JOURNAL OF CHEMOMETRICS, VOL. 11, 111–129 (1997)

COMPARISON OF MULTILAYER PERCEPTRON AND PROBABILISTIC NEURAL NETWORKS IN ARTIFICIAL VISION. APPLICATION TO THE DISCRIMINATION OF SEEDS

YOUNES CHTIOUI,[1] DOMINIQUE BERTRAND,[1] MARIE-FRANCOISE DEVAUX[1] AND DOMINIQUE BARBA[2]

[1] *Institut National de la Recherche Agronomique, Laboratoire de Technologie Appliquée à la Nutrition, BP 1627, Rue de la Géraudière, F-44316 Nantes Cedex 03, France*

[2] *Institut de Recherche et d'Enseignement Supérieur aux Techniques de l'Electronique, Laboratoire des Systèmes Electroniques et Informatiques, La Chantrerie, CP 3003, F-44087 Nantes Cedex 03, France*

SUMMARY

In classification problems the most commonly used neural network is probably the multilayer perceptron network (MLPN). The probabilistic neural network (PNN) is a possible alternative to the MLPN. The PNN is based on the Bayesian approach and a non-parametric estimation of the probability density functions of the qualitative classes. In this paper the performances of the PNN and the MLPN were compared on an illustrative application which consisted of the discrimination of seed species by artificial vision. The colour images of individual kernels of four species (two cultivated and two adventitious ones) were acquired. A set of 73 features characterizing the seed size, shape and texture was extracted. The data collection was divided into a training set of 1600 seeds and a test set of 800 seeds. A stepwise discriminant analysis made it possible to select the first four relevant variables among the 73 available ones. The MLPN incorrectly classified 44 and 28 seeds of the training and test sets respectively. Three configurations of the PNN were tested on the same data collection. The most sophisticated version of the PNN gave 17 and 19 misclassifications in the same data sets. The PNN presents an architecture in which all the units are operating in parallel and a hardware implementation of this kind of architecture is therefore possible. All the scaling parameters of the PNN can be determined from the training set. In contrast, there is no algorithm to automatically determine the structure of the MLPN. © 1997 by John Wiley & Sons, Ltd.

Journal of Chemometrics, Vol. **11,** 111–129 (1997) (No. of Figures: 9 No. of Tables: 4 No. of Refs: 29)

KEY WORDS probabilistic neural network; multilayer perceptron network; artificial vision; seed; classification

INTRODUCTION

Neural network models are based on the interconnection of a set of non-linear computational units called 'neurons'. Each neuron achieves a simple task. Over the last decade scientists have been proposing various architectures in order to model biological nervous systems for resolving numerical problems.

In the last few years, neural networks have been tested with a number of deterministic problems, e.g. pattern recognition.[1] They have also been used for recognition and synthesis of speech.[2] Neural networks can serve many purposes, e.g. classification and generalized regression. Generalized

Correspondence to: Younes Chtioui.

CCC 0886–9383/97/020111–19 $17.50

Received 1 April 1996
Accepted 20 June 1996

regression aims at predicting the value of one or more unknown variables from one or more measured variables. Classification with neural networks consists of assigning an unknown pattern to a qualitative group. The non-linear boundaries of classes are built during the training process.

There are various topologies and algorithms of neural networks, which make it possible to determine pattern statistics from a set of training samples and then classify new patterns on the basis of these statistics. According to their topologies, neural networks may require either unsupervised or supervised training. For example, the Carpenter and Grossberg neural network[3] is used to form clusters without any supervision. In this case no information concerning the correct class of each training pattern is provided to the network during the training phase. Networks trained with supervision, such as the Hopfield network,[4] the multilayer perceptron network,[5] and the probabilistic neural network,[6] are generally used as nonlinear classifiers. In these cases, the class of each training sample is known. The most widespread neural network architecture is the multilayer perceptron network (MLPN), which usually applies the well-known back propagation algorithm[7] as a learning rule. The main goal of the back propagation algorithm is to adjust the weights of the network in order to reduce the errors of classification of the training set. In this algorithm, the available learning patterns are presented, one after another, at the inputs of the network. Each corresponding output is assessed forwards. The network weights are gradually adjusted, from the output to the input layers, by taking the value of the observed error into account. This algorithm may be time-consuming and very slow to converge. Moreover, it may stick at a local minimum.

High parallelism and analogical VLSI (very large scale integration) implementation techniques are essential for high performance in pattern recognition. However, the back-propagation algorithm of the MLPN does not present a structure which can be easily implemented in a completely parallel manner. For this reason, neural networks which operate in parallel have been proposed. The probabilistic neural network (PNN) has been developed in order to respect the requirement of high parallelism. The PNN applies a comprehensive mapping strategy derived from the Bayesian decision rule[8] and from non-parametric estimators of probability density functions. It associates an unknown pattern to a class in order to minimize the estimated misclassification risk. The Bayesian classification rule was developed many decades ago, but its application required a lot of computational power, which was not available. At that time the practical interest of the Bayesian classification rule was therefore small. For this reason the method had been neglected by the statistician community and considered as a theoretical approach. Today, as available computers are more powerful, the method of mapping which is required by the Bayesian approach becomes feasible. The Bayesian classification rule was first applied in the area of neural networks by Specht[6] who invented the PNN model in early 1990.

In the present study, we tried to compare the performances of the MLPN and the PNN in artificial vision. In the field of artificial vision in chemistry, PNN have seldom been applied. The two kinds of neural networks were tested on an application problem which consisted of the discrimination of seeds in commercial lots. In many countries, seed lots cannot be commercialized if they contain some adventitious seeds. In specialized seed laboratories, the control is currently achieved by visual inspection. Visual inspection of seeds is time-consuming and difficult, because the number of registered varieties is constantly increasing. Techniques such as gel electrophoresis[9] and pattern recognition combined with image analysis have been attempted for automatic seed classification. However, gel electrophoresis is complex and requires sophisticated laboratory methods and skilled operators. According to the published results, attempts at seed classification which involved computer vision[10–13] have often been confined to methods which apply linear classifiers, e.g. discriminant analysis. Since discriminant analysis represents each qualitative group by a single centroid, it is not a relevant method when a population of seeds of a given species is multimodal. The main goal of the present work was to discriminate between four seed species (two cultivated and two adventitious species) and to compare the respective performances of the MLPN and the PNN.

THEORY

Let a pattern be denoted by a p-dimensional vector $\mathbf{x}=(x_1, x_2, \ldots, x_p)^T$, where p is the number of measured variables. Let us assume there are k pattern classes, numbered from 1 to k, and n training patterns. All the training data are gathered into a matrix $\mathbf{X}$ with n rows and p columns. Each row represents a training pattern and each column represents a measured variable. Let x_{ij} be the jth variable of row i. As the two classifiers presented here are supervised, the qualitative group of each training pattern is known. The aim of the classification is to predict the qualitative group of each test pattern by using neural networks.

