

where n is the number of bands, X is the data vector, U_i is the mean vector of class i, and Σ_i is the covariance matrix of class i,

$$X = \begin{bmatrix} x_i \\ x_i \\ \vdots \\ x_n \end{bmatrix} \quad U_i = \begin{bmatrix} \mu_{i1} \\ \mu_{i2} \\ \vdots \\ \mu_{in} \end{bmatrix}$$

$$\Sigma_i = \begin{bmatrix} \sigma_{i11} & \sigma_{i12} & \cdots & \sigma_{i1n} \\ \sigma_{i21} & \sigma_{i22} & \cdots & \sigma_{i2n} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{in1} & \sigma_{in2} & \cdots & \sigma_{inn} \end{bmatrix}. \tag{3}$$

The values in the mean vector, U_i, and the covariance matrix, Σ_i, are estimated from the training data by the unbiased estimators

$$\hat{\mu}_{ij} = \frac{1}{P_i} \sum_{l=1}^{P_i} x_{jl} \qquad j = 1, 2, \cdots, n, \tag{4}$$

and

$$\hat{\sigma}_{ijk} = \frac{1}{P_i - 1} \sum_{l=1}^{P_i} (x_{jl} - \hat{\mu}_{ij})(x_{kl} - \hat{\mu}_{ik}) \qquad j = 1, 2, \cdots, n; k = 1, 2, \cdots, n \tag{5}$$

where P_i is the number of training patterns in class i. Note that in order for the inverse of the covariance matrix to be calculated, P_i must be at least one greater than the number of image bands. Equation (2) can be reduced by taking the natural log and discarding the constant π term to

$$\begin{aligned} g_i(X) = {}& \log_e p(w_i) - \tfrac{1}{2} \log_e |\Sigma_i| \\ & - \tfrac{1}{2}(X - U_i)^T \cdot \Sigma_i^{-1}(X - U_i). \end{aligned} \tag{6}$$

If the *a priori* probabilities are assumed to be equal, the first term is a constant and can be ignored. The second term is a constant for each class. This leaves only the third term to be calculated for each pixel during classification. The discriminant $g_i(X)$ is calculated for each class and the class with the highest value is selected for the final classification map.

II. Configuring the Network

In this study, the standard backpropagation training routine, with an *adaptive* learning rate and momentum, was used. The particular adaptive algorithm was defined heuristically and has proven robust in our experience. The original idea was to keep the learning rate at a level just below the point at which it causes instability. This leads to faster, yet stable, training. After a user-defined number of training cycles, the mean square error is compared with that of the previous cycle. If the error has increased, the learning rate and momentum are halved. If the error has decreased, the learning rate and momentum are increased by 20%. This allows for accelerated convergence when the error is steadily decreasing.

For input to the network, the pixel data was scaled to the range zero to one. For training, an output of 0.9 was used to represent the correct class and an output of 0.1 was used to represent all other classes. The range of initial random weight values was chosen arbitrarily to be the interval

−0.1—−0.0001 and +0.0001—+0.1 (avoiding values too close to zero). The learning rate and momentum were 0.001 and 0.00005, respectively, with adaptation every four training cycles.

With these values set, the remaining parameters are the number of hidden layers and number of nodes per hidden layer. It was assumed that a 3–layer neural network (one hidden layer) would be sufficient for this type of classification and would provide similar discrimination abilities as the maximum-likelihood classifier. Thus, the sole parameter to be considered was the number of nodes in the single hidden layer. If we view both the neural network and maximum-likelihood classifiers as decision making functions defined by a number of parameters (degrees of freedom), then it is logical to use the same number of parameters in each for comparison. For a three layer network, assuming one input node per image band and one output node per class, the number of parameters as a function of network structure is

$$\begin{aligned} N_{\text{Net}} &= 3 + \#\ \text{weights} \\ &= 3 + \text{hidden layer nodes} \cdot (\text{bands} + \text{classes}) \end{aligned} \tag{7}$$

where the first (constant) term is the number of parameters needed to specify the size of each of the three layers.

For the maximum-likelihood classifier, the corresponding number of parameters is

$$\begin{aligned} N_{\text{ML}} &= 2 + \#\ \text{means} + \#\ \text{unique convariances} \\ &= 2 + \text{classes} \cdot \text{bands} + \tfrac{1}{2} \cdot \text{classes} \cdot (\text{bands}^2 + \text{bands}). \end{aligned} \tag{8}$$

Note that the number of parameters for the maximum-likelihood method is quadratic with respect to the number of bands, while the number of parameters for the neural network method will depend on the choice of hidden layer size, which is not necessarily dependent on the number of bands. These dependencies may be important with certain types of data, such as hyperspectral imagery.

Setting N_{NET} equal to N_{ML} and solving for the number of hidden layer nodes, we obtain

$$\#\ \text{hidden layer nodes} = \frac{2 + \text{classes} \cdot \text{bands} + \frac{1}{2} \cdot \text{classes} \cdot (\text{bands}^2 + \text{bands}) - 3}{\text{bands} + \text{classes}}. \tag{9}$$

Substituting the number of classes (12) and the number of bands (6) in our experiment, a value of 17.94 is obtained for the hidden layer size. Thus, 18 hidden layer nodes were used for the neural network.

III. Data Description

A Landsat Thematic Mapper (TM) image of Tucson, AZ was selected for the primary comparison between the neural network and maximum-likelihood classification methods. Familiarity with the area allowed for accurate class training and test site identification. The Apr. 1, 1987 scene had been georeferenced prior to the work on this project. A 900 × 900 portion of the six nonthermal bands was used for the classifications; band 4 is shown in the top portion of Fig. 1. Band 4 of an Oakland, CA TM image from Aug. 12, 1983, is also shown in the figure. This image was used for further comparisons.

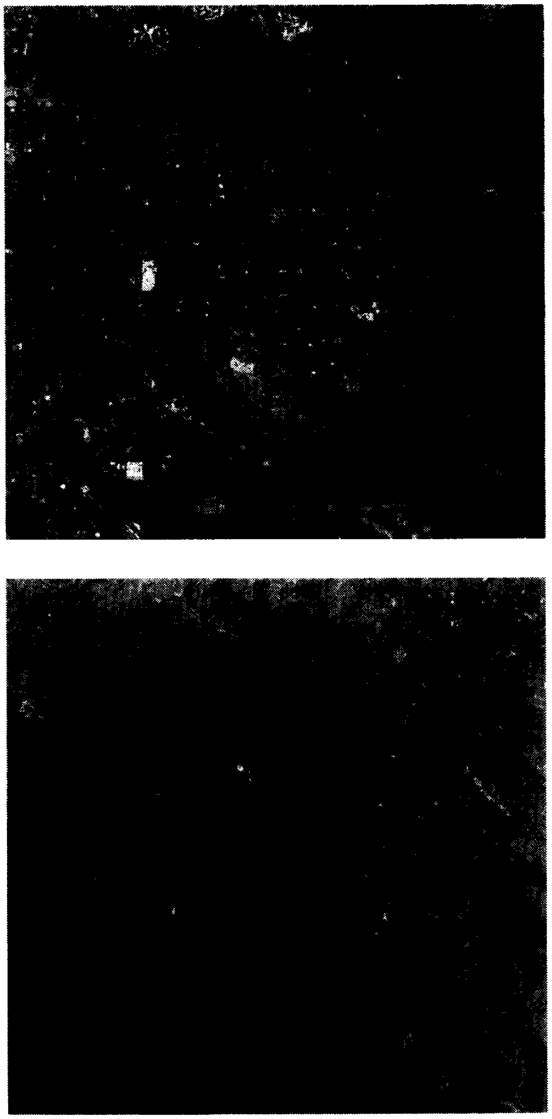


Fig. 1. Band 4 (near infrared) of the Landsat Thematic Mapper images of Tucson, AZ, and Oakland, CA used to compare the maximum-likelihood and neural network classifiers.

After some experimentation, it was determined that 12 classes covered the majority of urban land use features in the Tucson image (Table I). The labels used were similar to the Level I and II categories defined in [1]. These classes are meaningful to photointerpreters and land use mappers, but *are not necessarily spectrally homogeneous.* First a set of similar-sized training regions were defined by visual interpretation of the image. The same training sites were used by both the maximum-likelihood and neural network classifiers. Ad-

TABLE I
CLASSES USED IN MAXIMUM-LIKELIHOOD AND NEURAL NETWORK TRAINING, ALONG WITH THE NUMBER OF PIXELS USED FOR THE TRAINING AND TESTING OF THE CLASSIFIERS, AND THE MEANS AND STANDARD DEVIATIONS OF THE CLASSES AS DEFINED BY THE TRAINING DATA FOR THE LANDSAT TM IMAGE OF TUCSON, AZ

				Class Means						Class Standard Deviations					
	Class	# of Train Pixels	# of Test Pixels	Band 1	Band 2	Band 3	Band 4	Band 5	Band 7	Band 1	Band 2	Band 3	Band 4	Band 5	Band 7
1	Tarnac	66	194	167.9	91.3	115.1	97.6	168.9	111.4	5.5	4.2	5.8	5.1	10.4	6.9
2	Building	54	65	233.5	119.0	139.2	114.9	191.2	126.3	24.9	18.0	21.3	17.3	25.9	23.7
3	Grass	63	122	71.4	33.8	32.5	127.4	103.1	35.6	6.9	4.4	7.2	8.9	10.5	7.3
4	Foothills Natural Vegetation	81	298	92.4	46.0	55.4	66.3	114.3	60.2	6.5	4.5	6.5	6.3	11.2	6.5
5	Sand	69	80	138.8	75.7	99.4	95.7	171.7	106.2	8.8	5.7	8.1	7.8	17.6	12.9
6	Desert Scrub	81	319	106.8	55.8	72.6	76.8	135.2	78.7	3.8	2.0	3.2	2.5	4.6	3.4
7	Bare Soil	76	130	146.5	85.1	118.5	111.0	182.6	109.7	23.5	13.4	16.0	12.4	14.1	10.0
8	Urban Residential	130	307	122.7	60.1	71.8	77.3	109.2	60.3	26.0	13.9	16.9	10.9	16.1	9.0
9	Asphalt	72	79	88.8	37.0	40.9	34.1	51.6	31.9	10.3	6.6	9.0	9.1	16.4	10.3
10	Riparian Vegetation	81	288	69.3	28.6	30.7	53.3	87.7	42.1	4.4	3.2	5.3	6.2	13.9	8.0
11	Dense Urban	81	226	137.2	66.6	78.6	67.0	104.9	66.6	29.3	14.9	17.4	14.4	26.4	16.9
12	Shaded Foothills Natural Vegetation	61	143	74.7	32.4	36.0	45.2	75.3	36.5	7.8	5.2	7.3	7.6	17.2	8.8

ditional patterns were added to the "urban residential" class after initial experimentation because of its high variability. This resulted in better performance for both classifiers. Table I shows the mean and standard deviation of each class as defined by the training samples.

