Pergamon

Neural Networks

Neural Networks 11 (1998) 283–296

Contributed article

# Evolution and generalization of a single neurone: I. Single-layer perceptron as seven statistical classifiers

Šarūnas Raudys*

*Institute of Mathematics and Informatics, Akademijos 4, Vilnius 2600, Lithuania*

Received 3 January 1997; accepted 5 July 1997

**Abstract**

Unlike many other investigations on this topic, the present one considers the non-linear single-layer perceptron (SLP) as a process in which the weights of the perceptron are increasing, and the cost function of the sum of squares is changing gradually. During the backpropagation training, the decision boundary of of SLP becomes identical or close to that of seven statistical classifiers: (1) the Euclidean distance classifier, (2) the regularized linear discriminant analysis, (3) the standard Fisher linear discriminant function, (4) the Fisher linear discriminant function with a pseudoinverse covariance matrix, (5) the generalized Fisher discriminant function, (6) the minimum empirical error classifier, and (7) the maximum margin classifier. In order to obtain a wider range of classifiers, five new complexity-control techniques are proposed: target value control, moving of the learning data centre into the origin of coordinates, zero weight initialization, use of an additional negative weight decay term called ''anti-regularization'', and use of an exponentially increasing learning step. Which particular type of classifier will be obtained depends on the data, the cost function to be minimized, the optimization technique and its parameters, and the stopping criteria. 

*Keywords:* Single-layer perceptron; Statistical classification; Generalization error; Initialization; Overtraining; Dimensionality; Complexity; Sample size; Scissors effect

## 1. Introduction

The single-layer perceptron (SLP) is the key elementary component in multilayer feedforward networks used to solve real-world problems. The behaviour of a single neurone serves as the simplest prototype model for studying the characteristics of more general non-linear models such as multilayer perceptrons. Despite the fact that the SLP has been studied for quite a few years, there are still some phenomena which remain to be studied. These are, for example, the generalization error, dynamics of learning, regularization terms and overtraining. It is reasonable to study these problems first by using the very simple model of the single-layer perceptron. The non-linear SLP has much in common with a variety of conventional linear classification algorithms. Therefore it is important to follow parallels between statistical and neural net approaches, to find out which result from statistics can be used in neural net analysis.

More than two hundred algorithms for designing statistical classification rule have been proposed in the literature on statistical pattern recognition and discriminant analysis (DA). We mention below only those that are similar to the perceptron design algorithm, and can be useful as a source of practical results and interesting ideas for further investigations. The first known classification rule is the standard linear discriminant function (DF) proposed by Fisher in 1936 (Fisher, 1936). Anderson (1951) has shown that it can be obtained from optimal statistical decision function theory. There we assume multivariate Gaussian classes with a common covariance matrix (GCCM), and insert maximum likelihood sample estimates instead of unknown parameters into the model. Koford and Groner (1966) have shown that when we have the same number of learning vectors from both competing classes, the adaline linear classifier of Widrow and Hoff (1960) is identical to the Fisher linear rule. The Fisher rule requires a sample-based covariance matrix to be inverted. In the case of a small learning set and large dimensionality, the situation becomes problematic. One of the solutions is to use a pseudoinverse of the sample covariance matrix (Schurmann, 1977; Malinovskij, 1979; Duin, 1995). A variety of structures of the covariance matrix, including a diagonal, a block diagonal,

* Requests for reprints should be sent to Šarūnas Raudys. E-mail: raudys@ktl.mii.lt.

0893–6080/98/$19.00 
*PII* S0893-6080(97)00135-4

tree-type dependence models and others, have been proposed to overcome this kind of difficulty (see, e.g., review in Raudys, 1991). One of the most successful solutions is the addition of a small positive constant to diagonal elements of the covariance matrix. This technique was first used in regression (Hoerl and Kennard, 1970) and later in discriminant analysis (Di Pillo, 1979; see also Friedman, 1989; McLachlan, 1992) under the title of ''regularized discriminant analysis''.

The main objective in classifier design is to obtain a classifier that results in the minimum number of misclassification errors. Anderson and Bahadur (1962) obtained the optimal linear DF to classify two multivariate Gaussian populations with different covariance matrices. Their result was generalized to a more general class of distribution densities by Patterson and Mattson (1966). When one does not have the analytical form of distribution densities, one needs to use the learning-set data and minimize the empirical (learning-set) error rate. Amari (1967) suggested using a soft limiting ''arctangent'' function whose closeness to the threshold function can be controlled by a parameter. Do-Tu and Installe (1978) suggested to change this parameter gradually. Then the pattern error function (a contribution of an individual learning-set vector to the cost) is smooth at the beginning, but later on approaches the threshold function, and finally minimizes the number of empirical errors. A similar approach was used by Pietrantonio and Jurs (1972). A variety of other algorithms are based on the use of a sequential random search (Wolf, 1966), the Kiefer–Wolfowitz stochastic approximation (Yau and Schumpert, 1968), linear programming (Ibaraki and Muroga, 1970), the algebraic method (Warmack and Gonzales, 1973), linear transformations of the coordinate system (Miyake, 1979) and heuristic ideas (Vapnik and Chervonenkis, 1974). Ad hoc principles are used to overcome numerical difficulties.

When zero empirical classification error is obtained, the resulting discriminant function is no longer unique. Some additional criteria are introduced which favour an increase in the distance between the discriminant hyperplane and the learning-set vectors nearest to it. Examples are the tolerance function (Babu and Chen, 1971) and the ''margin'' — the Euclidean distance of the nearest learning-set observations from the separating hyperplane [Glucksman, 1966; Vapnik and Chervonenkis, 1974 (generalized portrait); Duin, 1995 (small learning sample classifier); Boser et al., 1992 (maximal margin classifier); Cortes and Vapnik, 1995 (support vector machine)].

Typically, in the study of the generalization error, the single-layer perceptron is analysed as a separate special specimen of the classification algorithm. Most often the activation function (or the pattern error function) is assumed to be the linear or the threshold function; sometimes it is assumed to be a soft limiting one. *In contrast with other investigations, the present paper does not consider the non-linear SLP as a single classifier*. We investigate the SLP in action. We stop and analyse the perceptron at different moments of the learning process. We regard the SLP classifier as developing during the training process. It can adapt to the complexity of the pattern recognition problem. On the way between the starting point and the minimum of the cost function, the weights of the perceptron are increasing. The actual slope of the activation function changes gradually. Therefore, the decision boundary of the SLP can become identical or close to that of seven classifiers, analysed in the statistical pattern recognition and discriminant analysis. The objective of this article is to show how a significant number of results from standard multivariate statistical analysis can be used in the generalization error analysis of simple artificial neural nets.

The article is split into two parts. In Part I, we show that the non-linear SLP is not a single classifier but a process. In Part II, we analyse the small learning-set properties of several well-known statistical classifiers that can be detected in the non-liner SLP training process. We analyse the former and new results from a fresh, unique point of view, using the terminology popular in the statistical mechanics approach, and demonstrate how theoretical results concerned with statistical classifiers can be used for purposeful control of SLP complexity.

Part I of the article is organized as follows. In Section 2, we enumerate the main results of the whole article. In Section 3, we present five parametric statistical classifiers based on the class distribution densities, and two non-parametric classifiers based on the decision rule approach. In Section 4 we show that in the backpropagation (BP) training of the non-linear SLP classifier, the weights gradually increase, and the cost function changes gradually. In Section 5 we show analytically that in adaptive SLP training, one can obtain decision boundaries for seven different statistical classifiers. Section 6 contains experimental illustrations, and Section 7 a discussion.

## 2. Main results

We analyse the non-linear SLP that has $p$ inputs $x_1,x_2\ldots,x_p$ and one output calculated according to the following equation

$$output = o\left(\mathbf{w}'\mathbf{x} + w_o\right) \tag{1}$$

where $w_o,\mathbf{w} = (w_1,w_2,\ldots,w_p)'$ are weights of the discriminant function and $o(g)$ is a non-linear activation (transfer) function. To simplify analytical formulae in our analysis we use the ''tanh'' function

$$o(g) = \tanh(g) = \left(e^{g} - e^{-g}\right)/\left(e^{g} + e^{-g}\right) \tag{2}$$

However, in simulation experiments, we used software with the closely related *sigmoid activation function*, $o(g) = 1/(1 + \exp(-g)) = (\tanh(g/2) + 1)/2$.

The SLP actually performs the classification of a vector $\mathbf{x} = (x_1,x_2,\ldots,x_p)'$ into one of two classes: $\pi_1$ and $\pi_2$. It is, in

fact, a modification of the prediction (regression) problem for the ''0–1'' loss function. We consider the case where perceptron weights are found in an iterative training procedure, where the following cost of the sum of squares

$$cost_l = \frac{1}{2(N_1+N_2)} \sum_{i=1}^{2} \sum_{j=1}^{N_i} \left( t_j^{(i)} - o\left( \mathbf{w}' \mathbf{x}_j^{(i)} + w_o \right) \right)^2 \qquad (3)$$

is minimized. In the above formula, $t_j^{(i)}$ is a desired output (a target) for $\mathbf{x}_j^{(i)}$, the $j$th learning-set observation vector from the $i$th class, $N_1$ is the number of learning vectors in $\pi_1$ and $N_2$ is the number of learning vectors in $\pi_2$. In regression the targets are continuous values. In classification, for the activation function (2), we usually use $t_j^{(1)} = 1$ and $t_j^{(2)} = -1$. We call these values *limiting* ones. Another choice is: $t_j^{(1)} = 0.8$ and $t_j^{(2)} = -0.8$. In simulations with the sigmoid function, we use $t_j^{(1)} = 0$ and $t_j^{(2)} = 1$ (limiting values), or $t_j^{(1)} = 0.1$ and $t_j^{(2)} = 0.9$ (recommended by Rumelhart et al., 1986). We mainly analyse the standard total gradient delta learning rule (backpropagation (BP)), where the weight vector is adapted according to the rule

$$\mathbf{w}_{(t+1)} = \mathbf{w}_{(t)} - \eta \frac{\partial cost_l}{\partial \mathbf{w}} \qquad (4)$$

with $\eta$ called a learning step.

In our analysis we have established the following facts.