Multilayer perceptron network

The MLPN is a feedforward network with one or more hidden layers of units between the input and the output. As an example, a three-layer perceptron network with one layer of hidden units is shown in Figure 1. It was proved that no more than three layers are required, because a three-layer network can generate any arbitrarily complex decision region.[14] The MLPN consists of an interconnection of small units called 'neurons', all the connections being weighted. The task of each unit is to add all the weighted inputs and then to apply a non-linear activation function on the weighted sum s. During the training, a qualitative group of each training pattern must be known. In 1986, Rumelhart, McClelland and Williams proposed the 'back propagation' algorithm, also called the 'generalized delta rule', as a learning algorithm. The back propagation is an iterative learning algorithm which generally uses the

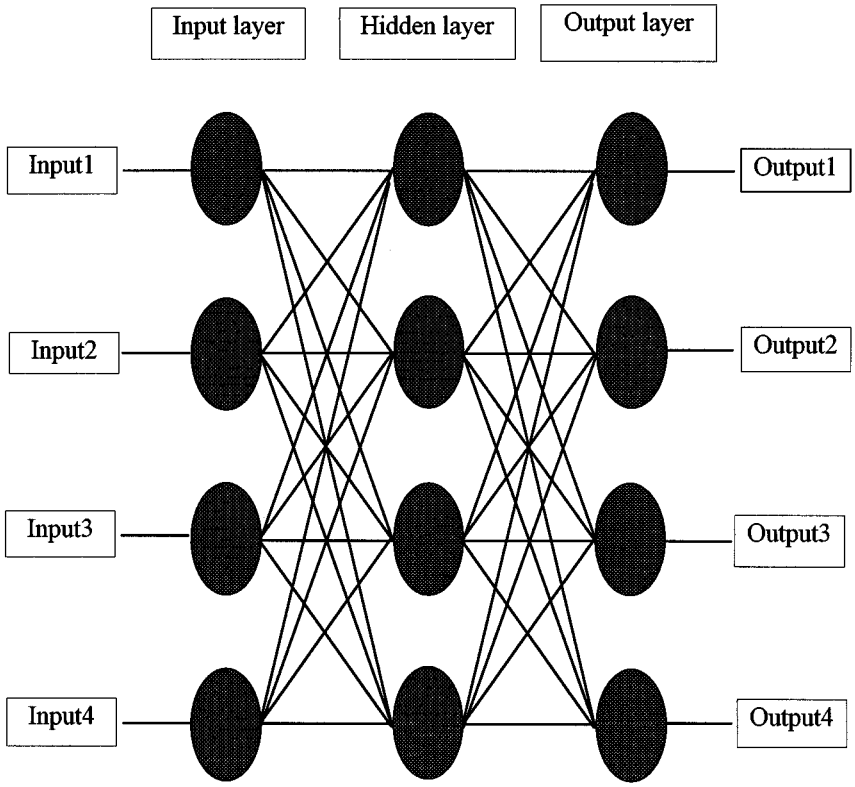

Figure 1. Structure of MLPN

sigmoid as an activation function. The sigmoid function is defined by

$$f(s)=\frac{1}{1+\mathrm{e}^{-as}} \tag{1}$$

and has the range $0<f(s)<1$, where a is a constant which controls the spread of the sigmoid function. The network is trained by initializing the weights to small random values. Each time an unknown training pattern p is presented to the inputs of the network, the corresponding output is assessed. The error function is formulated on the basis of the current output $\mathbf{o}_p$ and the target $\mathbf{t}_p$. It is defined by

$$e_p=\sum_{i=1}^{k}(o_{ip}-t_{ip})^2 \tag{2}$$

The main goal of the back-propagation learning rule is to gradually adjust the weights in order to minimize e_p. For this purpose the back propagation learning rule uses the gradient descent algorithm. The weights are iteratively adjusted according to

$$w_{ij}(t+1)=w_{ij}(t)+\eta g(e_p) \tag{3}$$

where η refers to the learning coefficient ($0<\eta<1$), $w_{ij}(t)$ is the weight between node i and node j at time t, and $g(e_p)$ is a term depending on the error function.[7] Moreover, equation (3) can be improved by introducing a 'momentum' factor. The momentum factor is to weight adaptation what the highpass filter is to analogical electronics. It amplifies large weight changes but attenuates small ones. Equation (3) becomes

$$w_{ij}(t+1)=w_{ij}(t)+\eta g(e_p)+\alpha[w_{ij}(t)-w_{ij}(t-1)] \tag{4}$$

where α is the momentum factor and has the range $0<\alpha<1$. The momentum factor is sometimes useful to escape from a local minimum. It may also speed up the convergence of the back-propagation algorithm.

Probabilistic neural network

The structure of the MLPN is quite different from that of the PNN.[15] The PNN takes its basic concept from the optimum Bayesian decision rule. Before describing the architecture of the PNN, it is necessary to explain the Bayesian approach.

Suppose we have a set of random p-dimensional patterns belonging to one among k different classes. Each class i has a *a priori* probability of occurrence h_i. The main goal of the Bayesian decision rule is to assign an unknown pattern $\mathbf{x}$ to a class to which this pattern is most likely to belong. In some studies, the cost of misclassification can be different according to the actual class of $\mathbf{x}$. In the present study, it was assumed that the costs of misclassifications were equal for all the classes. The Bayesian decision rule provides a means to estimate the *a posteriori* probability that the input pattern $\mathbf{x}$ belongs to each class and then to assign this input pattern to the class which presents the maximum *a posteriori* probability. The input pattern $\mathbf{x}$ is attributed to class i if

$$p(w=w_i/\mathbf{x})=\max_{j=\{1,2,\ldots,k\}}p(w=w_j/\mathbf{x}) \tag{5}$$

From the Bayes theorem we have:

$$p(w=w_i/\mathbf{x})=\frac{h_i f(\mathbf{x}/w=w_i)}{\sum_{j=1}^{k} h_j f(\mathbf{x}/w=w_j)} \tag{6}$$

where h_i refers to the *a priori* probability of population i and $f(\mathbf{x}/w=w_i)$ is the probability density function (PDF) for class i. By using equation (6), the assignment criterion defined in equation (5) can be replaced by

$$h_i f(\mathbf{x}/w=w_i)=\max_{j=\{1,2,\ldots,k\}} h_j f(\mathbf{x}/w=w_j) \tag{7}$$

The main key to using equation (7) is the calculation of the PDF for each class. In the literature, many non-parametric PDF estimators exist, such as the Rosenblatt estimator,[16] the Parzen estimator,[17] and the Loftsgaarden–Quesenberry estimator.[16] Parzen proposed a powerful technique to estimate the PDF in the univariate case. The principal advantage of Parzen's PDF estimator is that it is both unbiased and consistent. Parzen's PDF estimator is simply a weighted sum of small units centred at each training pattern:

$$f(\mathbf{x}/w=w_i)=\frac{1}{n_i}\sum_{j=1}^{n_i} W\left(\frac{\|\mathbf{x}-\mathbf{x}_j\|}{\sigma}\right) \tag{8}$$

where σ is a smoothing parameter, n_i is the number of patterns in class i, and W is a unit function, also called a kernel function, which must meet certain conditions.[17] In principle, many kernel functions can be used. The most widely applied is the Gaussian function, which is known to give satisfactory results in many situations. The utilization of the Gaussian function does not imply that the training set was supposed to present a normal distribution. Parzen has proved that the estimated PDF for a population converges to the actual PDF as the size of this population increases. This means that Parzen's PDF estimator is consistent in a quadratic sense:

$$\lim_{n_i\to+\infty} E[|f_{\text{estimated}}(\mathbf{x}/w=w_i)-f_{\text{actual}}(\mathbf{x}/w=w_i)|^2]=0 \tag{9}$$

Cacoullos[18] has extended Parzen's PDF estimator to the multivariate case. The estimated density function is now the sum of a multivariate kernel function centred at each training sample:

$$f(x_1, x_2, , , , x_p/w=w_i)=\frac{1}{n_i}\sum_{j=1}^{n_j} W\left(\frac{x_1-x_{j1}}{\sigma_{i1}}, \frac{x_2-x_{j2}}{\sigma_{i2}}, \ldots, \frac{x_p-x_{jp}}{\sigma_{ip}}\right) \tag{10}$$