Ideally, the ground truth at every pixel should be known to calculate the accuracy of the classification. Since this is impractical, a set of test regions were chosen using our knowledge of the area to estimate the overall accuracy. A few of the classes, such as "building," are sparsely distributed in the image. Therefore, in order to keep the classification accuracy calculations from being dominated by a few of the more prevalent classes, it was necessary to use very small test areas. Thus, only about 0.28% of the image pixels are used to evaluate the accuracy (0.11% are used to define the classes). Despite the small size of these regions, there is some heterogeneity within each, and a 100% overall accuracy would not be expected for any classifier.

IV. CLASSIFIER COMPARISON

The neural network and maximum-likelihood classifiers were implemented on a Sun SPARCstation 10 in the Digital Image Analysis Laboratory of the Department of Electrical and Computer Engineering at the University of Arizona. Both were written from scratch in the C language, using many of the same subroutines. This was done to provide a fair comparison of training and classification speed.

A. *Classification Accuracy*

The maximum-likelihood algorithm was applied to the six band Tucson image, producing the classification map in the upper left of Fig. 2. The accuracy of the training regions was 96.9% and the accuracy of the test regions was 89.5%. The training time (for the sake of comparison with the neural network method, this is considered to be the time of computing the class statistics) was negligible, while the classification time was 590 seconds.

Although the test site accuracy of the maximum-likelihood method is high, there are some major errors in the overall classification. The "urban residential" class is much more extensive than it should be. It is covering large areas at the expense of the more appropriate "foothills natural vegetation" (in the north and east) and "desert scrub" (in the south) classes. This is due to the fact that the "urban residential" class has a similar mean vector to these other classes, but a much higher standard deviation in each band (see Table I). It is really a mixed class consisting of the two competing vegetation classes as well as the "asphalt," "building," "grass," and "bare soil" classes. The other composite class, "dense urban," also has a high standard deviation and appears to take priority in some areas over other, more appropriate, but lower standard deviation classes.

The neural network classifier training stage was run for 50 000 iterations. The mean square error and test site classification accuracy are plotted versus iteration number in Fig. 3. Fig. 4 shows how the learning rate changes as a function of iteration number. It can be seen that the initial learning rate setting of 0.001 was inconsequential, as it was adjusted automatically to a more useful level soon after the beginning of training. The resulting classification map is shown in the middle left side of Fig. 2. The training site accuracy was 96.3% and the test site accuracy was 93.4%.

Fig. 3 indicates that the test site accuracy increases at nearly the same rate that the mean square error decreases—quickly at first, and then slowly, but steadily right through the end. This is encouraging because it links the minimization of mean square error of the neural network representation of the training patterns with the maximization of classification accuracy of areas not used in the training. This leads to two important conclusions. First, from an implementation standpoint, the representation of the image band values as single, scaled, floating point numbers in the range 0–1 is shown to be sufficient. Secondly, it shows that the network has enough generalization capability to extend what it has learned about the training patterns to the rest of the image. Had the test site

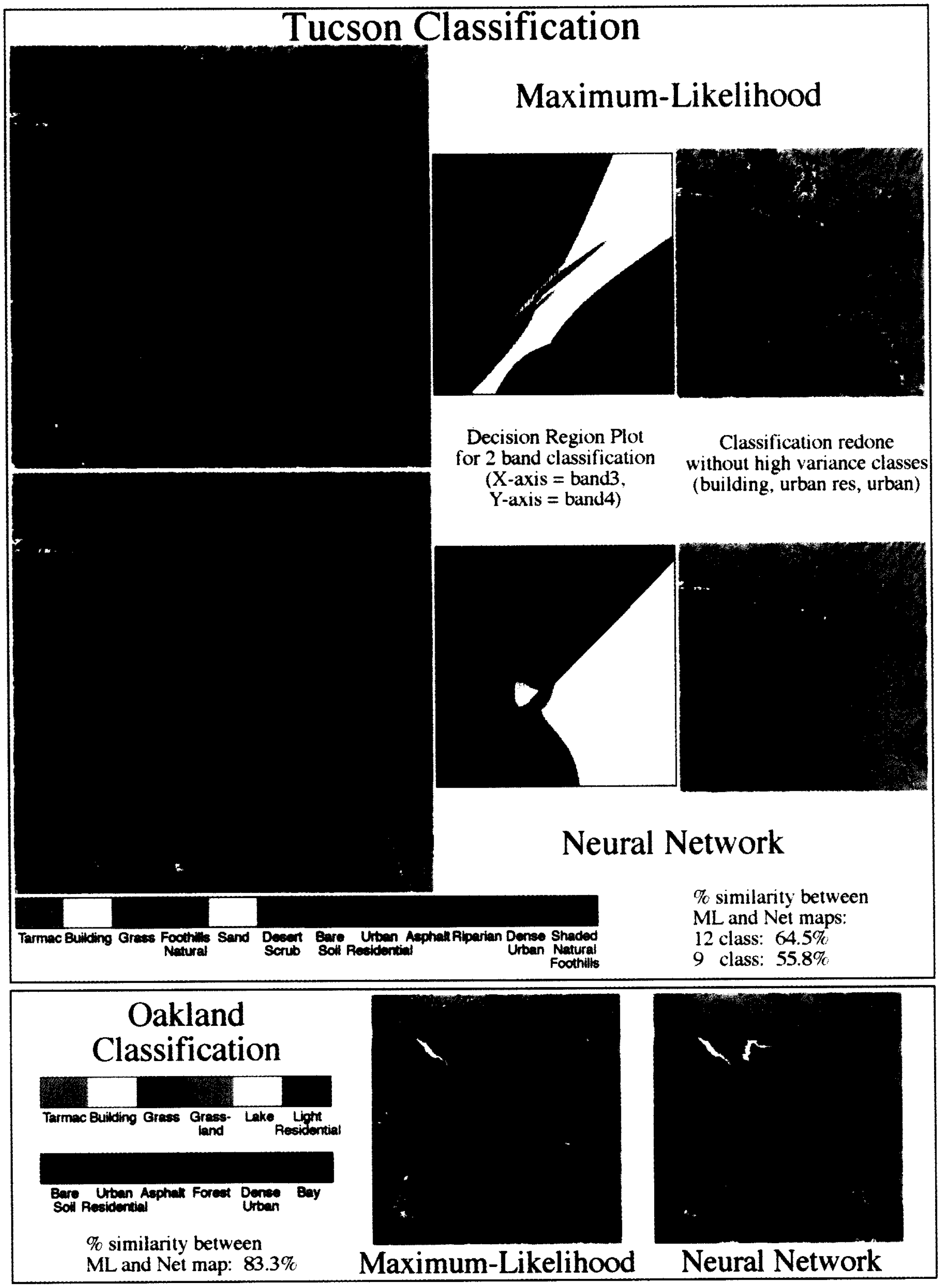

Tucson Classification
Maximum-Likelihood
Decision Region Plot for 2 band classification (X-axis = band3, Y-axis = band4)
Classification redone without high variance classes (building, urban res, urban)
Neural Network
Tarmac Building Grass Foothills Natural Sand Desert Scrub Bare Soil Urban Residential Asphalt Riparian Dense Urban Shaded Natural Foothills
% similarity between ML and Net maps:
12 class: 64.5%
9 class: 55.8%
Oakland Classification
Tarmac Building Grass Grass-land Lake Light Residential
Bare Soil Urban Residential Asphalt Forest Dense Urban Bay
% similarity between ML and Net map: 83.3%
Maximum-Likelihood
Neural Network

Fig. 2. Classification maps and decision region plots for the maximum-likelihood and backpropagation neural network classifiers. The upper left image is the maximum-likelihood classification map, produced from the six nonthermal Landsat TM bands of a scene of Tucson, AZ. The middle left image is the corresponding neural network classification map. To the right of each map is a plot of decision regions formed in the feature space for a two band classification (done so that the entire feature space can be visualized in 2-D). The x-axis of each plot is the pixel value in the red band, and the y-axis is the pixel value in the near infrared band. The origin of each plot is at the lower left. To the right of the decision region maps are the classifications redone with the high variance classes removed. The lower part of the image shows the maximum-likelihood and neural network classifications of the Oakland, CA, TM image. Unlike the Tucson case, these maps are very similar.