1. When starting training from zero or small weights, the magnitudes of the weights are increasing up to certain values. These values depend on the separability of the learning-set vectors and on the target values: for well-separated learning sets with zero empirical errors and limiting target values, the magnitudes can increase without unbound, and for badly separated and/or non-limit targets the final weights are smaller.
2. The statistical properties of the SLP classifier depend on the cost function which, in turn, depends on the magnitudes of the weights. At the start the weights are small, and the activation function acts as a linear one. Then we obtain the adaline cost function which leads to the standard Fisher linear DF. With an increase in the magnitudes of the weights, the activation function begins to act as a non-linear one, and reduces contributions of learning-set observation vectors distant from the discriminant hyperplane. Then we get a rule similar to the generalized Fisher classifier which is robust to outliers (i.e., atypical observations). When the weights are very large, the contributions of all learning-set observation vectors become close to either $+1$ or $-1$. In that case, we have a classifier close to that which minimizes empirical (training) error. If the empirical error is zero, we can obtain the maximal margin classifier for the limiting target values.
3. The classifier obtained depends on the number of iterations. If:

- the centre of the data, $1/2(\overline{\mathbf{x}}^{(1)} + \overline{\mathbf{x}}^{(2)})$, is moved to the zero point,
- $t_2 = -t_1 N_1/N_2$ (for $t_2 = -t_1$ this means we have an equal number of training samples from both pattern classes),
- we start training from zero weights, $w_{0(0)} = 0$, $\mathbf{w}_{(0)} = \mathbf{0}$, and
- we use total gradient training (*conditions* **E**),then, after the first iteration, we have the Euclidean distance (the nearest means) classifier. In subsequent iterations we get the regularized DA, and move further towards the standard Fisher DF. We can obtain the Fisher DF when we have close non-limiting target values, e.g., $t_1 = -t_2 = 0.1$. If $n = N + N_2 < p$, we are approaching the Fisher DF with a pseudoinversion of the covariance matrix.

4. Target values are, probably, the most important factor determining the character of the learning curve. Starting and final values of the learning step are very important too. Other factors that affect the result are: the data, its separability and its prior transformations; the regularization or anti-regularization procedure used; the local minima and flat area of the cost function in a multivariate weight space; and, of course, the number of iterations. In principle, we can obtain seven statistical classifiers in SLP training.
5. In the thermodynamic limit, for two GCCM classes, the average generalization error of the Euclidean distance classifier (EDC), the standard (F) and regularized (RDA) Fisher linear DF are expressed by the equations (Raudys, 1967; 1972; Raudys and Skurikhina, 1994):

$$EP_N^{(\mathrm{EDC})} \approx \Phi\left\{ -\frac{\delta^*}{2} \frac{1}{\sqrt{T_\mu^*}} \right\}, \quad EP_N^{(\mathrm{F})} \approx \Phi\left\{ -\frac{\delta}{2} \frac{1}{\sqrt{T_\mu T_\Sigma}} \right\},$$

$$EP_N^{(\mathrm{RDA})} \approx \Phi\left\{ -\frac{\delta_\lambda}{2} \frac{\sqrt{1+\lambda T_\lambda}}{\sqrt{T_\mu T_\Sigma}} \right\} \qquad (5)$$

where

$$\Phi\{a\} = \int_{-\infty}^{a} (2\pi)^{-1/2} \sigma^{-1} \exp\left\{ -t^2/(2\sigma^2) \right\} \mathrm{d}t, \quad N = N_1 = N_2$$

$\delta^*$, $\delta$ and $\delta_\lambda$ are certain functions of the parameters of the GCCM data model that determine the separability of the pattern classes; the term $T_\mu = 1 + (2p/(\delta^2 N))$ arises from an inexact sample estimation of mean vectors of the classes; the term $T_\Sigma = 1 + (p/(2N - p))$ arises from an inexact sample estimation of a covariance matrix; $\lambda$ is a regularization constant, the term $T_\lambda$ is also a function of parameters of the GCCM model. This term is trying to reduce the negative influence of $T_\Sigma$. The first two asymptotic formulae are rather exact for $p > 10$, while the formula for the regularized DA is valid only for very small $\lambda$.

6. In general, the GCCM class model is ''unfaithful'' for EDC. Therefore we have a term $T_\mu^* = 1 + (2p^*/(\delta^{*2} N))$, with $1 < p^* < \infty$ (Raudys, 1967). The term $T_\mu^*$ indicates

that the small-sample properties of this classifier are highly affected by true distribution densities of the classes. This conclusion is also valid for other *parametric* classifiers: in ''unfaithful'' cases, the generalization of parametric statistical classifiers depends on the data; in certain cases, the parametric classifiers can be extremely sensitive to the learning-set size.

7. In the multivariate spherical Gaussian case, the average generalization error of the pseudo Fisher linear DF is given by

$$EP_N^{(\mathrm{PF})} \approx \Phi\left\{ -\frac{\delta\sqrt{r/p}}{2} \frac{1}{\sqrt{(1+\gamma^2)T_\mu + \gamma^2 \frac{3\delta^2}{4Np}}} \right\} \tag{6}$$

where $r = N_1 + N_2 - 2$ is the rank of the sample covariance matrix $\mathbf{S}$; $\gamma = \sqrt{V_d}/E_d$; $E_d$ and $V_d$ are the mean and variance of $1/d$; and $d$ is an arbitrary eigenvalue of $\mathbf{S}$ chosen at random. The average generalization error has a peaking behaviour: with an increase in the learning-set size $N$, it first decreases, reaches the minimum and then begins increasing. The minimum error is obtained for $N = p/4$ ($n = p/2$) and the maximum error (0.5 for equiprobable classes) is obtained for $N = p/2$.

8. In the multivariate spherical Gaussian case, an increase in the average generalization error of the zero empirical error (ZEE) classifier trained by the Gibbs algorithm is proportional to $(p/N)^S$, where parameter $S < 1$. It increases with $\delta$ and the ratio $N/p$, and approaches 1 asymptotically as sample size $N$ increases. The generalization error is:

$$MEP_N^{(ZEE)} \approx \Phi\left\{ -\frac{\delta}{2} \frac{1}{\sqrt{1 + (1.6 + 0.18\delta)\left(\frac{p}{N}\right)^{1.8-\delta/5}}} \right\} \tag{7}$$

This classifier can perform well even if $n < p$. For $n \gg p$, the Fisher classifier performs better. The correct prior information supplied in the form of the prior distribution $q_{prior}(\mathbf{w})$ of the weights reduces the generalization error dramatically.

9. Theoretical and experimental analysis of the Gibbs algorithm in the case of spherical Gaussian data shows that an increase in the margin width decreases the generalization error averaged over a variety of classification problems and learning sets. For one particular problem and one learning set, simulations with the SLP exhibit peaking behaviour.
10. The simulations with the spherical Gaussian and GCCM data confirm that, depending on the learning level, the small-sample properties of the non-linear SLP classifier are determined by one of the seven statistical classifiers analysed in this paper.
11. The overtraining effect is caused by two factors. First of all, there is a difference between the cost function surfaces, obtained from the learning-set data, and that obtained from the test-set data (general population). The greater the difference, the greater the overtraining effect that can be expected. Another factor is the change in type of statistical classifier that occurs with an increase in the number of iterations. One of these classifiers appears to be the best one in the finite learning-set size situation. Overtraining can occur when the weights are small and the activation function acts linearly. Then we move from the EDC towards the RDA and the Fisher classifier. Overtraining can also happen later, when the weights are large and the activation function acts non-linearly. Then we move from the Fisher classifier towards the generalized Fisher classifier, or from the ZEE classifier towards the maximal margin classifier.
12. For the GCCM model with several eigenvalues of the covariance matrix close to zero, the small-sample properties of the Euclidean distance, the zero empirical error and the non-linear SLP classifiers with non-limiting targets are determined by the intrinsic dimensionality of the data. This conclusion does not apply to the standard Fisher linear DF.
13. A purposeful, conscious control of the SLP classifier complexity—obtained by determining optimal values of targets, the learning step and its change in the training process, the number of iterations, addition or subtraction of the regularization term—all help to reduce the generalization error. Correct initialization of weights and transformations that simplify the data structure are also useful. In certain experiments with artificial data, we have achieved the generalization error reduction up to 10 times or more.

## 3. Seven statistical classifiers

*3.1* The optimal Bayes decision rule for classifying two multivariate Gaussian classes that differ in mean vectors $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$, but share a common covariance matrix $\boldsymbol{\Sigma}$, is given by the following discriminant function (see, e.g., Fukunaga, 1990; McLachlan, 1992):

$$g(\mathbf{x}) = \left(\mathbf{x} - \frac{1}{2}(\mu_1 + \mu_2)\right)' \Sigma^{-1}(\mu_1 - \mu_2)$$

The classification is performed according to the sign of discriminant function (DF). After inserting sample mean vectors, $\bar{\mathbf{x}}^{(1)}$ and $\bar{\mathbf{x}}^{(2)}$, and the sample covariance matrix $\mathbf{S}$ instead of unknown $\boldsymbol{\mu}_1$, $\boldsymbol{\mu}_2$ and $\boldsymbol{\Sigma}$ into the above formula, one obtains the plug-in sample DF

$$g^{\mathrm{LDF}}(\mathbf{x}) = \mathbf{x}'\mathbf{S}^{-1}\left(\bar{\mathbf{x}}^{(1)} - \bar{\mathbf{x}}^{(2)}\right)$$

$$-\frac{1}{2}\left(\bar{\mathbf{x}}^{(1)} + \bar{\mathbf{x}}^{(2)}\right)'\mathbf{S}^{-1}\left(\bar{\mathbf{x}}^{(1)} - \bar{\mathbf{x}}^{(2)}\right) = w_o^{\mathrm{F}} + \mathbf{x}'\mathbf{w}^{\mathrm{F}}$$

where

$$\mathbf{w}^{\mathrm{F}} = \left(w_1, w_2, \ldots, w_p\right)' = \mathbf{S}^{-1}\left(\overline{\mathbf{x}}^{(1)} - \overline{\mathbf{x}}^{(2)}\right)$$

$$w_o^{\mathrm{F}} = -\frac{1}{2}\mathbf{w}^{\mathrm{F}\prime}\left(\overline{\mathbf{x}}^{(1)} + \overline{\mathbf{x}}^{(2)}\right) \tag{8}$$

$$\overline{\mathbf{x}}^{(i)} = \frac{1}{N_i}\sum_{j-1}^{N_i}\mathbf{x}_j^{(i)}$$

is a sample mean vector, and

$$\mathbf{S} = \frac{1}{N_1 + N_2 - 2}\sum_{i=1}^{2}\sum_{j=1}^{N_i}\left(\mathbf{x}_j^{(i)} - \overline{\mathbf{x}}^{(i)}\right)\left(\mathbf{x}_j^{(i)} - \overline{\mathbf{x}}^{(i)}\right)' \tag{9}$$

is a sample pooled covariance matrix, the maximum likelihood estimate of the covariance matrix $\mathbf{\Sigma}$ of two Gaussian classes. This linear classifier is called the *standard Fisher DF*. In DA it is sometimes called the Anderson classification statistics.

