INFORMATICA, 1992, Vol.3, No.3, 301-337

ON TRAINING SAMPLE SIZE AND COMPLEXITY OF ARTIFICIAL NEURAL NET CLASSIFIER

Anil JAIN
Department of Computer Science
Michigan State University
East Lansing, MI 48824, USA

Šarūnas RAUDYS
Department of Data Analysis
Institute of Mathematics and Informatics
2600 Vilnius, Akademijos St.4, Lithuania

Abstract. Small training sample effects common in statistical classification and artificial neural network classifier design are discussed. A review of known small sample results are presented, and peaking phenomena related to the increase in the number of features and the number of neurons is discussed.

Key words: feed forward neural nets, classification error training samples, dimensionality, complexity, peaning phenomenon.

1. Introduction. In recent years artificial neural networks (ANN) have become one of the most frequentely used classification techniques in pattern recognition. A classification error of ANN classifier is low, a recognition speed is high both in sequential and paralel implementation. A typical ANN classifier consists of several layers of neurons (see Fig. 1). Each (say β^{th}) neuron has several (say d) inputs $i_1, i_2, \ldots, i_d$, one output o_β, and performs an operation,

$$o_\beta = f_\beta(net_\beta), \tag{1}$$

where $net_\beta = \sum_{\alpha=1}^{d} w_{\beta\alpha} i_\alpha + w_{\beta_0}$, w_{β_1}, $w_{\beta_2}, \ldots, w_{\beta_o}$ are weights of β-th neuron and f_β is a nondecreasing and differentiable activation function, e.g.

$$f_\beta(net_\beta) = 1/(1 + e^{-net_\beta}). \tag{2}$$

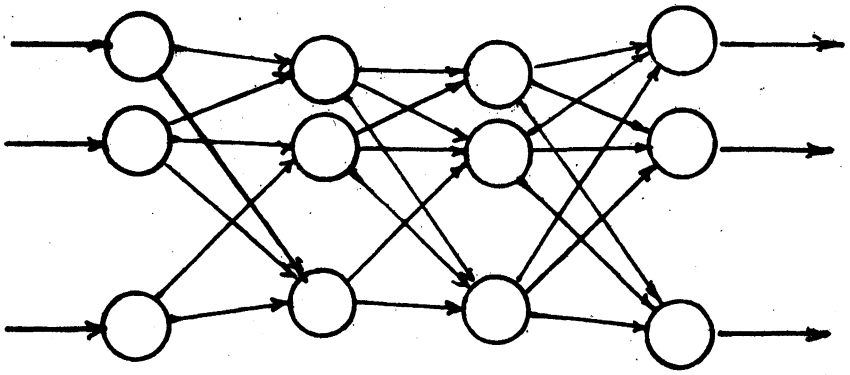

Fig. 1. An ANN With Two Hidden Layers.

The input nodes of the ANN classifier correspond to the components of the feature vector to be classified. The inputs to the neurons in each of the hidden layers consist of the outputs of the preceeding layers. The neurons in the output layer are usually associated with pattern class labels.

The important design issues in building an ANN classifier are to find an appropriate network topology (number of hidden layers, number of neurons in each layer) and to learn the weights w_{ij} for each neuron from the given training samples. If an one-layer ANN classifier with a single neuron and hard limiting threshold activation function (a simple perceptron) is used, then a simple linear discriminant function is realized and the resulting decision surface is a hyperplane [43]. On the other hand, a multilayer ANN with soft limiting threshold activation function can realize an arbitrarily complex decision surface [12, 27, 41, 42, 67]. A number of methods exist to train an ANN [10, 23, 24, 28, 42, 43, 59, 66]. The training methods used to design ANN classifier differ in the error function (3), in the type of the activation function f_β (net) and in an optimization technique used to determine the weights in the neural net. Let o_β be the actual output and t_β be the desired output of the β^{th} neuron on in the output layer of ANN. The most popular error function is the mean square error function, defined as

$$MSE = \frac{1}{n}\sum_{p=1}^{n} MSE_p = \sum_{p=1}^{n}\sum_{\beta=1}^{k} \varepsilon(o_{\beta p} - t_{\beta p}) = \frac{1}{n}\sum_{p=1}^{n}\sum_{\beta=1}^{k}(o_{\beta p} - t_{\beta p})^2, \quad (3)$$

where n is the number of training samples and k is the number of neurons in the output layer, and $\varepsilon()$ denotes the error function.

Instead square function $(o_{\beta p} - t_{\beta p})^2$ one can use a modulus function. In many training algorithms, the weights $w_0, w_1, \ldots, w_d$ are changed only when a training vector X_p is incorrectly classified (error-correction algorithm) [61, 63]. For example, in the relaxation algorithm REL,

$$\varepsilon(c) = \begin{cases} c^2, & \text{when } X_p \text{ is incorrectly classified,} \\ 0, & \text{otherwise,} \end{cases} \quad (4)$$

and in the fixed increment algorithm FIX

$$\varepsilon(c) = \begin{cases} |c|, & \text{when } X_p \text{ is incorrectly classified,} \\ 0, & \text{otherwise.} \end{cases} \quad (5)$$

When one wants minimize an empirical probability of misclassification a hard-limiting threshold function

$$\varepsilon(c) = \begin{cases} 1, & \text{when } X_p \text{ is incorrectly classified,} \\ 0, & \text{otherwise,} \end{cases} \quad (6)$$

should be employed.

In their classical paper Rumelhort, Hinton and Williams [59] recommended to use activation function (3) and the values of 0,1 and 0,9 as the fargets. The error function plot of such MSE_p versus net_p is presented in Fig. 2.

The standard learning procedure used to find weights $w_{\beta\alpha}$ involves the presentation of a set of pairs of input and output patterns and a changing the weights according to a value of the pattern error function. A sequential presentation of all training vectors will be called a learning sweep or training pass. Usually several tens or even thousands of learning sweeps are required in order to find the weights.

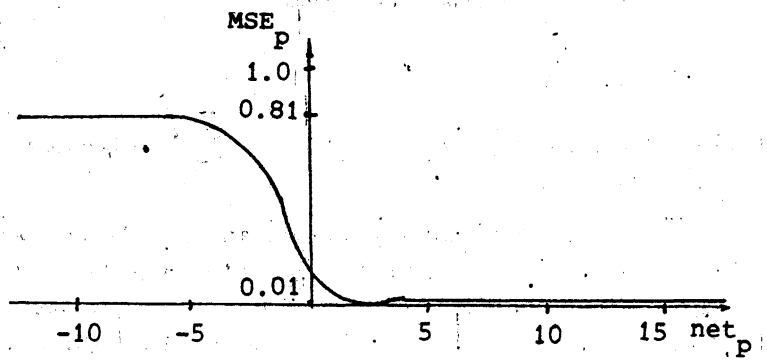

Fig. 2. Pattern error function MSE_p of the output layer versus net_p.

Thus the feed forward ANN classifiers and training methods used to design ANN pattern recognition system differ in:

1) an ANN architecture (number of inputs α, number of outputs k, number of hidden layers and number of neurons on these layers, connections between the neurons),
2) a type of pattern error function $\varepsilon(c)$,
3) a type of activation function $f(net)$,
4) an optimisation technique used to minimise the error function.

An information used to design ANN pattern recognizer consists of: a) prior information about the pattern recognition problem to be solved, b) training samples.

The prior information is used to choose the ANN architecture and the training algorithm; the training samples are used to specify values of weights of the ANN classifier with the given architecture.

The ANN classifier can be analyzed as a special class of statistical pattern classifiers which are derived from the training samples or data-driven, such as Parzen-window classifiers and k-NN classifiers [14]. It is well known that, in a finite training sample case, the expected classification error EP_n of a statistical pattern classifier increases due to an inexact determination of the parameters of the classifier. The finite number of training samples causes the following practical difficulties and constraints [8, 14, 15, 29, 31, 34,

49, 54, 64]:

1. The resubstitution error rate estimate has an optimistic bias.
2. A peaking in classification performance is observed as the number of features increases.
3. A need to use a simple classification algorithm instead of a complex one.
4. A need to reduce the number of features.
5. A need to find an optimal value of the smoothing parameter in the Parzen window classifier.
6. A need to determine an optimal value of the number of nearest neighbors (K) in the k-NN classifier.
7. To balance the plug-in sample discriminant functions in the case of unequal numbers of training samples per class.

The ANN classifiers are also constructed primarily from training samples. Therefore they will also suffer from most of the finite sample problems mentioned above.

The purpose of this paper is to analyze the small sample effects that occur in the design of the ANN classifiers.

In Sec. 2 we present some known results concerning the influence of the number of training samples on the accuracy of several parametric and nonparametric statistical classifiers. These results will be useful in analyzing similar small sample affects for the ANN classifiers. In Sec. 3 we discuss the classification accuracy and training time of the ANN classifiers. Sec. 4 deals with the problem of classification error estimation. In Sec. 5 and 6 we analyze the peaking phenomena which arise due to an increase in the number of inputs and the number of hidden layer nodes in multilayer ANN classifiers. Sec. 7 consists of discussion and suggestions for future research.

2. Effects of finite number of training samples on the performance of statistical pattern classifiers. It is generally recognized that more complex classification rules are more sensivite to training sample size [15, 21, 29, 42, 49, 54, 64]. In order to determine a complexity of the classification rules many authors [3, 4, 5, 40, 64] use a measure of richness of a class of classification

rules Γ called Vapnik-Chervonenkis (VC) dimension which is closely related to a maximal number of ways to divide n d-dimensional vectors into two parts by any rule from the class Γ. Blumer *et al.* [5] have proved a folowing theorem.