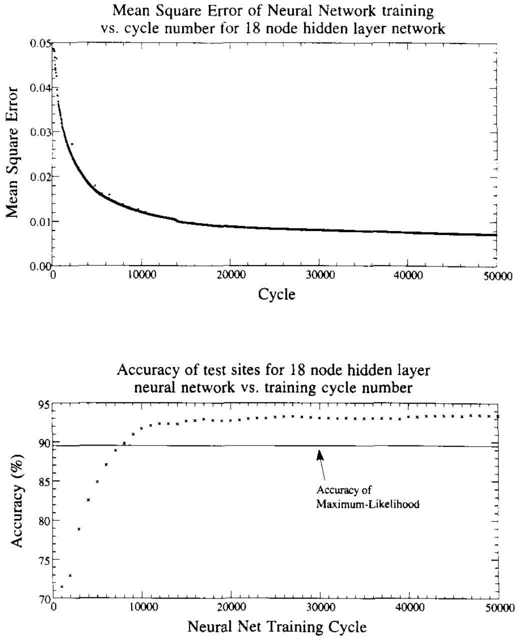


Fig. 3. Mean square error and test site accuracy for 50 000 iterations of neural network classifier training on the Tucson image. Mean square error is shown at 40 cycle intervals and test site accuracy at 1000 cycle intervals. The maximum-likelihood test site accuracy of 89.5% is shown for comparison.

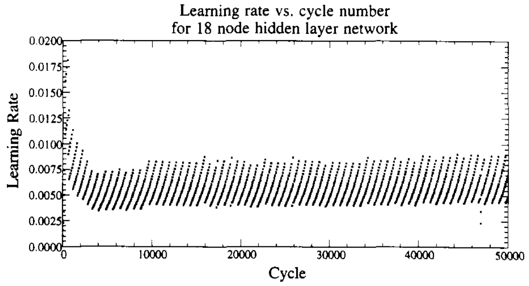


Fig. 4. Learning rate parameter for 50 000 iterations of neural network classifier training for the Tucson image. This plot illustrates the adaptive learning rate behavior implemented to decrease training time and ensure stability of training. The learning rate is shown at 40 cycle intervals.

accuracy not increased proportionally to the decrease in mean square error, then the network would have been specializing too much on the training patterns and would have been useless as a classifier, particularly for high variability classes.

The neural network achieves a higher test site accuracy than the maximum-likelihood method, but requires a much greater amount of time. As can be seen in Fig. 3, only 8000 training iterations were needed to equal the accuracy of maximum-likelihood. The time required to train for 8000 iterations, however, was 8577 seconds. The time required to produce the classification map was 385 seconds, about half that of maximum-likelihood. Thus, the overall classification time was about 15 times that of maximum-likelihood to produce the same test site accuracy. At the end of 50 000 cycles, the test site accuracy reached 93.4% and the accuracy of the training sites was 96.3%, slightly lower than the 96.9% reached by maximum-likelihood. At the equal test site accuracy point (8000 iterations), in fact, the network training site accuracy was only 94.8%. The closer test and training site accuracies indicate that the neural network has generalized better than the maximum-likelihood method. The second order statistics of the maximum-likelihood procedure describe the training sites, but do not apply as well to the test sites, which do not necessarily fit the assumed distribution of the training sites, even if they are normally distributed. The neural network's nonstatistical approach seems to have helped it discriminate these similar, but slightly differently distributed test sites. This may prove to be important when it comes to classifying multi-date imagery, or when extending the same classification to different images.

TABLE II

Confusion Matrices of the Test Site Results for the Maximum-Likelihood and Neural Network Classifications for the Tucson Image. The Top Class Numbers are the Actual Test Site Class and the Side Class Numbers are Those Determined by the Classification Methods. Thus, the Rows Show the Number of Pixels from Each Original Test Site that Make up the Final Classifier Class. The Columns Show the Distribution of the Original Test Site Pixels Among the Output Classes of the Classifiers. For Class Names See TABLE I

Confusion Matrix for Maximum-Likelihood:

	1	2	3	4	5	6	7	8	9	10	11	12
1:	151	0	0	0	0	0	0	0	0	0	0	0
2:	0	63	0	0	0	0	0	4	0	0	13	0
3:	0	0	122	0	0	0	0	0	0	0	0	0
4:	0	0	0	185	0	0	0	3	0	0	0	0
5:	0	0	0	0	78	0	2	1	0	0	0	0
6:	0	0	0	10	2	317	0	3	0	0	0	0
7:	0	0	0	0	0	1	128	0	0	0	0	0
8:	0	2	0	71	0	1	0	291	0	18	8	0
9:	0	0	0	0	0	0	0	0	74	0	6	3
10:	0	0	0	0	0	0	0	0	0	269	0	0
11:	43	0	0	0	0	0	0	5	5	0	199	2
12:	0	0	0	32	0	0	0	0	0	1	0	138

Overall Accuracy: 89.5%

Confusion Matrix for Neural Network:

	1	2	3	4	5	6	7	8	9	10	11	12
1:	191	0	0	0	0	0	0	0	0	0	1	0
2:	0	27	0	0	0	0	0	0	0	0	11	0
3:	0	0	122	0	0	0	0	0	0	0	0	0
4:	0	0	0	286	0	3	0	15	0	0	0	0
5:	2	0	0	0	76	0	8	3	0	0	0	0
6:	0	0	0	10	3	316	0	7	0	0	0	0
7:	0	21	0	0	1	0	122	0	0	0	0	0
8:	0	17	0	2	0	0	0	276	0	16	11	0
9:	0	0	0	0	0	0	0	0	78	0	4	6
10:	0	0	0	0	0	0	0	0	0	272	0	0
11:	1	0	0	0	0	0	0	6	1	0	199	0
12:	0	0	0	0	0	0	0	0	0	0	0	137

Overall Accuracy: 93.4%

In comparing the maximum-likelihood classification map with the neural network classification map (Fig. 2), it is apparent that there is more difference than the 4% difference in test site accuracy indicates. Only 64.5% of the pixels match in the two maps. This is also reflected in Table III-A, which shows a cross-classification matrix between the two classifiers. Qualitatively, the neural network output is the more accurate of the two. Most of the differences are in the "urban residential" and natural vegetation classes, for which the network was considerably more accurate. In the network map, the 'urban residential' class is confined to the proper true residential region, while the natural vegetation classes rightfully occupy areas the maximum-likelihood classifier mislabeled as "urban residential." Table III-A shows that one-third of the "urban residential" class in the maximum-likelihood map appears more accurately as "foothills natural vegetation" in the neural network map. The neural network is not as sensitive to high variance classes such as "urban residential." Most of the other classes have a very similar distribution in both maps, but there are minor differences in the smaller classes, and some of these differences are reflected in the test site confusion matrices (Table II), and the cross-classification matrix (Table III-A). The maximum-likelihood classifier did a poor job in delineating the tarmac (located just to the left and below the middle of the image). The network, however, performed more poorly on the "building" class. There are some significant differences in the distribution of the "bare soil" class, but without additional ground truth, a fair comparison could not be drawn in this case.

TABLE III
CROSS-CLASSIFICATION MATRIX OF MAXIMUM-LIKELIHOOD AND NEURAL NETWORK CLASSIFICATIONS OF (A) TUCSON AND (B) OAKLAND. THE VALUES ARE SHOWN AS FRACTIONS OF THE TOTAL NUMBER OF PIXELS IN THE IMAGE. *'S REPRESENT FRACTIONS SMALLER THAN 0.0005 (450 PIXELS). FOR CLASS NAMES, SEE TABLE I

		Maximum-Likelihood												
		1	2	3	4	5	6	7	8	9	10	11	12	Total (%)
	1:	**.001**	*	0	0	*	0	*	*	*	0	.002	0	0.4
	2:	0	**.002**	0	0	0	0	0	*	0	0	*	0	0.3
	3:	0	0	**.011**	0	0	0	0	*	0	*	*	0	1.1
	4:	0	0	.004	**.129**	0	.025	0	.151	0	.003	.005	.004	32.1
N	5:	*	*	0	.001	**.025**	*	.003	.002	0	0	.005	0	3.6
e	6:	0	*	*	.010	.013	**.085**	.002	.014	0	*	.017	*	14.2
t	7:	0	*	.001	*	.003	.001	**.009**	.003	0	*	.006	0	2.3
	8:	0	.001	.002	.002	.006	*	0	**.279**	*	*	.010	0	30.1
	9:	0	*	0	0	0	0	0	*	**.009**	0	.004	*	1.3
	10:	0	*	*	.003	0	0	0	.006	0	**.021**	*	.001	3.1
	11:	*	.001	0	0	.001	*	0	.008	.007	0	**.053**	0	7.0
	12:	0	0	*	.004	*	0	0	.009	*	*	.002	**.020**	3.5
Total (%):		0.1	0.5	1.9	15.2	5.0	11.2	1.4	47.7	1.7	2.5	10.5	2.4	

Cross-classification matrix for Tucson image

(a)

		Maximum-Likelihood												
		1	2	3	4	5	6	7	8	9	10	11	12	Total (%)
	1:	**.017**	.005	0	0	0	.001	*	.004	0	0	.007	0	3.5
	2:	0	**.004**	0	0	0	*	0	*	0	0	.001	0	0.5
	3:	0	0	**.006**	0	0	.002	0	0	0	*	0	0	0.9
	4:	*	.001	*	**.169**	0	.022	.001	.004	0	0	0	0	19.6
N	5:	0	*	0	0	**.008**	0	*	*	0	*	.007	.001	1.7
e	6:	0	*	.001	.010	0	**.231**	*	.006	0	.001	*	0	24.9
t	7:	*	.003	0	.003	0	.001	**.003**	.002	0	0	*	0	1.2
	8:	*	.001	0	*	0	.020	*	**.129**	0	0	.004	0	15.5
	9:	*	*	0	0	0	*	*	0	**.001**	0	.004	0	0.5
	10:	0	*	.002	*	0	.034	*	.001	0	**.161**	.001	0	19.8
	11:	.003	.003	0	0	0	*	*	.005	*	0	**.044**	0	5.5
	12:	0	.001	0	0	0	0	0	0	0	0	.006	**.059**	6.5
Total (%):		2.1	1.9	0.9	18.2	0.8	31.0	0.4	15.0	0.1	16.2	7.4	6.0	

Cross-classification matrix for Oakland image

(b)

The training site histograms indicated that some of the classes with a high variance are multimodal, arising from either within–class training site variability or mixtures of other classes, and are therefore improperly modeled as single Gaussians. To investigate the impact of this fact on the two classifiers, the classification was repeated without three of the high variance classes—"building," "urban residential," and "dense urban." The remaining nine classes were more spectrally pure. The resulting maps are shown on the right side of Fig. 2. It is clear that the problem with the natural vegetation has been corrected for the maximum-likelihood classifier. In fact, for these classes the maximum-likelihood map is now superior to that of the neural network. The "desert scrub" class is mostly confined to the southern, topographically flat, portion of the image, as intended. The foothills vegetation is more appropriately placed in the east and north parts of the image. The neural network output shows some mixing of these two classes in small areas. However, the maximum-likelihood classifier is now misclassifying large areas as "grass" (a threshold "unclassified" category could avoid this commission error). This is clearly an error since the "grass" training sites were defined on golf courses and are very different from the rest of the image. The neural network output for the city region in the center of the image is more realistic. The "grass" areas are unchanged from the original image and much of the "urban residential" class has been replaced by "asphalt," "tarmac" (concrete) and the natural vegetation classes, as expected.