Things are now becoming much more complicated, as we have to estimate a function of p variables. The density estimate for class i and for the particular case where the kernel function is a multivariate Gaussian function is given by:

$$f(x_1, x_2, \ldots, x_p/w=w_i)=\frac{1}{(2\pi)^{p/2}\sigma_{i1}\sigma_{i2}\ldots\sigma_{ip}n_i}\sum_{j=1}^{n_i} \mathrm{e}^{-D_i(\mathbf{x},\,\mathbf{x}_j)} \tag{11}$$

where

$$D_i(\mathbf{x}, \mathbf{x}_j) = \sum_{k=1}^{p} \left(\frac{x_k - x_{jk}}{\sigma_{ik}} \right)^2 \tag{12}$$

On the basis of equations (7) and (11), Specht has proposed a PNN which is a four-layer feedforward neural network. This network model presents a high degree of parallelism. Figure 2 shows an example of the architecture of a PNN which deals with a problem where four classes are to be discriminated and the dimensionality of each input pattern is also four. Each unit in the pattern layer represents a training pattern. The pattern layer assesses the distance between the input pattern and each training sample (equation (12)). The activation function, which is here the exponential function as in equation (11), is then applied. The summation layer associated to a given class sums the output of the pattern

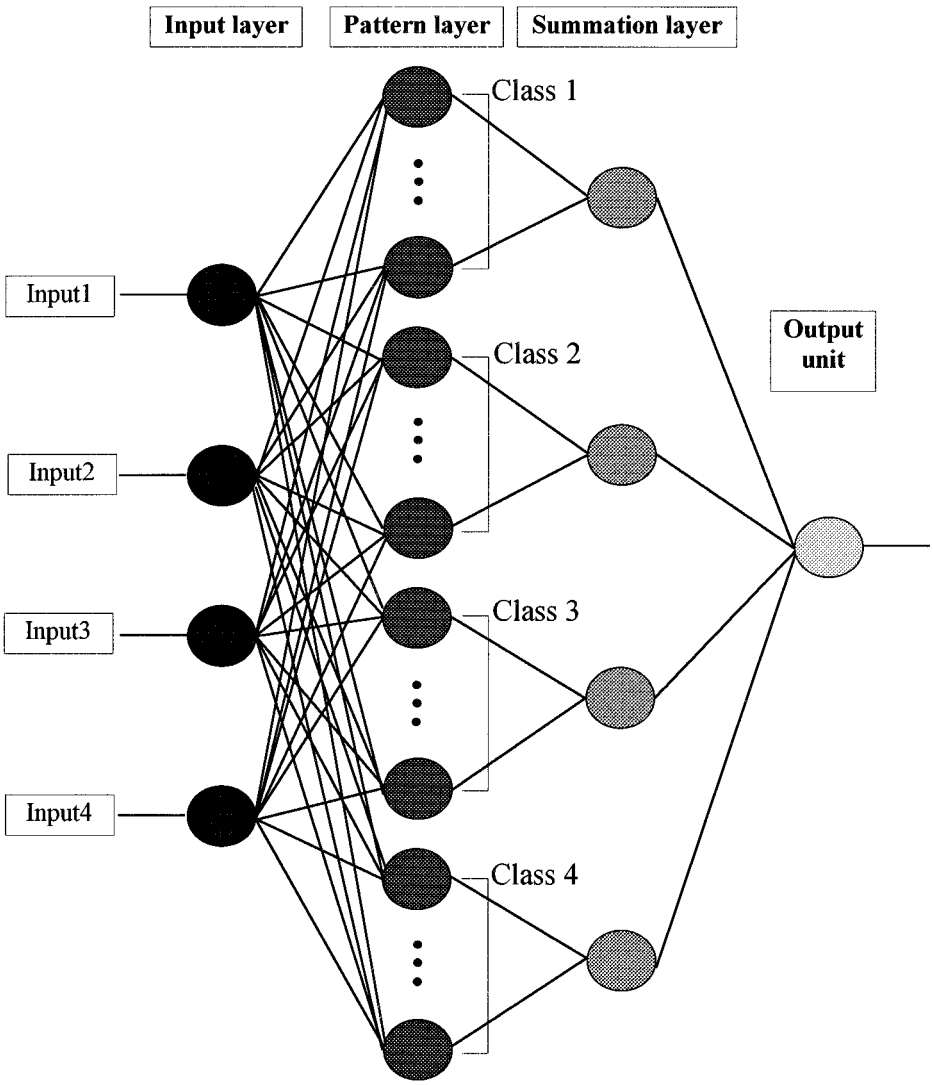

Figure 2. Structure of PNN

layer units belonging to that class. In this way the output of each summation unit is proportional to the PDF of the corresponding class. The output unit finds the maximum value of its inputs and returns the class number associated to this maximum (equation (7)).

The density estimates use smoothing parameters (σ_{ik}) which must be chosen in order to minimize the estimated PDF error. It is therefore necessary to define a criterion for evaluating the performance of a trial value of σ_{ik} and to choose an optimization algorithm enabling the minimization of this criterion. The usual criterion is the observed percentage misclassification of the training set. Each time a training sample is to be classified, the corresponding unit in the pattern layer is not taken into account in order to have a non-biased classifier. The estimation of σ_{ik} is achieved by an iterative optimization algorithm, the conjugate gradient algorithm. The method is described by Schioler and Hartmann[19] and Specht.[20]

Several options in the way of assessing the smoothing parameters can be selected. (i) We can define a single smoothing parameter for all the variables and all the populations. This sigma model has been called *BASIC*. (ii) Variability between variables can be taken into account by associating to each variable its own smoothing parameter (model called *SEPVAR*; p smoothing parameters to be estimated). (iii) Density estimates of heterogeneous classes can be assessed by a more complete sigma model in which a smoothing parameter is defined for each class and each variable ($k \times p$ smoothing parameters). This sigma model has been called *SEPCLASS*. In this study the performances of the PNN using these three sigma models were compared.

MATERIALS AND METHODS

Sample collection

The aim of this study was the automatic characterization of four seed species by a colour image analysis system. It is known that the growing location and environmental conditions have a considerable effect on the visual appearance of the seeds. Some seeds belonging to the same species present a wide heterogeneity in their morphometrical and colour features.

Samples of four seed species were provided by a French national seed-testing station (Station Nationale d'Essais de Semences, Beaucouzé, France). The four studied species were chosen because they corresponded to a real seed purity analysis problem. Wild oat and rumex seeds are very dangerous for crops and the European standards[21] include a rigorous identification of these wild seeds. For each kind of seed lot to be commercialized, these European standards give the maximum level of foreign material and adventitious seeds which is allowed in the lot. Seeds of red fescue (*Festuca rubra* L.), perennial rye grass (*Lolium perenne* L.), wild oat (mixture of three varieties: *Avena fatua* L., *Avena pubescens* L. and *Avena sterilis* L.) and rumex (mixture of three varieties: *Rumex crispus* L., *Rumex longifolius* L., and *Rumex obtusifolius* L.) were randomly picked from lots. The seeds of *Rumex longifolius* L. may be with or without an envelope, which drastically changes their external appearance. Red fescue and perennial rye grass are cultivated seed species, whilst wild oat and rumex are adventitious seeds which may devastate crops. Discrimination between the three varieties of rumex is not achieved in purity analysis of seeds, nor between the three varieties of wild oat.

Image acquisition

A set of numerical images was acquired by a high resolution three-CCD camera (KY-F55B, JVC Corp., Japan). A 35 mm macrolens (Nikkor AF, Nikon, Japan) was fitted on the camera. Lighting was achieved by two 18 W neon lamps placed at either side of the working surface. The camera surface was about 45 cm away from the working surface. Once the images were captured, they were digitized

by a frame grabber (VP1300-768-E-AT, Imaging Technology Inc., Bedford, U.S.A.). An image was just a matrix of size 512 rows by 768 columns, which represented a spatial resolution of 6 cm $\times$ 9 cm. Each pixel in a colour image is represented by a three-dimensional vector corresponding to red, green and blue luminance values. The luminance values at the co-ordinates of each pixel ranged from 0 to 255.

A total of 600 seeds were available for each of the four species. Sets of seeds were placed in the field of the camera in random orientation and in non-touching positions. The number of seeds which could be placed together in the field of the camera depended on the average size of each kernel. For seeds of small size, such as perennial rye grass, red fescue and rumex, a single image contained about 100 seeds whereas an image of wild oat kernels included only 20 seeds. Forty-eight images of sets of seeds were acquired and stored for further treatment.

Any colour can be reproduced by mixing an appropriate set of three primary colours. A few primary colour spaces have been proposed during the past, all of which attempt to represent the trichromatic nature of light for different purposes. In machine vision the popular choices of colour primaries are RGB (red, green and blue) co-ordinates or HSI (hue, saturation and intensity) co-ordinates. In this study, the RGB representation was used and no transformation was applied to enhance colour differences within the colour space.

Image processing

The initial treatment of the images consisted of reducing the noise. For this purpose a median filter[22] was applied on the images. In order to isolate the objects from the background, the colour images were binarized by a spatial segmentation technique.[23] The binarization algorithm proceeds as follows.

(i) Binarize the colour image into two regions by using an initial random threshold vector value. We could use the average grey levels for the three channels of the whole colour image.
(ii) Calculate the respective mean grey level $\langle \mathbf{z}_i \rangle$ and the number of pixels n_i of each region, $i \in \{1, 2\}$.
(iii) Calculate the spatial threshold value

$$t = \tfrac{1}{2}(\langle \mathbf{z}_1 \rangle + \langle \mathbf{z}_2 \rangle) + \frac{\langle \mathbf{z}_1 \rangle - \langle \mathbf{z}_2 \rangle}{2 \log(n_1 + n_2)} \log\left(\frac{n_2}{n_1}\right)$$

(iv) For each point z_0 in the image evaluate the quantity

$g = \frac{1}{9} \sum_{p=0}^{8} z_p$, where z_0, z_1, ..., z_8 are the observed grey levels at points in the 3×3 neighbourhood window of z_0. If g is less than t (i.e. $\forall k \in \{1,2,3\}$, $g_k < t_k$, where k is the index of the colour channel), then allocate z_0 to region 1, otherwise to region 2.
(v) Go to (ii) until the threshold is stable.

In order to characterize individual seeds, 73 features were then measured on each seed. These features included 25 size and shape parameters measured from the binarized image (Table 1) and 48 (16×3 colour channels) texture features (Table 2). The size and shape features are independent of grey value statistics and were therefore extracted only from the binarized image. However, the texture features were measured independently from the red, green and blue channels. Each texture feature was therefore a three-dimensional vector representing the values of the texture on the three primary

Table 1. Size and shape features used for seed characterization. All these features were measured from binarized image

Feature	Interpretation
Area	Sum of all pixels in the region of a selected seed
Perimeter	Sum of all pixels on the boundary of a selected seed
Length	Maximum length of the seed through the centroid
Width	
Thinness ratio (or circularity)	
Elongation	Ratio of the width to the length of a seed
First ten magnitude Fourier descriptors	
Seven invariant moments	
Eccentricity	
Spread	

channels. All the features were normalized in order to have invariance to rotation and translation of the seeds. The Fourier descriptors are attractive features which are assessed by the application of a monodimensional fast Fourier transform to the seed contour.[24] The number of Fourier descriptors depends on the length of the contour of the seed. Moreover, the phase components contained useless information about the orientation of the seed. Only the first ten magnitude components were kept. Image texture includes coarseness, fineness, regularity, irregularity, etc. These terms relate to the spatial distribution of the luminance signal in a neighbourhood. A region in which the grey levels change slowly with distance is characterized by a coarse texture. In the case of a fine texture, the grey levels tend to change rapidly with distance. The texture was characterized by three well-known methods: local histograms,[25] grey level co-occurrences,[26] and grey level run lengths.[27] The local histograms are obtained from the monodimensional distribution of the grey levels of the seeds. In order to summarize these histograms, parameters such as mean, variance, energy, kurtosis and skewness were extracted. Grey level co-occurrence and grey level run length approaches are based on the assessment of bidimensional distributions of the grey levels.

Variable selection

All the measured features were gathered into a matrix of size 2400 rows by 73 columns. Each row of the matrix represented a particular seed and each column represented a measured feature. This matrix was randomly divided into two matrices: a matrix of size 1600×73 for the training set and one of size 800×73 for the test set. Some variables could be highly correlated and therefore present no discriminant ability. It was worth selecting a relevant subset of variables. To this end we used stepwise discriminant analysis (SDA), which introduced the variables in a stepwise manner by maximizing a specific criterion.[28] Let $\mathbf{T}$ be the total covariance matrix of the training set and $\mathbf{B}$ the 'between' matrix which describes the variations between the groups. At each iteration, SDA introduces the variable which maximizes the trace of the matrix defined by $\mathbf{T}^{-1}\mathbf{B}$. At the mth iteration, m among p variables have already been selected by SDA. The procedure assesses the traces of all the matrices $\mathbf{T}^{-1}\mathbf{B}$ which can be obtained by using the m previously selected variables and one among the $p-m$ remaining variables. In this way the traces of $p-m$ matrices of the form $\mathbf{T}^{-1}\mathbf{B}$ are assessed. These traces are denoted t_i (i ranging from 1 to $p-m$). All the traces t_i are then compared. At iteration $m+1$ the variable which gives the maximum trace is selected. The procedure of SDA is able to order the variables according to their discriminant abilities. However, it gives no simple criterion to make an end of the variable selection at a given step. It was supposed that when the maximum value of t_i did not notably

increase from one step to another, the optimal subset of variables was found. In order to have a variable selection criterion which is independent of the step of SDA, we defined a relative variable selection criterion v as the ratio of the maximum trace to the sum of all the traces:

$$v = \frac{\max_{i \in \{1,2,\ldots,p-m\}} t_i}{\sum_{i=1}^{p-m} t_i} \tag{13}$$

From the examination of the evolution of v according to the step of SDA, a subset of relevant variables was selected and used as input for the tested networks. In this way a subset of variables was formed.

Table 2. Texture features used for seed characterization. Each of these features is a three-dimensional vector

	Feature	Interpretation
Local histogram features	Mean grey level	Gives the variance of the histogram. It is minimized when histogram elements are as equal as possible. This arises from a very homogeneous seed surface.
	variance	
	energy	
	Entropy	Measures the homogeneity of the histogram. It is maximized for uniform histograms
	Kurtosis	Characterizes the degree of asymmetry of the histogram around its mean
	Skewness	Measures the relative peakedness or flatness of the histogram
Grey level co-occurrence matrix features	Energy (or angular second moment)	Gives the variance of the matrix. It is minimized when matrix elements are as equal as possible. This arises from a very homogeneous textured surface
	Contrast	Gives the inertia of the matrix according to its main diagonal
	Correlation	Measure the resemblance between lines (respectively columns). It is maximized when values are uniformly distributed over the matrix
	Entropy	Gives a measure of the homogeneity of the matrix elements. It is maximized for a uniform matrix
	Inverse difference moment	Gives high values if elements are concentrated around the main diagonal of the matrix. This occurs in images with very smooth transitions in grey levels
Grey level run length matrix features	Short-run emphasis	Emphasizes the short run existing in the image. It gives high values for complex texture
	Long-run emphasis	Emphasizes the long run lengths of an image. Gives high values for homogeneous texture
	Grey level distribution	Gives high values if frequencies of occurrence of run length are distributed over very few grey levels
	Run length distribution	Gives high values if frequencies of occurrence of run length are distributed over very few run lengths. It should have its lowest value for a seed with the most linear structure
	Run percentages	Low for homogeneous texture

Classification by neural networks

The two neural networks were used as pattern classifiers. The MLPN had four inputs (representing the number of selected variables), one hidden layer with four units, and four outputs. Each output corresponded to one of the four qualitative groups representing the seed species. Each output node took values ranging from -1 to 1. The transfer function was the sigmoid. The momentum factor (α in equation (4)) and the learning coefficient (η in equation (3)) might have an important effect on the classification performances. In order to study their effects, they were varied from 0 to 1 in steps of 0·1. The MLPN was performed with the commercial software (NeuralWorks Explorer, NeuralWare Inc., Pittsburgh, U.S.A.).

The PNN models were implemented with the standard multivariate Gaussian function as a kernel function. The four studied species were considered to have the same *a priori* probabilities. Three models of the PNN were tested: *BASIC*, *SEPVAR* and *SEPCLASS*. For the *BASIC* sigma model, which needs a single smoothing parameter σ, the effect of values of σ on the classification performances was systematically studied by varying this parameter from 0·0001 to 40. In the second stage the golden section technique[29] was applied to assess the optimal value of σ. For the *SEPVAR* and *SEPCLASS* sigma models, the conjugate gradient technique[29] was used to estimate the optimal values of the smoothing parameters. Results of the MLPN and the PNN were compared with regard to their ability to classify both the training and the test sets.

RESULTS

Image examination of seeds

The seeds showed some differences in their appearance: see Figure 3. This figure represents the blue channel images of typical seeds. In order to simulate a real analysis of seeds, seeds were placed in random orientations and non-touching positions. This was possible because all the measured features were invariant to rotation and translation of the seeds.

The red fescue and perennial rye grass seeds presented approximately the same elongated shape and brown colour. It is not a trivial task to discriminate between them even by visual examination. The seeds of wild oat were larger and presented large variations in morphology and colour which ranged from pure yellow to brown. They also might present some pelosity. The rumex seeds presented two typical different appearances. In most cases the envelopes of rumex seeds were absent and the seeds showed a small and regular lozenge shape. The colour was almost evenly dark brown. In some other cases, the envelopes remained linked to the seeds. The size of rumex seeds with their envelopes was larger and the colour was lighter.

Variable selection by stepwise discriminant analysis

SDA was applied for the selection of a subset of relevant variables from the 73 measured ones. Many of the measured features were correlated. Figure 4 shows the evolution of the selection criterion v defined in equation (13) in relation to the number of introduced variables. This criterion decreased up to four introduced variables and then remained almost constant. This meant that only four variables, namely elongation, length, blue channel skewness (BS) and long run emphasis of the blue channel (BLRE), were relevant. The elongation is the ratio of the width to the length of the seed. This parameter is close to unity if the seed is circular. BS characterizes the degree of asymmetry of the grey level histogram of the blue channel. BLRE describes the homogeneity of the texture. It gives a high value for a homogeneous texture. The four selected variables corresponded to different kinds of

parameters which respectively describe the size, shape, local histogram and texture. Figure 5 shows the biplot of the first two selected variables, namely elongation and length. In this figure each point represents an individual seed. Adventitious species (rumex and wild oat) were well-separated from cultivated ones (red fescue and perennial rye grass). The representations of seeds of red fescue and perennial rye grass completely overlapped. This was to be expected because they presented almost the same morphometrical and colour features. Wild oat presented a wide range of variations in features and occupied a large area on the map. The representation of seeds of rumex could be divided into two classes which corresponded to the presence or absence of envelopes. BS and BLRE seemed to play a less important role in discrimination.

Classification with neural networks

The MLPN was tested with the previously defined training and test sets. The back-propagation algorithm was used to select the optimal weights. The MLPN was trained with 1600 seeds and tested on 800 other seeds. In order to study the relation between the training time and the classification performances, we trained the network by applying the training set many times. Each application of the whole training set is called a 'pass'. Furthermore, the momentum factor and learning coefficient values were varied from 0 to 1. The error probability of the classification of the training set was assessed for

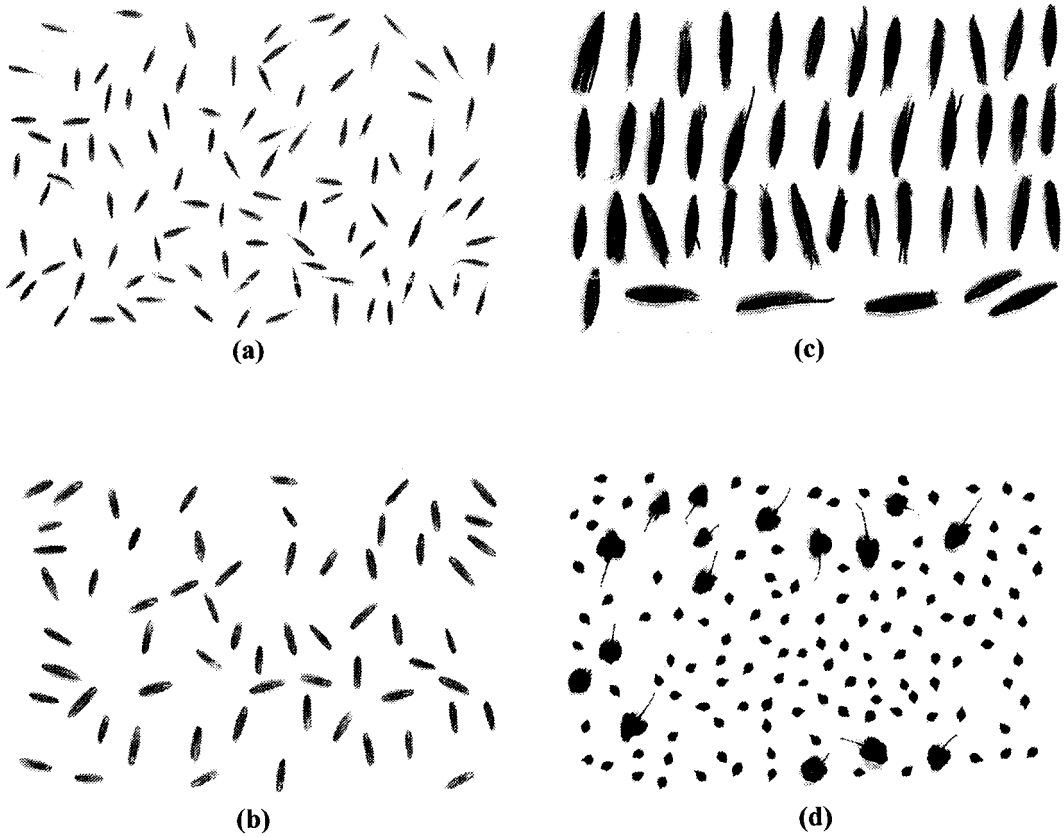

Figure 3. Example of blue channel images of seeds: (a) red fescue; (b) perennial rye grass; (c) wild oat; (d) rumex

each value of these parameters. Figure 6 showed that these parameters have a great effect on the classification results. For example, only 35 among 1600 seeds were misclassified when the momentum factor was set to 0·9 and the learning coefficient was set 0·9. However, 400 among 1600 seeds were not correctly classified when the momentum factor was 0·5 and the learning coefficient was 0·1. From Figure 6 it was shown that for this particular problem the optimal values of both the momentum and the learning coefficient were 0·9. These optimal values were used in the following experiments.

Figure 7 shows the percentage of misclassification as a function of the number of passes. The misclassifications for both the training and test sets dropped drastically from one to ten passes. With only one pass the network misclassified 401 and 200 seeds for the training and test sets respectively. The performances of the MLPN were almost constant when the number of passes exceeded ten. The classification results were therefore highly dependent on the number of passes through the training set. Table 3 gives, for the training and test sets, the confusion table in the case when the number of passes was equal to ten. The rows of this table represent the actual species of seeds and the columns the species predicted by the MLPN. Each block contains the number of corresponding patterns. The diagonal terms, showing the numbers of seeds correctly classified, are the most important figures. Classification errors were 44 among 1600 seeds for the training set and 28 among 800 seeds for the test set. Confusions were essentially between red fescue and perennial rye grass. As the appearances of these seeds were rather comparable, these confusions were not surprising. It must be noticed that some seeds of wild oat were classified as cultivated seeds. These misclassifications are costly in purity analyses, because wild oat seeds are injurious for field cultivation.

On the same data collection, three sigma models (*BASIC*, *SEPVAR* and *SEPCLASS*) of PNN were investigated. The optimal value of the smoothing parameter is problem-dependent. The *BASIC* model

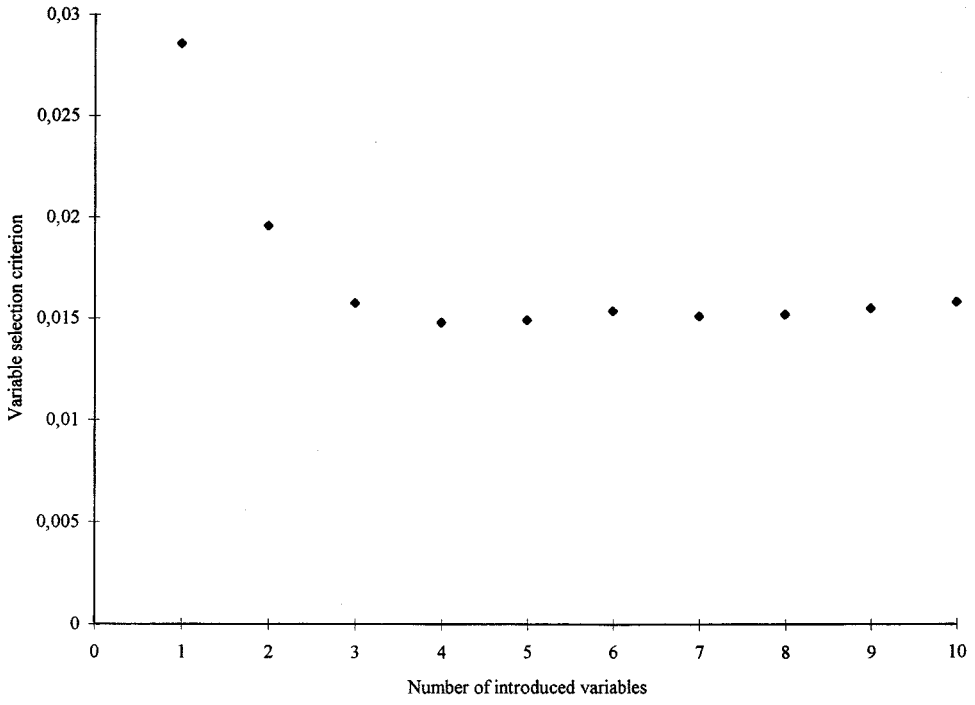

Figure 4. Variation in variable selection criterion as a function of number of variables introduced by stepwise discriminant analysis

requires the estimation of a single smoothing parameter σ. In a first experiment the value of σ was varied from 0·0001 to 40. Figures 8(a) and 8(b) show how the percentage of misclassification of the training set varied in relation to σ. The percentage misclassification was highly dependent on the value of σ . This parameter took relevant values only in a very narrow range from about 0·1 to 0·2 (Figure 8(b)). When σ was outside this interval, the percentage misclassification increased drastically and could reach as much as 75, which corresponded to random classification. This shows the high importance of optimizing the value of σ. A 'golden section technique' was applied for automatically finding the best value of σ. For this sigma model, 18 among 1600 seeds and 33 among 800 seeds were misclassified for the training and test sets respectively.

As the *SEPVAR* sigma-model required the estimation of four smoothing parameters, it was not possible to independently study the effect of each of them. The optimal values of the smoothing parameters were estimated by the conjugate gradient technique. The *SEPVAR* sigma model gave the worst classification results, with as many as 27 and 37 errors for the training and test sets respectively.

The SEPCLASS sigma model gave the best classification results. There were only 17 and 19 misclassified seeds for the training and test sets respectively. Table 4 shows the confusion table for this model. All the confusions were between seeds of red fescue and perennial rye grass. It should be noticed that all the adventitious seeds were correctly identified. The comparison between the four tested neural networks is summarized in Figure 9. The bars represent the numbers of errors for each of the PNN sigma models and for the MLPN. If the results obtained for only the test set were

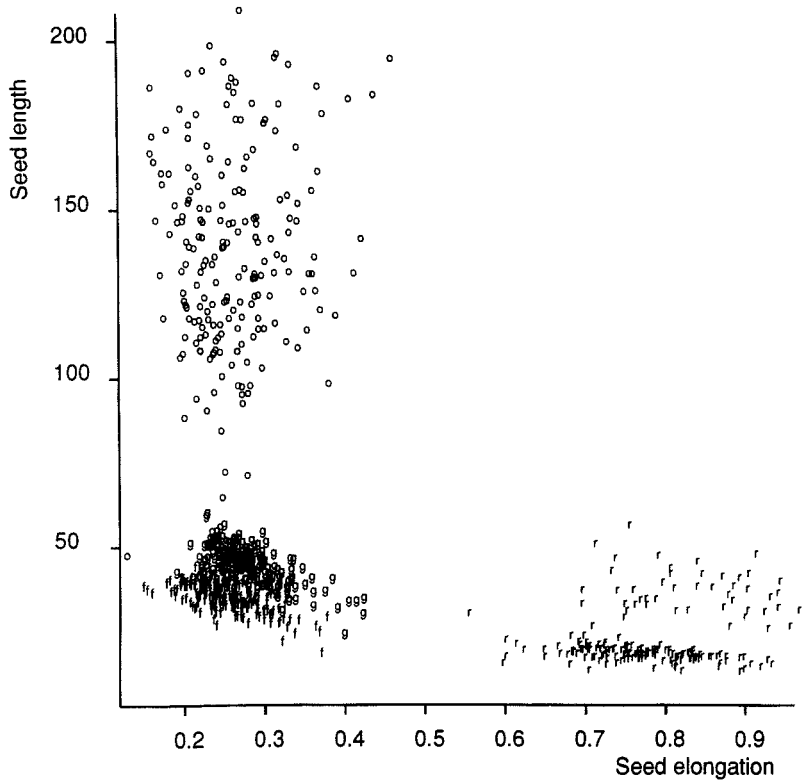

Figure 5. Biplot of first two variables selected by stepwise discriminant analysis. Each seed is represented by a single point: f, red fescue; g, perennial rye grass; o: wild oat; r, rumex

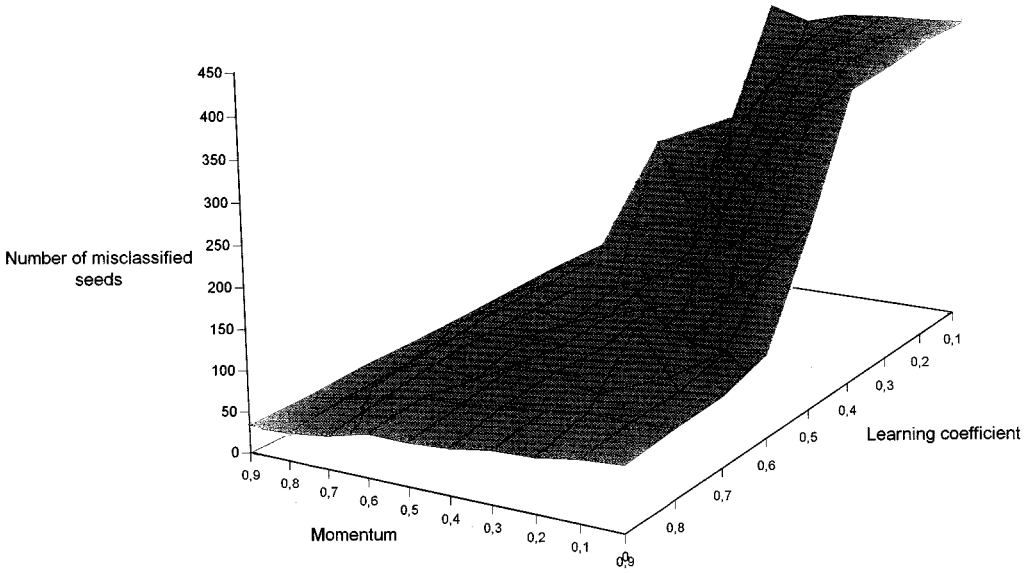

Figure 6. Variation in classification of training set with respect to values of the momentum factor and learning coefficient of MLPN

considered, the methods could be classified according to their decreasing performances in the order *SEPCLASS*, MLPN, *BASIC*, *SEPVAR*. In this study the PNN with the *SEPCLASS* sigma model outperformed the MLPN and the other two PNN sigma models. For the SEPCLASS sigma model the

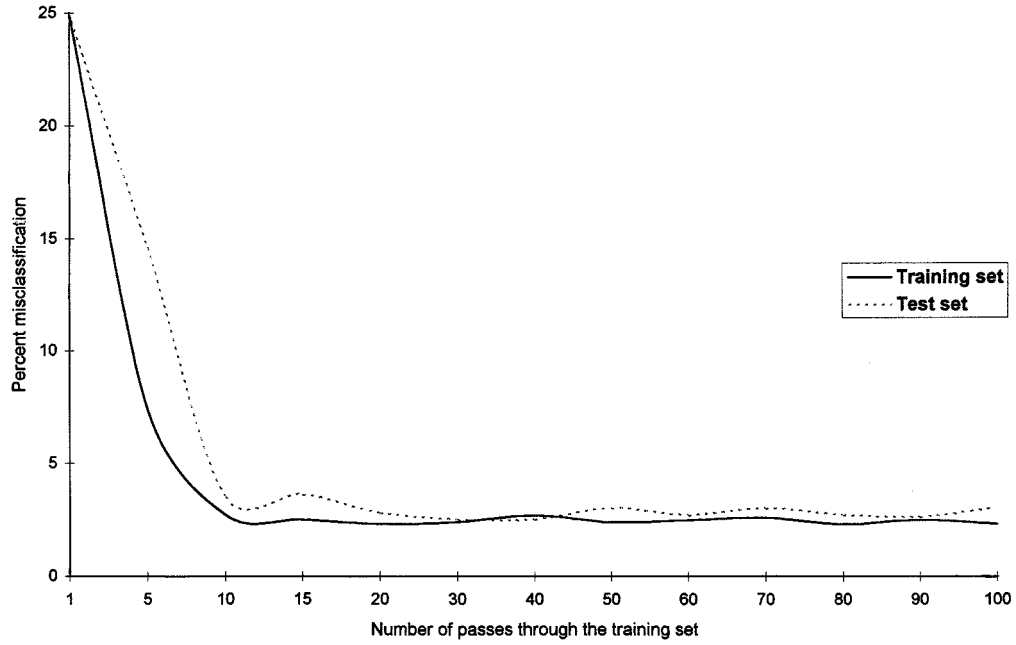

Figure 7. Classification results by MLPN for training and test sets as a function of number of passes through training set

Table 3. MLPN confusion table for training and test sets. Results are given for ten passes through training set

	Predicted species							
	Training set				Test set			
Actual species	Red fescue	Perennial rye grass	Wild oat	Rumex	Red fescue	Perennial rye grass	Wild oat	Rumex
Red fescue	385	14		1	188	12		
Perennial rye grass	27	373			15	185		
Wild oat		2	398		1		199	
Rumex				400				200

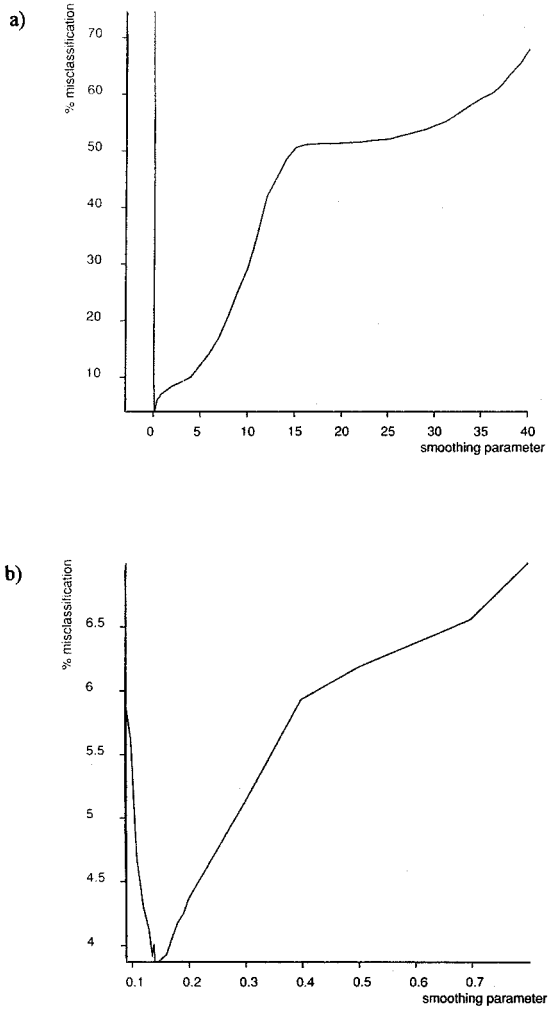

Figure 8. Effect of σ on misclassification percentage for PNN with *BASIC* sigma-model: (a) smoothing parameter ranging from 0·0001 to 40; (b) detail of same curve for smoothing parameter ranging from 0·1 to 0·6

Table 4. PNN confusion table for training and test sets. Results are given for *SEPCLASS* sigma model

	Predicted species							
	Training set				Test set			
Actual species	Red fescue	Perennial rye grass	Wild oat	Rumex	Red fescue	Perennial rye grass	Wild oat	Rumex
Red fescue	396	4			195	5		
Perennial rye grass	13	387			14	186		
Wild oat			400				200	
Rumex				400				200

smoothing parameters were dependent on the measured features and populations. It was therefore not unreasonable to associate its own smoothing parameter with each class and with each measured feature. In this sense the *SEPCLASS* sigma model takes into account the variability which may exist between the measured features of the available qualitative groups. Moreover, the training phase of *SEPCLASS* was less time consuming than that of the MLPN.

