Evaluation of Neural Pattern Classifiers for a Remote Sensing Application

Fischer, Manfred M. and Gopal, Sucharita and Staufer, Petra and Steinnocher, Klaus

Vienna University of Economics and Business, Boston University, Vienna University of Economics and Business, Vienna University of Economics and Business

1995

WSG Diskussionspapiere

WSG 46/95

Evaluation of Neural Pattern Classifiers for a Remote Sensing Application

Manfred M. Fischer, Sucharita Gopal,
Petra Staufer and Klaus Steinnocher

Institut für Wirtschafts- und Sozialgeographie

Wirtschaftsuniversität Wien

Department of Economic and Social Geography

Vienna University of Economics and Business Administration

WSG Discussion Papers

Abteilung für Theoretische und Angewandte Wirtschafts- und Sozialgeographie
Institut für Wirtschafts- und Sozialgeographie
Wirtschaftsuniversität Wien

Vorstand: o.Univ.Prof. Dr. Manfred M. Fischer
A - 1090 Wien, Augasse 2-6, Tel. (0222) 313 36 - 4836

Redaktion: Mag. Petra Staufer

WSG 46/95

Evaluation of Neural Pattern Classifiers for a Remote Sensing Application

Manfred M. Fischer, Sucharita Gopal,
Petra Staufer and Klaus Steinnocher

WSG-Discussion Paper 46

May 1995

Gedruckt mit Unterstützung
des Bundesministerium
für Wissenschaft und Forschung
in Wien

WSG Discussion Papers are interim reports presenting work in progress and papers which have been submitted for publication elsewhere.

ISBN 3 85037 051 8

Abstract

This paper evaluates the classification accuracy of three neural network classifiers on a satellite image-based pattern classification problem. The neural network classifiers used include two types of the Multi-Layer-Perceptron (MLP) and the Radial Basis Function Network. A normal (conventional) classifier is used as a benchmark to evaluate the performance of neural network classifiers. The satellite image consists of 2,460 pixels selected from a section (270 x 360) of a Landsat-5 TM scene from the city of Vienna and its northern surroundings. In addition to evaluation of classification accuracy, the neural classifiers are analysed for generalization capability and stability of results. Best overall results (in terms of accuracy and convergence time) are provided by the MLP-1 classifier with weight elimination. It has a small number of parameters and requires no problem-specific system of initial weight values. Its in-sample classification error is 7.87% and its out-of-sample classification error is 10.24% for the problem at hand. Four classes of simulations serve to illustrate the properties of the classifier in general and the stability of the result with respect to control parameters, and on the training time, the gradient descent control term, initial parameter conditions, and different training and testing sets.

Keywords: Neural Classifiers, Classification of Multispectral Image Data, Pixel-by-Pixel Classification, Backpropagation, Sensitivity Analysis

Evaluation of Neural Pattern Classifiers for a Remote Sensing Application

1. Introduction

Satellite remote sensing, developed from satellite technology and image processing, has been a popular focus of pattern recognition research since at least the 1970s. Most satellite sensors used for land applications are of the imaging type and record data in a variety of spectral channels and at a variety of ground resolutions. The current trend is for sensors to operate at higher spatial resolutions and for providing more spectral channels to optimize the information content and the usability of the acquired data for monitoring, mapping and inventory applications. At the end of this decade, the image data obtained from sensors on the currently operational satellites will be augmented by new instruments with many more spectral bands on board of polar orbiting satellites forming part of the Earth Observing System (Wilkinson et al. 1994).

As the complexity of the satellite data grows, so too does the need for new tools to analyse them in general. Since the mid 1980s, neural network (NN) techniques have raised the possibility of realizing fast, adaptive systems for multispectral satellite data classification. In spite of the increasing number of NN-applications in remote sensing (see, for example Key et al. 1989, Benediktsson et al. 1990, Hepner et al. 1990, Lee et al. 1990, Bischof et al. 1992, Heerman and Khazenie 1992, Civco 1993, Dreyer 1993, Salu and Tilton 1993, Wilkinson et al. 1994) very little has been done on evaluating different classifiers. Given that pattern classification is a mature area and that several NN approaches have emerged in the last few years, the time seems to be ripe for an evaluation of different neural classifiers by empirically observing their performance on a larger data set. Such a study should not only involve at least a moderately large data set, but should also be unbiased. All the classifiers should be given the same feature sets in training and testing.

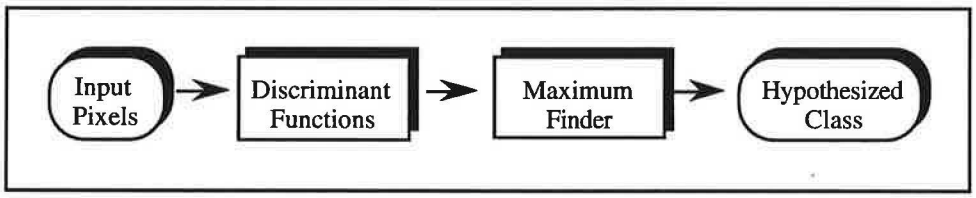
This paper addresses the above mentioned issue in evaluating the classification accuracy of three neural network classifiers. The classifiers include two types of the Multi-Layer Perceptron (MLP) and a Radial Basis Function Network (RBF). The widely used normal classifier based on parametric density estimation by maximum likelihood, NML, serves as benchmark. The classifiers were trained and tested for classification (8 a priori given classes) of multispectral images on a pixel-by-pixel basis. The data for this study was selected from a section (270 x 360 pixels) of a Landsat-5 Thematic Mapper scene (TM Quarter Scene 190-026/4; location of the center: 16° 23‘ E, 48° 14‘ N; observation date: June 5, 1985).

In section two of this paper, we will describe the structures of the various pattern classifiers. Then we will describe the experimental set-up in section 3, i.e. the essential organization of inputs and outputs, the network set-ups of the neural classifiers, a technique for addressing the problem of overfitting, criteria for evaluating the estimation (in-sample) and generalization (out-of-sample) ability of the different neural classifiers and the simulation set up (section 3). Four classes of simulations serve to analyse the stability of the classification results with respect to training time (50,000 epochs), the gradient descent control term (constant and variable learning schemes), the initial parameter conditions, and different training and testing sets. The results of the experiments are presented in section 4. Finally, in section 5 we give some concluding remarks.

2. The Pattern Classifiers

Each of our experimental classifiers consists of a set of components as shown in figure 1. The ovals represent input and output data, the rectangles processing components, and the arrows the flow of data. The components do not necessarily correspond to separate devices. They only represent a separation of the processing into conceptual units so that the overall structure may be discerned. The inputs may - as in the current context - come from Landsat-5 Thematic Mapper (TM) bands.

Figure 1: Components of the Pixel-by-Pixel Classification System



Each classifier provides a set of discriminant functions D_c ($1 \le c \le C$, C number of a priori given classes). There is one discriminant function D_c for each class c. Each one provides a single floating-point-number which tends to have a large number if the input pixel (i.e. feature vector $\mathbf{x}$ of the pixel, $\mathbf{x} \in \Re^n$) is of the class corresponding to that particular discriminant function. The C-tuple of values produced by the set of discriminant functions is sent to the 'Maximum Finder'. The 'Maximum Finder' identifies which one of the discriminant values $D_c(\mathbf{x})$ is highest, and assigns its class as the hypothesized class of the pixel, i.e. uses the following decision rule

$$\text{Assign } \mathbf{x} \text{ to class c if } D_c(\mathbf{x}) > D_k(\mathbf{x}) \text{ for } k=1, \ldots, C \text{ and } k \neq c \qquad (1)$$

Three experimental neural classifiers are considered here: multi-layer perceptron (MLP) classifiers of two types, MLP-1 and MLP-2, and one radial basis function (RBF) classifier. The normal

classifier NML serves as statistical benchmark. The following terminology will be used in the descriptions of the discriminant functions below:

n dimensionality of feature space (n representing the number of spectral bands used, n=6 in our application context),

$\Re^n$ the set of all n-tuples of real numbers (feature space),

x feature vector of a pixel ($\mathbf{x} = (x_1, \ldots, x_n) \in \Re^n$),

C number of a priori given classes ($1 \leq c \leq C$).

2.1 The Normal Classifier

This classifier (termed NML) which is most commonly used for classifying remote sensing data serves as benchmark for evaluating the neural classifiers in this paper. NML is based on parametric density estimation by maximum likelihood (ML). It presupposes a multivariate normal distribution for each class c of pixels. In this context, it may be worthwhile to mention first factors pertaining to any parametric classifier.