The biggest disadvantage of the neural network method is the lengthy training time. As stated above, to achieve the same test site accuracy as maximum-likelihood, the network took 15 times as long. The focus of this investigation is the comparison of the network and maximum-likelihood methods, so a detailed analysis of network structure and optimization has been left out. It was discovered during our research, however, that networks with as few as six hidden layer nodes produced maps with test site accuracies higher than maximum-likelihood. This smaller net required about half the time to train as the 18 node case. Thus, through node optimization, the total classification time for the network can be reduced to about 8 times that of maximum-likelihood.

A sometimes overlooked characteristic of the neural network classifier is that its results are subject to random variation. Since the weights of the network are randomized prior to training, the algorithm takes a different path to minimum error in each training run, and the final results can be different. To see if this would effect the overall classification the training was redone 15 times to 20 000 iterations. The resulting classification maps looked nearly identical. The test site accuracies ranged from 91.6–93.4%, with a mean of 92.5% and a standard deviation of 0.84 percentage points. Thus, it was determined that although there is some variability in the internal representation of the classifier, this variability did not manifest itself in a significant way in the final classification maps.

B. A Second Urban Land Use Classification

To generalize our analysis, we produced a second urban land use classification of the Oakland/East Bay area in Northern California (lower image of Fig. 1). The resulting maps are shown at the bottom of Fig. 2. Most of the classes are the same as those in the Tucson image. The residential class was split into two to accommodate the clear differences between the more urban and more vegetated (in the hills) residential areas of the city. Also, two water classes, "bay" and "lake," were introduced. In this case, the similarity between the two maps was 83.3%, nearly 20 percentage points higher than for the Tucson image (see Table III-B for a breakdown of the cross-classification between the two classifiers). Fig. 5 shows a small portion of the classification in detail. The maximum-likelihood classification erred in the placement of reservoir pixels into the "dense urban" class. Another characteristic of the maximum-likelihood map, also illustrated in Fig. 5, was the prevalence of edge effects, particularly around the shores of the reservoirs, where the mixed pixels were classified into unlikely categories such as "dense urban." The neural network labeled and delineated the reservoirs correctly. Overall, however, both classifiers performed similarly and were qualitatively accurate for this image.

C. Decision Space Characteristics

In comparing the capabilities of the maximum-likelihood and neural network classifiers, it is useful to examine their

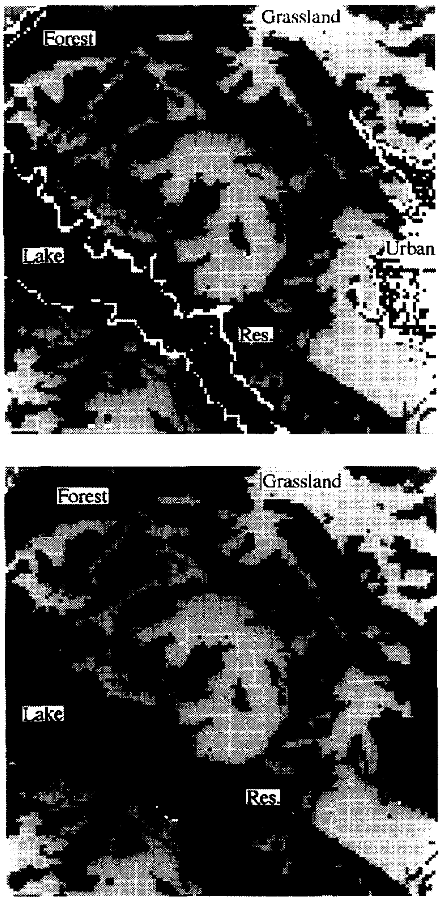


Fig. 5. Closeup of a portion of the Oakland classification (top image—maximum-likelihood, bottom image—neural network) showing the misclassification of the reservoir (lake) on the right as urban by the maximum-likelihood classifier. Also, the reservoir on the left shows the edge effects seen frequently in the maximum-likelihood classification. The neural network output has much less pronounced edge effects due to these mixed pixels.

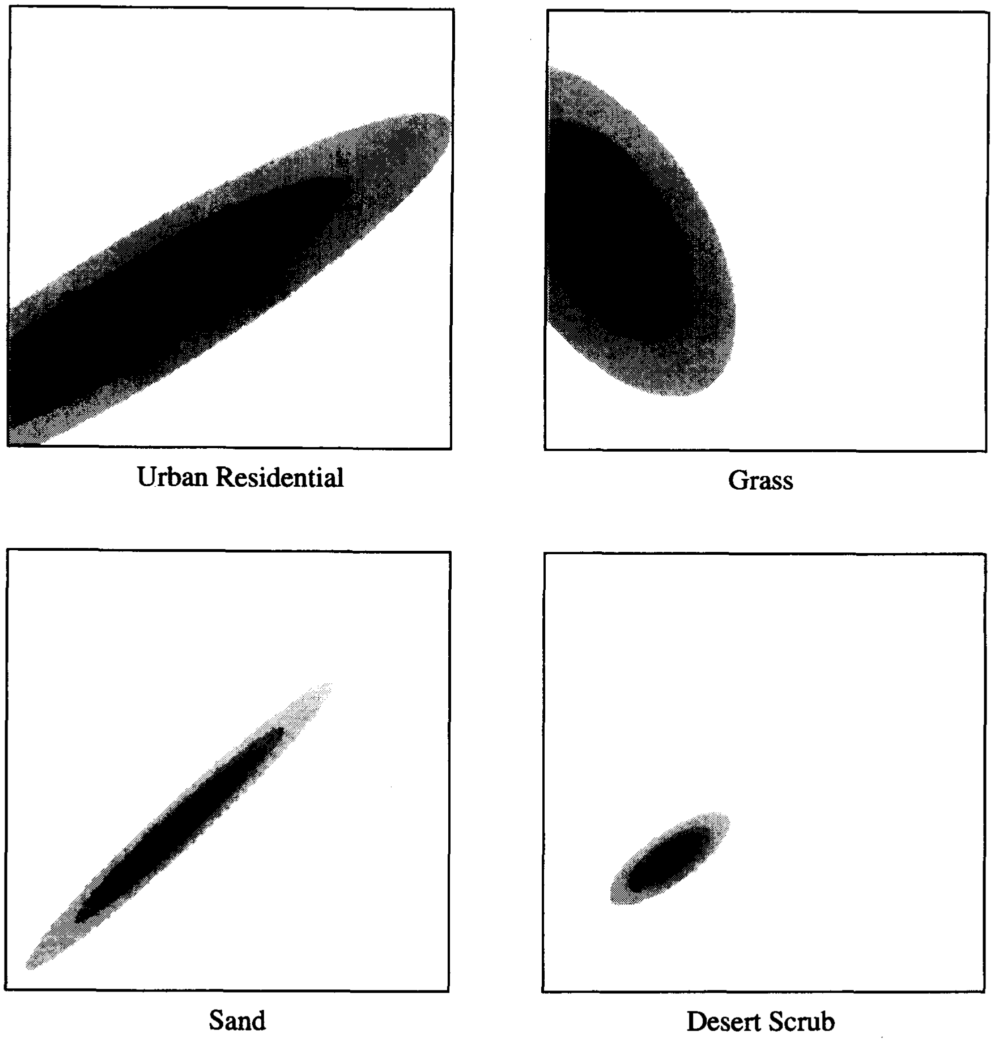
respective class density functions and decision regions. The original Tucson classification was redone with only two input bands, so that the entire decision space could be displayed in a 2-D decision region plot. Bands 3 and 4 were chosen since vegetation is easily separable from other classes in these two bands. As expected, the accuracies of both methods decreased due the smaller amount of spectral information. The maximum-likelihood test site accuracy was 77.8% and the neural network test site accuracy, after 25 000 training iterations, was 76.2%.

In the middle of Fig. 2 are the decision region plots for both methods. The values range from 0–255 from left to right in the x-axis (red band), and from 0–255 from bottom to top in the y-axis (near infrared band). In the maximum-likelihood case, the plot shows the highest surfaces of the intersecting probability density functions for each class. Fig. 6 shows the log of the Gaussian probability density (i.e., the discriminant function) plotted for four of the classes as calculated in the maximum-likelihood classifier. The same scaling is used in each plot, so that the relative heights of the Gaussians can be compared. This plot explains why the "grass" class appears in the lower right of the decision region plot in Fig. 2. The negatively correlated, high variance "grass" density is oriented nearly perpendicular to the other high variance class, "urban residential." Thus, its density actually surpasses that of the "urban residential" class in the lower right. However, the overall low values of these densities in the lower right of the plot are indications that the classification is not reliable in this region. In Fig. 7, a scattergram for the two image bands, shows the actual image pixel distribution on the same scales as in Figs. 2 and 6. It is apparent in this plot that the major differences between the maximum-likelihood and neural network decision regions, mostly located in the lower right, are not as significant as they appear, since few image pixels exist in that part of the feature space. And indeed if a threshold were applied, both algorithms would leave this part of the feature space unclassified.