If the VC dimension of a set of classification rules is V, then for any confidence level $\delta < 1$, generalisation error's limit $\varepsilon > 0$, training sample error $\hat{\varepsilon} > 0$ and training sample size n, greater than:

$$n \geqslant \max\left(\frac{8}{\gamma^2\varepsilon}\ln\left(\frac{8}{\delta}\right), \quad \frac{16V}{\gamma^2\varepsilon}\ln\left(\frac{16}{\gamma^2\varepsilon}\right)\right), \tag{7}$$

where $\gamma = 1 - \hat{\varepsilon}/\varepsilon$ the probability that there exists a classification rule in a class Γ with the generalisation error $p_t > \varepsilon$ is at most δ.

Many authors [3, 5, 40] agree VC dimension V for the two-layer feed forward ANN classifier with d input and h neurons in the hidden layer is equal to

$$(h+1)d, \tag{8}$$

i.e., a number of weights w to be determined from the training samples.

Equations 7 and 8 express qualitative and quantitative relation between the compexity of the classification rule, training sample size, training and generalisation classification errors. E.g., for the ANN with two inputs ($d = 2$) three neurons in the hidden layer ($h = 3$), when training sample error $\hat{\varepsilon} = 0.05$, the generalisation error's limit $\varepsilon = 0.15$, the reliability $\delta = 0.5$ we have $\gamma = 0,6667$, $V = 8$ and $n = \max(333, 10\,523) = 10\,523$ samples.

For the ANN classifier with $d = 20$, $h = 20$ when $\hat{\varepsilon} = 0.018$, $\varepsilon = 0.05$, $\delta = 0.5$ we have $\gamma = 0.64$, $V = 420$ and $n = \max(1083, 2\,185\,606) = 2\,185\,606$ samples.

Numerous simulation studies show, however, that above theoretical estimates of the number of training samples required to achieve valid generali ation are too high. E.g., for ANN with $d = 2$, $h = 3$, $\hat{\varepsilon} = 0$ we got the generalisation error $P_t = 0.06$ with significantely smaller number of training vectors ($n = 80$). Baum [3] reported experiments with $d = 20$, $h = 20$, $\hat{\varepsilon} = 0.016$ and $p_t = 0.05$

when $n = 8800$. Possible reasons of too high values of n in theoretical Eq. 7 are that a derivation of these equations requires a use of some upper bounds several times, the equations are derived for most unfavorable conditions. In real applications of the ANN classifiers conditions are not so unfavorable, therefore even qualitative relation between the generalisation error, complexity and training sample size can be totaly different than that predicted by Eq. 7 and 8.

In this context an analysis done in statistical pattern recognition using traditional methods of multivariate statistical analysis become very important.

One of the most popular and simplest statistical pattern classifier is the Fisher linear discriminant function (LDF)

$$g(x) = \sum_{\beta=1}^{d} w_\beta x_\beta + w_0 = net, \tag{9}$$

where $x_1, x_2, \ldots, x_d$ denote the d features, and $w_0, w_1, \ldots, w_d$ are constants.

For a two-class problem, if $g(X) > 0$ then the feature vector $X = (x_1, x_2, \ldots, x_d)^T$ is allocated to class π_1, otherwise to class π_2. The linear discriminant function linearly maps the training patterns from each class on the real line defined by the weight vector $W = (w_0, w_1, w_2, \ldots, w_d)^T$. Fisher [19] chose the weight vector such that the mean squared deviation of the projected training patterns around their class mean vectors (within-class scatter) is minimized with respect to the seperation between the class mean vectors (between-class scatter). The weights of this LDF are identical to those obtained from the 'plug-in' decision rule for the case of two Gaussian class-conditional density functions when the unknown mean vectors and the common covariance matrix are replaced by their sample estimates. The same weight vector can also be obtained by the least-mean-square-error adaptation algorithm for an equal number of observations ($N_1 = N_2$) from both classes [22, 36]. Therefore, the linear discriminant function is, in fact, an one-layer perceptron with a linear activation function $f(net) = net$ and

trained by the standard delta rule learning algorithm [59] (where MSE criterion is minimised).

The standard LDF has been analyzed by a number of authors [17, 45, 46, 55, 68]. The expected probability of misclassification, EP_N, of LDF can be written as

$$EP_N = q_1 P\Big\{g(X) \le 0 \mid X \in \pi_1\Big\} + q_2 P\Big\{g(X) < 0 \mid X \in \pi_1\Big\}, \quad (10)$$

where q_1 and q_2 are prior probabilities of classes π_1 and π_2, respectively. The expected probability of misclassification, EP_n, (the generalisation error in the ANN terminology) of the LDF depends on the number of training samples per class N_1 and N_2, dimensionality of the feature vector d, and the asymptotic probability of misclassification, P_∞ (Table 1).

$$P_\infty = \lim_{N\to\infty} EP_n.$$

Pikelis [46], Wyman, Young and Turner [68] compared a number of asymptotic expansions of the expected probabilty of misclassification for LDF and found that Deev's expansion [13] is very exact. In the case of Gaussian class-conditional density functions with a common covariance matrix, the first term on the right hand side of Eq. 10 can be computed as follows [13]

$$P\{g(X) > 0 | X \in \pi_2\} = \phi\left\{-\frac{\delta}{2}\frac{1+\frac{N_2-N_1}{N_1N_2}\frac{d-1}{\delta^2}}{\sqrt{\frac{N_1+N_2-1}{N_1+N_2-d}\frac{N_1+N_2+1}{N_1+N_2}\left(1+\frac{N_1+N_2}{N_1N_2}\frac{d-1}{\delta^2}\right)}}\right\}, \quad (11)$$

where N_i is the number of training samples from class π_i, $\phi(c)$ is Laplace's probability integral and δ^2 is the squared Mahalanobis distance, $\delta^2 = (\mu_1 - \mu_2)^t \sum^{-1}(\mu_1 - \mu_2)$. The other term in Eq. 10 $P\{g(X) \le 0 | X \in \pi_1\} = 1 - P\{g(X) > 0 | X \in \pi_1\}$ can be obtained from Eq. 11 by interchanging N_1 and N_2.

Eq. 11 shows that if $d \to (N_1 + N_2 - 2)$, i.e., when the estimated covariance matrix becomes singular, the misclassification error increases enormously. Due to the nonlinear nature of the Laplace integral $\phi(c)$, the term

Table 1. The values of the relative increase in the expected probability of misclassification $\kappa = EP_n/P_\infty$ of standard linear discriminant function for two Gaussian classes with common covariance matrix

P_∞	0.2	0.1	0.03	0.01	0.001	d
$N\backslash\delta$	1.68	2.56	3.76	4.65	5.50	
2	2.00	3.26	8.27	20.6	59.3	
3	1.64	2.22	4.15	8.00	17.9	
6	1.31	1.50	2.00	2.74	4.07	
15	1.12	1.17	1.30	1.45	1.66	3
30	1.06	1.08	1.14	1.20	1.27	
150	1.01	1.02	1.03	1.04	1.05	
3	2.11	3.64	9.90	25.7	76.1	
4	1.80	2.56	5.62	11.9	28.9	
5	1.69	2.21	4.01	7.37	15.4	
10	1.32	1.51	2.00	2.66	3.78	5
25	1.13	1.18	1.31	1.45	1.64	
50	1.06	1.09	1.14	1.20	1.27	
250	1.01	1.02	1.03	1.04	1.05	
5	2.04	3.38	8.57	20.9	58.2	
6	1.85	2.78	6.00	18.8	30.9	
8	1.63	2.19	3.87	6.84	13.5	
16	1.32	1.51	1.97	2.59	3.58	8
40	1.13	1.18	1.31	1.45	1.63	
80	1.07	1.09	1.14	1.20	1.27	
400	1.01	1.02	1.03	1.04	1.05	
30	2.05	3.39	8.40	19.8	52.0	
50	1.62	2.15	3.61	5.95	10.6	
100	1.33	1.51	1.93	2.47	3.27	50
250	1.14	1.19	1.31	1.44	1.61	
1000	1.04	1.05	1.07	1.10	1.13	

$$\left(\frac{N_i - N_j}{N_i N_j}\right)\left(\frac{d-1}{\delta^2}\right), \quad i = 3 - j, \quad j = 1,2$$

will increase the classification error when $N_1 \neq N_2$. This degradation in classification performance due to unequal numbers of training samples from different classes is more pronounced in higher dimensional cases when a quadratic discriminant function is used [50, 54].

The generalisation error of the ANN classifier depends on the type of error criteria $\varepsilon()$ used in the training process. Smith [61] derived an approximate formula to calculate the expected probability of misclassification of MSE, REL and FIX adaptation algorithms for linear discriminant functions.

$$EP_n^{\infty} = P_{\infty}^{\alpha} + \frac{1}{2N}(A^{\alpha} + B^{\alpha}(d-1)), \tag{12}$$

where parameters A^{α} and B^{α} depend on the type, α, of the error function (MSE, REL, FIX) and on the asymptotic probability of misclassification (see Table 2).