Let L(c|k) denote the loss (classification error) incurred assigning a pixel to class c rather than to class k. Let us define a particular loss function in terms of the Kronecker symbol δ_{ck}

$$L(c \mid k) = 1 - \delta_{ck} = \begin{cases} 0 & c=k \\ 1 & \text{otherwise} \end{cases} \qquad (2)$$

This loss function implies that correct classifications yield no losses, while incorrect classifications produce equal loss values of 1. In this case the optimal or Bayesian classifier is that one which assigns each input **x** ('feature vector' of a pixel), to that class c for which the a posteriori probability p(c|**x**) is highest, i.e.

$$p(c \mid \mathbf{x}) \geq p(k \mid \mathbf{x}) \qquad k=1, \ldots, C \qquad (5)$$

According to Bayes rule

$$p(c \mid \mathbf{x}) = \frac{p(c)\ p(\mathbf{x} \mid c)}{p(\mathbf{x})} \qquad (4)$$

where p(c) denotes the a priori probability of class c and p(**x**) the mixture density $\int p(\mathbf{x})\, d\mathbf{x}$ with **x** belonging to the training set $S \subseteq \Re^n$. For a pattern classification problem in which the a priori

probabilities are the same, p(c) can be ignored. For the normal classifier NML each class c is assumed to have a conditional density function

$$p(\mathbf{x}|c) = (2\pi)^{-\frac{n}{2}} |\Sigma_c|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\mu_c)^T \Sigma_c^{-1} (\mathbf{x}-\mu_c)\right\} \qquad c=1, \ldots, C \qquad (5)$$

with μ_c and Σ_c being the mean and associated covariance matrix for class c. The first term on the right-hand side of (5) is constant and may be discarded for classification. By replacing the mean vectors μ_c and the covariance matrices Σ_c with their sample estimates, $\mathbf{m}_c$ and $\mathbf{S}_c$, squaring and taking logarithms the set of NML-discriminant functions is given by

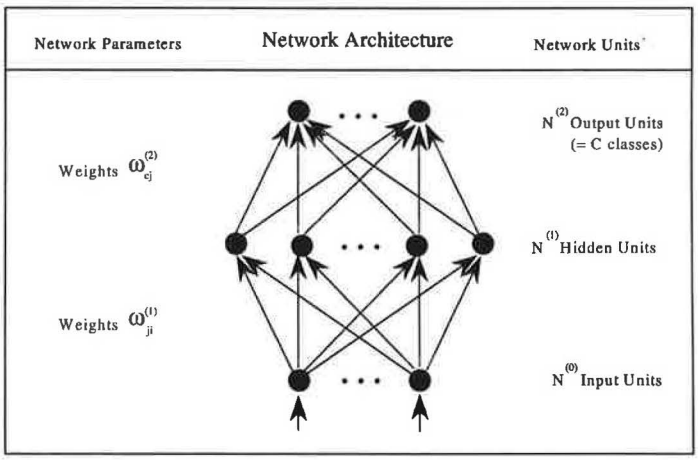
$$D_c(\mathbf{x}) = 2 \log \hat{p}(c) - \log|\mathbf{S}_c| - (\mathbf{x}-\mathbf{m}_c)^T \mathbf{S}_c^{-1} (\mathbf{x}-\mathbf{m}_c) \qquad (6)$$

where $\hat{p}(c)$ denotes the estimate of p(c).

2.2 The Multi-Layer Perceptron Classifiers

Multi-layer perceptrons are feed-forward networks with one or more layers of nodes between the input and output nodes. These additional layers contain hidden (intermediate) nodes or units. We have used MLPs with three layers (counting the inputs as a layer), as outlined in figure 2.

Figure 2: Architecture of a $N^{(0)} : N^{(1)} : N^{(2)}$ Perceptron



Let $N^{(k)}$ denote the number of units in the k-th layer (k = 0, 1, 2). The number of inputs, $N^{(0)}$[=n] and the number of outputs, $N^{(2)}$[=C] are determined by the application at hand, and in our study are

six for the input layer (one for each spectral channel TM1, TM2, TM3, TM4, TM5 and TM7) and eight for the output layer (representing the eight a priori categories of the pixels). The parameter with respect to the network architecture outlined in figure 2 is the number $N^{(1)}$ of non-linear hidden units that are fully connected to the input units and with the output units. Output and hidden units have adjustable biases (left out of consideration in figure 2). The weight $\omega_{ji}^{(l)}$ connects the i-th node of the (l-1)-th layer to the j-th node of the l-th layer ($l = 1, 2$; $1 \le i \le N^{(l-1)}$, $1 \le j \le N^{(l)}$). The weights can be positive, negative or zero.

Let us define $b_c^{(l)}$ the bias term of the i-th node of the l-th layer ($l = 1, 2$), and $\Psi(\mathbf{x})$ the non-linear hidden unit activation function, then the set of discriminant functions are of the form:

$$D_c(\mathbf{x}) = \frac{\exp\left\{b_c^{(2)} + \sum_{j=1}^{N^{(1)}} \omega_{cj}^{(2)} \; \psi\left(b_j^{(1)} + \sum_{i=1}^{N^{(0)}} \omega_{ji}^{(1)} \; x_i\right)\right\}}{\sum_{I=1}^{N^{(2)}} \exp\left\{b_l^{(2)} + \sum_{j=1}^{N^{(1)}} \omega_{lj}^{(2)} \; \psi\left(b_j^{(1)} + \sum_{k=1}^{N^{(0)}} \omega_{jk}^{(1)} \; x_k\right)\right\}} \qquad c=1,\ldots, C \qquad (7)$$

It is worthwhile to note that classifiers of type (7) use a softmax output unit activation function (see Bridle 1989). This activation function is a composition of two operators: an exponential mapping, followed by a normalisation to ensure that the output activations are non-negative and sum to one. The specification of the activation function Ψ is a critical issue in successful application development of a MLP classifier. We have experimented with two types of sigmoid functions, the most widely used non-linear activation functions: asymmetric and symmetric sigmoid functions. We use logistic activations for defining MLP-1 and hyperbolical tangent (tanh) activations for MLP-2.

The activation S_h of a logistic (sigmoid) hidden unit is given by

$$S_h = \left\{1 + \exp\left[-a\left(b_h^{(1)} + \sum_{i=1}^{N^{(0)}} \omega_{hi}^{(1)} \; x_i\right)\right]\right\}^{-1} \qquad (8)$$

which performs a smooth mapping $(-\infty, +\infty) \rightarrow (0,1)$. The slope 'a' can be absorbed into weights and biases without loss of generality and is set to one.

The activation T_h of a tanh hidden unit is given by

$$T_h = \tan\left[a\left(b_h^{(1)} + \sum_{i=1}^{N^{(0)}} \omega_{hi}^{(1)} \; x_i\right)\right] \qquad (9)$$

performing a smooth mapping $(-\infty, +\infty) \rightarrow (-1, +1)$. We here also set a=1.

For the training of the weights of MLP networks, a reasonable procedure is the use of an optimization algorithm to minimise the mean-square-error (least mean square error function) over the training set between the discriminant values actually produced and the target discriminant values that consist of the appropriate strings of 1s and 0s as defined by the actual classes of the training pixels. For example, if a training vector is associated to class 1, then its target vector of discriminant values is set to (1,0, ..., 0).

Networks of the MLP type are usually trained using the error backpropagation algorithm (see Rumelhart et al. 1986). Error backpropagation is an iterative gradient descent algorithm designed to minimise the least square error between the actual and target discrimination values. This is achieved by repeatedly changing the weights of the first and second parameter layer according to the gradient of the error function. The updating rule is given by

$$\omega_{rs}^{(k)}(t+1) = \omega_{rs}^{(k)}(t) + \eta \frac{\partial E}{\partial \omega_{rs}^{(k)}} \qquad k=1,2 \qquad (10)$$

Where E denotes the least mean square error function to be minimised over the set of training examples, and η the learning rate, i.e. the fraction by which the global error is minimised during each pass. The bias value b_h is also learned in the same way. In the limit, as η tends to zero and the number of iterations tends to infinity, this learning procedure is guaranteed to find the set of weights which gives the least mean square error (see White 1989).

2.3 The Radial Basis Function Classifier

In the MLP classifiers, the net input to the hidden units is a linear combination of the inputs. In a Radial Basis Function (RBF) network the hidden units compute radial basis functions of the inputs. The net input to the hidden layer is the distance from the input to the weight vector. The weight vectors are also called centres. The distance is usually computed in the euclidean metric. There is generally a bandwidth σ associated with each hidden unit. The activation function of the hidden units can be any of a variety of functions on the non-negative real numbers with a maximum at zero, approaching zero at infinity, such as the Gaussian transfer function.

We have experimented with a RBF classifier which uses softmax output units and Gaussian functions in the hidden layer. The following notation is necessary to describe the classifier. Let

$$\mathbf{c}^{(k)} = \left(c_1^{(k)}, \ldots, c_n^{(k)}\right)^T \in \Re^n \qquad k=1, \ldots, N^{(1)}$$

denote the centre vector of the k-th hidden unit and

$$\sigma^{(k)} = \left(\sigma_1^{(k)}, \ldots, \sigma_n^{(k)}\right)^T \in \Re^n \qquad k=1, \ldots, N^{(1)}$$

its width vector, while $b_k^{(l)}$ and $\omega_{lk}^{(2)}$ with $1 \le l \le N^{(2)} =: C$ and $1 \le l \le N^{(1)}$ are the bias term to the k-th node of the l-th layer and the weight connecting the l-th output node to the k-th hidden node, respectively.

Then the discriminant functions are given by:

$$D_c(\mathbf{x}) = \frac{\exp\left\{b_c^{(2)} + \sum_{k=1}^{N^{(1)}} \omega_{ck}^{(2)}\ \phi_k(\mathbf{x})\right\}}{\sum_{l=1}^{N^{(2)}} \exp\left\{b_c^{(2)} + \sum_{k=1}^{N^{(1)}} \omega_{lk}^{(2)}\ \phi_k(\mathbf{x})\right\}} \qquad c=1,\ldots, C \qquad (11)$$

where each hidden unit j computes the following radial basis function:

$$\phi_k(\mathbf{x}) = \exp\left(-\sum_{i=1}^{N^{(0)}} \left(\frac{x_i - c_i^{(k)}}{\sigma_i^{(k)}}\right)^2\right) = \prod_{i=1}^{N^{(0)}} \exp\left(-\left(\frac{x_i - c_i^{(k)}}{\sigma_i^{(k)}}\right)^2\right) \qquad k=1, \ldots, N^{(1)} \qquad (12)$$

The centres $\mathbf{c}^{(k)}$, widths $\sigma^{(k)}$, output bias nodes $b_c^{(2)}$ and output node weights $\omega_{lk}^{(2)}$ may be considered as trainable weights of the RBF network. They are trained initially using the cluster means (obtained by means of the K-means algorithm) as the centre vectors $\mathbf{c}^{(k)}$. The width vectors $\sigma^{(k)}$ are set to a single tunable positive value. Note that no target discriminant values are used to determine $\mathbf{c}^{(k)}$ and $\sigma^{(k)}$, while training of the output weights and bias proceeds by optimization identical to that described for the MLP classifiers.

The crucial difference between the RBF and the two MLP classifiers lies in the treatment of the inputs. For the RBF classifier, as can be seen from (12), the inputs factor completely. Unless all inputs x_i ($1 \le i \le n$) are reasonably close to their centres $c_i^{(k)}$, the activation of hidden unit k is close to zero. A RBF unit is shut off by a single large distance between its centre and the input in any one of the dimensions. In contrast, in the case of the MLP classifiers, a large contribution by one weighted output in the sum of (7) or (8) can often be compensated for by the contribution of other weighted inputs of the opposite sign. This difference between MLP and RBF classifiers increases with the dimensionality of the feature space.

3. Experimental Set up

3.1 The Data and Data Representation

The data used for training and testing the classification accuracy of the classifiers was selected from a section (270 x 360 pixels) of a Landsat-5 Thematic Mapper (TM) scene. The area covered by this imagery is 8.1x10.8 km^2 and includes the city of Vienna and its northern surroundings. The spectral resolution of each of the six TM bands (TM1, TM2, TM3, TM4, TM5, TM7) which were used in this study was eight bits or 256 possible digital numbers. Each pixel represents a ground area of 30x30 m^2. The purpose of the multispectral image classification task was to distinguish between eight land cover categories as outlined in table 1.

One of the authors, an expert photo interpreter with extensive field experience of the area covered by the image, used ancilliary information from maps and orthophotos (from the same time period) in order to select suitable training sites for each class. One training site was selected for each of the eight categories of land cover [single training site case]. This approach resulted in a database consisting of 2,460 pixels (about 2.5 percent of all the pixels in the scene) that are described by six-dimensional feature vectors and their class membership (target values). The set was divided into a training set (two thirds of the training site pixels) and a testing set by stratified random sampling, stratified in terms of the eight categories. Thus each training/test run consists of 1,640 training/820 testing vectors. This moderately large size for each training run makes the classification problem non-trivial at the one hand, but still allows for extensive tests on in-sample and out-of-sample performance of the classifiers.

Table 1: Categories Used for Classification and Number of Training/Testing Pixels

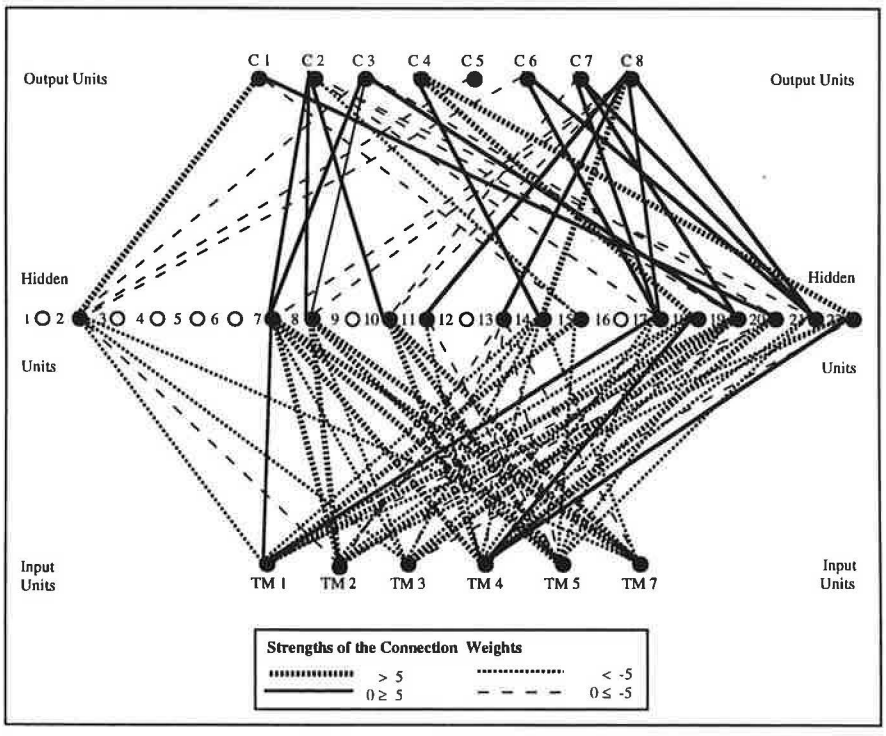
Category Number	Description of the Category	Pixels	
		Training	Testing
C1	Mixed grass and arable farmland	167	83
C2	Vineyards and areas with low vegetation cover	285	142
C3	Asphalt and concrete surfaces	128	64
C4	Woodland and public gardens with trees	402	200
C5	Low density residential and industrial areas (suburban)	102	52
C6	Densely built up residential areas (urban)	296	148
C7	Water courses	153	77
C8	Stagnant water bodies	107	54
Total Number of Pixels for Training and Testing		1,640	820

Data preprocessing (i.e. filtering or transforming the raw input data) plays an integral part in any classification system. Good preprocessing techniques reduce the effect of poor quality (noisy) data and this usually results in improved classification performance. In this study, the classifiers implemented in the experiments use gray coded data. The gray scale values in each spectral band were linearly compressed in the (0.1, 0.9) range to generate the input signals.

3.2 Network Set Up of the Neural Classifiers and the Overfitting Problem

The architecture of a neural classifier is defined by the arrangement of its units, i.e. the set of all weighted connections between units (see figure 2). This arrangement (i.e. the topology) of the network of a classifier is very important in determining its generalization ability. Generalization refers to the ability of a classifier to recognize patterns outside the training set. An important issue for good generalization is the choice of the optimal network size. This means finding the optimal number of hidden units, since inputs and outputs are defined by the problem at hand. There are some rules of thumb which often fail drastically since they ignore both the complexity of the task at hand and the redundancy in the training data (Weigend 1993). The optimal size of the hidden layer is usually not known in advance.

Figure 3: The Pruned MLP-1 with 14 'Degrees of Freedom' and 196 Parameters



The number of hidden units when the minimum is arrived may be viewed as a kind of measure of the degree of freedom of the network (Gershenfield and Weigend 1993). If the hidden layer is chosen to be too small, it will not be flexible enough to discriminate the patterns well, even in the training set. If it is chosen too large, the excess freedom will allow the classifier to fit not only the signals, but also the noise. Both, too small and too large hidden layers thus lead to a poor generalization capability in the presence of noise (Weigend et al. 1991).

This issue of overfitting or in other words the problem of estimating the network size has been widely neglected in remote sensing applications, up to now. Recently, several techniques have been proposed to get around this problem. To be relieved from the uncertainty of a specific choice of a validation set of the cross-validation approach (see Fischer and Gopal 1994) we have chosen in this study another approach, a network pruning or weight-elimination technique to overcome the problem of overfitting. This technique starts with an oversized network and attempts to minimise the complexity of the network (in terms of connection weights) and the standard sum squared error function by removing 'redundant' or least sensitive weights (see Weigend et al. 1991).