As shown in Fig. 2 and Table III-A, the maximum-likelihood classifier placed too many pixels into the "urban residential" class, particularly at the expense of the "foothills natural" and "desert scrub" classes. The reason for this is apparent in the decision region plots of Fig. 2 and the log probability density plots of Fig. 6. Most of the classes are clustered near the diagonal of the two spectral bands. The density of the "urban residential" class, because of its high variance, is wider and has a lower peak than the densities of all the low variance classes in this cluster. Other classes are chosen only where their narrow probability density functions protrude above the wide "urban residential" density. Thus, for instance, the "urban residential" decision region surrounds the "desert scrub" region. The small variance of the "desert scrub" class, shown graphically in Fig. 6, constrains it to a very small decision region. The same problem occurs in the 6-dimensional case. Thus, the statistical assumption of the maximum-likelihood classifier is at the heart of the poorer performance in the classification of this image. It might be possible to improve this performance by decreasing the *a priori* probability of the "urban residential" class, thereby lowering its density function and, in comparison, raising those of the surrounding classes, but such "tuning" would be laborious and unrealistic in terms of the true prior probabilities. Also, as seen in the 9-class maps at the right of Fig. 2, removing these high variance classes results in better performance for some of the classes. However, we no longer have the desired urban land use map.

Values similar to the class probability densities can be obtained from the neural network classifier. In a complex classification problem with overlapping distributions, the error during backpropagation training never converges to zero and the network outputs for each class are continuous between 0 and 1. In [11], [28], it is shown that the network outputs can be estimates of the *a posteriori* probabilities of the classes. This will be explored in a later section.

The neural network outputs are shown in Fig. 8 as a function of the pixel value for comparison to the log probability densities of the maximum-likelihood classifier (Fig. 6). It



Urban Residential

Grass

Sand

Desert Scrub

Fig. 6. Natural log of the Gaussian probability density functions as calculated by the maximum-likelihood classifier for the Tucson image. These are feature space representations of the discriminant functions of the classes. The same normalization was used for each plot and they are shown with the same contour scheme. The axes are the same as the those of the decision region plots in Fig. 3—the red band is represented on the x-axis (0–255 from left to right) and the near infrared band is represented on the y-axis (0–255 from bottom to top). Since these were derived from a two band classification, the entire input feature space is represented.

is clear that these "probability" values (perhaps more appropriately, *likelihood* measures) are not constrained to any particular function. The function peaks generally match those of maximum-likelihood and the functions are mostly monotonic, but the similarity to Gaussian probability densities ends there. The maximum-likelihood classifier provides information about each class, from which a decision is made concerning most likely class membership. The training data for a particular class affects the statistics of the training class only. The neural network, because of its fully-interconnected nature, produces a fundamentally different type of classification. The warped appearance of the network class output functions is due to the competitive learning that takes place during backpropagation training. As one class is trained it suppresses the output value of all the other classes, and, in this way, the classes compete with one another for "territory" in the feature space. Thus, the network produces "probability densities" that have sharper inter-class transitions and do not overlap nearly as much as those of maximum-likelihood. These "densities" are warped around one another and fit together in a jigsaw fashion to form the final decision regions of the classifier (Fig. 2). Because of this, the network is able to avoid the problem of high variance classes. The lower variance vegetation classes can carve an adequate space out of the larger "urban residential" area through the competitive process, and the resulting classification is more accurate.

D. Reduced Training Set Size

One of the most difficult aspects of supervised classification is the selection of representative training sites for each class.

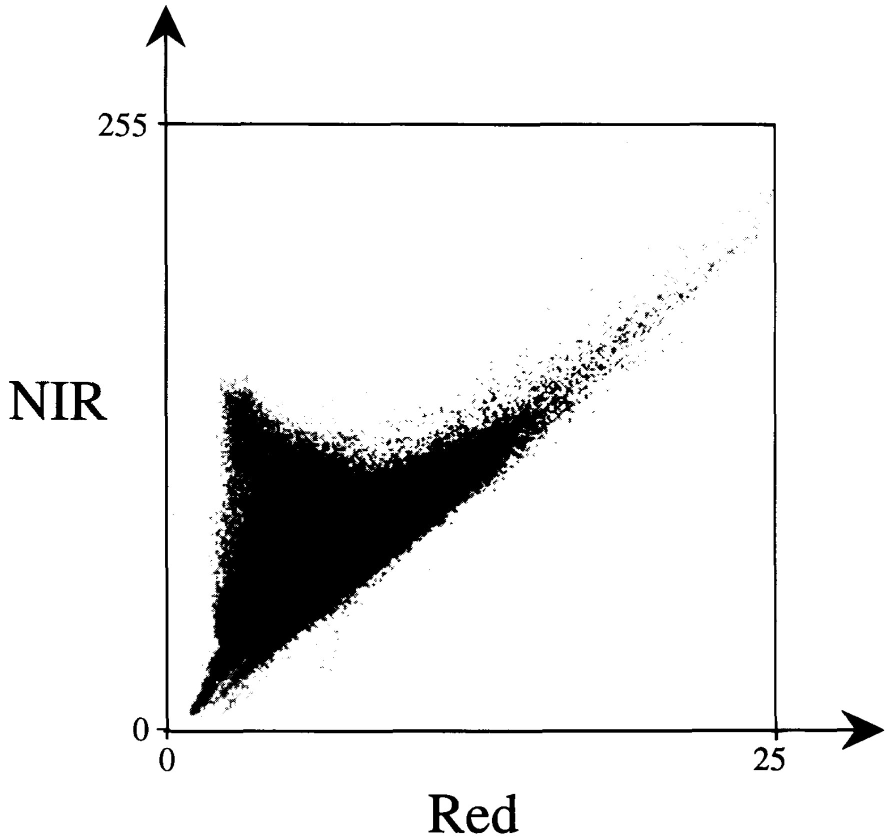


Fig. 7. Scattergram of bands 3 and 4 of the Tucson Landsat Thematic Mapper image. Band 3 (red) is represented on the x-axis and band 4 (NIR) is represented on the y-axis. This plot shows the image data distribution for the two band classification example. The axes have the same range as in Figs. 2, 6, and 8.

Much time and expense can be saved if fewer samples are required for the classifier. An additional motivation is the potential to reduce neural network training time.

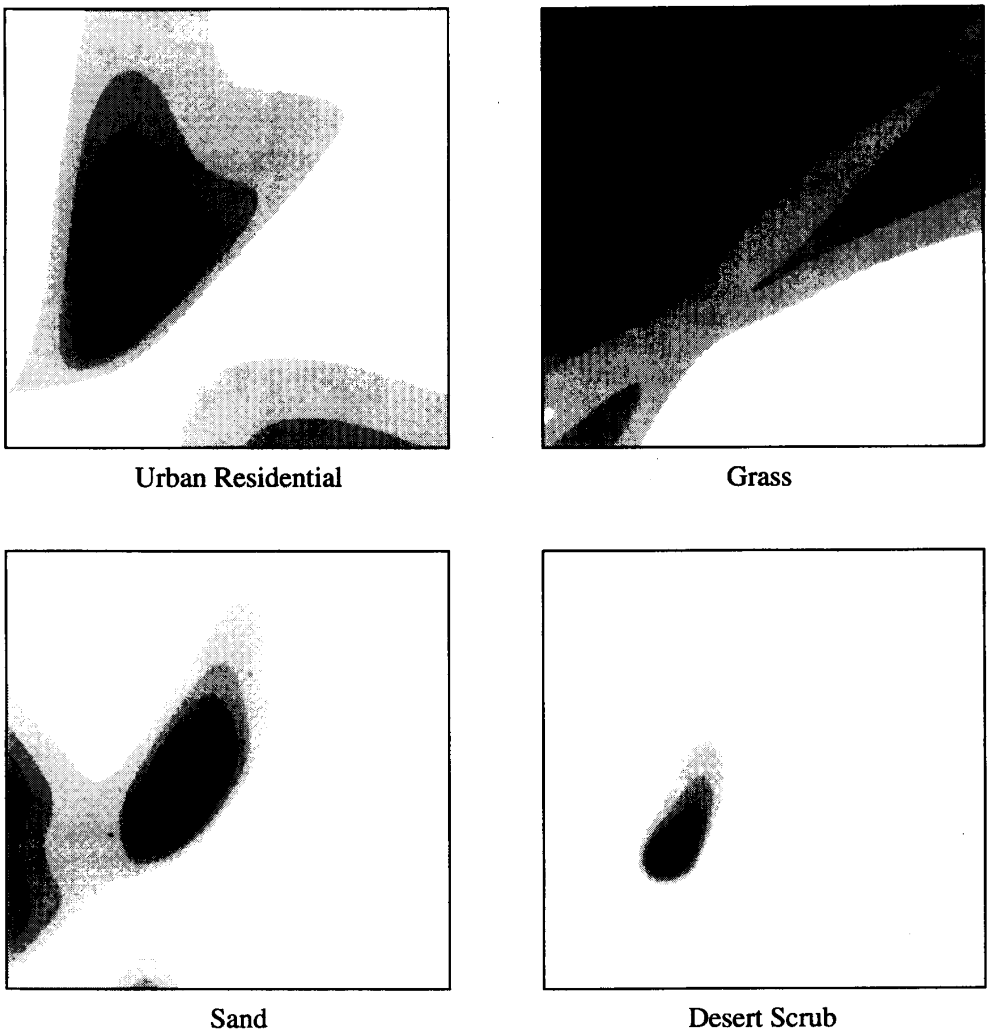
Two subsets of the original 915 pixel Tucson training data set were produced and applied to both the maximum-likelihood and neural network classifiers. The first subset is a reduction by one-third to 24 or 25 pixels per class, for a total of 297 training patterns. The second subset represents another 1/3 reduction, yielding 9 pixels per class, for a total of 108 training patterns. This is just above the minimum number of training patterns required to estimate the mean vector and covariance matrix for the maximum-likelihood method [32]. Table IV shows the results of the training and test site accuracy, and the overall classification time for both methods. The neural network was trained for 8000 iterations, which was the amount required for the network to achieve maximum-likelihood test site accuracy with the original 915 pixel training data.