DISCUSSION AND CONCLUSIONS

The performances of two network topologies were compared on the basis of a practical pattern recognition problem. The results of the discrimination of four species of seeds from their colour image

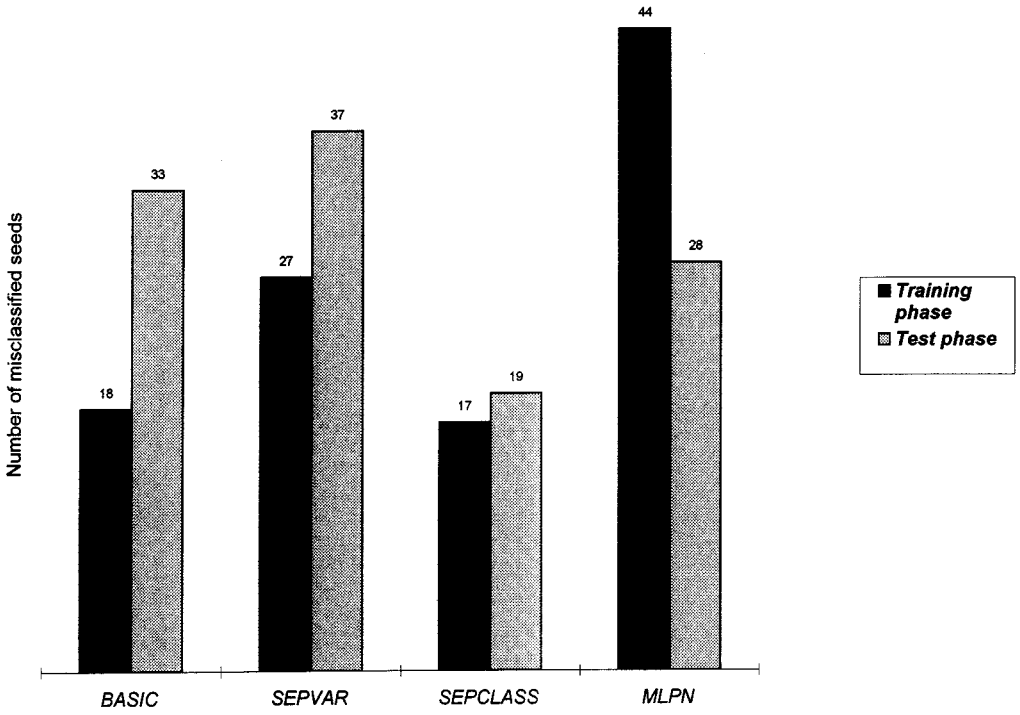

Figure 9. Comparison between performances of three PNN sigma models and MLPN. Results of MLPN are given for ten passes through training set. Training set size, 1600 seeds; test set size, 800 seeds

features by the PNN were better than those obtained by the MLPN. The PNN with the *SEPCLASS* sigma model configuration classified correctly 98·93% and 97·62% for the training and test sets respectively, whereas the MLPN gave 97·25% and 96·5% percent of correct classifications. All the adventitious seeds were correctly identified by the PNN.

The MLPN uses the training patterns only for the optimization of the weights. Once the network is trained, the training patterns are no longer used. One drawback of the MLPN is the requirement to define the optimal network topology. There is no algorithm which automatically defines the optimal MLPN structure for a given problem. Moreover, the back propagation is a heuristic algorithm, which makes it possible to gradually approach the solution and usually requires a lot of computation time. Many passes through the training set are required to adjust the network weights. Even if it has been proven that the back propagation always converges, it can stick at a local minimum. Moreover, we do not really understand how it converges. For this reason, this algorithm has never been used in applications which require a lot of reliability. It is impossible to examine the trained network in order to analyse both the predictive importance and the respective role of each variable. In contrast, the PNN, since it is based on the Bayesian classification rule, makes it possible to mathematically assess the error probability values.

In the PNN all the training patterns take part in the classification process of an unknown pattern. It is required to retain the whole training data. On the other hand, all the PNN scaling parameters can be automatically assessed using the training set, whilst there is no algorithm to select the optimal structure of the MLPN. If new learning patterns are available, they can be added to the pattern layer. It is generally not necessary to readjust the smoothing parameters and the updating of the network is immediate. The PNN is based on the Bayesian decision rule and the confidence figures can be easily computed. This is the main advantage of the PNN with respect to other neural network models.

Image analysis in combination with a probabilistic neural network showed promise for designing an automatic system for seed discrimination. It can be an alternative to the current manual seed purity analysis. The measurement of new relevant seed features, e.g. qualitative features, may substantially improve seed recognition. Further research is also needed to reduce the size of the training set of the PNN. This can be achieved by the application of a clustering algorithm on the predictive variables. In this way, only a consistent subset of the training set will be used.

ACKNOWLEDGEMENTS

The authors wish to thank D. Demilly and M. Marino for providing the samples and also for their technical assistance.

REFERENCES

1. S. M. Peeling, R. K. Moore and M. J. Tomlinson, in *Proc. IOA Autumn Conf. on Speech and Hearing*, (1986).
2. R. P. Lippmann, *Neural Comput.* **1**, 1–38 (1989).
3. R. P. Lippmann, *IEEE ASSP Mag.*, April, 4–22 (1987).
4. J. J. Hopfield, *Proc. Natl. Acad. Sci., USA*, **79**, 2554–2558 (1982).
5. D. E. Rumelhart and J. L. McClelland, *Parallel Distributed Processing*: *Explorations in the Microstructure of Cognition*, MIT Press, Cambridge, MA (1986).
6. D. F. Specht, *Neural Netw.* **3**, 109–118 (1990).
7. J. McClelland and D. Rumelhart, *Explorations in Parallel Distributed Processing*, MIT Press, Cambridge, MA (1988).
8. A. M. Mood and F. A. Graybill, *Introduction to the theory of Statistics*, Macmillan, New York (1962).
9. C. W. Wrigley, J. C. Autran and W. Bushuk, *Adv. Cereal Sci. Technol.* **5,** 211–259 (1981).
10. D. G. Myers and K. J. Edsall, *Plant Var. Seeds*, **2**, 109–116 (1989).
11. Y. Chtioui, D. Bertrand, Y. Dattée and M. F. Devaux, *J. Sci. Food Agric.* **71**, 441–443 (1996).

12. I. Zayas, Y. Pomeranz and F. S. Lai, *Cereal Chem.* **66**, 233–237 (1989).
13. P. E. H. Petersen and G. W. Krutz, *Seed Sci. Technol.* **20**, 193–208 (1992).
14. K. Funahashi, *Neural Netw.* **2**, (1989).
15. D. F. Specht, in *Proc. IEEE Int. Conf. on Neural Networks*, Vol. 1, pp. 525–532, IEEE, New York (1988).
16. T. M. Cover, in *Frontiers of Pattern Recognition*, ed. by S. Watanabe, pp. 83-98, Academic, New York (1972).
17. E. Parzen, *Ann. Math. Stat.* **33**, 1065-1076 (1962).
18. T. Cacoullos, *Ann. Inst. Stat. Math.* (*Tokyo*), **18**, 179–189 (1966).
19. H. Schioler and U. Hartmann, *Neural Netw.* **5**, 903–909 (1992).
20. D. Specht, in *Proc. Int. Joint Conf. on Neural Networks*, Baltimore, MD, USA, 1992.
21. S. R. Draper, *Seed Sci. Technol.* **13**, Suppl. 2, 1–61 (1985).
22. R. C. Gonzalez and P. Wintz, *Digital Image Processing*, Addison-Wesley, Reading, MA (1987).
23. K. V. Mardia and T. J. Hainsworth, *IEEE Trans. Patt. Machine Intell.* **PAMI10**, 919–927 (1988).
24. C. T. Zahn and R. Z. Roskies, *IEEE Trans. Comput.* **C-21**, 269–281 (1972).
25. M. D. Levine, *Vision in Man and Machine*, McGraw-Hill, New York (1985).
26. R. M. Haralick, K. Shanmugan and I. Dinstein, *IEEE Trans. Syst., Man, Cyber.* **SMC-3**, 610–621 (1973).
27. M. M. Galloway, *Comput. Graph. Image Process.* **4**, 172–179 (1975).
28. J. M. Romeder, *Méthodes et Programmes d'Analyse Discriminante*, Dunod, Paris (1973).
29. W. H. Press, B. Flannery, S. Teukolsky and W. Vetterling, *Numerical Recipes in C*, Cambridge University Press, New York (1992).