In the finite learning-set case, the sample mean vectors $\overline{\mathbf{x}}^{(1)}$ and $\overline{\mathbf{x}}^{(2)}$, the sample covariance matrix $\mathbf{S}$ are inexact estimates of unknown $\boldsymbol{\mu}_1$, $\boldsymbol{\mu}_2$ and $\mathbf{\Sigma}$. Therefore the sample-based classification rule is conditioned by a particular random learning set. Its performance—a conditional probability of misclassification (generalization error)—will be higher than that of the Bayes optimal DF. When prior probabilities of the classes are different and $N_2 \neq N_1$, the sample plug-in DF with weights (Eq. (8)) is not an optimal rule among all possible sample-based classifiers. In statistical pattern recognition, a Bayes approach for designing optimal sample-based classification rules is developed. In this approach, $\boldsymbol{\mu}_1$, $\boldsymbol{\mu}_2$ and $\mathbf{\Sigma}$ are supposed to be random variables, and the prior distribution $q_{prior}(\boldsymbol{\mu}_1,\boldsymbol{\mu}_2,\mathbf{\Sigma})$ of these variables is supposed to be known. Then we seek a posterior (predictive) density of the vector $\mathbf{X}$, and we design a classifier for this density estimate. For a uniform prior distribution of $\boldsymbol{\mu}_1$, $\boldsymbol{\mu}_2$ and $\mathbf{\Sigma}$, the predictive density was determined as (Geisser, 1964; Keehn, 1965)

$$f\left(\mathbf{X}|\mathbf{x}_1^{(1)}, \mathbf{x}_2^{(1)}, \ldots, \mathbf{x}_{N_2}^{(2)}, \pi_i\right) \propto \left(\frac{N_i}{N_i + 1}\right)^{p/2}$$

$$\left(1 + \frac{N_i\left(\mathbf{x} - \overline{\mathbf{x}}^{(i)}\right)'\mathbf{S}^{-1}\left(\mathbf{x} - \overline{\mathbf{x}}^{(i)}\right)}{\left(N_i + 1\right)\left(N_i - 2\right)}\right)^{-\left(N_1 + N_2 - 3\right)/2} \tag{10}$$

The use of the density estimate (Eq. (10)) to design the classification rule results in a quadratic discriminant function. When $N_2 = N_1 = N$ and the prior probabilities of the classes are equal, this optimal ''Bayes predictive classifier'' is linear and becomes equivalent to the Fisher linear discriminant (Gupta, 1977). This means the *Fisher linear DF* is the optimal sample-based classification rule in the sense that it *yields the minimal classification error for a set of classification problems defined by a uniform prior distribution* of $\boldsymbol{\mu}_1$, $\boldsymbol{\mu}_2$ and $\mathbf{\Sigma}$. No other sample-based classification rule will yield a smaller generalization error.

If $N_1$ and $N_2$ are small in comparison with the number of dimensions $p$, then there arise problems associated with the inversion of the sample covariance matrix $\mathbf{S}$. There are several ways to overcome this kind of difficulty. We describe only three of them.

*3.2* One of the approaches is to assume $\mathbf{\Sigma} = \mathbf{I}\cdot\sigma^2$ ($\sigma$ is a scalar). This means we can assume the pattern classes to be spherical Gaussian. Therefore, when classifying according to the sign of the DF, one can omit the covariance matrix; i.e., we have the following DF

$$g^{\mathrm{E}}(\mathbf{x}) = \mathbf{x}'\left(\overline{\mathbf{x}}^{(1)} - \overline{\mathbf{x}}^{(2)}\right) - \frac{1}{2}\left(\overline{\mathbf{x}}^{(1)} + \overline{\mathbf{x}}^{(2)}\right)'\left(\overline{\mathbf{x}}^{(1)} - \overline{\mathbf{x}}^{(2)}\right) \tag{11}$$

This classifier can be obtained from heuristic (ad hoc) considerations, if we suppose that pattern vectors of each class are similar among themselves and close to the ''typical member'' of this class (i.e., the centre with sample mean vector $\overline{\mathbf{x}}^{(j)}$), and perform the classification according to the distance to the mean $\overline{\mathbf{x}}^{(j)}$. Therefore this classifier is called *the Euclidean distance or the nearest means classifier*. Note that for a uniform prior distribution of the difference $\boldsymbol{\mu} = \boldsymbol{\mu}_1 - \boldsymbol{\mu}_2$, EDC is the optimal Bayes predictive classification rule for spherical Gaussian patterns (Abramson and Braverman, 1962). This means that, in a variety of classification problems defined by the prior distribution of $\boldsymbol{\mu}$, no other sample-based classifier will perform better. It is very important to remember this in the analysis of the overtaining effect.

*3.3* Another approach to overcoming difficulties associated with the matrix inversion in regression and DA is to use the shrinkage (ridge) estimate of the covariance matrix

$$\mathbf{S}_{\mathrm{R}} = \mathbf{S} + \lambda\mathbf{I}$$

instead of estimate (9). Here $\mathbf{I}$ is the $p \times p$ identity matrix and $\lambda$ is a positive regularization constant.

In this case the weight vector of the linear discriminant function becomes

$$\mathbf{w}^{\mathrm{RDA}} = (\mathbf{S} + \lambda\mathbf{I})^{-1}\left(\overline{\mathbf{x}}^{(1)} - \overline{\mathbf{x}}^{(2)}\right) \tag{12}$$

This procedure is called *regularized linear discriminant analysis* (RDA). When $\lambda$, the regularization constant, is extremely small, its influence is insignificant and we get the conventional Fisher linear DF. When $\lambda$ is very large, the influence of sample estimate $\mathbf{S}$ disappears and we have the Euclidean distance classifier.

*3.4* If $N_1 + N_2 = n < p + 2$, the covariance matrix becomes singular. Therefore, instead of the conventional matrix inversion, use of a *pseudoinversion* of this matrix is sometimes suggested. The sense of the pseudoinversion of matrix $\mathbf{K}$ consists of a singular value decomposition of matrix $\mathbf{K}$:

$$\mathbf{TKT}' = \begin{bmatrix}\mathbf{d} & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{bmatrix}$$

where

$$\mathbf{T} = \begin{bmatrix}\mathbf{t}_1\\ \mathbf{t}_2\end{bmatrix}$$

is an orthogonal matrix such that

$$\mathbf{TKT}' = \begin{bmatrix} \mathbf{d} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{13}$$

and $\mathbf{d} = \mathbf{t}_1\mathbf{K}\mathbf{t}_1{}'$ is an $r \times r$ diagonal matrix composed of $r = N_1 + N_2 - 2$ non-zero eigenvalues of $\mathbf{K}$. Thus, the pseudoinverse of $\mathbf{K}$ is

$$\mathbf{K}^* = \mathbf{T}' \begin{bmatrix} \mathbf{d}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \mathbf{T} \tag{14}$$

This classifier is called *the Fisher linear DF with pseudo-inversion* or, simply, a Pseudo Fisher classifier:

$$\mathbf{w}^{\text{Fpseudo}} = \mathbf{K}^* \left(\overline{\mathbf{x}}^{(1)} - \overline{\mathbf{x}}^{(2)}\right), \tag{15}$$

where the data is centred a priori, i.e., $\overline{\mathbf{x}}^{(2)} = -\overline{\mathbf{x}}^{(1)}$, and

$$\mathbf{K} = \frac{1}{N_1 + N_2} \sum_{i=1}^{2} \sum_{j=1}^{N_i} \mathbf{x}_j^{(i)} \left(\mathbf{x}_j^{(i)}\right)', \tag{16}$$

*3.5 A generalization of the Fisher criterion* to obtain the weights of the linear discriminant function was proposed by Randles et al. (1978). They suggested minimizing

$$T_{gen} = \frac{1}{N_1 + N_2} \sum_{i=1}^{2} \sum_{j=1}^{N_i} \varphi\left(\frac{t_j^i - \mathbf{w}'\mathbf{x}_j^{(i)} - w_o}{\sqrt{\mathbf{w}'\mathbf{S}^{-1}\mathbf{w}}}\right) \tag{17}$$

where $t_j^{(i)}$ is a class index of the training pattern vector $\mathbf{x}_j^{(i)}$: $t_j^{(i)} = +1$ and $t_j^{(2)} = -1$, and $\varphi(\varepsilon)$ is a non-decreasing odd and non-constant function: e.g., $\varphi(\varepsilon) = (-1)^{i-1} \tanh(\varepsilon)$ ($\omega$ is a positive constant). In fact, while designing his linear DF, Fisher (1936) used the linear function $\varphi(\varepsilon) = \varepsilon$. The use of the non-linear function $\varphi(\varepsilon)$ is a generalization of the Fisher criterion. Because of the saturation of $\varphi(\varepsilon)$, *the solution becomes more robust to outliers* than the standard Fisher discriminant. For numerical optimization of the above criterion, the function $\varphi(\varepsilon)$ should be smooth and differentiable. The properties of the DF are similar to the soft limiting activation function (Eq. (2)) used in ANN design. In an exceptional case, where $\varphi(\varepsilon)$ is a threshold function, the minimization of criterion $T_{gen}$ results in minimization of the number of misclassifications in the learning set, i.e., the empirical error.

*3.6* Most often, the objective of classifier design is to obtain a discriminant function that yields the minimum classification error. Thus, in training, usually one attempts to minimize the number of vectors misclassified—the empirical error. In statistical pattern recognition, one often minimizes the cost function

$$Cost = \frac{1}{N_1 + N_2} \sum_{i=1}^{2} \sum_{j=1}^{N_i} \varphi\left(t_j^{(i)}, \left(\mathbf{w}'\mathbf{x}_j^{(i)} + w_o\right)\right) \tag{18}$$

where $t_j^{(i)}$ is a class index of the training pattern vector $\mathbf{x}_j^{(i)}$: $t_j^{(1)} = +1$ and $t_j^{(2)} = -1$, and $\varphi(\varepsilon)$ is a pattern error function. If $\varphi\left(t_j^{(i)}, \left(\mathbf{w}'\mathbf{x}_j^{(i)} + w_o\right)\right)$ is the threshold function:

$$\varphi\left(t_j^{(i)}, g\right) = \begin{cases} 1 & \text{for } g < 0 \\ 0 & \text{otherwise} \end{cases}$$

we minimize the empirical error. However, this function is not differentiable. In the adaline, one uses a quadratic function $\varphi(d) = d^2$. It is important to remark that minimization of this and other differentiable cost functions does not imply minimization of the classification error. Amari (1967) suggested using a function

$$\theta(d) = \arctan\left(d/d_0\right)$$

where $d = |t_j^{(i)} - \left(\mathbf{w}'\mathbf{x}_j^{(i)} + w_o\right)| / \sqrt{\mathbf{w}'\mathbf{w}}$ is the distance between the vector $\mathbf{x}_j^{(i)}$ and a discriminant hyperplane $\mathbf{w}'\mathbf{x}_j^{(i)} + w_o = 0$, and $d_0$ is a sufficiently small constant. For small $d_0$, the function $\varphi(d)$ is similar to the threshold function, and roughly, we minimize the empirical classification error. This leads to the *minimal empirical error (MEE) classifier*.

The minimization of Eq. (18) is performed in an iterative way. It is, in fact, a version of the generalized Fisher criterion (Randles et al., 1978). Do-Tu and Installe (1978) suggested changing a slope of the function $\varphi(d)$ so that, with an increase in the number of iterations, this function would gradually approach the threshold function and the cost function (Eq. (18)) would eventually minimize the empirical error (a window function technique). A dozen algorithms by other authors were also mentioned in Section 1.

*3.7* When the zero empirical classification error is obtained, the resulting discriminant function is no longer unique. To obtain a unique rule, some additional criteria are introduced that favour an increase in the distance between the discriminant hyperplane and the learning-set vectors closest to it. In one of the first known algorithms, in order to find the weight vector $\mathbf{w}$, Vapnik and Chervonenkis (1974) suggested minimizing the quadratic form

$$\mathbf{w}'\mathbf{w} \tag{19}$$

under the constraints

$$\mathbf{w}'\mathbf{x}_j^{(1)} \geq \Delta \quad \text{and} \quad \mathbf{w}'\mathbf{x}_j^{(2)} < -\Delta \quad (j = 1, 2, \ldots, N_i;\ i = 1, 2)$$

where $\Delta$ is a positive constant, a bound for the margin (''generalized portrait'' method).

To this end, they used quadratic optimization techniques. After optimization a finite number of vectors, called ''supporting vectors'', determine the position of the discriminant hyperplane. In one of the latest algorithms, more neighbouring learning vectors than the ''supporting vectors'' contribute to determination of the final position of the hyperplane. In this article, we call all the classification algorithms with an increasing margin *''maximal margin'' classifiers*.

## 4. Weight and cost function dynamics in BP training

We demonstrate that *the criterion (Eq. (3)) used to find weights of the non-linear SLP classifier changes during the training process*. Let $\bar{v}_i = 1/N_i \sum_{j=1}^{N_i} \left(w_o + \mathbf{w}'\mathbf{x}_j^{(i)}\right)$ be average values of a discriminant $g_j^{(i)}\left(\mathbf{x}_j^{(i)}\right) = w_o + \mathbf{w}'\mathbf{x}_j^{(i)}$.

We use absolute values of $\bar{v}_1$ and $\bar{v}_2$ as indicators that show a similarity of the soft limiting activation function $\tanh\left(w_o+\mathbf{w}'\mathbf{x}_j^{(i)}\right)$ to the hard limiting threshold function. Inspection of curve tanh($g$) indicates that, for very small absolute values $\bar{v}_1$ and $\bar{v}_2$, the values $g_j^{(i)}$ vary in the neighbourhood of 0 (the zero point). Then the activation function acts as a linear function. For extremely large $\bar{v}_1$ and $\bar{v}_2$, contributions $\tanh\left(w_o+\mathbf{w}'\mathbf{x}_j^{(i)}\right)$ of all learning vectors $\mathbf{x}_j^{(i)}$ are close to either $-1$ or $+1$. Then the activation function ''tanh'' is actually acting like a threshold function. Obviously, the values $\bar{v}_1$ and $\bar{v}_2$ depend on the magnitudes of the weights.

Typically, during perceptron training, one starts from very small weights and normalizes the data. Some authors suggest having data values in the interval $(-1,+1)$; others suggest moving the centre of the data, $1/2\left(\bar{\mathbf{x}}^{(1)}+\bar{\mathbf{x}}^{(2)}\right)$, to the zero point and making unit variances of all the features. For very small initial weights, the scalar products $g_j^{(i)}\left(\mathbf{x}_j^{(i)}\right)$ are close to zero. Therefore, during the first iteration, the activation function acts as a linear function, i.e., $o\left(g_j^{(i)}\right)=g_j^{(i)}$, giving $\partial o(g)/\partial g=1$. If $\eta$, the learning step, is small, we obtain small weights and small values of $g_j^{(i)}\left(\mathbf{x}_j^{(i)}\right)$ in subsequent iterations. In fact, the activation function ''$\tanh\left(g_j^{(i)}\right)$'' will act as a linear function.

With an increase in the number of iterations $t$, the cost function (Eq. (3)) tends to its minimum and the weight vector $\mathbf{w}_t$ to its optimal value $\hat{\mathbf{w}}$, where the minimum of Eq. (3) is obtained. We show that final average values $\bar{v}_i=1/N_i\sum_{j=1}^{N_i}g_j^{(i)}$ depend on the target values.

Let the activation function be the linear function $o\left(g_j^{(i)}\right)=g_j^{(i)}$, $t_j^{(1)}=[1]$, $t_j^{(2)}=-1$, $N_2=N_1=N$ and $n=2N>p+1$, so that the sample covariance matrix $\mathbf{S}$ is not singular. Equating the derivatives

$$\frac{\partial cost}{\partial w_o}=-\frac{1}{2N}\sum_{i=1}^{2}\sum_{j=1}^{N}\left(t_j^{(i)}-w_o-\left(\mathbf{x}_j^{(i)}\right)'\mathbf{w}\right)$$

$$=-\left(t_1+t_2\right)/2+w_o+\left(k_1\bar{\mathbf{x}}^{(1)}+k_2\bar{\mathbf{x}}^{(2)}\right)'\mathbf{w} \qquad (20)$$

and

$$\frac{\partial cost_l}{\partial \mathbf{w}}=-\frac{1}{2N}\sum_{i=1}^{2}\sum_{j=1}^{N}\mathbf{x}_j^{(i)}\left(t_j^{(i)}-w_o-\left(\mathbf{x}_j^{(i)}\right)'\mathbf{w}\right)$$

$$=-\left(t_1\bar{\mathbf{x}}^{(1)}+t_2\bar{\mathbf{x}}^{(2)}\right)/2+\left(\bar{\mathbf{x}}^{(1)}+\bar{\mathbf{x}}^{(2)}\right)w_o/2+\mathbf{K}\mathbf{w} \qquad (21)$$

where

$$\mathbf{K}=\frac{1}{2N}\sum_{i=1}^{2}\sum_{j=1}^{N}\mathbf{x}_j^{(i)}\left(\mathbf{x}_j^{(i)}\right)',K_i=N_i/(N_1+N_2)$$

to zero, and solving the resulting linear equations, we can show that for small weights (linear activation function), the minimum of cost function (3) is obtained at

$$w_o=0,\quad \mathbf{w}=k\mathbf{S}^{-1}\Delta\bar{\mathbf{x}} \qquad (22)$$

where

$$k=2/\left(D^2+4(N-1)/N\right)$$

$$\Delta\bar{\mathbf{x}}=\bar{\mathbf{x}}^{(1)}-\bar{\mathbf{x}}^{(2)}$$

and

$$D^2=\Delta\bar{\mathbf{x}}'\mathbf{S}^{-1}\Delta\bar{\mathbf{x}}$$

is the sample Mahalanobis distance. For simplicity, we assumed $\bar{\mathbf{x}}^{(1)}+\bar{\mathbf{x}}^{(2)}=0$. This assumption does not affect $\mathbf{w}$.

Then the average values are given by $\bar{v}_i-1/N_i\sum_{j=1}^{N_i}g_j^{(i)}=\pm 1/\left(1+D^2\right)$. We have $|\bar{v}_i|=0.2$ for $D=1$, $|\bar{v}_i|=0.5$ for $D=2$ and $|\bar{v}_i|=0.8$ for $D=4$. The values $\bar{v}_1$ and $\bar{v}_2$ become close to the desired outputs 1 and $-1$ for very large $D$. In the case of a *non-linear activation function*, higher order statistical moments of the learning set affect cost function (3). Let us analyse the GCCM data model and assume all sample statistical moments of the learning sets $\mathbf{X}_1^{(1)},\ldots,\mathbf{X}_N^{(1)}$ and $\mathbf{X}_1^{(2)},\ldots,\mathbf{X}_N^{(2)}$ to be identical to that of the GCCM model with the parameters $\bar{\mathbf{x}}^{(1)}$, $\bar{\mathbf{x}}^{(2)}$ and $\mathbf{S}$; i.e., density $f\left(\mathbf{X}_j^{(i)}\right)=N\left(\mathbf{X},\bar{\mathbf{x}}^{(i)},\mathbf{S}\right)$. The optimal weight vector for this model is the Fisher DF. Its weight vector $\mathbf{w}=c\mathbf{w}^{\mathrm{F}}$ and the threshold $w_o=cw_o^{\mathrm{F}}$ can be changed arbitrarily by scaling by any positive constant $c$. However, a change in $c$ changes cost (3). We evaluate average values $\bar{v}_1$ and $\bar{v}_2$ where the minimum of the cost for the ''tanh'' activation function is obtained. The expectation of cost function (3) with respect to the set of $2N$ random learning vectors $\mathbf{X}_1^{(1)},\ldots,\mathbf{X}_N^{(2)}$ is

$$E_{t\mathbf{X}}cost(\mathbf{w})=\int\ldots\int cost\left(\mathbf{w},\mathbf{X}_1^{(1)},\ldots,\mathbf{X}_N^{(2)}\right)f\left(\mathbf{X}_1^{(1)},\ldots,\mathbf{X}_N^{(2)}\right)$$

$$\mathrm{d}\mathbf{X}_1^{(1)}\ldots\mathrm{d}\mathbf{X}_N^{(2)}=\frac{1}{4}\int\Big\{\left(t_1-\tanh\left(c\left(\mathbf{w}'\mathbf{X}+w_o\right)\right)\right)^2N\left(\mathbf{X},\bar{\mathbf{x}}^{(1)},\mathbf{S}\right)$$

$$+\left(t_2-\tanh\left(c\left(\mathbf{w}'\mathbf{X}+w_o\right)\right)\right)^2N\left(\mathbf{X},\bar{\mathbf{x}}^{(2)},\mathbf{S}\right)\Big\}\,\mathrm{d}\mathbf{X} \qquad (23)$$

A numerical minimization of Eq. (23) with respect to the parameter c results in: for $D=1$ $|\bar{v}_i|=0.25$, for $D=2$ $|\bar{v}_i|=1$, for $D=4$ $|\bar{v}_i|=4$, and for $D=6$ $|\bar{v}_i|=9$. We see that the values $\bar{v}_1$, $\bar{v}_2$ and the actual similarity of the activation function tanh($g$) to the threshold function depend on the separability of the learning set: i.e., $D$, the sample Mahalanobis distance. For close pattern classes we obtain small weights, and therefore we cannot minimize the empirical classification error; however, for distant pattern classes, with an increase in the number of iterations, we minimize the empirical classification error more and more exactly.

Three important remarks.

1. The sample estimate $D^2$ is a biased estimate of the true squared sample Mahalanobis distance $\delta^2$:

$$ED^2=\delta^2T_\mu T_\Sigma \qquad (24)$$

where terms $T_\mu$ and $T_\Sigma$ have been defined by Eq. (5) in

Section 2. Thus, for finite values of the learning-set size $N$, we have higher separability of the learning sets. For very small learning sets, when $N_2 + N_1 \to p$, both $ED^2$ and $|\bar{v}_i|$ increase abruptly, and cost function (3) actually starts minimizing the empirical classification error.The conclusion that, for the soft limiting non-linear activation function and very distant classes, the weights can increase without bound is easy to understand on the basis of intuitive arguments. Let the empirical error be zero. This means that values $g_j^{(1)}$ are positive and values $g_j^{(2)}$ are negative. In order to minimize cost function (3) the training algorithm will move $o\left(g_j^{(1)}\right)$ to $t_1 = +1$ and $o\left(g_j^{(2)}\right)$ to $t_2 = -1$. This is possible only by increasing the magnitudes of the weights. Obviously, the weights will not increase without bound if $|t_i| < 1$.

2. While minimizing cost function (3) by the gradient-type backpropagation algorithm, certain numerical difficulties arise. In the case of small empirical error, the weights become large, *the neurone becomes aged and begins to produce almost categorical answers*: outputs $o\left(w_o + \mathbf{w}'\mathbf{x}_j^{(i)}\right)$ are close to either $+1$ or $-1$. Then the gradient of the cost function is near to zero. Therefore in BP training with constant learning step $\eta$, the training process slows down and afterwards actually stops. *In order to force the backpropagation algorithm to adjust the weights continuously and increase the margin, we need to increase the magnitude of the learning step*. Analysis shows that, in order to ensure a linear increase in the magnitude of the weights, the learning step $\eta$ should increase *exponentially* with an increase in iteration number $t$:

$$\eta = \eta_{start} \cdot \alpha^t \tag{25}$$

where $\alpha$ is a positive constant, slightly larger than 1 (e.g., in our simulation studies $\alpha$ was taken between 1.001 and 1.3).

In the practical application of this approach, one faces difficulties associated with the accuracy of calculations. Very large magnitudes of the weight vector actually cause zero values of the cost function, and the training process stops.

3. If the empirical classification error is large (e.g., close to or larger than 0.25), then we have small weights, and we cannot minimize the empirical error. To force the classifier to minimize the empirical frequency of misclassification, we can add to the cost function (3) a ''*negative weight decay*'' — $\lambda_1 \mathbf{w}'\mathbf{w}$ or $+ \lambda_2(\mathbf{w}'\mathbf{w} - h^2)^2$, the so called ''*anti-regularization*'' term (Raudys, 1995). This term forces the learning algorithm to increase the magnitude of the weights and, consequently, to increase the actual slope of the activation function.

## 5. Seven types of statistical classifier in SLP design

*5.1* In this section we show that, in BP training, one can in principle obtain seven different statistical classifiers. *First, consider a case where at the beginning of training the weights are small and the activation function acts as the linear function*. Thus $o(g) = g$, and $\partial o(g)/\partial g = 1$. To obtain a more general result, we temporarily reject the assumption $N_2 = N_1$. Then instead of Eqs. (20) and (21) we have

$$\left.\frac{\partial cost}{\partial w_o}\right|_{w_o = w_{o(t)}} = -\left(t_1 k_1 + t_2 k_2\right) + w_{o(t)} + \left(k_1 \bar{\mathbf{x}}^{(1)} + k_2 \bar{\mathbf{x}}^{(2)}\right)' \mathbf{w}_{(t)} \tag{26}$$

$$\left.\frac{\partial cost_l}{\partial \mathbf{w}}\right|_{\mathbf{w} = \mathbf{w}_{(t)}} = -\left(t_1 k_1 \bar{\mathbf{x}}^{(1)} + t_2 k_2 \bar{\mathbf{x}}^{(2)}\right)$$

$$+ \left(k_1 \bar{\mathbf{x}}^{(1)} + k_2 \bar{\mathbf{x}}^{(2)}\right) w_{o(t)} + \mathbf{K}\mathbf{w}_{(t)} \tag{27}$$

where $\mathbf{K}$ was defined in Eq. (16).

If the prior weights are zero, i.e., $w_{0(0)} = 0$ and $\mathbf{w}_{(0)} = \mathbf{0}$, then after the first learning iteration one has

$$w_{0(1)} = \eta\left(t_1 k_1 + t_2 k_2\right) \quad \text{and} \quad \mathbf{w}_{(1)} = \mathbf{w}_{(0)} - \eta \frac{\partial cost_l}{\partial \mathbf{w}_{(0)}}$$

$$= \eta\left(t_1 k_1 \bar{\mathbf{x}}^{(1)} + t_2 k_2 \bar{\mathbf{x}}^{(2)}\right) \tag{28}$$

If $t_2 N_2 = -t_1 N_1$, then

$$\mathbf{w}_{(1)} = \eta t_1 k_1 \Delta\bar{\mathbf{x}}, \quad w_{0(1)} = 0 \tag{29}$$

*This is is the weight vector of the Euclidean distance classifier* designed for the centred data, i.e., $\bar{\mathbf{x}}^{(1)} + \bar{\mathbf{x}}^{(2)} = \mathbf{0}$. The classification according to the sign of discriminant function (29) is asymptotically optimal, when the classes are spherical Gaussian $N(\boldsymbol{\mu}_i, \mathbf{I}\sigma^2)$ and also in many other situations. It is a nice property of the single-layer perceptron that it should become a comparatively good statistical classifier just after the first learning iteration! To achieve this, one has to fulfil conditions ***E*** enumerated in Section 2. We see that there are several arguments for using the centred data $\left(\bar{\mathbf{x}}^{(2)} = -\bar{\mathbf{x}}^{(1)}\right)$ with $N_2 = N_1$.

**5.2** In further analysis, we assume that these assumptions are again fulfilled, $t_j^{(1)} = 1$ and $t_j^{(2)} = -1$, and analyse a change in the weight vector after the second and next iterations. The usage of total gradient adaptation rule (Eq. (4)) with gradient given by Eqs. (26) and (27) after the second iteration results in:

$$w_{0(2)} = 0$$

and

$$\mathbf{w}_{(2)} = \mathbf{w}^{(1)} - \eta \left.\frac{\partial cost_l}{\partial \mathbf{w}}\right|_{\mathbf{w} = \mathbf{w}_{(1)}} = \frac{1}{2}\eta\Delta\bar{\mathbf{x}} - \eta\left(-\frac{1}{2}\Delta\bar{\mathbf{x}} + \mathbf{K}\mathbf{w}_{(1)}\right)$$

$$= \left(\mathbf{I} - \left(\mathbf{I} - \eta\mathbf{K}\right)^2\right)\frac{1}{2}\mathbf{K}^{-1}\Delta\bar{\mathbf{x}}$$

After further iterations, $w_{0(2)} = 0$,

$$\mathbf{w}_{(t)} = \left(\mathbf{I} - \left(\mathbf{I} - \eta\mathbf{K}\right)^t\right)\frac{1}{2}\mathbf{K}^{-1}\Delta\bar{\mathbf{x}} \tag{30}$$

where

$$\mathbf{K}=\frac{1}{2N}\sum_{i=1}^{2}\sum_{j=1}^{N}\mathbf{x}_j^{(i)}\left(\mathbf{x}_j^{(i)}\right)'=\frac{N-1}{N}\mathbf{S}+\frac{1}{4}\Delta\bar{\mathbf{x}}\Delta\bar{\mathbf{x}}'$$

The matrix **K** is supposed to be non-singular and have an inverse.

Employing the first terms of expansions

$$\left(\mathbf{I}-\eta\mathbf{K}\right)^t=\mathbf{I}-t\eta\mathbf{K}+\frac{1}{2}t(t-1)\eta^2\mathbf{K}^2-\ldots$$

and

$$(\mathbf{I}-\beta\mathbf{K})^{-1}=\mathbf{I}+\beta\mathbf{K}+\ldots$$

for small $\eta$ and $t$, we obtain

$$\mathbf{w}_{(t)}=\left(t\eta\mathbf{K}-\frac{1}{2}t(t-1)\eta^2\mathbf{K}\right)\frac{1}{2}\mathbf{K}^{-1}\Delta\bar{\mathbf{x}}$$

$$=\frac{1}{2}t\eta\left(\mathbf{I}-(t-1)\eta\mathbf{K}\right)\Delta\bar{\mathbf{x}}=\frac{1}{2}t\eta\left(\mathbf{I}+\frac{1}{2}(t-1)\eta\mathbf{K}\right)^{-1}\Delta\bar{\mathbf{x}}$$

$$=\frac{t}{t-1}\left(\mathbf{I}+\frac{(t-1)\eta}{2}\left(\frac{N-1}{N}\mathbf{S}+\frac{1}{4}\Delta\bar{\mathbf{x}}\Delta\bar{\mathbf{x}}'\right)\right)^{-1}\Delta\bar{\mathbf{x}}$$

Assuming the matrix $\mathbf{I}\lambda+\mathbf{S}$ to be non-singular, after some matrix algebra we get

$$\mathbf{w}_{(t)}=\left(\mathbf{I}\frac{2}{(t-1)\eta}\frac{N}{N-1}+\mathbf{S}\right)^{-1}\Delta\bar{\mathbf{x}}\frac{tN}{(t-1)(N-1)}k_{\mathrm{R}} \quad (31)$$

where

$$k_{\mathrm{R}}=\frac{2\eta t}{\left(D_{\mathrm{R}}^2+2\eta(t-1)(N-1)/n\right)}$$

$$D_{\mathrm{R}}^2=\Delta\bar{\mathbf{x}}'\mathbf{S}_{\mathrm{R}}^{-1}\Delta\bar{\mathbf{x}}$$

and

$$\mathbf{S}_{\mathrm{R}}=\mathbf{S}+\mathbf{I}\frac{2}{(t-1)\eta}\frac{N}{N-1}$$

The weight vector $\mathbf{w}_{(t)}$ is equivalent to that resulting from *the regularized linear discriminant analysis* (Eq. (12)) with the regularization parameter $\lambda=(2/(t\text{-}1)\eta)(N/N-1)$. The regularization parameter $\lambda$ changes during training: it decreases with increase in the number of iterations.

*5.3* The above expressions indicate that in training when $t\to\infty$

$$\left(\mathbf{I}\frac{2}{(t-1)\eta}\frac{N}{N-1}+\mathbf{S}\right)\to\mathbf{S},\quad k_{\mathrm{R}}\to 1$$

*the resulting classifier approaches the Fisher DF*. This conclusion follows also from Eq. (22).

*5.4* While deriving weight vectors given by Eqs. (22) and (31), the sample covariance matrix **S** was assumed to be non-singular. When the number of learning samples is $2N<p+2$, this matrix is always singular. Weight vector (30) can be written in the following way:

$$\mathbf{w}_{(t)}=\frac{1}{2}\sum_{s=1}^{t}C_t^s\eta^s(-\mathbf{K})^{s-1}\Delta\bar{\mathbf{x}} \quad (32))$$

This representation does not require the matrix **K** to be non-singular. Let the orthogonal $p\times p$ matrix **T** satisfy representation (13). Then

$$\mathbf{w}_{(t)}=\frac{1}{2}\sum_{s=1}^{t}C_t^s(-\eta)^s\mathbf{T}'(\mathbf{T}\mathbf{K}\mathbf{T}')^{s-1}\mathbf{T}\Delta\bar{\mathbf{x}}=\frac{1}{2}\sum_{s=1}^{t}C_t^s(-\eta)^s\mathbf{T}'$$

$$\begin{bmatrix}\mathbf{d} & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{bmatrix}^{s-1}\mathbf{T}\Delta\bar{\mathbf{x}}=\frac{1}{2}\mathbf{T}'\left(\sum_{s=1}^{t}C_t^s(-\eta)^s\begin{bmatrix}\mathbf{d} & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{bmatrix}^s\right)$$

$$\begin{bmatrix}\mathbf{d}^{-1} & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{bmatrix}\mathbf{T}\Delta\bar{\mathbf{x}}=\frac{1}{2}\mathbf{T}'\left(\mathbf{I}-\left(\mathbf{I}-\eta\begin{bmatrix}\mathbf{d} & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{bmatrix}\right)^t\right)\mathbf{T}\mathbf{T}'$$

$$\begin{bmatrix}\mathbf{d}^{-1} & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{bmatrix}\mathbf{T}\Delta\bar{\mathbf{x}}=\frac{1}{2}\mathbf{T}'\left(\mathbf{I}-\left(\mathbf{I}-\eta\begin{bmatrix}\mathbf{d} & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{bmatrix}\right)^t\right)\mathbf{T}\mathbf{w}^{\mathrm{Fpseudo}}$$

where

$$\mathbf{w}^{\mathrm{Fpseudo}}=\mathbf{T}'\begin{bmatrix}\mathbf{d}^{-1} & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{bmatrix}\mathbf{T}\Delta\bar{\mathbf{x}}=\mathbf{K}^*\Delta\bar{\mathbf{x}}$$

has been defined in subsection 3.4.

For small values of $\eta$, with an increase in the number of training iterations $t$,

$$\left(\mathbf{I}-\left(\mathbf{I}-\eta\begin{bmatrix}\mathbf{d} & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{bmatrix}\right)^t\right)\to\mathbf{I}$$

Then $\mathbf{w}_{(t)}\to\mathbf{w}^{\mathrm{Fpseudo}}$ (the weight vector of the *Fisher linear DF with pseudoinversion*). This conclusion is correct while the inputs $g=\mathbf{w}'\mathbf{x}+w_o$ of the activation function $o(g)$ vary in the linear interval. Consequently, we can obtain this classifier when we have *non-limit* target values ($|t_i|<1$).

*5.5* Suppose the *target values* differ from the limiting values (e.g., $t_1=0.9$ and $t_2=-0.9$ for the $\tanh(g)$ activation function). Then, for correctly classified vectors, the smallest deviations $\left(t_j^{(i)}-\tanh\left(\mathbf{w}'\mathbf{x}_j^{(i)}+w_o\right)\right)^2$ will be obtained for medium size weights. For the medium size weights, the activation function becomes a non-linear function, similarly for the non-decreasing odd and non-constant function $\varphi(\varepsilon)$ in Eq. (17). Thus the minimization of cost function (3) results in a linear classifier that is very similar to *generalized discriminant analysis*, the Amari (1967) algorithm, and the window function technique discussed earlier in the previous section.

*5.6* When one uses limit values of targets ($t_1=+1$ and $t_2=-1$ for the $\tanh(g)$ activation function), we have seen that for two *distant* classes we can obtain very high weights after applying the minimizing cost function (3). In that case, for all the training-set vectors, *the activation function is essentially acting as a hard limiting threshold function*.

This means that if one uses global minimization techniques (which enables us to avoid local minima), *we obtain a classifier similar to the minimum empirical error classifier*.

*5.7* When the number of dimensions exceeds the number of training samples and limiting values of the targets are used, the proper training of a perceptron will always lead to zero empirical error. Zero empirical error can also be obtained also when $n = N_1 + N_2$ exceeds the number of dimensions $p$, the distance between the pattern classes being sufficiently large. Now let the empirical error be zero, the targets $t_1 = 1$ and $t_2 = -1$, and denote by $D(\mathbf{x}^*,\mathbf{w})$ an Euclidean distance between the discriminant hyperplane $g(\mathbf{x}) = \mathbf{w}'\mathbf{x} + w_o = 0$ and the learning-set vector $\mathbf{x}^*$ closest to it. Let $D(\mathbf{x}_+^+,\mathbf{w})$ be the Euclidean distance between the discriminant hyperplane and the second learning-set vector $\mathbf{x}_+^+$ closest to it and different from $\mathbf{x}^*$. Then, asymptotically, with an increase in the magnitude of the weights $\|\mathbf{w}\|$, the ratio $\left(t_+^+ - o\left(\mathbf{w}'\mathbf{x}_+^+\right)\right)^2 / \left(t_*^* - o\left(\mathbf{w}'\mathbf{x}_*^*\right)\right)^2$ diminishes to zero. This implies that the relative contribution of the second learning-set vector $\mathbf{x}_+^+$, closest to the decision hyperplane, becomes insignificant. The learning algorithm tends to put the decision hyperplane further from *the closest* learning-set vector $\mathbf{x}^*$. When the learning process is over, several vectors $\{\mathbf{x}^*\}$ are at the same distance from the discriminant hyperplane $g(\mathbf{x}) = \mathbf{w}'\mathbf{x} + w_o = 0$. Only these learning-set vectors $\{\mathbf{x}^{(*)}\}$ closest to the discriminant hyperplane (Cortes and Vapnik, 1995 call them supporting patterns) contribute to a value of the cost function and to the final determination of the hyperplane location. Thus, we obtain the *maximum margin classifier*.

## 6. Simulations

The aim of this section is to illustrate a variety of possible statistical classifiers that can be obtained in SLP training, and the ways of controlling the type of classifier obtained. For this we use the simplest bivariate artificial data sets. More details on classifier complexity control are presented in Part II.

*6.1* In Fig. 1 we show the distribution of two bivariate Gaussian classes $N(\boldsymbol{\mu}_i,\boldsymbol{\Sigma})$ — two small ellipses — contaminated with 10% additional Gaussian noise $N(\mathbf{0},\mathbf{N})$. The noise patterns are denoted by '' $*$ '' and '' $+$ '', and the signal classes by two ellipses. Both signal classes have different means $\boldsymbol{\mu}_1 = -\boldsymbol{\mu}_2 = \boldsymbol{\mu}$ and share the common covariance matrix $\boldsymbol{\Sigma}$:

$$\mu = \begin{bmatrix} 0.10 \\ 0.05 \end{bmatrix}, \quad \Sigma = \begin{bmatrix} 0.040 & 0.018 \\ 0.018 & 0.01 \end{bmatrix} \quad \mathbf{N} = \begin{bmatrix} 1.0 & 0.5 \\ 0.5 & 1.0 \end{bmatrix}$$

We call this *data set A1*.

The SLP classifier with the sigmoid activation function and targets $t_1 = 0$, $t_2 = 1$ was trained for $s = 1000$ iterations by the standard BP in the batch mode, with learning step $\eta = 5$; the learning-set size being $N_1 = N_2 = N = 250$. In all the experiments reported in this paper, we used centred learning data; the starting learning vector $\mathbf{w}_{(0)} = \mathbf{0}$. After the first iteration (boundary 1 in Fig. 1) we obtain EDC (boundary 2), yielding 22% of errors. After 250 iterations the resulting classifier (boundary 4) became very close to the Fisher linear DF (boundary 3) and yielded 12% of classification errors. After a further 750 iterations with learning step $\eta = 5$, the decision boundary of the SLP classifier actually became identical with the Fisher linear DF (boundary 3). Only a significant increase in the learning step (up to $\eta = 100$) moved the decision boundary (boundary 5; after 1000 iterations) to boundary 6 of the minimum empirical error classifier with 5.5% classification error.

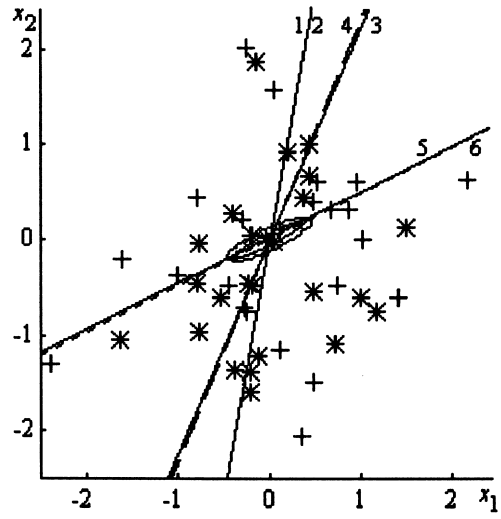


Fig. 1. Distribution of two Gaussian pattern classes contaminated with additional noise and positions of the discriminant lines: 1 — SLP after the first iteration, 2 — EDC, 3 — Fisher linear DF, 4 — SLP after 250 iterations, 5 — SLP at the end of the training process, 6 — MEE classifier.

*6.2* More details on finding the maximal margin can be found in Fig. 2. Each class here consists of a mixture of two Gaussian densities (*data B*). Each subclass is distributed on a separate line in the bivariate space. All four lines are parallel. After the SLP training with an exponentially increasing learning step, the decision boundary was placed approximately halfway between the vectors of two closest subclasses of the opposite classes. This gave the maximum margin classifier. The remaining vectors from other two subclasses do not affect the position of the decision boundary.

*6.3* The details on using the exponentially increasing learning step in SLP training are presented in Fig. 3. There we see *the process of change in the magnitude of the weights* of the SLP classifier during $t_{max} = 800$ training iterations with 100-variate GCCM data $N(\boldsymbol{\mu}_i,\boldsymbol{\Sigma})$: $\boldsymbol{\mu}_2 = -\boldsymbol{\mu}_1 = \boldsymbol{\mu} = (\mu_1,\mu_2,\ldots,\mu_{100})'$; randomly $\chi_1^2$ distributed components $\mu_i$ were normalized ($\boldsymbol{\mu}'\boldsymbol{\mu} = 4$) and ranked: $\mu_1 > \mu_2 > \ldots > \mu_{100}$; $\boldsymbol{\sigma} = ((\sigma_{ij}))$, $\sigma_{ii} = 1$, when $i = j$ and $-\sigma_{ij} = 0.3$, when $i \neq j$; $N = 100$.

In this high dimensional case, we have two distant classes and a comparatively small number of learning vectors.

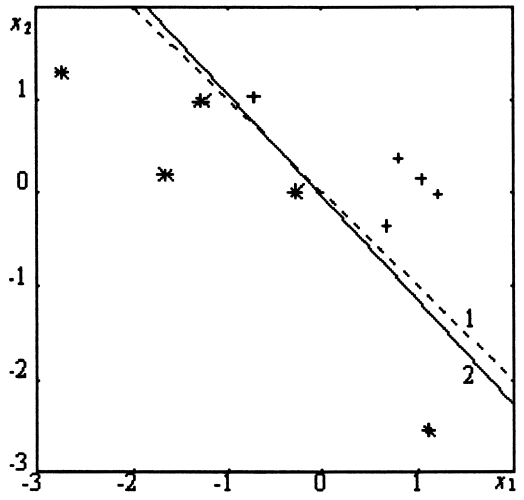


Fig. 2. The SLP as a maximum margin classifier. Data are a mixture of Gaussian subclasses on four parallel lines; 1 — the discriminant line after training by varying $\eta$ ($\eta = 0.5 \cdot 1.05^t$), only four vectors (from the closest subclasses) contribute to the determination of exact position of the boundary, 2 — the discriminant line after SLP training with $\eta = 0.5$.

Therefore, the learning-set vectors are linearly separable and, after achieving zero empirical error, we obtain a certain margin between the decision boundary and the learning-set vectors. Two strategies of control of learning step $\eta$ were compared in this experiment. In the first test, the parameter $\eta$ was increased exponentially with the iteration number $t$, according to Eq. (25). In the second and the third tests, we used two constant values of $\eta$ (0.01 and 0.1).

Obviously, the value of $\eta$ essentially affects the magnitudes of the weights and the margins. Consequently, it affects the learning process and the statistical properties of the classification rule obtained. In order to obtain maximal margins, one needs to have large weights. To ensure weight growth, we have to increase the learning step (see the previous section). Constant values of the learning step ensure the quality of the learning process only for the first few iterations, while the weights are small and the activation function (Eq. (2)) is unsaturated.

*6.4* If the empirical classification error is large (e.g., close to or larger than 0.25), then the small weights prevent us from obtaining the minimum empirical error classifier even when the targets acquire their limit values (+1 and −1 for the tanh($g$) activation function). To force the classifier to minimize the empirical frequency of misclassification, we add the ''*anti-regularization*'' term to cost function (3). This technique is illustrated in Figs. 4 and 5. In this experiment, we have the same kind of data as in Fig. 1; however, the parameters differ:

$$\mu = \begin{bmatrix} 0.030 \\ 0.015 \end{bmatrix}, \quad \Sigma = \begin{bmatrix} 0.040 & 0.0196 \\ 0.0196 & 0.01 \end{bmatrix}$$

$$\mathbf{N} = \begin{bmatrix} 1.0 & -0.7 \\ -0.7 & 1.0 \end{bmatrix}$$

We call this *data set A2*.

We should like to draw attention to two facts that, in bivariate data *A2*, we have a comparatively large number of learning vectors ($N_1 = N_2 = N = 250$) and that the ''signal'' and ''noise'' components have opposite correlations. The variance of ''noise'' is much larger than that of the ''signal''. Therefore the direction of the decision boundary of the EDC (Graph 1) and that of the Fisher DF (Graph 2) differs substantially from that of the boundary of the optimal linear classifier (Graph 3). In traditional training (conditions *E*, $\eta = 10$), the boundary of the SLP moves from (1) towards (2) for a while, and then rotates back a

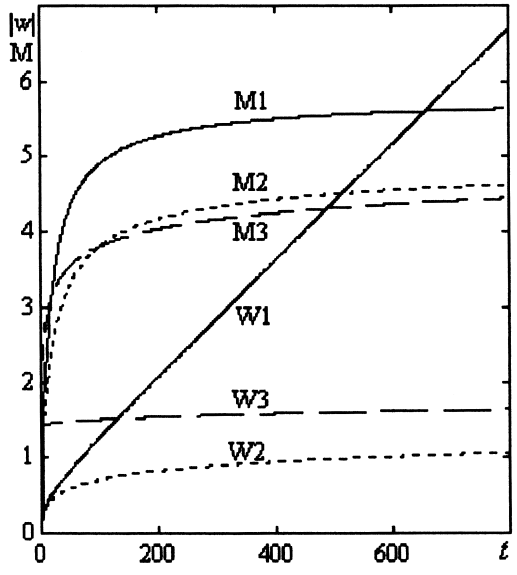


Fig. 3. Magnitude of the first weight $|w_1|$ and the margin $M$ versus $t$, the number of iterations: 1 — $\eta = 0.01 \cdot 1.05^t$, 2 — $\eta = 0.01$, 3 — $\eta = 0.1$.

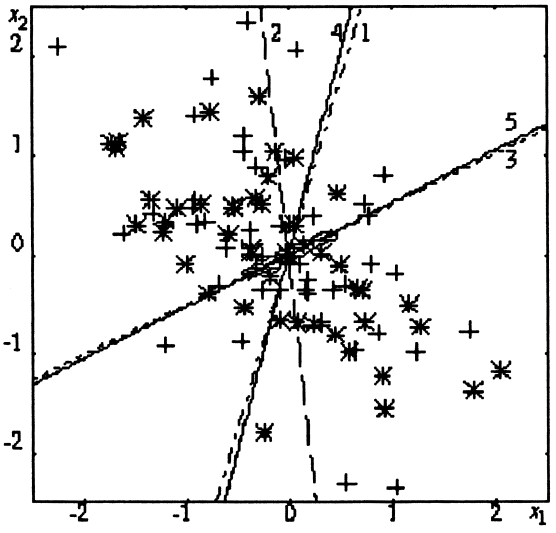


Fig. 4. Distribution of two Gaussian pattern classes contaminated with additional noise, and positions of the discriminant lines: 1 — EDC, 2 — Fisher linear DF, 3— optimal linear DF, 4 — SLP with conventional training, 5 — SLP with anti-regularization.

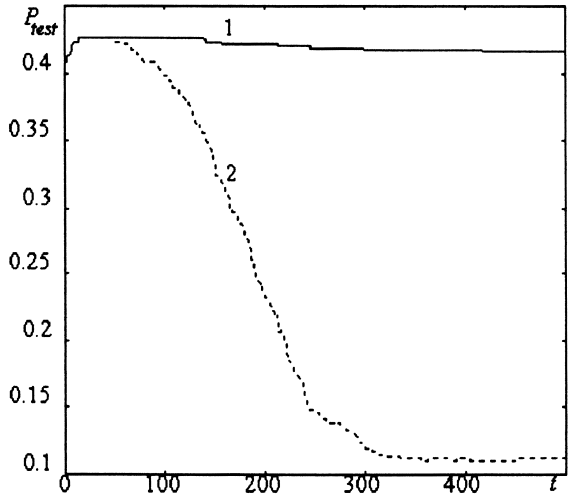


Fig. 5. The empirical and generalization errors versus $t$, the number of iterations: 1 — SLP with conventional training, 2 — SLP with anti-regularization.

little bit as long as it settles at (4). Graph 4 (after 5000 iterations) is close to the decision boundary of the EDC. After the first iteration, the classification error is 0.41 and, while approaching the Fisher DF, it first increases and afterwards decreases, fixing itself at 0.42.

The use of the additional anti-regularization term $+1.6(\mathbf{w}'\mathbf{w} - 25^2)^2$ in the cost function does not change the learning curve at the beginning but later on, when the weights increase substantially, the decision boundary begins approaching the optimal boundary. Graph 5 in Fig. 4 shows the position of the boundary after 500 iterations. As a result, we obtain 11% of classification errors, the same as that obtained with use of the optimal linear classifier designed to classify only the Gaussian ''signal'' patterns (Fig. 5). We see, in the case of a large number of highly contaminated learning-set observation vectors, that utilization of an additional anti-regularization term can pull the discriminant hyperplane near to that of the minimum empirical error classifier.

## 7. Conclusions

We have established that, during adaptive training, the weights of the SLP classifier increase gradually, and one can obtain seven statistical classifiers of different complexity:

- the Euclidean distance classifier,
- the regularized linear discriminant analysis schema,
- the Fisher linear discriminant function,
- the Fisher linear discriminant function with pseudoinversion,
- the generalized Fisher discriminant function,
- the minimum empirical error classifier and
- the maximum margin classifier.

The analysis performed indicates that, despite its apparent simplicity, *the SLP trained by adaptive optimization techniques is, in fact, a very rich family of linear classifiers. There exists no unique single-layer perceptron classifier. On the contrary, there is a great number of classifiers that can be obtained during training*. We can assume that, in principle, more variants on the known classifiers can be obtained. Possibly, there exists a close link between multilayer perceptrons and statistical techniques, too. This is a subject for further study.

Several means of controlling the learning process are described in the literature. These means are associated either with the cost function or with the optimization technique used. The most popular cost function is the sum of squares. Other types of cost function can undoubtedly originate more types of classifier. Different regularization terms are frequently added in order to control the solution obtained.

The most widespread optimization technique is the gradient-type backpropagation algorithm. This algorithm is controlled by weight initialization, the number of iterations and the learning-step parameter. Adding noise to SLP inputs and/or outputs is a popular approach to influence the learning process. It has a similar effect to the utilization of the conventional regularization term. In the present paper, it has been shown that all the means enumerated influence the type of classifier obtained in each training experiment. For example, the number of iterations is an almost universal factor for controlling the type of classifier. One of the alternatives to the BP technique is the conjugate gradient technique — a modification of the second-order (Newton) method. For a linear activation function the cost function (3) is quadratic. Then use of a second-order optimization technique, such as the Newton method, can lead to the Fisher classifier in a single iteration, avoiding the Euclidean distance classifier, and the regularized DA. Therefore, *in this sense, gradient BP training can become preferable*, since it does not require an additional regularization term, such as the ''weight decay'' term added to the cost function.

In order to get a wider range of classifiers in each training experiment, in addition to a variety of known complexity control techniques *five new* ones were proposed:

1. moving the learning data centre, $1/2\left(\overline{\mathbf{x}}^{(1)} + \overline{\mathbf{x}}^{(2)}\right)$, into the origin of the coordinates,
2. zero weight initialization,
3. target value control,
4. use of the additional negative weight decay term called ''anti-regularization'' and
5. use of an exponentially increasing learning step.

All the factors enumerated act simultaneously, and often (although not always) the influence of one factor can be compensated by others. There are several directly uncontrolled factors. These are false local minima and high-dimensional extremely flat areas of the cost function, where the training process almost stops. Adding noise to inputs of the perceptron or to its weights, as well as a

constant or temporal increase in the learning step, can help to move the perceptron weight vector from these inappropriate areas.

In each single training experiment, one cannot optimize over all seven types of classifier. Which particular type of classifier will be obtained depends on: the data, the cost function to be minimized, the optimization technique and its parameters, and the stopping criteria. If the conditions $E$ are fulfilled (batch-mode training, $\bar{\mathbf{x}}^{(1)} + \bar{\mathbf{x}}^{(2)} = \mathbf{0}$, $w_{\theta(0)} = 0$, $\mathbf{w}_{(0)} = \mathbf{0}$, $t_2 = -t_1 N_1/N_2$), then, after the first BP step, we always obtain the EDC. However, too high a value of the learning step $\eta$ can lead to a situation when, after the first learning iteration, the activation function has already become saturated and further training becomes impossible. So we obtain only one classifier: the EDC.

In training with a small learning step, one always obtains the regularized linear DA and approaches the Fisher classifier. In zero empirical error, and in the case of limiting target values, one skips the Fisher DF with pseudoinversion and goes in the direction of the maximum margin classifier directly. The standard Fisher DF or the Fisher DF with pseudoinversion can be obtained if one uses non-limiting target values (e.g., $|t_i| = 0.8$).

The maximal margin can be obtained only when the learning sets of both classes are linearly separable, limiting values of the targets are used, and at the end of the BP training, a very large learning step $\eta$ is used. Other necessary conditions are the need to avoid false local minima and a sufficiently large number of learning iterations.

The presence of a number of statistical classifiers appearing in SLP training raises the problem of which classifier to choose for practical use. From the statistical viewpoint, these classifiers differ in their complexity. Consequently, the choice of the ''best'' classifier depends on the learning-set size and on the complexity of the pattern classification problem (the data). In Part II, we consider generalization properties of the statistical classifiers mentioned and that of the SLP, the overtraining effect, and different means that can be used for complexity control of SLP classifiers in situations with a small learning sample size.

## Acknowledgements

The author thanks Valdas Dičiūnas and an anonymous referee for useful remarks, and E.R. Davies for his aid in preparing the final version of the paper.

## References

Abramson, N., & Braverman, D. (1962). Learning to recognise patterns in a random environment. *IRE Transactions on Information Theory, IT-8*, 58–63.

Amari, S. (1967). A theory of adaptive pattern classifiers. *IEEE Transactions on Electronic Computers, EC-16*, 299–307.

Anderson, T.W. (1951). Classification by multivariate analysis. *Psychometrika, 16*, 31–50.

Anderson, T.W., & Bahadur, R.R. (1962). Classification into two multivariate normal distributions with different covariance matrices. *Ann. Math. Stat., 33*, 420–431.

Babu, C.C., & Chen, W.C. (1971). An optimal algorithm for pattern classification. *Int. J. Control, 13 (3)*, 577–586.

Boser, B., Guyon, I. and Vapnik, V. (1992). A training algorithm for optimal margin classifiers. In *Proc. of the 5th Annual Workshop on Computational Learning Theory, ACM: Pittsburgh, Vol. 5* (pp. 144–152).

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning, 20*, 273–297.

Di Pillo, P.J. (1979). Biased discriminant analysis: evaluation of the optimum probability of misclassification. *Commun. Statist. Theory and methods, A8 (14)*, 1447–1457.

Do-Tu, H., & Installe, M. (1978). Learning algorithms for non-parametrical solution to the minimum error classification problem. *IEEE Transactions on Computers, C-27*, 648–659.

Duin, R.P.W. (1995, June). *Small sample size generalization*. Paper presented at 9th Scandinavian Conference on Image Analysis (SCIA '95), Uppsala, Sweden, Vol. 2, 957–964.

Fisher, R.A. (1936). The use of multiple measurements in taxonomic problems. *Ann. of Eugenics (London), 7 (2)*, 179–188.

Friedman, J.M. (1989). Regularized discriminant analysis. *J. American Statistical Association, 84*, 165–175.

Fukunaga, K. (1990). *Introduction to statistical pattern recognition*. New York: Academic Press.

Geisser, S. (1964). Posterior odds for multivariate normal classifications. *J. Royal Stat. Soc., Ser. B, 21 (1)*, 69–76.

Glucksman, H. (1966). On improvement of a linear separation by extending the adaptive process with a stricter criterion. *IEEE Transactions on Electronic Computers, EC-15 (6)*, 941–944.

Gupta, A.K. (1977). On the equivalence of two Classification Rules. *Biometrical Journal, 19 (5)*, 365–367.

Hoerl, A.E., & Kennard, R.W. (1970). Ridge regression: biased estimation for orthogonal problems. *Technometrics, 12*, 55–67.

Ibaraki, T., & Muroga, S. (1970). Adaptive linear classifier by linear programming. *IEEE Transactions on Systems Science and Cybernetics, SSC-6*, 53–62.

Keehn, D. (1965). A note on learning for Gaussian properties. *IEEE Transactions on Information Theory, IT-11*, 126–131.

Koford, J.S., & Groner, G.F. (1966). The use of an adaptive threshold element to design a linear optimal pattern classifier. *IEEE Transactions on Information Theory, IT-12*, 42–50.

Malinovskij, L.G. (1979). *Hypotheses on subspaces in the problem of discriminant analysis of normal populations* (pp. 195–206). Moscow: Nauka (in Russian).

McLachlan, G.J. (1992). *Discriminant analysis and statistical pattern recognition*. New York: Wiley.

Miyake, A. (1979). Mathematical aspects of optimal linear discriminant function. In *COMPSAC'79, Proc. IEEE Comput. Soc. 3rd Int. Comput. Software and Applic. Conference, Chicago, IL* (pp. 161–166). New York, N.Y.: IEEE.

Patterson, D.W., & Mattson, R.L. (1966). A method of finding linear discriminant functions for a class of performance criteria. *IEEE Transactions on Information Theory, IT-12*, 380–387.

Pietrantonio, H., & Jurs, P.C. (1972). Iterative least squares development of discriminant functions for spectroscopic data analysis by pattern recognition. *Pattern Recognition, 4*, 391–400.

Randles, R.H., Brofitt, J.D., Ramberg, I.S., & Hogg, R.V. (1978). Generalised linear and quadratic discriminant functions using robust estimates. *J. of American Statistical Association, 73 (363)*, 564–568.

Raudys, S. (1967). On determining the training sample size of a linear classifier. In N. Zagoruiko (Ed.) *Computing systems, Vol. 28* (pp. 79–87). Novosibirsk: Nauka, Institute of Mathematics, Academy of Sciences USSR (in Russian).

Raudys, S. (1972). On the amount of a priori information in designing the

classification algorithm. *Proc. Acad. Sci. USSR, Technical. Cybernetics*, *4*, 168–174. (in Russian).

Raudys, S. (1991). Methods for overcoming dimensionality problems in statistical pattern recognition. A review. *Zavodskaya Laboratorya*, **3**, 45 and 49–55 (in Russian). Paris: EC2&Cie.

Raudys, S. (1995, October). A negative weight decay or antiregularisation. In *Proc. ICANN'95, Vol. 2*, Paris (pp. 449–454).

Raudys, S. and Skurikhina, M. (1994, May). Small sample properties of ridge estimate of the covariance matrix in statistical and neural net classification. In *New trends in probability and statistics, multivariate statistics and matrices in statistics, Proc. of the 5th Tartu Conference, Vol. 3*, Tartu-Puhajarve, Estonia (pp. 237–245). Vilnius: TEV.

Rumelhart, D.E., Hinton, G.E. and Williams, R.L. (1986). Learning internal representations by error propagation. In *Parallel distributed processing: explorations in the microstructure of cognition, Vol. 1* (pp. 318–362). Cambridge, MA: Bradford Books.

Schurmann, J. (1977). *Polynomklassifikatoren fur Zeichenerkennung*. Munchen/Wien: R. Oldenbourg Verlag.

Vapnik, V.N. and Chervonenkis, D.Ya. (1974). *Theory of pattern recognition — statistical learning problems.* Moscow: Nauka (in Russian).

Warmack, R.E., & Gonzales, R.C. (1973). An algorithm for optimal solution of linear inequalities and its application to pattern recognition. *IEEE Transactions on Computers, C-22*, 1065–1075.

Widrow, B. and Hoff, M.E. (1960). Adaptive switching circuits. *WESCON Convent. Record.*.

Wolf, A.C. (1966). The estimation of the optimum linear decision function with a sequential random method. *IEEE Transactions on Information Theory, IT-12*, 312–315.

Yau, S.S., & Schumpert, J.M. (1968). Design of pattern classifiers with the updating property using stochastic approximation techniques. *IEEE Transactions on Computers, C-17*, 908–1003