The smaller training sets result in poorer performance for both classifiers. Table IV shows that for each 1/3 reduction in training set size, test site accuracies decrease about 15 percentage points for maximum-likelihood and 9 percentage points for the neural network. Since both classifiers require an adequate number of samples in each class to describe decision boundaries in the feature space, the decreased accuracy is expected.

TABLE IV
TRAINING AND TEST SITE ACCURACIES FOR MAXIMUM-LIKELIHOOD AND NEURAL NETWORK CLASSIFIERS FOR THE ORIGINAL AND TWO REDUCED TRAINING SAMPLE SETS FOR THE TUCSON IMAGE. THE NETWORK HAD 18 HIDDEN LAYER NODES AND WAS TRAINED FOR 8000 ITERATIONS. ALL TIMES ARE FOR A SINGLE USER PROCESS RUNNING ON THE DIGITAL IMAGE ANALYSIS LAB'S SUN SPARCSTATION 10

	Maximum-Likelihood	Neural Network
915 pattern set: Train Site Accuracy	99.0%	94.8%
Test Site Accuracy	89.5%	89.8%
Total Training and Classification Time	590 s	8962 s
297 pattern set: Train Site Accuracy	99.0%	95.6%
Test Site Accuracy	75.5%	81.3%
Total Training and Classification Time	590 s	3769 s
108 pattern set: Train Site Accuracy	100.0%	98.1%
Test Site Accuracy	60.6%	72.8%
Total Training and Classification Time	590 s	1625 s

It might seem intuitive, however, that the neural network would require more samples than maximum-likelihood since it assumes no statistical distribution and thus would need more information to define these decision regions. In general, nonparametric classifiers require more training samples to accurately describe the natural data distribution, since they start with no prior assumptions [18]. Reference [32], how-



Urban Residential

Grass

Sand

Desert Scrub

Fig. 8. Neural network output values for the two band classification of the Tucson image. These show the continuous distribution of output values for the network over the entire feature space. All plots are shown with the same contouring as the maximum-likelihood plots (Fig. 6) and have the same axes as Figs. 6 and 7. The red band is represented on the x-axis and the near infrared band is represented on the y-axis.

ever, makes the opposite claim that the maximum-likelihood assumption of a specific distribution requires more samples to accurately define the parameters of the distribution (e.g., the mean and covariance). The results from our data set seem to support this claim. A possible explanation for this seemingly contradictory observation is that the neural network is actually using *more* samples to define each class than maximum-likelihood. The argument is as follows. The maximum-likelihood classifier models each class by its own statistics, independent of all other classes. The neural network, on the other hand, considers all class training data when defining the "probability density" for each class. This is accomplished through the competitive training property discussed previously. When all the other classes are being trained, the output for a given class is set low (e.g., 0.1), thus, in effect, providing information about where that class *does not* exist in the feature space. This is used in addition to the training data for the class itself, which indicates where the class *does* exist. Thus the network is using the training data more efficiently than maximum-likelihood to determine the natural distributions of the classes and is more robust when faced with a decrease in training samples.

E. Class Probabilities and Mixing

Both the maximum-likelihood and neural network classifiers calculate values for every class at every pixel. The classification maps are produced by choosing the highest of those values. The magnitude of a given value represents a certainty measure for that class. In addition, the values of the other classes might be useful for identifying the existence and composition of mixed pixels—those that represent more than one ground cover class. At the very least this information can indicate the closeness of a decision between two classes and thus problems with the classification can be examined. In

[10], it is shown that the probability values produced during the maximum-likelihood classification provide a measure of classification quality and can be used to map gradients from one class to another. In [7] it is suggested that the fuzzy output values of a neural network likewise can be perceived as a certainty measure and used to enhance the classification. References [11], [28] bring the two methods together by showing that there is a relationship between many neural networks and minimum-error Bayesian classifiers. This work is summarized in [12]. For a large enough training set, and providing that the backpropagation algorithm does not get stuck in a local minimum, the classifier does in fact approximate the *a posteriori* class probabilities. The results presented here tend to substantiate this relationship.

There are two distinct, useful quantities that can be derived from the maximum-likelihood classification procedure. During classification, the log of the multivariate Gaussian probability density function is calculated for each class and the class with the highest value is chosen. If these values are normalized by the highest possible value of the function *over all classes*, a log probability density image can be created. The pixel values in this image represent an absolute (i.e., independent of other classes) confidence measure for class membership, which allows direct comparison of likelihoods among all classes for a given pixel. The second quantity is the *a posteriori* probability. This can be determined by using Bayes theorem on the discriminant values calculated during classification. The resulting 0–1 values can be scaled to the range 0–255 for display. Recall the maximum-likelihood decision rule

$$\text{Decide input vector } X \in w_i \text{ if and only if}$$
$$p(X|w_i)p(w_i) \geq p(X|w_j)p(w_j), \qquad \text{for all } j \neq i.$$

Using Bayes theorem, the probability of class i given the input vector X (i.e., the *a posteriori* probability) is

$$p(w_i|X) = \frac{p(X|w_i)p(w_i)}{p(X)} \tag{10}$$

where

$$p(X) = \sum_j p(X|w_j)p(w_j). \tag{11}$$

Since the *a priori* values, $p(w_i)$, are assumed to be equal, the probability for a given class is simply the discriminant value divided by the sum of the discriminant values for all the classes. In contrast to the probability density value, which provides a measure of the confidence of membership to a single class, the *a posteriori* probability quantifies the relative probability of a pixel belonging to a class on the assumption that it actually belongs to one of the classes.

Fig. 9 shows the maximum-likelihood log probability density maps of the original Tucson image for the classes "urban residential" and "riparian." Fig. 10 shows the associated *a posteriori* probability images for these classes. The low contrast of the log probability density maps is due to the normalization factor, chosen to maintain the relative likelihood among the classes. Each image was normalized by the highest density value over all classes as described above, and the resulting 0–1 value was scaled from 0–255. It can be seen in the *a posteriori* probability maps, however, that the resulting classification was fairly confident for most pixels. Values in these plots near the middle of the grayscale range would indicate a close choice between two or more classes. As expected, the "riparian" class was easily separable, while the "urban residential" class had a higher rate of close calls. The latter is due to the high variance of the class and the proximity of its mean to those of several other classes (see Figs. 2 and 6).

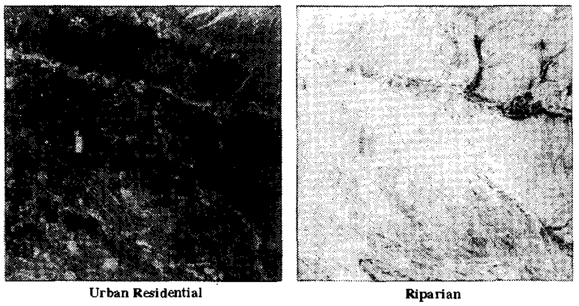


Fig. 9. Maximum-likelihood log probability density maps, with relative normalization. These are the discriminant values used to make a class selection. Darker pixels represent higher likelihood.

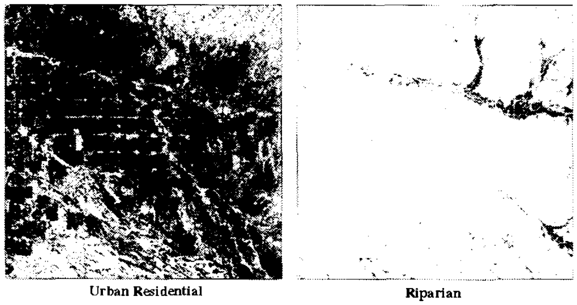


Fig. 10. Maximum-likelihood *a posteriori* probability maps. Darker pixels represent higher probabilities.

The neural network outputs for each class appear to be intermediate between the log probability density and *a posteriori* probability values of maximum-likelihood. To produce network "probability" maps the output of the desired class is simply scaled from its 0–1 range to a 0–255 range and displayed. Fig. 11 shows the results for the classes "urban residential" and "riparian" for the neural network after 8000 iterations (same test site accuracy as maximum-likelihood). Unfortunately, the value of the neural network output for each class has no precise mathematical meaning. Reference [28], while showing that there is a relationship between these outputs and the *a posteriori* probabilities derived from Bayesian classifiers in some cases, indicates that this relationship is poorly understood. Thus, the only way to examine the significance of these results is by comparing them to the ground truth for the area.

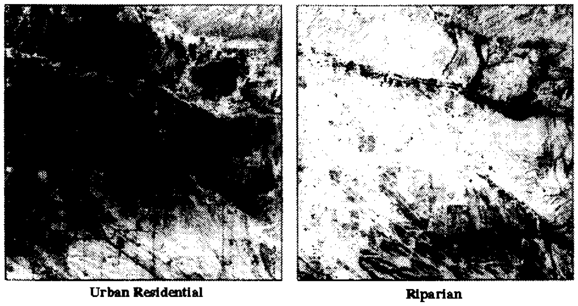
A detailed pixel mixture investigation was beyond the scope of this research, but some meaningful inferences can be made by comparing neural net and maximum-likelihood probability maps and by examining a small area in detail with the help of an aerial photo of part of the Tucson image.