We deliberately have chosen an oversized, fully connected MLP-1 network with 22 hidden units and a variable learning rate. The 338 weights were updated after each 3 patterns, presented in random order (stochastic approximation). In the first 17,000 epochs, the procedure eliminated the weights between the eight output units and eight hidden units. Since these eight units did not receive the signals in the backward pass anymore, their weights to the input subsequentially decayed. In this sense, the weight-elimination procedure can be thought of as unit-elimination, removing the least important hidden units. The weights and biases of the pruned MLP with 14 remaining hidden units are given in appendix A. The architecture of the pruned MLP-1 is outlined in figure 3. The size of the network declined from 338 to 196 free parameters.

In contrast to MLP-classifiers, RBF networks are self-pruning to some degree. Unimportant connections are effectively pruned away by the training process leaving a large width. Each large width effectively deletes one connection from an input to one RBF and reduces the number of active patterns by two.

3.3 Performance Measures

The ultimate performance measure for any classifier is its usefulness to provide accurate classifications. This involves in-sample and out-of-sample classification accuracy. Four standard measures will be used to measure various aspects of classification accuracy:

- the **classification error** (also termed confusion) **matrix** (f_{lk}) with f_{lk} (l,k=1, ..., C) denoting the number of pixels assigned by the classifier to category l and found to be actually in (ground truth) category k,

- the **map user's classification accuracy**, υ_k, for the ground truth category k=1, ..., C

$$\upsilon_k := \frac{f_{kk}}{f_{\bullet k}} := \frac{f_{kk}}{\sum_{i=1}^{C} f_{ik}} \tag{13}$$

- the **map producer's classification accuracy** π_l for the classifier's category l=1, ..., C

$$\pi_l := \frac{f_{ll}}{f_{l\bullet}} := \frac{f_{ll}}{\sum_{j=1}^{C} f_{lj}} \tag{14}$$

- the total **classification accuracy** τ [or the total classification error τ' defined as $\tau' = (100 - \tau)$]

$$\tau := \frac{\sum_{i=1}^{C} f_{ii}}{f_{\bullet\bullet}} := \frac{\sum_{i=1}^{C} f_{ii}}{\sum_{k=1}^{C}\sum_{l=1}^{C} f_{lk}} \tag{15}$$

3.4 Experimental Simulation Set Up

Neural networks are known to produce wide variations in their performance properties. This is to say that small changes in network design, and in control parameters such as the learning rate and the initial parameter conditions might generate large changes in network behaviour. This issue, which is the major focus of our simulation experiments, has been highly neglected in remote sensing applications up to now. In real-world applications, it is, however, a central objective to identify intervals of the control parameters which give robust results, and to demonstrate that these results persists across different training and test sets.

In-sample and out-of-sample performance are the two most important experimentation issues in this study. In-sample performance of a classifier is important because it determines its convergence ability and sets a target of feasible out-of-sample performance which might be achieved by fine-tuning of the control parameters (Refenes et al. 1994). Out-of-sample performance measures the ability of a classifier to recognize patterns outside the training set, i.e. in the testing set strictly set apart from the training set. The performance depends on many factors, such as

- the gradient descent control term,
- initial parameter conditions, and
- training and testing sets.

Consequently, it is important to analyse the stability with respect to such control parameters. Several other important issues are not considered in this study, such as for example the issue of how the convergence speed can be improved. We have not used any acceleration scheme of backpropagation such as momentum. We also do not discuss the dependence of the performance on the size of the training/testing sets.

For our MLP-simulations we used parameter values initialised with uniformly distributed random values in the range between -0.1 and +0.1. If the initial weights are too large, the hidden units are saturated, and the gradient is also very small. The initial values for the RBF-centres were obtained from a K-means algorithm and the widths from a nearest neighbour heuristic. All the simulations were carried out on a Sun SPARCserver 10-GS with 128 MB RAM. The simulations described are performed using the epoch-based stochastic version of backpropagation, where the weights are updated after each epoch of three (randomly chosen) patterns in the training set. This version is opposed to the batch version, where the weights are updated after the gradients have accumulated over the whole training set, and to the pattern based version, where the weights are updated after the presentation of each pattern. The supervised learning minimised the standard objective (error) function, the sum of square of the output errors. Training and testing sets were chosen as simple random sample in each stratum of the eight training sites.

4. Classification Results

4.1 Overall Results: Performance of the Neural Classifiers with a Fixed Hidden-Layer Size

The purpose of the first experiment is to compare the in-sample and out-of-sample performance of the three neural classifiers each with 196 parameters, where the degrees of freedom are equal to 14. Thus, we were able to analyse the effect of different hidden unit activation functions, the sigmoid (logistic), the hyperbolic tangent (tanh) and the radial basis activations, upon performance. All other factors including initial conditions are fixed in these simulations (η=0.8). The results are outlined in table 2 and show that the two MLP-classifiers trained more slowly than the RBF-classifier, but clearly outperform RBF (measured in terms of τ). The RBF-classifier does not train and generalize as accurately as the MLP-networks. Its results, however, strongly depend on the initial conditions for the RBF centres and widths. It is important to bear in mind that no attempts

have been made here to optimise the results of this classifier with respect to these parameters. There seems to be much unexplored potential to improve the performance of this classifier. MLP-1 and MLP-2 generally train and generalize at the same rate, but MLP-1's training is faster, by about 30 percent.

Table 2: Summary of Classification Results

Classifier	Classification Accuracy τ		Convergence Time
	In-Sample	Out-of-Sample	(CPU-Time [sec.])
MLP-1	92.13	89.76	15.1
MLP-2	90.91	90.00	21.0
RBF	80.00	75.61	10.6
NML	90.85	85.24	1.4

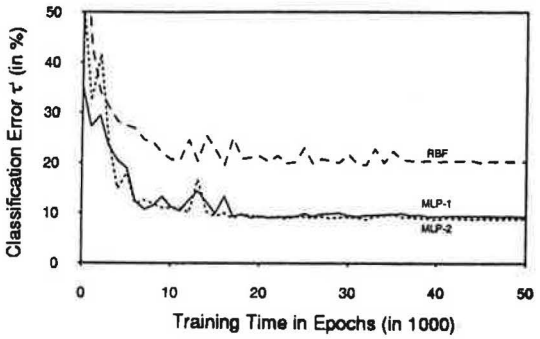
Thus, the best overall result is provided by the MLP-1 classifier with 14 hidden units and 196 free parameters, followed by MLP-2, and RBF. Both MLP classifiers outperform the NML classifier in terms of generalization capabilities. The superiority of the MLP classifiers over RBF may be, moreover, underlined by considering the in-sample and out-of-sample classification error matrices (see appendix B), the map user's and map producer's accuracies in appendix C. Even though trained on 1,640 pixels only, the MLP-1 classifier can be used to classify the 97,200 pixels of the whole image. The raw satellite image and the MLP-1 classified image are displayed in figure 4.

4.2 Stability with Training Time

Figure 5 shows the in-sample performance for the two versions of the multi-level perceptron, MLP-1 and MLP-2, and the radial basis function classifier as a function of training time in epochs (η=0.8, trained for 50,000 epochs, and equal random initialisations). The in-sample performance tends to converge asymptotically at a minimum that is found at about 17,000 epochs in the case of the MLP-classifiers and about 36,000 epochs in the case of RFB.

There are some regions with temporary performance drops. At least, in the case of the MLP-classifiers we do not think that these can be interpreted as signs of overtraining, because they appear rather early in the training process. More probably, their existence implies that the network is still undertrained, and the better solutions are yet to come for larger numbers of epochs. This behaviour persists across the three different neural classifiers.

Figure 5: In-Sample-Performance of MLP-1, MLP-2, and RBF
(as a function of training time in epochs)



4.3 Stability with Initial Conditions

Backpropagation is known to be sensitive to the values of initial conditions of the parameters. The number of free parameters of MLP-1 is 196. The objective function has multiple local minima and is sensitive to details of initial values. A relatively small change in the initial values for the parameters generally results in finding a different local minimum. In this type of experiment we used three different sets of initial conditions. Initial weights were chosen from a uniform random distribution in (-0.1, +0.1).

Figure 6: The Effect of Different Initial Parameter Conditions on the Performance of MLP-1

(a) In-Sample Performance **(b) Out-of-Sample Performance**

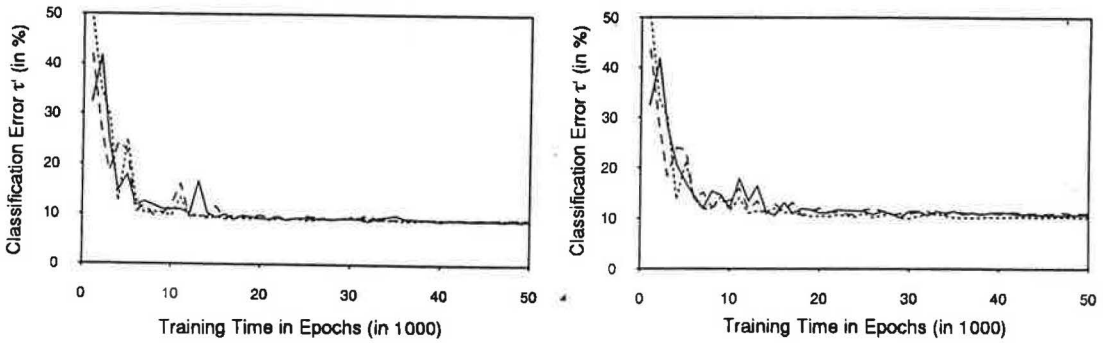


Figure 6 shows the in-sample and out-of-sample classification error curves for the three trials. It is clear, that different initial conditions can lead to more or less major differences in the starting stage of the training process. After about 15,000 epochs the differences in performance more or less vanish. Nevertheless, it is important to stress that the issue of stability with initial conditions deserves consideration when training a classifier in a real-world application context.

4.4 Stability with the Gradient Descent Control Term η

The choice of the control parameter for the gradient descent along the surface essentially influences the magnitude of weight changes and, thus, is crucial for learning performance. But it is difficult to find appropriate learning rates. On one hand, a small learning rate implies small changes even though greater weight changes would be necessary. On the other hand, a greater learning rate implies greater weight changes. Greater weight changes might be required because of the speed of convergence on the network stability. Larger learning rate values might also assist the classifier to escape from a local minimum.

It is important to examine how the classification results vary with the gradient descent control term. A stability analysis with respect to this parameter shows that both in-sample and out-of-sample performance of the classifier remain very stable in the range of η=0.4 to η=0.8, while a small change from η=0.4 to η=0.2 yields a dramatic loss in classification accuracy (see table 3). The optimal learning rate is the one which has the largest value that does not lead to oscillation, and this is η=0.8 in this experiment. Figure 7 shows that a variable learning rate adjustment (declining learning rate: η=0.8 until 5,000 epochs, η=0.4 until 15,000 epochs, then η=0.1 until 35,000 epochs and thereafter η=0.00625) might lead to faster convergence, but only to a slightly better generalization performance.

Figure 7: The Effect of Different Approaches to Learning Rate Adjustment on (a) In-Sample Performance and (b) Out-of-Sample Performance of MLP-1: Constant (η=0.8) Versus Variable Learning Rate Adjustment

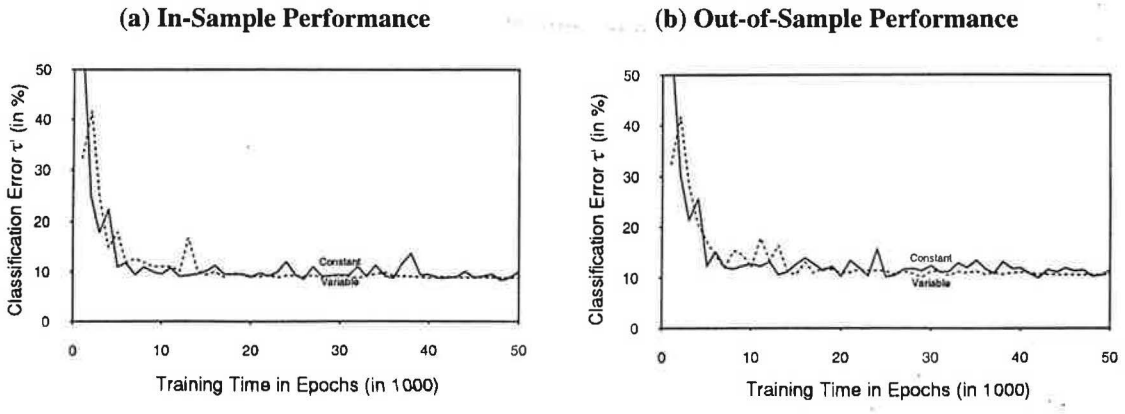


Table 3: Stability of Results with the Gradient Descent Control Parameter as Function of Training Time in Epochs

Epochs ($\times 10^3$)	Control Parameter η	In-Sample Performance (in terms of τ)	Out-of-Sample Performance (in terms of τ)
3	0.2	16.6	12.5
	0.4	73.7	72.3
	0.6	78.2	78.5
	0.8	82.2	78.5
6	0.2	17.60	12.5
	0.4	90.17	88.2
	0.6	86.93	86.0
	0.8	88.28	84.9
9	0.2	21.56	12.5
	0.4	89.37	88.2
	0.6	90.22	87.5
	0.8	89.97	87.6
12	0.2	21.56	12.5
	0.4	88.37	85.4
	0.6	88.38	86.5
	0.8	90.92	86.8
15	0.2	22.54	12.7
	0.4	90.06	89.1
	0.6	88.93	87.9
	0.8	89.86	87.3
18	0.2	24.50	13.1
	0.4	89.55	87.3
	0.6	89.96	87.1
	0.8	90.51	88.5
21	0.2	24.50	13.1
	0.4	90.77	87.7
	0.6	91.48	88.3
	0.8	90.22	86.6
24	0.2	31.51	15.4
	0.4	91.47	88.2
	0.6	90.69	88.0
	0.8	87.87	84.3
27	0.2	31.51	15.4
	0.4	91.11	89.0
	0.6	89.96	87.2
	0.8	88.95	88.2
30	0.2	31.51	15.4
	0.4	90.81	89.2
	0.6	90.29	87.9
	0.8	90.59	87.5

4.5. Stability of Results with Different Training and Testing Samples

All the simulations we mentioned so far were performed for the same training and test data sets, obtained by stratified random sampling. To examine the effect of different training and test data sets on the performance, we used three randomly selected trials with stratification to generate

training and testing sets of 1,640 and 820 pixels, respectively. In figure 8 we see only minor differences. The in-sample performance of the classifier did not alter significantly after 15,000 epochs. The out-of-sample performance of two trials was rather similar after 36,000 epochs. However, one of the trials shows a different pattern in out-of-sample performance. If the training and test samples were randomly drawn without stratification, major differences in performance might arise between the trials (see figure 9).

Figure 8: The Effect of Selected Randomly Chosen Training/Testing Set Trials with Stratification on (a) In-Sample Performance and (b) Out-of-Sample Performance of MLP-1with Variable Learning Rate Adjustment

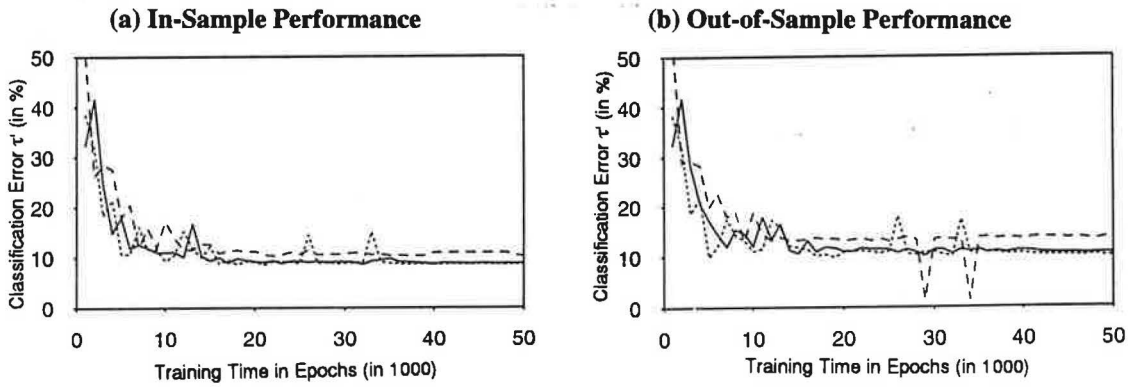
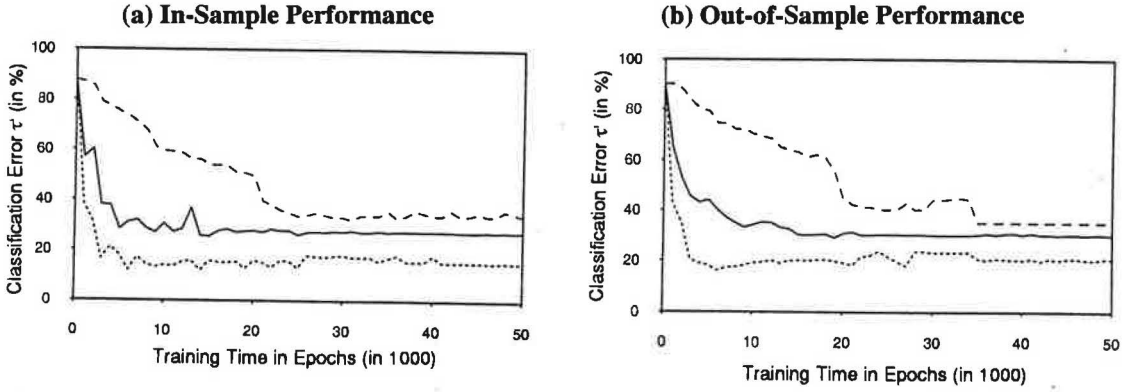


Figure 9:The Effect of Selected Randomly Chosen Training/Testing Set Trials without Stratification on (a) In-Sample Performance and (b) Out-of-Sample Performance of MLP-1with Variable Learning Rate Adjustment



5. Conclusions

One major objective of this paper was to evaluate the classification accuracy of three neural classifiers, MLP-1, MLP-2 and RBF, and to analyze their generalisation capability and the stability of the results. We illustrated that both in-sample and out-of-sample performance depends upon fine-tuning of control parameters. Moreover, we were able to show that even a simple neural learning procedure such as the backpropagation algorithm outperforms by about 5 percent the conventional classifier in generalisation that is most often used for multispectral classification on a pixel-by-pixel basis, the NML classifier. The non-linear properties of the sigmoid (logistic) and the hyperbolic tangent (tanh) activation functions in combination with softmax activations of the output units allow neural network based classifiers to discriminate the data better and generalize significantly better, in the context of this study.

We strongly believe that with careful network design and multiple rather than single training sites and with a more powerful learning procedure, the performance of the neural network classifiers can be improved further, especially the RBF classifier. In this respect, other techniques than the K-means procedure might be more promising to use in order to obtain the initial values for the RBF centres and widths.

We hope that the issues addressed in this paper will be beneficial not only for designing neural classifiers for multispectral classification on a pixel-by-pixel basis, but also for other classification problems in the field of remote sensing, such as classification of multi-source data or multi-angle data.

Acknowledgement

The authors gratefully acknowledge Professor Karl Kraus (Department of Photogrammetric Engineering and Remote Sensing, Vienna Technical University) for his assistance in supplying the remote sensing data used in this study. This work is supported by a grant from the Austrian Fonds zur Förderung der Wissenschaftlichen Forschung (P-09972-TEC) and the US-National Science Foundation (SBR-930063).

References

Benediktsson, J.A., Swain, P.H. and Ersory, O.K. (1990): Neural network approaches versus statistical methods in classification of multisource remote sensing data, *IEEE Transactions on Geoscience and Remote Sensing*, vol. 28(4), pp. 540-551.

Bischof, H., Schneider, W. and Pinz, A.J. (1992): Multispectral classification of Landsat-images using neural networks, *IEEE Transactions on Geoscience and Remote Sensing*, vol. 30 (3), pp. 482-490.

Bridle, J.S. (1989): Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition, in Fougelman-Soulie, F. and Hérault, J. (eds.): *Neuro-Computing: Algorithms, Architectures and Applications*, New York: Springer.

Civco, D.L. (1993): Artificial neural networks for land-cover classification and mapping, *International Journal of Geographical Information Systems*, vol. 7(2), pp. 173-186.

Dreyer, P. (1993): Classification of land cover using optimized neural nets on SPOT data, *Photogrammetric Engineering and Remote Sensing*, vol. 59(5), pp. 617-621.

Fischer, M.M. and Gopal, S. (1994): Artificial neural networks. a new approach to modelling interregional telecommunication flows, *Journal of Regional Science* (in press).

Gershenfield, N.A. and Weigend, A.S. (eds.) (1993): *Time Series Prediction: Forecasting the Future and Understanding the Past. Reading* (MA): Addison-Wesley.

Heerman, P.D., and Khazenie, N. (1992): Classification of multispectral remote sensing data using a backpropagation neural network, *IEEE Transactions on Geoscience and Remote Sensing*, vol. 30(1), pp. 81-88.

Hepner, G.F., Logan, T., Ritter, N. and Bryant, N. (1990): Artificial neural network classification using a minimal training set: Comparison to conventional supervised classification, *Photogrammetric Engineering and Remote Sensing*, vol. 56 (4), pp. 469-473.

Key, J., Maslanic, A. and Schweiger, A.J. (1989): Classification of merged AVHRR and SMMR Arctic data with neural networks, *Photogrammetric Engineering and Remote Sensing*, vol. 55(9), pp. 1331-1338.

Lee, J., Weger, R.C., Sengupta, S.K. and Welch, R.M. (1990): A neural network approach to cloud classification, *IEEE Transactions on Geoscience and Remote Sensing*, vol. 28(5), pp. 846-855.

Refenes, A.N., Zapranis, A. and Francis, G. (1994): Stock performance modeling using neural networks: A comparative study with regression models, *Neural Networks*, vol. 7(2), pp. 375-388.

Rumelhart, D.E., Hinton, G.E. and Williams, R.J. (1986): Learning internal representation by error propagation, in Rumelhart, D.E., McClelland, J.L. and PDP Research Group (eds.): *Parallel Distributed Processing: Explorations in the Microstructures of Cognition*, pp. 318-362. Cambridge (MA): MIT Press.

Salu, Y. and Tilton, J. (1993): Classification of multispectral image data by the binary diamond neural network and by nonparametric, pixel-by-pixel methods, *IEEE Transactions on Geoscience and Remote Sensing*, vol. 31(3), pp. 606-617.

Weigend, A.S, Huberman, B.A. and Rumelhart, D.E. (1991): Predicting sunspots and exchange rates with connectionist networks, in Eubank, S. and Casdagli, M (eds.): *Proceedings of the 1990 NATO Workshop on Nonlinear Modelling and Forecasting*, pp. 1-36. Redwood City (CA): Addison-Wesley.

Weigend, A.S. (1993): Book Review: John A. Hertz, Anders S. Krogh and Richard G. Palmer, Introduction to the Theory of Neural Computation, *Artificial Intelligence*, vol. 62, pp. 93-111.

White, H. (1989): Some asymptotic results for learning in single hidden-layer feedforward network models, *Journal of the American Statistical Association*, vol. 84, pp. 1003-1113.

Wilkinson, G.G., Fierens, F. and Kanellopoulos, I. (1994): Integration of neural and statistical approaches in spatial data classification, *Geographical Systems - The International Journal of Geographical Information, Analysis, Theory and Decision* (in press).

Appendix A: Parameters of the MLP-1 Classifier after Weight Elimination

The classifier was trained for 17,000 epochs with backpropagation and a constant learning rate of 0.8. The connection weights and biases of the network are given below in table A1. When simulated serially on a SPARCserver 10-GS, the training took 15.1 CPU-minutes. Once the parameters have been determined, predictions are extremely fast.

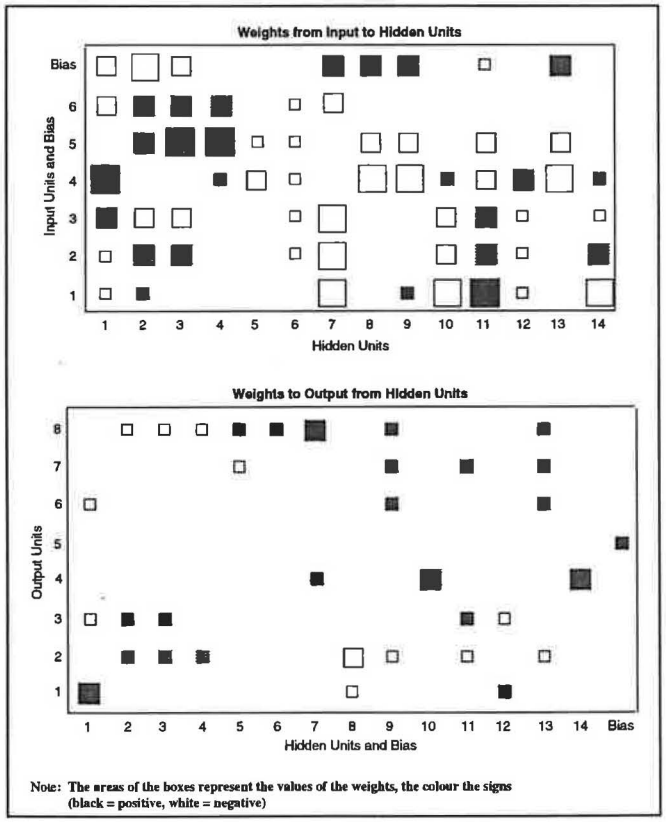
Table A1: Weights of the MNP-1-Classifier after Weight Elimination (17 x 10^3 epochs)

Weights from	Input Unit i1		Input Unit 2		Input Unit 3		Input Unit 4		Input Unit 5		Input Unit 6		Bias Unit	
	Initial	Final	Initial	Final	Initial	Final	Initial	Final	Initial	Final	Initial	Final	Initial	Final
to Hidden Unit 1	-0.2654	-4.2068	-0.1314	-3.5575	-0.2212	-5.9796	0.3784	18.6484	0.3578	-0.0732	0.0869	-8.5990	0.2003	-8.3436
to Hidden Unit 2	0.1594	3.0924	0.4070	5.8573	0.2456	5.3472	0.0073	-0.3563	0.2850	7.5201	0.2688	5.5217	0.1842	-10.4762
to Hidden Unit 3	0.0531	1.8249	0.3094	5.8297	0.2921	5.2104	0.1607	-0.1826	-0.0433	12.0609	0.3005	7.7699	0.4015	-9.0623
to Hidden Unit 4	0.1994	0.4149	0.0774	0.7208	-0.2140	0.2577	-0.2098	4.3713	0.0882	10.3742	0.1856	6.9198	-0.2269	-1.1575
to Hidden Unit 5	-0.1601	0.3377	-0.3399	-0.3897	0.1033	0.1062	-0.4065	-5.9631	-0.3450	-4.9175	0.3534	-0.2195	-0.1589	0.7615
to Hidden Unit 6	-0.0044	-1.3902	-0.2401	-2.8702	-0.1541	-3.1082	0.0318	-3.2401	-0.0878	-3.6032	0.1167	-2.4680	-0.0406	-0.4469
to Hidden Unit 7	-0.3718	-15.9215	-0.3073	-14.7156	0.1205	-11.5417	-0.1708	-0.6922	0.0414	-1.6728	-0.3274	-5.5068	0.0237	7.1074
to Hidden Unit 8	0.3438	7.9521	0.3005	1.2669	0.3396	1.7644	0.1011	-14.2799	0.0615	-7.5545	0.3542	-0.0473	0.1576	8.3054
to Hidden Unit 9	0.0437	2.4169	0.0576	0.5233	0.1545	0.9377	-0.0733	-15.7223	0.2209	-7.2733	0.0608	-0.8447	0.2412	6.7711
to Hidden Unit 10	-0.3722	-13.5282	-0.2948	-9.6025	0.0408	-6.4631	0.1526	3.2262	0.2933	-0.4754	-0.0902	-1.7938	-0.0380	-0.1079
to Hidden Unit 11	0.0069	12.3658	0.1574	9.6851	0.3950	8.1292	-0.3037	-8.5565	-0.3066	-6.0183	-0.1253	1.2778	0.1470	-3.8310
to Hidden Unit 12	0.0750	-3.5478	-0.0813	-2.2013	0.2052	-2.2538	-0.1013	7.9774	-0.0173	0.1960	0.2132	-0.3957	-0.1855	0.6626
to Hidden Unit 13	-0.0528	1.5297	0.1934	0.5204	-0.3384	0.2251	-0.1238	-15.4543	0.0033	-6.3996	0.3912	-0.0856	-0.3394	7.0218
to Hidden Unit 14	-0.1923	-12.4562	-0.1975	-8.4983	0.0904	-4.8614	-0.3738	2.8841	-0.0765	-0.0992	0.1709	-0.8173	-0.1162	0.0340

Weights to	Output Unit 1		Output Unit 2		Output Unit 3		Output Unit 4		Output Unit 5		Output Unit 6		Output Unit 7		Output Unit 8	
	Initial	Final	Initial	Final	Initial	Final	Initial	Final	Initial	Final	Initial	Final	Initial	Final	Initial	Final
from Hidden Unit 1	0.0296	7.6672	-0.1638	0.1116	-0.0154	-2.1449	0.0306	1.3780	-0.1633	-0.2221	-0.1745	-2.0476	0.1451	-1.2741	-0.1667	-0.6374
from Hidden Unit 2	0.0313	-0.3342	0.0924	2.7388	-0.1079	3.7358	-0.0066	-1.3653	0.0776	0.7831	0.1954	0.0913	-0.0674	-0.5317	-0.1269	-2.1005
from Hidden Unit 3	-0.0727	-0.4291	0.0232	4.5698	0.2100	2.4571	-0.0060	-1.2327	0.1543	1.1922	0.1215	-0.0193	-0.2042	-1.1493	-0.1241	-2.6817
from Hidden Unit 4	0.1569	0.2916	0.1540	2.9387	0.1912	0.4205	0.0663	0.0305	-0.0894	0.3293	-0.2090	-0.5576	-0.0788	-2.0048	0.1540	-2.7553
from Hidden Unit 5	-0.1165	-0.5384	-0.0406	-1.1709	0.0888	-0.2694	-0.1419	-0.5978	0.1315	-0.0037	-0.0932	0.2617	-0.1675	0.5696	0.1175	2.2892
from Hidden Unit 6	0.0895	0.0044	-0.0086	-0.3923	0.0505	-0.4396	-0.0500	0.0434	-0.0742	-0.0712	-0.0528	0.0778	0.0694	0.2861	0.1096	2.6504
from Hidden Unit 7	-0.1257	0.2249	-0.1955	-1.7252	-0.1904	-1.7241	0.1733	4.7005	0.1701	0.1637	0.0223	-0.4417	-0.1298	-0.1447	0.0983	6.7245
from Hidden Unit 8	-0.1848	-3.3815	-0.1404	-6.3693	-0.1170	0.3268	-0.1681	-0.9975	0.0472	-0.1072	0.0691	1.6311	0.2088	1.8163	0.0129	0.8942
from Hidden Unit 9	-0.1908	-1.6496	-0.1914	-3.2548	0.1779	0.0828	-0.1485	-0.8071	0.0005	-0.2282	-0.0661	4.1493	0.2081	3.6822	0.1130	2.2155
from Hidden Unit 10	0.0026	1.0520	-0.1997	-1.3107	-0.0280	-1.1155	0.1611	6.2207	-0.1762	-0.3579	-0.0653	-1.1377	0.1879	-0.4309	-0.2095	-0.4046
from Hidden Unit 11	-0.0783	-1.0328	0.1830	-2.4157	0.1434	4.4601	-0.0311	-1.0268	-0.1831	-0.2165	0.0293	0.7224	0.1890	3.0847	-0.1837	0.0306
from Hidden Unit 12	0.1461	2.1664	-0.0339	0.4392	-0.0863	-4.0775	0.0292	0.4963	-0.0933	0.2816	-0.1719	-0.5003	-0.1109	-0.5863	-0.0690	-0.4879
from Hidden Unit 13	-0.2124	-1.6818	-0.1096	-2.9845	0.0001	0.0142	-0.1844	-0.7775	-0.0521	-0.2195	0.1410	4.0468	0.1640	3.6123	0.0173	2.0987
from Hidden Unit 14	0.0992	0.5098	-0.0655	-0.4612	-0.0141	-1.2092	0.2045	5.7548	0.0216	0.0802	-0.2110	-0.4714	-0.0020	-0.2936	-0.0036	0.1820
from Bias Unit	0.1249	0.2723	0.1453	0.7989	0.1157	-0.4729	0.1530	-0.2961	0.0318	3.4917	-0.1816	0.1615	-0.0615	-0.3898	0.1679	-0.6033

Interpretation of these weights sheds light on which spectral channels are important for particular surface categories. Similarly, the connection weights indicate, for each output category, the degree of information redundancy among channels in the input data. Channels which are only weakly weighted add little additional information to the classification process. The identification of the exact role of the hidden units is difficult, as they often represent generalisations of the input patterns. Figure A1 shows with which input data channel each hidden node is associated in the trained network (top) and with which hidden unit each output class is related (bottom). The unit labelled 'bias' has output +1 and so represents the bias term. The areas of the boxes represent the values, the colour the signs (black = positive, white = negative). Following the connections through these two boxes, thus, indicates which input channels are linked to particular output categories.

Figure A1: Weights of the MLP-1-Classifier after Weight Elimination (17 x 10^3 epochs)



Appendix B: In-Sample and Out-of-Sample Classification Error Matrices of the Neural Classifiers

An error matrix is a square array of numbers set out in rows and columns which expresses the number of pixels assigned to a particualr category relative to the actual category as verfied by some reference (ground truth) data. The columns represent the reference data, the rows indicate the classification generated. It is important to note that differences between the map classification and reference data might be not only due to classification errors. Other possible sources of errors include errors in interpretation and delineation of the reference data, changes in land cover between the data of the remotely sensed data and the data of the reference data (temporal error), variation in classification of the reference data due to inconsistencies in human interpretation etc.

Table B1: In-Sample Performance: Error Classification Matrices (f_{lk}) of the Neural and the Statistical Classifiers

(a) MLP-1

Classifier's Categories	Ground Truth Categories								
	C1	C2	C3	C4	C5	C6	C7	C8	Total
C1	157	10	0	0	0	0	0	0	167
C2	1	282	0	0	2	0	0	0	285
C3	0	0	128	0	0	0	0	0	128
C4	4	0	0	389	9	0	0	0	402
C5	0	0	2	2	98	0	0	0	102
C6	0	0	1	0	0	260	25	10	296
C7	0	0	0	0	0	60	93	0	153
C8	0	0	0	0	0	3	0	104	107
Total	162	292	131	391	109	323	118	114	1,640

(b) MLP-2

Classifier's Categories	Ground Truth Categories								
	C1	C2	C3	C4	C5	C6	C7	C8	Total
C1	156	9	0	2	0	0	0	0	167
C2	4	280	0	1	0	0	0	0	285
C3	0	0	126	2	0	0	0	0	128
C4	4	0	0	384	14	0	0	0	402
C5	0	0	2	4	96	0	0	0	102
C6	0	0	1	0	0	253	25	17	296
C7	0	0	0	0	0	60	93	0	153
C8	0	0	0	0	0	4	0	103	107
Total	164	289	129	393	110	317	118	120	1,640

(c) RBF

Classifier's Categories	Ground Truth Categories								
	C1	C2	C3	C4	C5	C6	C7	C8	Total
C1	141	22	0	4	0	0	0	0	167
C2	14	263	0	4	0	4	0	0	285
C3	0	0	115	13	0	0	0	0	128
C4	9	0	0	349	44	0	0	0	402
C5	0	0	12	12	78	0	0	0	102
C6	0	0	5	0	0	189	71	31	296
C7	0	0	0	0	0	73	80	0	153
C8	0	0	0	0	0	10	0	97	107
Total	164	285	132	382	122	276	151	128	1,640

(d) NML

Classifier's Categories	Ground Truth Categories								
	C1	C2	C3	C4	C5	C6	C7	C8	Total
C1	161	5	0	1	0	0	0	0	167
C2	0	284	0	0	1	0	0	0	285
C3	0	0	124	0	4	0	0	0	128
C4	0	4	0	385	13	0	0	0	402
C5	0	0	0	0	102	0	0	0	102
C6	0	0	3	0	0	214	62	17	296
C7	0	0	0	0	0	37	116	0	153
C8	0	0	0	0	0	3	0	104	107
Total	161	293	127	386	120	254	178	121	1,640

Table B2: Out-of-Sample Error Classification Matrices (f_{lk}) of the Neural and the Statistical Classifiers

(a) MLP-1

Classifier's Categories	Ground Truth Categories								
	C1	C2	C3	C4	C5	C6	C7	C8	Total
C1	79	4	0	0	0	0	0	0	83
C2	1	134	6	0	1	0	0	0	142
C3	0	0	64	0	0	0	0	0	64
C4	3	2	0	194	1	0	0	0	200
C5	0	3	0	0	49	0	0	0	52
C6	0	0	0	0	0	115	30	3	148
C7	0	0	0	0	0	29	48	0	77
C8	0	0	0	0	0	1	0	53	54
Total	83	143	70	194	51	145	78	61	820

(b) MLP-2

Classifier's Categories	Ground Truth Categories								
	C1	C2	C3	C4	C5	C6	C7	C8	Total
C1	79	4	0	0	0	0	0	0	83
C2	1	140	0	0	1	0	0	0	142
C3	0	0	64	0	0	0	0	0	64
C4	1	2	0	193	3	0	0	1	200
C5	0	1	0	0	51	0	0	0	52
C6	0	0	0	0	2	110	32	4	148
C7	0	0	0	0	0	29	48	0	77
C8	0	0	0	0	0	1	0	53	54
Total	81	147	64	193	57	140	80	58	820

(c) RBF

Classifier's Categories	Ground Truth Categories								
	C1	C2	C3	C4	C5	C6	C7	C8	Total
C1	35	21	0	27	0	0	0	0	83
C2	5	137	0	0	0	0	0	0	142
C3	0	4	60	0	0	0	0	0	64
C4	24	6	4	163	0	0	0	3	200
C5	0	26	4	0	22	0	0	0	52
C6	0	0	0	0	0	104	35	9	148
C7	0	0	0	0	0	29	48	0	77
C8	0	0	0	0	0	0	3	51	54
Total	64	194	68	190	22	133	86	63	820

(d) NML

Classifier's Categories	Ground Truth Categories								
	C1	C2	C3	C4	C5	C6	C7	C8	Total
C1	80	3	0	0	0	0	0	0	83
C2	0	141	0	0	0	0	0	0	142
C3	0	0	62	0	1	1	0	0	64
C4	1	3	0	191	5	0	0	0	200
C5	0	5	0	0	47	0	0	0	52
C6	0	0	1	0	2	73	64	8	148
C7	0	0	0	0	0	24	53	0	77
C8	0	0	0	0	0	2	0	52	54
Total	81	152	63	191	56	100	117	60	820

Appendix C: In-Sample and Out-of-Sample Map User's and Map Producer's Accuracy

Table C1: In-Sample Classification Accuracy π and υ for the Pattern Classifiers

Category	Name	Map User's Accuracy π				Map Producer's Accuracy υ			
		MLP-1	MLP-2	RBF	NML	MLP-1	MLP-2	RBF	NML
C1	Rural Landscape	94.0	93.4	84.4	96.4	96.9	95.1	86.0	95.1
C2	Vineyards	98.9	98.2	92.3	99.6	96.6	96.9	92.3	96.9
C3	Asphalt	100.0	98.4	89.8	96.9	97.7	97.7	87.1	97.7
C4	Woodland	96.8	95.5	86.8	95.8	99.5	97.7	91.4	97.7
C5	Low Residential	96.1	94.1	76.5	100.0	89.9	87.3	63.9	87.3
C6	Densely Built Up	87.8	85.5	63.9	72.3	80.5	79.8	68.5	79.8
C7	Water Courses	60.8	60.8	52.3	75.8	78.8	78.8	53.0	78.8
C8	Stagnant Water	97.2	96.3	90.7	97.2	91.2	85.8	75.8	85.8

Table C2: Out-of-Sample Classification Accuracy π and υ for the Pattern Classifiers

Category	Name	Map User's Accuracy π				Map Producer's Accuracy υ			
		MLP-1	MLP-2	RBF	NML	MLP-1	MLP-2	RBF	NML
C1	Rural Landscape	95.2	95.2	42.2	96.4	95.2	97.5	54.7	98.8
C2	Vineyards	94.4	98.6	96.5	99.3	93.7	95.2	70.6	92.8
C3	Asphalt	100.0	100.0	93.8	96.9	91.4	100.0	88.2	98.4
C4	Woodland	97.0	96.5	81.5	95.5	100.0	100.0	85.8	100.0
C5	Low Residential	94.2	98.1	42.3	90.4	96.1	89.5	100.0	83.9
C6	Densely Built Up	77.7	74.3	70.3	49.3	79.3	78.6	78.2	73.0
C7	Water Courses	62.3	62.3	62.3	68.8	61.5	60.0	55.8	45.3
C8	Stagnant Water	98.1	98.1	94.4	96.3	86.9	91.4	81.0	86.7

Manfred M. Fischer is the professor and chair of the Department of Economic Geography at Wirtschaftsuniversität in Vienna. He is the chair of the Commission on Mathematical Models of the International Geographical Union and is on the Executive Committee of the European Regional Science Association. He is member of the editorial board of several journals including Environment and Planning A, Geographical Analysis, The Annals of Regional Science, Papers in Regional Science and Sistemi Urbani, and he is one of the founding editors of the Geographical Systems, The International Journal of Geographical Information, Analysis, Theory and Decision. His research includes geoinformation processing and artificial intelligence, spatial econometrics and spatial modeling, pattern recognition, decision theory and micro-behavioural modeling. Dr. Fischer received the MA and Ph.D. degrees in geography and mathematics from the University of Erlangen, Germany. He was an associate professor at the University of Vienna before assuming his present position at the Wirtschaftsuniversität.

Sucharita Gopal received the Ph.D. in geography from the University of California, Santa Barbara in 1989. Since then she has carried out research in the area of spatial cognition, fuzzy sets and spatial accuracy, geographical information systems, and neural network applications. She is an assistant professor in the Department of Geography at Boston University.

Petra Staufer is an assistant and second year graduate student at the Department of Economic Geography at Wirtschaftsuniversität in Vienna. Her current research interests are in the field of neurocomputing, pattern recognition and geographical information systems. She received the MA degree in geography from the University of Vienna in 1993.

Klaus Steinnocher received the Ph.D. from the Institute of Photogrammetry and Remote Sensing at the Technical University of Vienna in 1994. He is currently employed at the Department of Environmental Planning at the Austrian Research Center, Seibersdorf. His research interests include classification and post classification methods in remote sensing, integration of geographical information systems and remote sensing, and environmental modelling.