Table 2. The coefficients A and B in Eq. 12 [61]

Algorithm		MSE		REL		FIX	
δ	P_{∞}	A	B	A	B	A	B
1.0	0.309	.0880	.4400	.0886	.0441	.0152	.505
2.0	0.159	.1210	.2420	.1400	.2610	.1960	.317
4.0	0.023	.0540	.0675	.1680	.1820	.2180	.232

Note, theoretically MSE criteria correspond to the standard Fisher LDF. Consider a classification problem for which the Mahalanobis distance $\delta = 2$, dimensionality $d = 10$, and training sample size $N = N_1 = N_2 = 20$, then Eq. 12 results in $EP_N^{MSE} = 0.2165$ for the standard delta rule with MSE criterion (the exact value from Pikelis' Table [46] is 0.219 and Deev's [13] main term in Eq. 11 gives 0.217); $EP_N^{REL} = 0.221$ for the relaxation error criteria (Eq. 4), and $EP_n^{FIX} = 0.235$ for the fixed increment criteria (Eq. 5). With an increase in the Mahalanobis distance δ (or a decrease in the

asymptotic probability of misclassification P_∞), a role of training samples distant from a classification hyperplane in determining the weights of the linear discriminant function is diminished and, as a result, the differences between the expected errors EP_n^{MSE}, EP_n^{REL} and EP_n^{FIX} increase. When one uses minimum empirical classification error criteria (Eq. 6), the classification error is significantly higher than that of the classifiers which use the MSE, REL and FIX criteria [64]. Therefore, the standard delta rule where the mean-square-error criterion is minimized is the most preferable learning rule to design linear classifiers in terms of their sensitivity to the training sample size.

The above results concerning the linear discriminant function are valid when only a single layer ANN performs the classification, and when the linear activation function $f(net) = net$ is used. When one uses a softlimiting activation function in a two-layer ANN, then Eq. 1 in fact performs a feature extraction procedure. In the case of inaccurate determination of the weights of the "feature extractors", given that

$$o_\beta = f_\beta\Big(\sum_{\alpha=1}^{d} w_\beta i_\alpha + w_{\beta_\alpha}\Big) \tag{13}$$

a set of new "features" $o_1, o_2, \ldots$ define a new feature space where simple classes can sometimes be comparatively easily separated by adjusting the weights of the next layer of neural elements. If the training algorithm adjusts the weights of the output neural elements first, and if there are only a few hidden layer elements (the new "features"), then the ANN can be comparatively well trained by a small number of training samples even in a extremely high dimensional case. Here, the sensitivity of two-layer ANN classifiers to the finiteness of the training sample size is determined mainly by the number of hidden layer neurons. No detailed analysis however has been done in this area.

It is well known that a multi-layer ANN classifier can form complex decision boundaries similar to nonparametric Parzen window or k-NN classifiers [27, 41, 42]. Therefore, a knowledge of the sensitivity of the nonparametric statistical classifiers to the finite-

ness of the training sample size can serve as a guide in the analysis of multi-layer ANN classifiers. Unfortunately, very little is known about the behavior of nonparametric statistical classifiers in the finite sample case.

Raudys [50] obtained some results by means of simulation studies for nonparametric Parzen window classifiers. The classification problem involved two multivariate Gaussian populations with identity covariance matrices. He used the following window function

$$K\left(\frac{X-X_p}{\lambda}\right) = k' * \exp\left\{-\frac{(X-X_p)^T(X-X_p)}{\lambda^2}\right\}, \qquad (14)$$

where λ is the window width, X_p is a training sample, and k' is a constant.

The relative increase in the classification error (EP_n/P_∞) is presented in Table 3 for two values of the smoothing parameter ($\lambda = 0.1$ and 0.8). The increase in the error rate of PW classifier to the training sample size drops when the value of the smoothing parameter increases. When $\lambda \to 0$, PW classifier with the Gaussian window function performs no "smoothing" and PW classifier becomes similar to an 1-NN classification rule. Thus, values of $\kappa = EP_n/P_\infty$ presented for $\lambda = 0.1$ are practically the values of κ for the 1-NN classifier. Note that the increase in the generalisation error rate of the nonparametric statistical pattern classifiers to the number of the training samples increases with an increase in number of features d is more significant than for parametric MSE adaptation rule [54]. The decrease of the generalisation error with an increase in the training sample size is also slower for nonparametric classifiers (usually of order $1/\sqrt{n}$ or $1/\sqrt[3]{n}$). Therefore, in order to design complex decision boundaries in a high dimensional feature space with the help of nonparametric statistical classifiers, a large number of training samples is required. We suspect that in a two-category case, the complex multi-layer neural net classifiers with a large number of hidden neurons and inputs can have similar behavior.

The search for the appropriate architecture and the weights of an ANN classifier is an optimization problem for a given error

Table 3. The values of the relative increase in the expected probability of misclassification $\kappa = EP_n/P_\infty$ of Parzen window classifier for two Gaussian classes with common identity covariance matrix. Results are reported for two values of the smoothing parameter: $\lambda = 0.1/0.8$

P_∞	0.1	0.01	d
$N\backslash\delta$	2.56	4.65	
2	1.97/1.96	3.53/3.51	
3	1.90/1.80	2.92/2.86	
6	1.78/1.68	2.61/2.51	3
15	1.64/1.46	2.32/2.18	
30	1.50/1.23	2.15/1.71	
150	1.39/1.06	1.53/1.20	
3	2.15/2.12	3.55/3.51	
5	1.98/1.94	3.24/3.15	
10	1.87/1.80	3.07/2.87	5
25	1.71/1.58	2.56/2.38	
50	1.66/1.44	2.16/1.90	
250	1.62/1.12	1.86/1.27	
5	2.28/2.26	4.29/4.21	
8	2.13/2.05	3.48/3.41	
16	1.95/1.90	3.18/3.08	8
40	1.91/1.76	3.72/2.46	
80	1.84/1.65	2.28/2.10	
400	1.81/1.29	2.14/1.54	

function. This optimization problem itself in fact involves selecting a variant of an ANN classifier among an immense number of all possible ANN classifiers with a given architecture. Raudys [53, 56, 57] analyzed a problem where the best model needs to be selected from an infinite general population (M) of the models. Let the i-th model $M_i \in M$ be characterized by some value of the error function P_i, and there exists an estimate $\widehat{P}_i$ of P_i. In the ANN design $\widehat{P}_i$ is, e.g., MSE found after the minimisation of Eq. 3 and P_i is the gener-

alisation error. It is assumed that $P_1, P_2, \ldots$ are random variables in some interval $(P_{\min}, P_{\max})$ with a density function $f(P)$. Similarly, $\widehat{P_1}, \widehat{P_2}, \ldots$ are random variables with the conditional density function $f(\widehat{P}_i|P_i)$. Due to inaccuracies in the estimates $\widehat{P_1}, \widehat{P_2}, \ldots, \widehat{P}_m$ the selection of the best model according to the estimates $\widehat{P_1}, \widehat{P_2}, \ldots, \widehat{P}_m$ results in an increase in the value of the true error function P_{true} compared with the value of the error P_{ideal} in an ideal selection procedure which uses only exact values $P_1, P_2, \ldots P_m$.

The true error P_{true} exceeds the error in ideal selection P_{ideal}. Both errors P_{true} and P_{ideal} are random variables due a random character of best model selection procedure. With increase in number of models compared, m, a mean value EP_{ideal} appraches minimal value $P_{\min}$. A mean value EP_{true} diminishes too, however a difference $EP_{\text{true}} - EP_{\text{ideal}}$ remains. It is proportional to a standard deviation (SD) of the estimates $\widehat{P}_1, \ldots, \widehat{P}_m$. It we use an empirical error counting error estimates, then

$$SD(\widehat{P}_j) = \sqrt{\frac{P_j(1-P_j)}{n}}, \tag{15}$$

where n is a number of samples used to obtain estimates $\widehat{P}_j$ (in ANN training procedure n is the number of training samples). When the number m is very great EP_{ideal} is very close to $P_{\min}$, however the difference $EP_N - P_{\min}$ constitutes several times of the standard deviation: when, e.g., $m = 10^6$ and $P_{\min} = 0.2$ then $EP_n - P_{\min} \approx SP(\widehat{P}_j)$ and when $P_{\min} = 0.01$ $EP_n - P_{\min} \approx (4 \div 5)SP(P_j)$ [53].

The theoretical values of the relative increase in the classification error due to an inaccurate selection of the best model provide only guidelines for real model selection tasks. They show, however, that when the sample size is very small then the increase in the true classification error is rather significant and there is only a small chance that a good algorithm (model) will be selected. The same conclusion is valid for the ANN optimization problem. The performance of the complex multi-layer ANN classifier trained on a small number of samples will differ significantly from that of the ANN classifier ideally trained on an arbitrarily large number of samples.

Above theoretical considerations show the increase in classification error due to the finiteness of training sample size is proportional to $\sqrt{1/n}$.

Unfortunately, the above theoretical model does not allow us to evaluate the influence of dimensionality, number of hidden layers, and other parameters of the architecture of a neural net clasifier. In order to get some general view on the influence of the complexity of ANN and the training sample size on the generalisation error Baum [3] performed special simulation studies. He used a target ANN with 20 inputs and 10 neurons in the hidden layer in order to partition 20-variate uniformly distributed random vectors into two classes. He used these vectors to train another neural nets with $H = 10, 20, 40$ and 80 neurons in the hidden layer. For $H = 10$ and training sample size $n = 4400$ he got the generalisation error $P_t = 0.108$; for $H = 20$ and $n = 2200, 4400, 8800$ he got $P_t = 0.177, 0.122, 0.050$ respectively; for $H = 40$, $n = 4400$, 00, 600 he got $P_t = 0.130$, 0.096, 0.062 and for $H = 80$, $n = 4400$, 8800, 17600, 35200 he got $P_t = 0.143$, 0.123, 0.071, 0.051. He concluded these results are roughly consistent with the heuristics $P_t = d \cdot h/n$, based on theory mentioned in the begining of this Sec. (see, e.g., Eq. 7 where n and ε–generalisation error) are opposite proportional.

A more careful analysis of Baum's empirical results shows that the decrease in the generalisation error P_t with increase in training sample size is of order $1/\sqrt{n}$ or $1/\sqrt[3]{n}$ what is characteristic to the nonparametric Parzen window or k-NN classifiers or to training procedures where training is performed via selection of the best model. Therefore we have chosen a heuristic of the following form

$$P_t = (k_1 + k_2(H)^{k_3})/\sqrt{n}, \tag{16}$$

which with $k_1 = 6$, $k_2 = 0.4$ and $k_3 = 1/2$ appeared to be more accurate for Baum's [3] empirical data (there are only two anomalies: for $n = 8800$ $P_t = 0.05$ for $H = 20$ and $P_t = 0.123$ for $H = 80$). Note the empirical estimates were greately influenced by imperfect training since in all cases zero empirical error was not achieved. Empirical Eq. 16 represents only one type of the target net and

can not pretend for wide generalisations. E.Baum [3] investigated an influence of a complexity of the pattern recognition problem (the complexity of the target net in his experiments) and found the generalisation error is more influenced by the complexity of the problem than by the complexity of the trainee ANN classifier.

Above empirical results leads to an interesting conclusion: for particular concrete pattern recognition problem quantitative and even qualitative relations between training sample size n, number of dimensions d, number of neurons in hidden layer H and the generalisation error can be different from that predicted theoreticaly for most unfavourable case of real distributions.

Multi-layer neural network classifiers correspond to a wide class of classification rules. By changing the shape of activation function $f(\cdot)$ in Eq. 1, and the architecture of the ANN, one can obtain classification algorithms with different features. The sensitivity (increase in the classification error) of the ANN to the finiteness of the training sample size will lie between:

a) sensitivity of single threshold linear element with the number of features equal to the number of neurons in the highest hidden layer and trained by delta adaptation algorithm;

b) the sensitivity of the 1-NN classifier or a classifier formally obtained by minimizing inaccurate sample-based classification error criteria.

Very little work has been done in the statistical analysis of the ANN classification algorithms.

3. The classification accuracy and training time of artificial neural networks. It is generally known [42] that traditional multi-layer artificial neural net classifiers trained by the back propagation algorithm require a great number of sweeps of the training sample data in order to minimize empirical classification error. Other types of artificial neural nets (kernel classifiers, exemplar classifiers, decision tree classifiers) do not form *a priori* definite structure of decision boundary and train relatively rapidly [42]. Thus, the long training time is one of the principal characteristics of the back propagation classifiers.

For a given pattern recognition problem and a fixed set of training samples, the complexity of the decision region required to obtain a good classification accuracy of the training samples is fixed. Therefore, an ANN with more hidden layers and more hidden units will have more degrees of freedom and will require less accurate determination of the weights of the hidden and output units of the neural net. Therefore, in the fixed training sample it will be easier to train a complex neural network than a simple one. This theoretical observation is supported by numerous experimental studies [7, 26, 38, 44, 59]. In an analysis of the XOR problem with a two-layer neural network, Rumelhart *et al.* [59] propose an empirical equation to determine the number, S, of presentations of training samples as a function of the number H of the hidden nodes:

$$S = 280 - 33 \log_2 H. \tag{17}$$

Kung and Hwang [38] present a graph of the convergence time (i.e., training sweeps) versus the number of hidden nodes per layer when various numbers of hidden layers are used (Fig. 3). There were $n = 8$ pairs of randomly generated 12-dimensional input patterns. They observed that a net with more hidden units per layer and with more layers led to a smaller number of training sweeps. They also noted an abrupt reduction in the number of training sweeps around $n - 1$ hidden nodes (n is the number of training patterns). This observation agrees with a well-known fact in statistical pattern recognition theory, that in a d-dimensional case, a hyperplane can discriminate perfectly any $d + 1$ points with arbitrary class labels [9]. Therefore, when the number of hidden units is equal to $n-1$ there is no need to adjust the weights of the hidden layer units!

With an increase in the number of training samples, the complexity of the decision boundaries required for perfect discrimination of the training samples increases. Therefore, "training times are typically longer when complex decision boundaries are required and when networks have more hidden layers" [42].

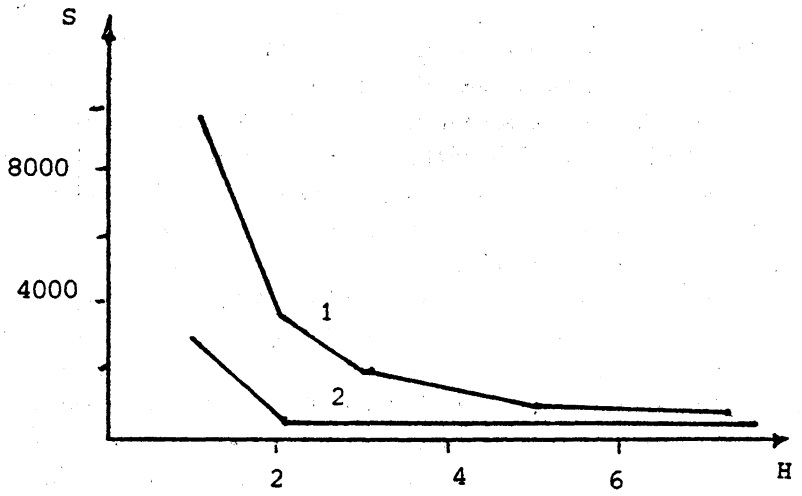

Fig. 3. The convergence time (i.e., number of the training sweeps S) versus the number of hidden nodes per layer in a three-layer (1) and a five-layer (2) ANN [38].

It is generaly recognized the error function (3) of the multilayer ANN classifier is a multiextremal one. The training sample size plays here an important role. Especially a global minimum problem arises in a case when we have more parameters to determine than the number of training samples. Then random initialisation of the weights often leads to a deep local minimum far from the global one. A grafical representation of this phenomena is presented in Fig. 4a, where we have two graphs of changing of MSE (3) and the generalisation error P_t on the line connecting two local minima points in multivariate ANN weights space. The training sample consisted of $n = 8+8 = 16$ two-variate vectors distributed uniformly on two $2\pi/3$ concentric arcs. ANN had two inputs, 3 neurons in the hidden layer and two neurons in the output layer. Note minimum points for MSE and P_t criteria differ in the small training sample case. This observation will be important in further analysis. In larger training sample case $n = 40+40 = 80$ vectors the local minima are not so deep (see Fig. 4b). Therefore an increase in the number

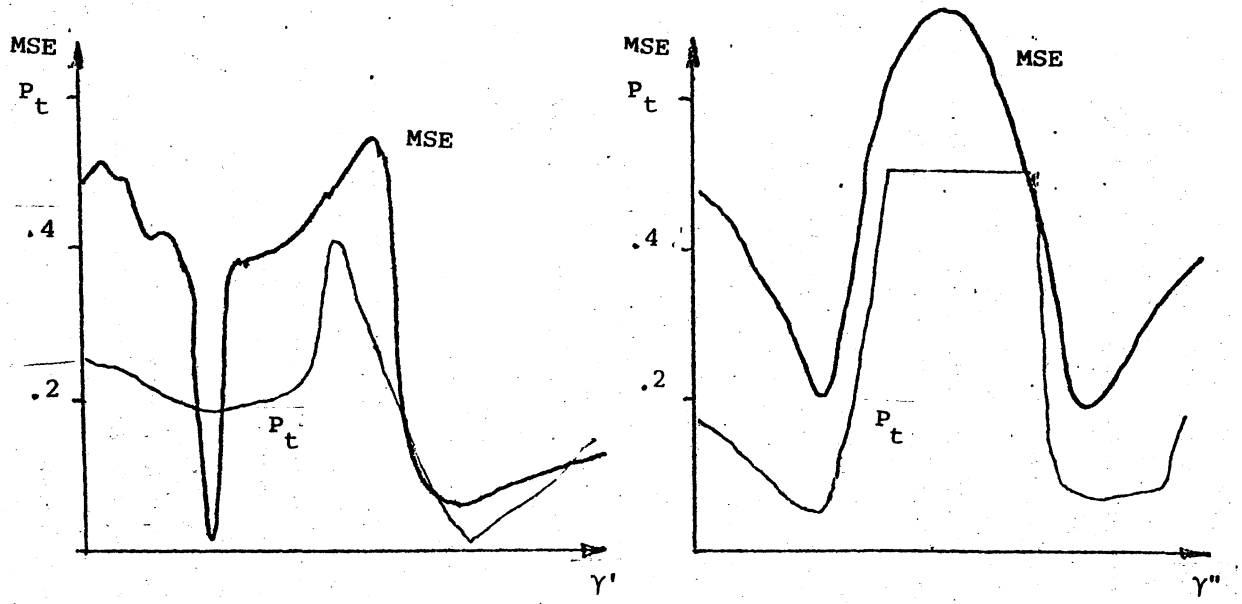

Fig. 4. A change of the mean square error (use) and the generalisation error P_t on lines connecting to local minimum points in multivariate ANN weghts' space [58].

of training samples flattens the local minima and makes the weight initialisation problem more easy to solve. Therefore to avoid the difficulty being trapped in the local minima of the error function we propose to add a small random noise to training sample vectors in each training sweep or to use specially selected weight vectors [58]. Other proposals are:

1) to mix training vectors in each sweep,

2) to use ANN with a probabilistic input-output relation,

3) to utilise a stochastic dynamic by adding a noise term and relaxation term to original back propogation learning algorithm (see, e.g., [60]).

At the end of Sec. 2, in the analysis of the accurracy of the best classifier selection problem on the basis of inaccurate sample-based estimates $\hat{P}_1, \hat{P}_2, \ldots$, we observed that the difference between the true (test sample estimate) classification error of the "best" classifier and the ideal one (obtained when selection is performed on the basisof exact values $P_1, P_2, \ldots$) decreases with an increase in the number of samples used to obtain the estimates $\hat{P}_1, \hat{P}_2, \ldots$ Besides the true classification error P_{true} and the ideal classification error P_{ideal} in the selection process, there exists an apparent error P_{apparent}, i.e. the minimal value among the estimates $\hat{P}_1, \hat{P}_2, \ldots, \hat{P}_m$. (here m is the number of classifiers compared empirically). The mean value of the apparent error (EP_{apparent}) is less than that of the ideal error EP_{ideal}.

The difference $EP_{\text{ideal}} - EP_{\text{apparent}}$ decreases with an increase in the number of samples used to obtain estimates $\hat{P}_1, \hat{P}_2, \ldots, \hat{P}_m$. It is important to note that in the best classifier selection problem, while using random optimization search procedure, the values of the true, ideal, and apparent errors depend on the number m, i.e., the size of the set from which the "best" classifier is selected. All three error rates decrease with increase in m; the apparent error decreases fastest, and the true error slowest (see, for example, curves in Fig. 5 obtained for some theoretical model of $f(\hat{P}, P)$. In experiments with feature selection, when the number m is sufficiently large, the true error practically ceases to decrease, and merely os-

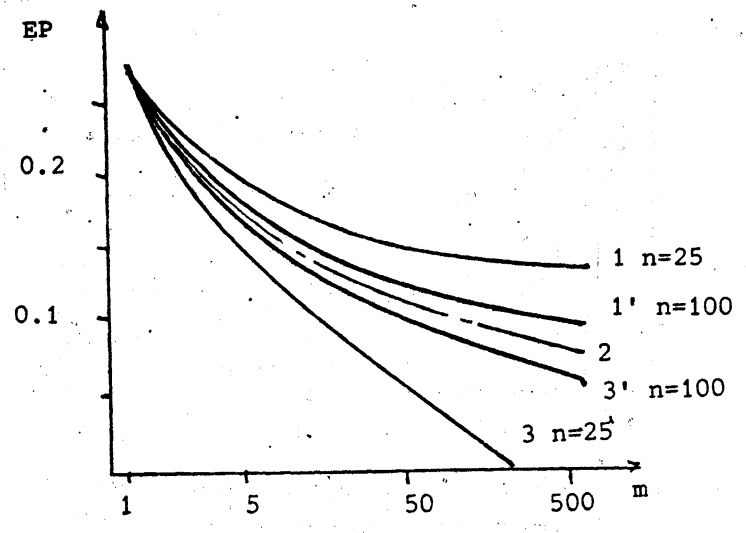

Fig. 5. The mean values of the true EP_{true} (1,1'), ideal (2) and apparent EP_{apparent} (3,3') in best classifier selection versus number, m, of classifiers compared (Beta-Binomial distribution of $(\hat{P}_j, P_j)$ [53]).

cillates [53]. Therefore, in finite design sample size, there is no need to analyze a very large number of classifiers if they were randomly chosen from the set of all possible classifiers. The same conclusion can be drawn for ANN optimization, where the empirical optimisation criteria (3) is not accurate due to a finiteness of the training sample size and due an use of an unacurate error function MSE instead of probability of misclassification (we can see a difference between MSE and P_t easily from Fig. 4a). The optimization itself is a selection of the best classifier from the infinitely number of possible classifiers with a given architecture. Simulation studies confirm this theoretically obtained conclusion. In Fig. 6 we present two such pairs of graphs obtained by means of simulation. Similar graphs were obtained by Le Cun *et al.* [10, 18] while solving a handwritten digit recognition problem by means of a 3-layer artificial neural network.

Therefore, in the ANN training problem, an excessive amount

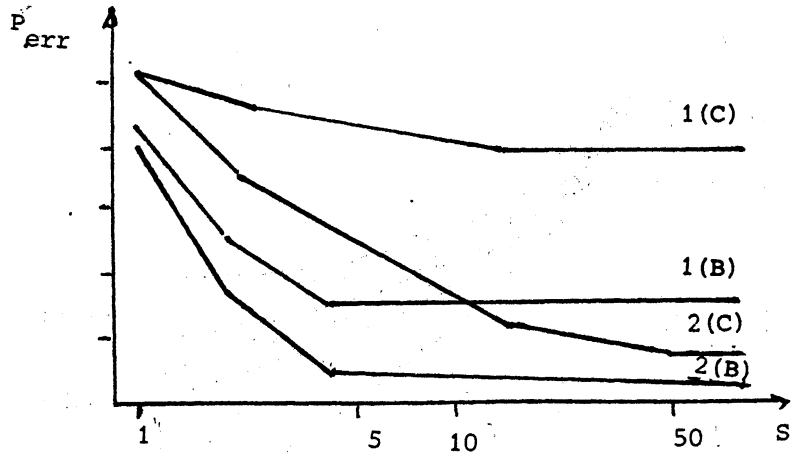

Fig. 6. True (expected=test sample) error (1(B), 1(C)) and apparent (training sample) error (2(B), 2(C)) versus number of sweeps in the back propagation algorithm ($H = 4$, $d = 12$, $n_{\text{training}} = 100$, two spherically Gaussian populations $N(\mu_i, I\delta_i^2)$; $\mu_i = (\mu_{i1}, \ldots, \mu_{id})^T$; data B: $\mu_{ij} = (-1)^i/j$, $\delta_i^2 = 1$; data C: $\mu_1 = \mu_2$, $\delta_i^2 = 4^{i-1}$).

of optimization (training) of the neural network weights is not necessary if the training sample size is small. The optimal number of sweeps required to minimize the true classification error (the test sample estimate) depends on the number of training samples. It increase with an increase in sample size; however, theoretical results which can be recommended for practical use do not exist yet.

The above analysis shows that instead of minimizing the apparent (training sample) error, one should minimize the true (test sample) error. A small oscillation of the empirical error function, usually occuring in the training process, is a positive phenomenon. Control of the variance of the oscillation of the empirical (training sample) error with respect to the training sample size will lead to the design of more effective and rapid ANN training algorithms.

4. Estimation of the classification error. A central problem in classifier design is the estimation of classification error. A

number of techniques exist to estimate the classification error in statistical pattern recognition [21, 25, 30, 62]. It is well known that the resubstitution estimate is optimistically biased. Use of the resubstitution method to estimate error rate of an ANN classifier in the small design sample case will also result in a biased estimate. Therefore, in a finite design sample case, the hold-out method (where independent test samples are used to estimate the classification error) is preferable.

Dutta and Shekhar [16] present the resubstitution ($\widehat{P_R}$) and hold-out ($\widehat{P_H}$) estimates, obtained with a three-layer ANN classifier in 6 and 10 variable cases when the total number of training samples from 4 classes was 30 and the test sample size was 17.

$\widehat{P_R} = 0.20$ for $d = 6$ and $\widehat{P_R} = 0.076$ for $d = 10$,

$\widehat{P_H} = 0.235$ for $d = 6$ and $\widehat{P_H} = 0.176$ for $d = 10$.

We notice a significant difference between $\widehat{P_R}$ and $\widehat{P_H}$ in the 10-dimensional case. The difference $(\widehat{P_H} - \widehat{P_R})$ can serve as a criterion to evaluate the sufficiency of the number of the training samples. Thus, for $d = 6$ we can conclude that the number of the training samples is sufficient, since $\widehat{P_R} = 0.2$ and $\widehat{P_H} = 0.235$, and for $d = 10$ the number of the training samples is not sufficient, since $\widehat{P_R} = 0.076$ and $\widehat{P_H} = 0.176$.

It is important to remember that the estimates $\widehat{P_R}$ and $\widehat{P_H}$ are random variables. Their standard deviations can be approximately evaluated by the following equation [20, 47, 51]:

$$SD(\widehat{P_e}) = \sqrt{\frac{E\widehat{P_e}(1 - E\widehat{P_e})}{n_t}}, \tag{18}$$

where E denotes the expection operator and n_t is the number of observations used to obtain the error estimate $\widehat{P_e}$ (here $e = R$ or H).

In solving practical pattern recognition problems, a researcher needs to select the "best" classifier and evaluate its performance empirically. Even when independent test samples are used to estimate the classification error of the best classifier, the result will still be optimistically biased [53, 56]; the test samples will take the role of additional training samples. If several classifiers are eval-

uated and compared, then the bias mentioned above can become significant; the bias can exceed the standard deviation (Eq. 18) of the estimate $\widehat{P_H}$ used to select the best version [57]. Therefore, one has to remember that in evaluation of the performance of the ANN classifier, an independent test sample, *never* used before, should be used to get an unbiased error rate of the final version of the ANN.

5. Peaking in the classification performance with increase in dimensionality. It is well known that introducing new features never increases the classification error of the optimal Bayes classifier. However, in a finite training sample case, or when one uses an inappropriate statistical model for the real data, the addition of new features can actually increase the classification error. The additional discriminatory information provided by the new features is outweighed by the increase in the inaccuracy of parameter estimates needed in the classification rule. Thus a peaking phenomenon is observed: addition of new features decreases the classification error at first, then the error levels off, and begins to increase [1, 14, 15, 29, 32, 34, 39, 40, 49, 52]. The peaking phenomenon is also observed in the design of ANN classifiers. A single linear threshold element trained by the delta rule in a finite training sample case will have the same behaviour as the standard linear discriminant function. In the latter case, the optimal number of features, d_{opt}, is a function of the asymptotic probability of error P_∞, the number of features d, and the training sample size N. If "best" features (providing the most discriminatory information) are added first and these best features are significantly better than the worst ones, then d_{opt} will be small. If the discriminatory information provided by the individual features is approximately equal, or if we include them in the classifier in a random order then $d_{opt} \approx N - 1$ [32].

When one uses a piecewise linear classifier with H linear hyperplanes, then the "effective" training samples used to determine the weights of each linear boundary segment is approximately [33]

$$N^* = N/H, \tag{19}$$

where N is the number of training samples per class.

A two-layer neural net classifier with a hard-limiting activation function gives a piecewise-linear decision boundary. One can, therefore, expect that when H is small, the number of observations used to adjust the weights of each hidden layer element will be approximately equal to $N^* = N/H$, where H is the number of neurons in the hidden layer.

The classification error of a two-layer neural net classifier ideally trained on an arbitrarily large number of training samples will not increase with the addition of new features or the number of nodes in the input layer. Suppose the asymptotic PMC, P_∞, of this ideally trained two-layer ANN classifier can be determined by the equation [32]:

$$P_\infty = \phi(-\delta\sqrt{1-\gamma^d}), \tag{20}$$

where $0 < \delta$ and $0 < \gamma < 1$.

Then, substitution of $N^* = N/H$ into Eq. 6 results in

$$EP_n = \phi\left\{ -\frac{\delta}{2}\sqrt{\frac{(1-\frac{dH}{2N})(1-\gamma^d)}{1+\frac{2dH}{N\delta^2(1-\gamma^d)}}} \right\}. \tag{21}$$

This function has a minimum with respect to d and, therefore, the optimal number of features [32] can be written as,

$$d_{opt} \approx \frac{N}{H} - \frac{\gamma^{-\frac{2N}{H}} + 2(\frac{N}{H}-2)\ln\gamma - 1}{4\ln\gamma[(\frac{N}{m}-2)\ln\gamma - 1]}. \tag{22}$$

In Fig. 7 we present several graphs that illustrate peaking phenomena. A two-layer ANN classifier with H neurons in the hidden layer was trained by back propagation rule with a sigmoidal activation function and was used to classify two spherically Gaussian pattern classes. A set of 1000 independent test samples was used to estimate the error rate of the ANN classifier. Graphs were obtained by averaging the results of sixty Monte Carlo trials with different training sample sets of fixed size and different initial weights. Similar results, which demonstrate the peaking with increase in dimensionality, were obtained by Kohonen *et al.* [37].

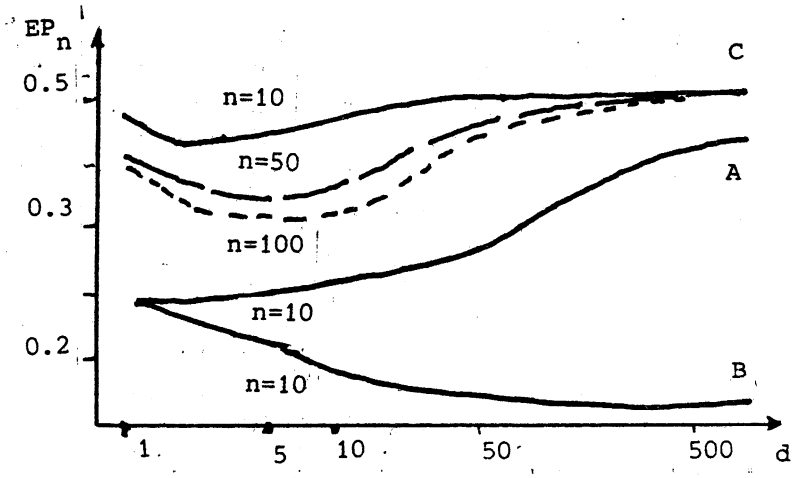

Fig. 7. Expected probability of misclassification EP_n versus dimensionality d (Two classes of spherically Gaussian data $N(\mu_i, I\delta_i^2)$, $\mu_i = (\mu_{i1}, \ldots, \mu_{id})^T$;
data A: $\delta_i = 1$, $\mu_{ij} = (-1)^i/\sqrt{j}$, $H = 2$;
data B: $\delta_i = 1$, $\mu_{ij} = (-1)^i/j$, $H = 2$;
data C: $\mu_1 = \mu_2$, $\delta_i^2 = 4^{i-1}$, $H = 8$).

In a extremely highdimensional case some, training samples will partially be used to determine the weights of each linear boundary segment and Eq. 22 wil not be valid. Then the sensitivity of the two-layer ANN classifier to the finiteness of the training sample size will be determined mainly by the number of hidden layer neurons (see discussion at the end of Sec. 2). Here, the dimensionality of the input patterns will play a less significant role than in the ANN classifier with hard-limiting activation function. Graph B which was obtained for a case where the discriminiting power of the features drops very slowly with the increase in the number of features does not peak at all.

6. Effect of the number of neurons in the hidden layer on the performance of a neural net classifier. It is obvious that the classification error of an ideally trained neural net classifier cannot be increased by introducing new hidden layer neural

elements. With an increase in the number of hidden layer elements, the classification error of the idealy trained ANN classifier, P_∞, will fall sharply at first, then the decrease of the classification error P_∞ will slow, and eventually, the addition of new elements will not effect P_∞. Suppose the classification error of an ideally trained ANN classifier P_∞ can be determined by the equation

$$P_\infty = \phi(-\delta_0\sqrt{1-\gamma^H}), \tag{23}$$

where H is the number of elements in the hidden layer, $0 < \delta$ and $0 < \gamma < 1$.

As in previous Sec. 4, we have assumed that the "effective" number of observations used to adjust the weights of each of the H hidden layer neural elements is determined by Eq. 19. Then, by using the hardlimiting threshold activation function and the generalized delta back propagation algorithm, the increase in classification error of the 2-layer ANN classifier due to finiteness of the number of training vectors can approximately be determined by Eq. 11 with $N^* = N/H$ instead of N :

$$EP_n = \phi\left\{-\frac{\delta}{2}\sqrt{\frac{(1-\frac{dH}{2N})(1-\gamma^H)}{1+\frac{2dH}{N\delta^2(1-\gamma^H)}}}\right\}. \tag{24}$$

Simple numerical analysis of Eq. 24 indicates the existence of peaking phenomena as the number of hidden layer neural elements increases. Several graphs that illustrate the peaking phenomena while increasing the number of hidden units for two classes of Gaussian data are presented in Fig. 8.

Equations 19 and 24 are valid only when the number H of nets in the hidden layer is small. With an increase in H, different neural elements in the hidden layer will be trained partially with the same training samples. Therefore, the "effective" number of training samples used to determine the weights of each linear boundary segment $N^* > N/H$, and addition of the number of hidden nodes H slows down the increase in the expected classification error. These theoretical considerations are supported by numerous experimental studies (Fig. 6 [6, 35, 40, 48]). Khotanzad and

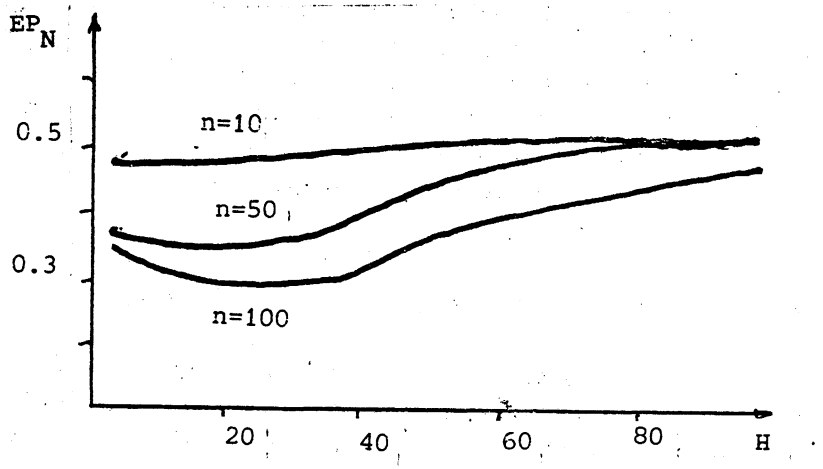

Fig. 8. Expected probability of misclassification EP_N versus the number of neurons H in the hidden layer (Two classes of spherically Gaussian populations, $N(0, I \cdot 4^{i-1})$, $i = 1, 2$, number of variables $d = 8$, training sample size N=10, 50, 100).

Lu [35] trained a two-layer ANN classifier to recognize the English alphabet. Training data consisted of differently positioned, scaled, and oriented 64 × 64 binary images of each of twenty six English characters, 12 images per character. The performance of the ANN classifier peaked at approximately 50 neurons (2% error) in the hidden layer, and with an increase in H, saturated at an error level of 5%.

7. Discussion. Artificial neural net classifiers can be analyzed as a special class of statistical pattern recognition algorithms. In the finite training sample case, some kinds of unexpected and counter intuitive behaviour can sometimes be observed in the design of the ANN classifiers. Some of these are listed below.

1. Increase in the true (test sample or generalisation) classification error due to finiteness of the training sample size;
2. Optimistic bias in the apparent (training sample) error;
3. A presence of deep local minima of the pattern error func-

tion in small sample case;

4. An initial sharp decrease of the true classification error till some limit, while the apparent (training sample) classification error follows to decrease gradually with an increase in number of sweeps of the ANN weight adaptation algorithm;
5. Increase in the number of sweeps required to minimize the true and apparent classification error rates of the ANN classifier with an increase in the number of training samples;
6. An optimistic bias in the error rate of the best version of an ANN classifier selected from several competing models based on finite size test sample data;
7. A peaking in classification performance when increasing the number of features and the number of neural elements in hidden layers of the ANN classifier.

The sensitivity of the generalisation error (GE) of the ANN classifier to the finiteness of the training sample size differs from that of conventional statistical classifiers. A very simple ANN is similar to piecwise linear statistical classifier and therefore its generalisation error is proportional to $1/n$. Decision boundaries of the complex multilayer ANN becomes more similar to boundaries of nonparametric Parzen window or k-NN classifiers and here increase in the GE is proportional to $1/n$ or $1/\sqrt[3]{n}$. Contrary to nonparametric classification rules in the ANN classifiers we have parametrisations in local areas of d-variate space of features and thus here the increase in the generalisation error often is less expressed than for nonparametric classifiers case. Therefore the multi-layer ANN classifiers have the desirable property of reducing the small training sample effects when the number of features and/or the number of neurons in the hidden layer is very large in comparison with the training sample size.

It is worth paying attention once more to the original E.Braum's [3] conclusion, that the increase in the generalisation error of the ANN classifier depends more on the complexity of the

pattern recognition problem than on complexity of the trainee ANN classifier. Therefore in advance it is imossible to predict the dependence of the generalisation error of the ANN classifier on the number of training samples. Some amount of experimental studies have to be made, or one have to use upper bounds.

In the multilayer ANN classifier design the weights of the lowest hidden layer's neurons are features common for all categories. They are trained by using training samples of all categories. Thus, the number of samples used to train the weights common for all the categories is larger than the number of samples used to train the majority of parametric and nonparametric statistical classifiers, where distribution densities of each class usually are estimated separately. Therefore, we can expect that multilayer ANN classifier will outperform conventional statistical classifiers in a high category case, and when the number of weights in the output layer is not too high.

A most sucessfull application of ANN classifiers to real-world problems have been achieved using large networks with large number of parameters (tens of thousands and more) to be determined in the training process [see, e.g., 18, 65]. Estimation of a large number of parameters requires a large number of training samples. In the finite training sample case the estimation of an excessive number of parameters causes undesired effects mentioned above. Therefore while choosing the architecture of the ANN classifier one should pay attention to the training sample size. There are several approaches which help to reduce a number of parameters to be estimated from the training samples [11].

A first approach is problem depending and consists of use of some a prior knowledge about the task and build it into the network. Although in the general case specifying such knowledge may be difficult it appears feasible on some highly regular tasks such as image and speech recognition [18].

A second technique is problem-independent and consis in successive deleting "useless" weights during training. This approach is similar to successive feature elimination (backward selection) pro-

cedure widely used in regression an statistical pattern recognition.

A third technique is again problem dependent and consists in having several connections (links between neurons) be controlled by a single parameter (weight). This technique is especially effective in image recognition when the multilayer ANN classifier has many identical neurons [11, 18]. It seems to be one of the most attractive characteristics of the ANN classifiers in comparison with conventional statistical classifiers enabling us to reduce the number of parameters to be adjusted from the training samples data dramatically.

While designing the ANN pattern recognition systems a priori knowledge on the problem to be solved should he incorporated into ANN as much as possible. A priori knowledge is very useful while designing the architecture of the ANN. Another nontraditional source of additional information lies in that slightly distorted patterns usually are recognized by humans without loss of recognition accuracy. Therefore the adding of the small noise to each training vector in each training sweep not only increases a stability of the training algorithm but also increases in some sense the training sample size.

Above review of research done on statistical pattern recognition and ANN in small training sample size case have showed an efficiency of theoretical and experimental analysis done for concrete types of distributions of the classes. In further analysis of small trainig sample effects a special attention to

a) the complexity of the pattern recognition problem to be solved with trainee ANN classifier,

b) shapes of the pattern error and activation function,

c) number of the classes, should be paid.

Acknowledgements. Jianchang Mao and Marina Skurikhina carried out several experiments reported in this paper. The first author would like to acknowledge the visiting scholarship provided by the Department of Computer Science, Michigan State University.

REFERENCES

[1] Allais, D.C. (1966). The problem of too many measurements in pattern recognition. *IEEE Int. Con. Rec.,* (part 7) 124–130.

[2] Amari (1967). A theory of adaptive pattern classifiers. *IEEE trans. Electron. Comput.,* **EC-16**, 623–625.

[3] Baum, E.B. (1990). When are k-nearest neighbor and back propogation accurate for feasible sized sets of examples. In L.B.Almeida and C.J.Wellekens (Eds.), *Proceedings of the EURASP workshop on neural networks.*

[4] Baum, E.B., and D.Haussler (1989).What size net gives valid generalization? *Neural Computation,* **1**, 151–160.

[5] Blumer, A., A.Ehrenfeucht, D.Haussler and M.Warmuth (1989). Learnability and the Vapnik-Chervonenkis dimension. *J. of the Association for Computing Machinery,* **36**, 929–965.

[6] Bounds, D.G., and P.J.Lloyd (1988). A multilayer perceptron network for the diagnosis of low back pain. In *Proc. 1988 IEEE Intl Conf. on Neural Networks.* San Diego, California. pp. II481–II489.

[7] Burr, D.J. (1988). Experiments on neural net recognition of spoken and written text. *IEEE Trans. ASSP.,* **36**(7), 1162–1168.

[8] Chandrasekaran, B., and A.K.Jain (1979). On balancing decision functions. *J. Cybernet. Inform. Sci.,* **2**, 12–15.

[9] Cover, T.M. (1965). Geometrical and statistical properties of systems of linear inequalities with application in pattern recognition. *IEEE Trans. Elect. Comp.,* **14**, 326–334.

[10] Cun, Le Y. (1989). *Generalisation and Network Design Strategies..* Computer Science. Dept., Univ. of Toronto, Tr CTG-TR-89-4.

[11] Cun, Le Y., J.Decker and S.Solla (1990). Optimal brain damage. In D.Tourecky (Ed.), *Neural Information Processing Systems,* Vol.2.Morgan Kautman.

[12] Cybenko, G. (1989). Approximation by superpositions of sigmoidal function. *Mathematics of Control, Signals and Systems,* **2**(4).

[13] Deev, A.D. (1970). Representation of statistics of discriminant analysis and asymptotic expansions in dimensionalities comparable with sample size. *Report of Academy of Sciences of the USSR,* **195**(4), 756–762 (in Russian).

[14] Devijver, D.A., and J.Kitter (1983). *Pattern Classification, A Statistical Approach.* Wiley, N.Y.

[15] Duin, R.P.W. (1978). *On the Accuracy of Statistical Pattern Recognizers.* Dutch Efficiency Bureau, Pijnacker, The Netherlands.

[16] Dutta, S., and S.Shekhar (1988). Bond rating: A non-conservative application of neural networks. In *Proc. 1988 IEEE Intl Conf. on Neural Networks.* San Diego, California. pp. 443–450.

[17] Estes, S.E. (1965). *Measurement Selection for Linear Discriminant Used in Pattern Recognition.* PhD thesis, Stanford University, Stanford Calif.

[18] Cun, Le Y. et al. (1989). Handwritten digit recognition: Applications of neural network chips and automatic learning. *IEEE Communications Magazine,* 41–46.

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

[20] Foley, D.H. (1972). Considerations of sample and feature size. *IEEE Trans. Inf. Theory,* IT-18(5), 618–626.

[21] Fukunaga, K. (1986). Statistical pattern recognition. In T.Y.Young and K.S.Fu (Eds.), *Handbook of Pattern Recognition and Image Processing.* Academic, New York. pp. 3–32.

[22] Gallinari, P., S.Thiria and F.Fogellman-Soulie (1988). Multilayer perceptrons and data analysis. In *Proc. 1988 IEEE Intl Conf. on Neural Networks.* San Diego, California. pp. I391–I399.

[23] Galuskhin, A.I. (1974). *Synthesis of Multilayer Pattern Recognition Systems.* Energia, Moscow.

[24] Glaz, A.B. (1988). *Parametric and Structuric Adaptation of Decisian Rules in Pattern Recognition Problems.* Zinatne, Riga.

[25] Hand, D.J. (1986). Recent advances in error rate estimation. *Pattern Recognition Letters,* 5, 335–346.

[26] Higashino, J., B.L.Degreefand and E.H.J.Person (1990). Numerical analysis and adaptation method for learning rate of back propagation. In *Proceedings of IEEE IJCNN'90,* Intl Neural Network Society and IEEE, Lawrence Erlbaum Associates, Inc. pp. I627–I630.

[27] Huang, W.M., and R.P.Lippmann (1988). Neural net and traditional classifiers. In D.Anderson (Ed.), *Neural Info. Processing Syst..* American Institute of Physics, New York. pp. 387–396.

[28] Ivakhanenko, A.G. (1975). *Perceptrons – a pattern recognition system.* Naukova Dumka, Kijev.

[29] Jain, A.K., and B.Chandrasekaran (1982). Dimensionality and sample size considerations in pattern recognition practice. In P.R.Krishnaiah and L.N.Kanal (Eds.), *Handbook of Statistics,* Vol.2. North Holland. 835–855.

[30] Jain, A.K., R.C.Dubes and C.C.Chen (1987). Bootstrap techniques for error estimation. *IEEE Trans. Pattern Anal. Machine Intell.,* PAMI-9(9), 628–630.

[31] Jain, A.K., and M.D.Ramaswami (1988). Classifier design with Parzen windows. In E.S.Gelsema and L.N.Kanal (Eds.), *Pattern Recognition and Artificial Intellegence.* Elsevier. pp. 211–228.

[32] Jain, A.K., and W.G.Waller (1978). On the optimal number of features in the classification of multivariate Gaussian data. *Pattern Recognition,* **10**, 365–374.

[33] Juskevicius, K (1983). Investigation of the sensitivity of a minimum distance piecewise linear classifier to the limitation of learning sample size. In Š. Raudys (Ed.), *Statisticheskie problemy upravlenija,* Vol.61. Inst. Math. Cybern. Lith. Acad. Sci., Vilnius. pp. 89–129 (in Russian).

[34] Kanal, L., and B.Chandrasekaran (1971). On dimensionality and sample size in statistical pattern recognition. *Pattern Recognition,* **3**, 238–255.

[35] Khotanzad, A., and J.H.Lu (1988). Distortion invariant character recognition by a multilayer perceptron and back-propagation learning. In *Proc. 1988 IEEE Intl Conf. on Neural Networks.* San Diego, California. pp. I625–I632.

[36] Koford, J.S., and G.F.Groner (1966). The use of an adaptive threshold element to design a linear optimal pattern classifier. *IEEE Trans. Inf. Theory,* **IT-12**, 42–50.

[37] Kohonen, T., G.Barna and R.Chrisley (1988). Statistical pattern recognition with neural networks: Benchmarking studies. In *Proc. 1988 IEEE Intl Conf. on Neural Networks.* San Diego, California. pp. I61–I68.

[38] Kung, S.Y., and J.N.Hwang (1988). An algebraic projection analysis for optimal hidden units size and learning rates in back propagation learning. In *Proc. 1988 IEEE Intl Conf. on Neural Networks.* San Diego, California. pp. I363–I370.

[39] Lbov, G.S. (1965). On representativeness of the sample size while choosing the effective measurement system. In N.G.Zagoraiko (Ed.), *Computing Systems,* Issue 22, Inst. of Math. Press, Novosibirsk. pp. 39–58.

[40] Lee, W.T., and M.F.Tenorio (1991). On optimal adaptive classifier design criterion. In *Proceedings of IJCNN,* **8**(7). Seattle, WA, IEEE Publ.

[41] Lippmann, R.P. (1988). An introduction to computing with neural nets. *IEEE ASSP Mag.,* 4(2), 4–22.

[42] Lippmann, R.P. (1989). Pattern classification using neural networks. *IEEE Communications Magazine,* 47–64.

[43] Minsky, M., and S.Papert (1969). *Perceptrons: An Introduction to Computational Geometry.* MA:MIT Press, Cambridge.

[44] Nisikawa, Y., H.Kita and A.Kawamura (1990). A neural network which divides and learns environments. In *Proceedings of IEEE IJCNN'90..* Intl Neural Network Society and IEEE, Lawrence Erlbaum Associates, Inc. pp. I684–I687.

[45] Pikelis, V. (1974). *Analysis of learning speed of three linear classifiers.* PhD thesis, Inst. of Math. and Cyb., Acad. of Sci. of Lithuania, Vilnius (in Russian).

[46] Pikelis, V. (1976). Comparison of methods of computing the expected classification errors. *Automation and Remote Control*, 5, 59–63 (in Russian).

[47] Pivoriunas, V., and Š.Raudys (1978). On the accuracy of "leaving-one-out estimate". In Š.Raudys (Ed.), *Statisticheskie problemy upravlenija*, Vol.27. Inst. Math. Cybern. Lith. Acad. Sci., Vilnius. pp. 53–70 (in Russian).

[48] Ramani, N., P.H.Patric, W.G.Hanson and H.Anderson (1990). Fish detection and classification using a neural-network-based active sonal system-preliminary results. In *Proceedings of IEEE IJCNN'90*, Vol.2. Intl Neural Network Society and IEEE, Lawrence Erlbaum Associates, Inc. pp. 527–530.

[49] Raudys, Š. (1970). On the problems of sample size in pattern recognition. In *Proc. 2nd All Union Conf. on Statistical Methods in Control Theory*. Nauka, Moscow. pp. 64–67 (in Russian).

[50] Raudys, Š. (1977). *Statistical classification in the case of essentially limited sample size*. USSR PhD thesis, Inst. of Math. and Cyb., Lithuanian Acad. of Sci., Vilnius (in Russian).

[51] Raudys, Š. (1978). Comparison of the estimates of the probability of misclassification.In *Proc. 4th Int Joint Conf. Pattern Recognition*. Kyoto. pp. 280–282.

[52] Raudys, Š. (1979). Determination of optimal dimensionality in statistical pattern recognition. *Pattern Recognition*, 11, 263–270.

[53] Raudys, Š. (1981). Influence of sample size on the accuracy of model selection in pattern recognition. In Š.Raudys (Ed.), *Statisticheskie problemy upravlenija*, Vol.50. Inst. Math. Cybern. Lith. Acad. Sci., Vilnius. pp. 9–30 (in Russian).

[54] Raudys, Š., and A.K.Jain (1991). Small sample size effects in statistical pattern recognition: Recommendations for practitioners. *IEEE Trans. on PAMI*, 3, 252–264

[55] Raudys, Š., and V.Pikelis (1980). On dimensionality, sample size, classification error and complexity of classification algorithm in pattern recognition. *IEEE Trans. on PAMI.*, PAMI-2(3), 242–252.

[56] Raudys, Š., and V.Pikelis (1982). Collective selection of the best version of a pattern recognition system. *Pattern Recognition Letters*, 1, 7–13.

[57] Raudys, Š., V.Pikelis and D.Stasaitis (1986). An influence of the number of initial and final features, and correlation between them on the accuracy of features selection. In Š.Raudys (Ed.), *Statisticheskie problemy upravlenija*, Vol.74. Inst. Math. Cybern. Lith. Acad. Sci., Vilnius. (in Russian).

[58] Raudys, Š., and M.Skurykhina (1991). A role of number of training samples on weight inifialisation of artificial neural net classifier. In *Proc. of International Conference on Neural networks*. Rostov na Donu, Russia.

[59] Rumelhart, D.E., G.E.Hinton and R.J.Williams (1986). Learning internal

representations by error propagation. In D.E.Rumelhart and J.L.KcClelland (Eds.), *Parallel Distributed Processing.* MA:MIT Press, Cambridge.

[60] Sakaguchi, H. (1990).Stochastic dynamic and learning rules in layered neural networks. *Progres in Theoretical Physics,* **83**, 693–700.

[61] Smith, F.W. (1972). Small-sample optimality of design techniques for linear classifiers of Gaussian patterns. *IEEE Trans. Inf. Theory,* **IT-18**(1), 118–126.

[62] Toussaint, G.T. (1974). Bibliography on estimation of misclassification. *IEEE Trans. on Information Theory,* **IT-20**(4), 472–479.

[63] Tsypkin, Y.Z. (1966). Adaptation, learning and self-learning in automatic systems. *Automatic and Remote Control,* **1**, 23–61.

[64] Vapnik, V.N. (1979). *Estimation of Dependences Based on Empirical Data.* Nauka, Moscow. (Springer NY, 1982).

[65] Waibel, A. (1989). Consonant recognition by modular construction of large phonemic time-delay neural networks. In Tourecky (Ed.), *Neural Information Processing Systems,* Vol.1. Denver, Morgan Kaufman. pp. 215–223.

[66] Widrow, B., and R.C.Winter (1988). Neural nets for adaptive filtering and adaptive pattern recognition. *IEEE Trans. Comput.,* 25–39.

[67] Wieland, A., and R.Leighton (1987). Geometric analysis of neural network capabilities. In *IEEE 1st Int'l Conf. on Neural Networks,* Vol.3. pp. 385.

[68] Wyman, F., D.Young and D.Turner (1990). A comparison of asymptotic error rate expansions for the sample linear discriminant function. *Pattern Recognition,* **23**, 775–783.

Received May 1992

A. Jain received a B.Tech. degree in 1969 from the Indian Institute of Technology, Kanpur, and the M.S. and Ph.D. degrees in electrical engineering from the Ohio State University, in 1970 and 1973, respectively. He joined the faculty of Michigan State University in 1974, where he currently holds the rank of Professor in the Department of Computer Science. He received the best paper award in 1987 and certificates for outstanding contributions in 1976 and 1979 from the Pattern Recognition Society. Dr. Jain is the Editor-in-Chief of the *IEEE Transactions on Pattern Analysis and Machine Intelligence* and is on the Editorial Boards of

Pattern Recognition, Pattern Recognition Letters, Journal of Intelligent Systems, and *Journal of Mathematical Imaging and Vision.* He is the co-author of *Algorithms for Clustering Data,* Prentice-Hall, 1988, has edited the book *Real-time Object Measurement and Classification,* Springer, 1988, and co-edited the book, *Analysis and Interpretation of Range Images,* Springer, 1990. Dr. Jain is a fellow of the IEEE. He has been a member of Programm commite of number of international scientific conferences. Dr. Jain received the Distinguished Faculty Award from Michigan State University in 1989 and served as the Distinguished Visitor of the IEEE Computer Society (1988–90).

Š. Raudys received the M.S. degree in electrical and computer engineering from Kaunas Polytechnical Institute in 1963, and the Candidate of Sciences and Doctor of Sciences degree from the Institute of Mathematics and Cybernetics, Academy of Sciences, Lithuania, in 1969 and 1978, respectively.

He is currently Head of the Department of Data Analysis in the Institute of Mathematics and Cybernetics and Professor in the Department of Control Systems at Kaunas Polytechnical Institute, Lithuania. His current research interests include statistical pattern recognition, artificial neural nets, expert systems, machine learning, and data analysis methods.

Dr.Raudys is a member of the Classification Societies of the USSR and France. He is an Associate Editor of the Pattern Recognition Journal. He has been a member of the Program Committee of INTERFACE-90 and other Soviet and international conferences.