Recall that each output is trained to be 0.9 if it matches the input training data and 0.1 otherwise. During classification, the outputs produce values ranging from 0–1 due to the sigmoid activation functions in the output layer nodes. The goal is to understand the significance of these fuzzy, continuous outputs and see if they can provide any information beyond the hard classification presented previously. The two classes shown in the figures were included because they indicate some differences between the maximum-likelihood and neural network results. The network, because of its mutually exclusive training (see Section IV-C), produces more separated classes. This is indicated in the "probability" maps by a sharper delineation between the areas labeled as the class of interest and the areas labeled as other classes. This property results in more homogeneous class regions and resistance to edge effects, as seen in the Oakland image (Fig. 5).

The maximum-likelihood classifier had the most trouble with the "urban residential" class, choosing it over the natural vegetation classes incorrectly for large areas. Examination of the *a posteriori* probability map (Fig. 10) shows that the statistical classifier was confident, but incorrect, in these areas at the top left and lower right of the image. The network map for this class shows the great difference between the two methods. The network has low values for the "urban residential" class in the incorrectly classified areas of the maximum-likelihood map. Additionally, examination of the upper portion of the image reveals the network picking out individual houses or groups of houses amongst the Tucson foothills as part of the "urban residential" class while giving immediately neighboring areas lower values. During neural network training, these built-up areas were correctly placed into the "urban residential" class just as with maximum-likelihood. However, the surrounding areas, which more closely resemble the spectrally similar natural vegetation were placed in those classes, and the "urban residential" class values were suppressed (i.e., set to 0.1). The mutually exclusive training of the neural network has allowed it to pick out these similar classes with greater accuracy since it is not assuming a broad distribution for the "urban residential" class.

The second comparison is for the "riparian" class. Examination of the class probability maps of both methods reveals a wealth of detail on both large and small washes coming out of the Santa Catalina (top) and Rincon (right) mountains. Note particularly the fine detail in the wash at the bottom, 1/3 from the right and the abundance of canyon streams and washes detected in the Santa Catalina mountains at the top right. These were found to correspond well with the canyon bottoms in the original image and topographic maps. The neural network does a slightly better job at separating these regions from clearly nonriparian areas, such as the city itself. Again, this might be due to the mutually exclusive training property of the neural network, in that even other natural vegetation areas are suppressed during the training for this class, resulting in the high contrast seen in Fig. 11.

An aerial photograph of part of Tucson, taken near the time of the satellite image, allows for a more detailed analysis of the class values for a small, well-characterized region. Fig. 12 shows a portion of this image, obtained in December of 1986, depicting Reid park and the Randolph golf course in central Tucson. This area is located to the left of the center of the satellite image (Fig. 1). Several of the defined classes should be present in this area to some degree or another. Fig. 13 depicts the maximum-likelihood *a posteriori* probability and probability density maps and the neural network class "probability" maps side-by-side for several of these classes. The 70 x 70 pixel area shown in the figure corresponds to the aerial photo image in Fig. 12. In Fig. 13, the maximum-likelihood log probability densities have been normalized *individually for each class*, rather than relatively, as in Fig. 9. This was done to allow more consistent comparison with the neural network output, which is inherently normalized the same way between 0 and 1 for each class.


Urban Residential

Riparian

Fig. 11. Neural network class output maps. Darker pixels represent higher outputs.

Fig. 12. Aerial photograph taken in December 1986, about 4 months before the Landsat TM image was acquired. This portion depicts Reid park and the Randolph golf course in central Tucson.

The fairways of the Randolph golf course provide a good test area for the measurement of class mixing. The fairways are separated by rows of trees, which should classify into the "riparian" class, since the "riparian" training area is the only one characterized by large trees. The maximum-likelihood probability density maps show "grass" dominating the golf course region with a few areas of higher likelihood for the "riparian" class, some of which match the rows of trees. The neural network output, on the other hand, preserves much of the detail of the fairways in the "grass" class, while more strongly delineating many of the tree rows in the "riparian" class. Both classifiers seem capable of showing both the

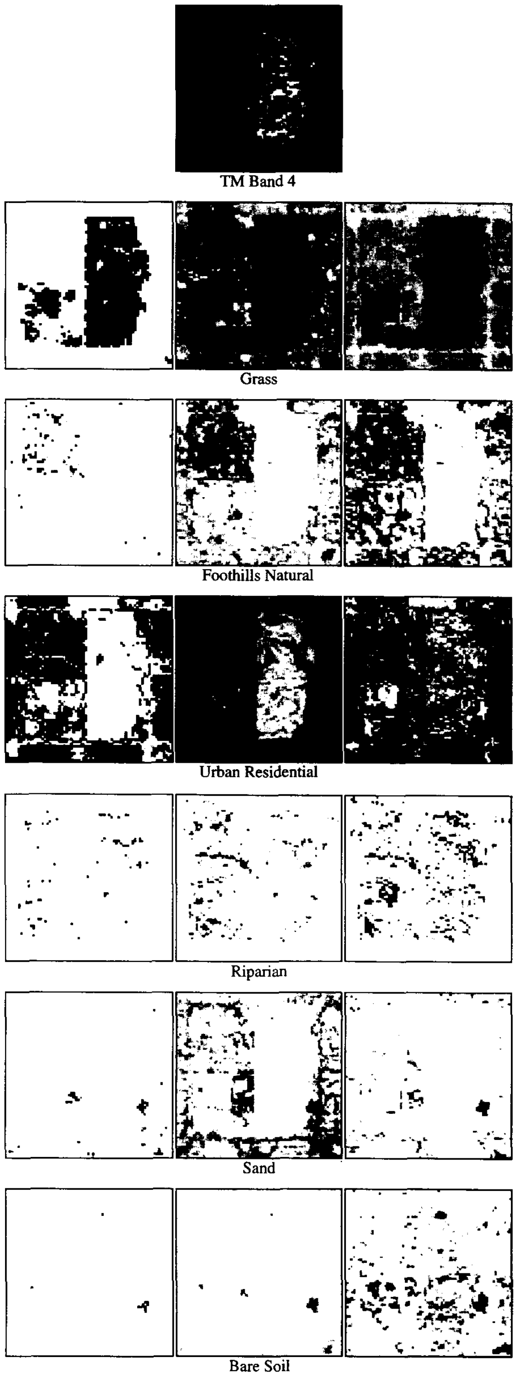


Fig. 13. This 70 × 70 region of the Tucson image corresponds to the area shown in the aerial photograph in Fig. 12. The top image contains the original band 4 (near infrared) data for the area. The following rows compare the maximum-likelihood *a posteriori* probability (left), the maximum-likelihood log probability density value (center), and the neural network class output (right).

existence and, through continuous values, the mixing of these two classes. The extent to which these values can be used to quantify class mixtures for each pixel is still unclear.

This detection of detail is evident in some of the other classes depicted in Fig. 13. For both classifiers, the "foothills natural" class, a denser form of natural desert vegetation (in comparison to "desert scrub"), is found amongst the residential areas. While at first this may seem incorrect, further examination of the residential areas reveals substantial yard areas between houses, many of which probably contain some form of natural Tucson desert vegetation. Some also contain grass as noted before. However, as can be seen in the "urban residential" maximum-likelihood *a posteriori* probability and neural network class output images, this predominant class (which is actually a mixture of many of the other classes) has a higher value and is correctly selected during an exclusive classification. Thus, both classifiers appear to be able to suggest the existence of lower priority classes while maintaining the primary goal of determining the major land use class. However, the "urban residential" class shows a fairly uniform probability density value for this region. The network output values for this class are much more varied, possibly indicating a greater level of class discrimination. This detail is consistent with the idea that the network produces an estimate of *a posteriori* class probabilities [12], [28]. However, the network clearly does not produce the same *a posteriori* probabilities as the maximum-likelihood method. The lack of a statistical assumption in the neural network allows it to produce more mutually exclusive class discriminant functions, resulting in more detail at the pixel level and possibly at subpixel levels. The maximum-likelihood classifier, because it depends on to the assumption of normal (and potentially high variance) distributions, can mask this fine detail in the classification. The *a posteriori* probabilities of maximum-likelihood reflect reality only if the assumed distributions are correct. If the network can produce similar values through the synthesis of more accurate distributions, it would be a valuable tool.

V. Summary and Conclusions

We set out in this paper to compare the neural network and maximum-likelihood classifiers in detail to better understand how and why they are different. Some of the differences we found may be attributed to the fact that the neural network classifier is nonparametric (distribution-free) and the maximum-likelihood classifier is parametric (distribution-dependent). There exist other, less common, nonparametric classifiers that could have been used for comparison to the neural network. The maximum-likelihood algorithm was chosen because it represents a widely-used "standard" for comparison that yields minimum total classification error for Gaussian class distributions.

Supervised land use classifications of two TM scenes, one of Tucson, AZ, and the other of Oakland, CA, were produced. Because of our familiarity with the site, the Tucson results were analyzed in several ways, including test site accuracies, spectral space decision regions, classifier "fuzzy" outputs, and visually. The Oakland results were analyzed visually, and those findings were correlated with those from the Tucson image. A cross-classification matrix was computed for each scene to view the differences between the classifiers. The land use categories chosen are similar to the Anderson Level I and Level II categories [1], which are heterogeneous in terms of spectral signatures, yet discernible to the human interpreter.

It is impossible to draw generally applicable conclusions from the analysis of only two scenes (this is a pervasive problem in most remote sensing analyses), so our results, like those of others, need to be qualified. However, we have looked at these particular datasets in enough detail to believe the general remarks below will be useful to others. The results discussed in this paper indicate the following:

- The primary computational difference between the algorithms is speed. The back-propagation approach to neural network training is extremely computation-intensive, taking at least an order of magnitude more time than the total classification time for maximum-likelihood. Although this situation may be alleviated with other, more efficient training algorithms and parallel implementation, it remains the single most important drawback to the routine use of neural networks. However, it was found that the classification time, once training is complete, is less for the neural network. The feed-forward classification of the Tucson image required about half the time for an 18 node hidden layer network as it does for maximum-likelihood. This is important for signature extension over multiple images or multitemporal images, where the initial long training period of the neural network will be amortized by shorter classification times over the lifetime of the trained classifier. The gain may be reduced, however, by the effort required to normalize all images for atmospheric changes and sun angle differences.
- The neural network approach, being nonparametric, is more robust to training site selection and class definition, and more easily accommodates a heterogeneous label such as "urban residential" to produce a visually and numerically correct map, even with smaller numbers of training pixels. The maximum-likelihood algorithm, on the other hand, is sensitive to the purity of the class signatures (a theoretical requirement for minimum error), and performs poorly if they are not pure. It is common practice to use an iterative "cleaning" procedure for maximum-likelihood training sites to remove outlier pixels and thereby form more homogeneous signatures. Although restricting ourselves to spectrally pure classes may result in a better output for maximum-likelihood, the fact is that we desire to produce a land *use* map for urban areas, and this type of map requires mixed classes such as "dense urban" and "residential." One way to handle these classes with the maximum-likelihood classifier is to model each with multiple Gaussians, corresponding to the component land *cover* classes, so that the training sites would not produce a single high variance Gaussian as seen in the original classification. An expert system can then be used to produce the desired urban land use categories from a set of spatial context rules applied to the land cover categories [24], [33]. This approach, however, requires more analyst care and attention to the training data, plus the additional expert system stage. Such laborious efforts appear to be less necessary for neural network training.
- The fuzzy output of the neural network, while not being directly related to classification probability, clearly does depend on the likelihood of the different classes. The results discussed here indicate it may be at least as good an indicator of significant class mixing as maximum-likelihood class density and *a posteriori* probability, but further study is necessary to understand this more completely.

Our work, as well as that of others, shows that the neural network classifier is a useful tool for remotely-sensed image classification. The biggest drawback to the method is the large training times necessary for mean square error minimization. Implementation of the training and classification algorithms on a massively parallel computing system would greatly enhance the applicability of the method. As computers become more powerful and processing speed increases, computationally intensive applications such as neural networks become more attractive. This increased speed, coupled with the neural network's flexible decision region capability and ability to use small training sets, give the neural network multispectral image classifier the potential to become a standard tool in remote sensing.

REFERENCES

[1] 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," USGS Professional Paper No 964, U.S. Geologic Survey, U.S. Government Printing Office, Washington, DC, 1976.

[2] 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. Geosci. Remote Sensing*, vol. 28, pp. 540–551, July 1990.

[3] J. A. Benediktsson, P. H. Swain, O. K. Ersoy, and D. Hong, "Classification of very high dimensional data using neural networks," in *Proc. IGARSS*, College Park, MD, May 1990, pp. 1269–1272.

[4] H. Bischof, W. Schneider, and A. J. Pinz, "Multispectral classification of landsat-images using neural networks," *IEEE Trans. Geosci. Remote Sensing*, vol. 30, pp. 482–490, May 1992.

[5] M. Caudill, "Neural networks primer: Part III," *AI Expert*, June, 1988, pp. 53–59.

[6] D. L. Civco, "Landsat TM image classification with an artificial neural network," in *Proc. ASPRS-ACSM Ann. Meeting*, Baltimore, MD, vol. 3, pp. 67–77, 1991.

[7] ——, "Artificial neural networks for land-cover classification and mapping," *Int. J. Geographical Inform. Syst.*, vol. 7, no. 2, pp. 173–186, 1993.

[8] M. S. Dawson, A. K. Fung, and M. T. Manry, "Sea ice classification using fast learning neural networks," in *Proc. IGARSS*, Houston, TX, May 1992, pp. 1070–1071.

[9] P. Dreyer, "Classification of land cover using optimized neural nets on SPOT data," *Photogrammetric Eng. Remote Sensing*, vol. 59, no. 5, pp. 617–621, 1993.

[10] G. M. Foody, N. A. Campbell, N. M. Trodd, and T. F. Wood, "Derivation and applications of probabilistic measures of class membership from the maximum-likelihood classification," *Photogrammetric Eng. Remote Sensing*, vol. 58, no. 9, pp. 1335–1341, 1992.

[11] H. Gish, "A probabilistic approach to the understanding and training of neural network classifiers," in *Proc. IEEE Int. Conf. Acoust., Signal, Speech Processing*, Albuquerque, NM, vol. 3, 1990, pp. 1361–1364.

[12] S. Haykin, *Neural Networks: A Comprehensive Foundation*. New York, NY: Macmillan, 1994.

[13] P. D. Heermann and N. Khazenie, "Classification of multispectral remote sensing data using a back-propagation neural network," *IEEE Trans. Geosci. Remote Sensing*, vol. 30, pp. 81–88, Jan. 1992.

[14] D. R. Hush and B. G. Horne, "Progress in supervised neural networks," *IEEE Signal Processing Mag.*, vol. 10, no. 1, pp. 8–39, 1993.

[15] I. Kanellopoulos, A. Varfis, G. G. Wilkinson, and J. Mégier, "Neural network classification of multi-date satellite imagery," in *Proc. IGARSS*, Espoo, Finland, June 1991, pp. 2215–2218.

[16] I. Kanellopoulos, G. G. Wilkinson, and J. Mégier, "Integration of neural network and statistical image classification for land cover mapping," in *Proc. IGARSS*, Tokyo, Japan, Aug. 1993, pp. 511–513.

[17] J. Key, J. A. Maslanik, and A. J. Schweiger, "Classification of merged AVHRR and SMMR arctic data with neural networks," *Photogrammetric Eng. Remote Sensing*, vol. 55, no. 9, pp. 1331–1338, 1989.

[18] ——, "Neural network versus maximum likelihood classifications of spectral and textural features in visible, thermal, and passive microwave data," in *Proc. IGARSS*, College Park, MD, May 1990, pp. 1277–1280.

[19] R. K. Kiang, "Classification of remotely sensed data using OCR-inspired neural network techniques," in *Proc. IGARSS*, Houston, TX, May 1992, pp. 1081–1083.

[20] H. Li, Z. Liu, and W. Sun, "A new approach to pattern recognition of remote sensing image using artificial neural network," in *Proc. IGARSS*, Tokyo, Japan, Aug. 1993, pp. 713–715.

[21] R. P. Lippmann, "An introduction to computing with neural networks," *IEEE Acoust., Speech, Signal Processing Mag.*, Apr. 1987, pp. 4–22.

[22] Z. K. Liu and J. Y. Xiao, "Classification of remotely-sensed image data using artificial neural networks," *Int. J. Remote Sensing*, vol. 12, no. 11, pp. 2433–2438, 1991.

[23] G. E. McClellan, R. N. DeWitt, T. H. Hemmer, L. N. Matheson, and G. O. Moe, "Multispectral image-processing with a three-layer backpropagation network," in *Proc. 1989 Int. Joint Conf. Neural Networks*, Washington, DC, vol. 1, pp. 151–153, 1989.

[24] G. Mehldau and R. A. Schowengerdt, "A C-extension for rule-based image classification systems," *Photogrammetric Eng. Remote Sensing*, vol. 56, no. 6, pp. 887–892, 1990.

[25] N. J. Mulder and L. Spreeuwers, "Neural networks applied to the classification of remotely sensed data," in *Proc. IGARSS*, Espoo, Finland, June 1991, pp. 2211–2213.

[26] Y.-H. Pao, *Adaptive Pattern Recognition and Neural Networks.* Reading, MA: Addison-Wesley, 1989.

[27] J. D. Paola and R. A. Schowengerdt, "A review and analysis of backpropagation neural networks for classification of remotely sensed multispectral imagery," *Int. J. Remote Sensing*, to be published.

[28] M. D. Richard and R. R. Lippmann, "Neural network classifiers estimate Bayesian *a posteriori* probabilities," *Neural Computation*, vol. 3, pp. 461–483, 1991.

[29] N. D. Ritter and G. F. Hepner, "Application of an artificial neural network to land-cover classification of thematic mapper imagery," *Computers Geosci.*, vol. 16, no. 6, pp. 873–880, 1990.

[30] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," in D. E. Rumelhart and J. L. McClelland, Eds., *Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Volume 1: Foundations.* Cambridge, MA: MIT Press, 1986, pp. 318–362.

[31] N. Short, "A real-time expert system and neural network for the classification of remotely sensed data," in *Proc. ASPRS-ACSM Ann. Meet.*, Baltimore, MD, vol. 3, pp. 406–418, 1991.

[32] P. H. Swain, "Fundamentals of pattern recognition in remote sensing," in P. H. Swain and S. M. Davis, Eds., *Remote Sensing: The Quantitative Approach.* New York: McGraw-Hill, 1978, pp. 137–187.

[33] S. W. Wharton, "A context-based land-use classification algorithm for high-resolution remotely sensed data," *J. Appl. Photographic Eng.*, vol. 8, no. 1, pp. 46–50, 1982.

Justin D. Paola received the B.S. degree in electrical engineering and computer science at the University of California, Berkeley and the M.S. degree in electrical and computer engineering from the University of Arizona, Tucson, in 1990 and 1994, respectively. He is pursuing the Ph.D. degree from the latter unversity.

He is currently a Research Assistant at the Digital Image Analysis Laboratory in the Department of Electrical and Computer Engineering at the University of Arizona. His research interests include image processing for remote sensing applications, image compression, neural networks, and high performance computing.

Robert A. Schowengerdt received the B.S. degree in physics from the University of Missouri, Rolla and the Ph.D. degree in optical sciences from the University of Arizona in 1968 and 1975, respectively.

He is currently an Associate Professor in Electrical and Computer Engineering, Optical Sciences and Arid Lands Studies at the University of Arizona, and is the author of numerous technical papers, including a textbook, *Techniques for Image Processing and Classification in Remote Sensing*, published in 1983. In 1989, he was a Fulbright Senior Scholar in image processing at the Australian Defence Force Academy in Canberra. His current research interests are in hyperspectral sensors and image analysis, and parallel computing algorithms for image processing, recognition and classification.

Dr. Schowengerdt is presently an Associate Editor for Image Processing of the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING.