1-1-1991

On Optimal Adaptive Classifier Design Criterion- How many hidden units are necessary for an optimal neural network classifier?

Wei Tsih Lee
Purdue University

Manoel Fernando Tenorio
Purdue University

On Optimal Adaptive Classifier Design Criterion-

How many hidden units are necessary for an optimal neural network classifier?

Wei-Tsih Lee
Manoel Fernando Tenorio

TR-EE-91-5
January 1991

On Optimal Adaptive Classifier Design Criterion-

How many hidden units are necessary for an optimal neural network classifier ?

Wei-Tsih Lee
Parallel Distributed Structures Lab.
School of Electrical Engineering
Purdue University
W. Lafayette, IN. 47907
lwt@ed.ecn.purdue.edu

Manoel Fernando Tenorio
Parallel Distributed Structures Lab.
School of Electrical Engineering
Purdue University
W. Lafayette, IN. 47907
tenorio@ee.ecn.purdue.edu

Abstract

A central problem in classifier design is the estimation of classification error. The difficulty in classifier design arises in situations where the sample distribution is unknown and the number of training samples available is limited. In this paper, We present a new approach for solving this problem. In our model, there are two types of classification error: approximation and generalization error. The former is due to the imperfect knowledge of the underlying sample distribution, while the latter is mainly the result of inaccuracies in parameter estimation, which is a consequence of the small number of training samples. We therefore propose a criterion for optimal classifier selection, called the Generalized Minimum *Empirical Criterion (GMEE). The GMEE criterion consists of two terms, corresponding to the estimates of two types of error. The first term is the empirical error, which is the classification error observed for the training samples. The second is an estimate of the generalization error, which is related to the classifier complexity. In this paper we consider the Vapnik-Chervonenkis dimension (VCdim) as a measure of classifier complexity. Hence, the classifier which minimizes the criterion is the one with minimal error probability. Bayes consistency of the GMEE criterion has been proven.*

As an application, the criterion is used to design the optimal neural network classifier. A corollary to the Bayes optimality of neural network-based classifiers has been proven. Thus, our approach provides a theoretic foundation for the connectionist approach to optimal classifier design. Experimental results are given to validate the approach, followed by discussions and suggestions for future research.

I. Introduction

The pattern classifier is an integral component of any perceptual system. The patterns are problem dependent; pixels in image segmentation [1, 2], and acoustic features in speech recognition [3] are but two examples. In order to make these systems capable of dealing with real world problems, several fundamental issues in classifier design needed to be addressed: the underlying distribution of features is, in general, unknown, and the number of available training samples is finite [4, 5]. To meet these challenges, a new criterion for classifier design is required.

Recently, Lippmann [6] pointed out the importance of matching the complexity of a classifier to the training data. A properly matched classifier has the following advantages: good generalization ability, thus, preventing over-fitting of training data; computational efficiency; and improved memory utilization in the training and recognition stages. The idea of matching classifier complexity to sample size stems from the sample-based approach to classifier design [7, 8]. From a collection of classifiers Γ, the approach taken in [7] was to maximize the success rate criterion to choose the best classifier; the criterion selects a classifier which maximizes the number of correct classifications among the training samples. However, since no proper measure of classifier complexity has ever been developed, adjusting classifier complexity (or equivalently, adapting the size of Γ) to the sample size can only be done heuristically [7]. Vapnik and Chervonenkis [8, 9] proposed the growth function as a measure of the separating ability of the decision rules (or classifiers). The growth function of a decision rule S being equal to n means no more than n samples can be partitioned in an arbitrary way by S. Basically, the concept is combinatorial in nature. Vapnik and Chervonenkis showed that the finite valued growth function of a decision rule is a sufficient condition for the uniform convergence of the empirical events to their probability, which can be interpreted as the convergence of

empirical error of a classifier to its error probability in the context of pattern recognition [9], and used the minimum classification error criterion to choose the best classifier [9]. No mention of adapting classifier complexity was made. However, in the finite sample case, Vapnik [9] proposed using the "Structured Minimum Empirical Error" to choose the best classifier, which was the first criterion developed to constrain the size of the space of classifiers, and can be considered a version of the "method of sieves" [10] in classifier design. It was not until Devroye [11] clearly suggested how classifier complexity should change with sample size such that asymptotic optimal performance of classifiers selected could be achieved. Furthermore, he investigated the complexity of various classifiers in terms of Vapnik and Chervonenkis dimension [12, 13, 14, 15].

The goal of this paper is to propose a criterion, the Generalized Minimum Empirical Criterion (GMEE), as a principle for data dependent classifier design. The criterion can be derived from the results of classification error analysis. In our analysis, there are two types of classification error: approximation error and generalization error. Depending on the size of Γ, the Bayes classifier may or may not be included in Γ. Hence, if we pick the classifier f^* with the smallest error probability from Γ, it can only be considered an approximation to the Bayes classifier. The approximation error is defined as the difference in error probability between f^* and the Bayes classifier. Generalization error results from inaccuracies in the estimates of classifier parameters due to the finite number of training samples.

The GMEE criterion consists of two parts, which correspond to the estimates of two types of classification error. By considering the two types of classification error as a whole, the classifier which minimizes the GMEE criterion is that with minimum classification error. Since the second term in the criterion relates the classifier complexity to sample size, it can serve as a term controlling the growth in complexity of classifiers

considered. Hence, the GMEE criterion is a criterion which adapts the size of Γ to sample size. The idea is similar to the "Structured Minimum Empirical Error" criterion [9], and to the complexity regulization criterion in [16]. However, the GMEE criterion is more flexible and can be used in both the finite and infinite sample cases. The organization of the paper is as follows: Section II discusses our model for classifier design, analysis of classification error, and the derivation of the GMEE criterion. The consistency property of the GMEE criterion has been proven in section III. In section IV, we briefly review neural networks and provide results for analyzing the approximation and generalization capability of networks as classifiers. In section V, the GMEE criterion is, then, applied to the optimal design of neural network-based classifiers. Two examples are given to demonstrate the performance of the criterion. A discussion and conclusions follow in section VI.

II. Formulation of Pattern Recognition Problems

We consider the following pattern recognition problem: Given n pairs of training samples $\{(x_1,y_1), (x_2,y_2),..., (x_n,y_n)\}$, find a classifier f which best maps feature vectors $x_i \in X$ to classes $y_i \in Y$. That is,

$$f : X \rightarrow Y \tag{1}$$

where feature space X has distribution D, and $Y= \{1,2,..,M\}$ are the allowed classes for vectors in X. The empirical error ν (or error frequency) is the ratio of classification errors to the total number of samples, i.e.,

$$\nu = \frac{\sum_{i=1}^{n} 1_{(y_i \neq f(x_i))}}{n} \tag{2}$$

where $1_{(y_i \neq f(x_i))}$ is the indicator function, which is one when the classifier decision is different from the true class and is zero otherwise. The error probability is

$$P_f(\text{error}) \equiv P(y \neq f(x)) = \int 1_{(y \neq f(x))} dP(x) \tag{3}$$

For simplicity, we have used P(x) to denote the class mixture distribution. The Bayes error is the minimum achievable error probability, i.e.,

$$P_{Bayes}(\text{error}) \equiv \inf_f P_f(\text{error}) \tag{4}$$

and the Bayes classifier is that which attains this bound. As mentioned above, to make full use of information, we need to adapt the size of Γ to the number of samples. To derive such a criterion, we start with an analysis of classification error.

II.1. Classification Error Analysis

There exist various approaches to estimating the expected performance of a classifier; notably, the holdout [17], resubstitution [18], and leave-one-out methods [19]. An excellent review of work in this area is given in Toussaint [20]. As pointed out in [21], the major difficulties in evaluating classifier performance are that the true sample distribution is unknown and that the number of available samples is finite.

These two problems cause two different types of classification error, which we have labelled the approximation error and the generalization error.

II.1.1. Approximation Error

The absence of the information concerning the underlying sample distribution makes it impossible to know the Bayes classifier f_{Bayes}, or even whether f_{Bayes} is in Γ. If $f_{Bayes} \notin \Gamma$, then f^*, the classifier in Γ with minimum error, is at best an approximation to f_{Bayes}, giving rise to an approximation error, hence the name. More formally, the approximation error is defined as:

$$P^{\Gamma}_{\text{Approximation}}(\text{error}) \equiv P(f^*(x) \neq f_{Bayes}(x)) = \int 1_{(f^*(x) \neq f_{Bayes}(x))} dP(x) \quad (5)$$

Hence, the error probability of f^* can be written as

$$P_{f^*}(\text{error}) = P_{f_{Bayes}}(\text{error}) + P^{\Gamma}_{\text{Approximation}}(\text{error}) \quad (6)$$

Intuitively, it is not surprising that the approximation error decreases as the size of Γ increases. Moreover, if the Bayes classifier is included in Γ, then the approximation error is zero.

Example 1.1: Let samples be drawn from two classes, each represented by a 2 dimensional Gaussian distribution having a different covariance matrix. The Bayes classifier will be a quadratic polynomial. If Γ is the collection of linear classifiers, then the approximation error will be the value of the integration over the area, with respect to the mixture distribution of the two classes, where the output of the linear classifier is not coincident with the Bayes classifier. However, if we extend Γ to quadratic polynomials, the approximation error is zero.

II.1.2. Generalization Error

In practice, there are always a finite number of samples available for training and testing. The limited amount of available data causes inaccuracies in estimates of classifier parameters. Random variation over the finite training set also degrades the performance of the resulting classifier. It has been observed that discrepancies exist between the empirical error and the classification error for testing data [14, 21, 22]. Since there are only a finite number of testing samples, the generalization error defined in [14] will be a random variable for both training and testing samples. To avoid such complications here, we define the generalization error to be the discrepancy between the empirical error and the classifier error probability.

Since the empirical error is a random quantity dependent on the finite collection of training samples, the generalization error is thus also a random variable dependent on the number of samples and on the classifier complexity [22]. In general, complex classifiers require more training samples to ensure reliable parameter estimation. One rule of thumb is to keep the ratio of classifier parameters to the number of samples constant [14, 23].

II.2. Estimates of the Approximation Error and Generalization Error

II.2.1. Classifier Complexity

In [7], the maximum sample success rate (or minimum empirical error rate) criterion was used to select the best-count classifier f from Γ. The sample success rate for arbitrary $f \in \Gamma$ converges to its success probability uniformly as number of samples approaches infinity, provided that Γ consists of the collection of linear, m-linear, or m-convex classifiers. (Note: In [7], an m-linear classifier is defined as a classifier with partition regions constructed by m half-spaces; more generally, an m-convex classifier has partition regions constructed from m measurable convex sets.)

Vapnik and Chervonenkis [8] established the same convergence result over extended collections of classifiers, including those with finite growth function. The theorem is given in the appendix. For completeness, the definition of the growth function is given below:

Let $X_r = x_1, x_2, \ldots, x_r$ be a set of r samples from a distribution D in R^n. Each two-class classifier $f \in \Gamma$ can be considered as a mapping, assigning x to class one if $f(x) \geq 0$, or to class two if $f(x) < 0$. Let A_f be the subspace induced by f, i.e., $A_f = \{ x \in R^n \mid f(x) \geq 0 \}$. Each f divides X_r into two subsequences: one consisting of samples in A_f, the other of samples not in A_f. Note also that each classifier $f \in \Gamma$ induces a subspace A_f in R^n.

Let S be the collection of sets A_f. The index of the system S with respect to the samples $x_1, x_2,..., x_r$ is the number of different subsequences partitioned by $A_f \in S$, and will be denoted $\Delta^S(x_1, x_2,..., x_r)$. The growth function $m^S(r)$ is obtained by taking the maximum index among all possible samples of length r, i.e., $m^S(r) = \max_r \Delta^S(x_1, x_2,..., x_r)$. In [12], the growth function of set is used to characterize the metric entropy property of the space of functions under the name of Vapnik-Chervonenkis dimension (VCdim). A set S has VCdim = k if $m^S(r) = 2^k$. We will follow this line of thinking, and consider the VCdim as characterizing the richness of Γ, and hence, the complexity of a classifier.

II.2.2. Estimate of Approximation Error

By Theorem 1 in the appendix, the empirical error can serve as an estimate of the error probability of a classifier. We will show how the empirical error can be used as an estimate of the approximation error. We first consider the convergence property of the minimum empirical error criterion, a result similar to that in [7].

Lemma 1: If Γ is a collection of classifiers with finite VCdim, $f^* \in \Gamma$ is the minimum empirical error classifier, and $P_{f^*}(\text{error})$ is the error probability of f^*, then the empirical error $\nu_{f^*}(\theta)$ converges to the minimum error probability $\inf_{f \in \Gamma} P_f(\text{error})$. Moreover, $P_{f^*}(\text{error})$ converges to $\inf_{f \in \Gamma} P_f(\text{error})$ uniformly.

Proof: From Theorem 1 in the appendix, for every $f \in \Gamma$,

$$\nu_f(\theta) \text{ convergences to } P_f(\text{error}) \text{ uniformly.} \qquad (7)$$

Define a function $g = -f$.

$$\left| \nu_{f^*}(\theta) - \inf_{f \in \Gamma} P_f(\text{error}) \right| = \left| \inf_{f \in \Gamma} \nu_f(\theta) - \inf_{f \in \Gamma} P_f(\text{error}) \right|$$

$$= \left| - \sup_{g \in \Gamma} \nu_g(\theta) + \sup_{g \in \Gamma} P_g(\text{error}) \right|$$

$$\leq \sup_{g \in \Gamma} \left| \nu_g(\theta) - P_g(\text{error}) \right| \tag{8}$$

Since $\nu_g(\theta)$ converges to $P_g(\text{error})$ uniformly by (7), the right hand side of (8) converges to zero. This implies

$$\nu_{f^*}(\theta) \text{ convergences to } \inf_{f \in \Gamma} P_f(\text{error}) \text{ uniformly.} \tag{9}$$

Moreover,

$$\left| P_{f^*}(\text{error}) - \inf_{f \in \Gamma} P_f(\text{error}) \right| \leq \left| P_{f^*}(\text{error}) - \nu_{f^*}(\theta) \right| + \left| \nu_{f^*}(\theta) - \inf_{f \in \Gamma} P_f(\text{error}) \right| \tag{10}$$

The first term on the right hand side converges to zero uniformly by (7), and the second term converges to zero uniformly by (9). Hence,

$$P_{f^*}(\text{error}) \text{ convergences to } \inf_{f \in \Gamma} P_f(\text{error}) \text{ uniformly.} \tag{11}$$

Q.E.D.

Remark:

From Lemma 1, the minimum empirical error is an estimate of the error probability of f^*. If the Bayes classifier in included in Γ, then $\inf_{f \in \Gamma} P_f(\text{error})$ is equal to $P_{Bayes}(\text{error})$. Hence, the error probability of the minimum empirical classifier converges to the Bayes error.

Recall that the error probability of f^* can be decomposed into two parts as in (6). The first part, the Bayes error, is a constant once the distribution of the samples is specified. The second part, the approximation error, is a variable dependent upon the richness of Γ. Hence, it is reasonable to consider the empirical error as an estimate of the approximation error.

II.2.3. Estimate of the Generalization Error

Next, we consider estimating the generalization error using the "Structured Minimum Empirical Error" approach suggested in [9]. By expanding the expression in (A.1), Vapnik obtains the following, which holds with probability $1-\eta$,

$$\nu(\theta) - 2\sqrt{\frac{\beta(ln\frac{2n}{\beta} + 1) - ln\frac{\eta}{9}}{n}} < P(\theta) < \nu(\theta) + 2\sqrt{\frac{\beta(ln\frac{2n}{\beta} + 1) - ln\frac{\eta}{9}}{n}}, \qquad (12)$$

After eliminating the constant terms, the following holds with high probability for properly chosen λ, and sufficiently large sample sizes:

$$\nu(\theta) - \lambda\sqrt{\frac{\beta}{n}ln\frac{n}{\beta}} < P(\theta) < \nu(\theta) + \lambda\sqrt{\frac{\beta}{n}ln\frac{n}{\beta}} \qquad (13)$$

By our definition of the generalization error, the second term $\lambda\sqrt{\frac{\beta}{n}ln\frac{n}{\beta}}$ can serve as an estimate of the generalization error.

II.3. Generalized Minimum Empirical Error Criterion

From (13), the error probability of a classifier is bounded from above by the sum of the estimated approximation and generalization errors. Thus, to minimize the error probability, both types of error must be minimized simultaneously. Hence, we define the Generalization Minimum Empirical Error (GMEE) as follows:

$$\{\ \nu(\theta) + \lambda\sqrt{\frac{\beta}{n} ln\frac{n}{\beta}}\ \} \tag{14}$$

where $\nu(\theta)$ is the empirical error, which is the estimate of the approximation error, θ a vector of classifier parameters, β is the VCdim of the classifier, n is number of the samples, and *ln* refers to the natural logarithm. λ is a positive number, and can be interpreted as a priori knowledge about how close the estimate $\lambda\sqrt{\frac{\beta}{n} ln\frac{n}{\beta}}$ is to the true generalization error.

As mentioned above, the approximation capability of classifier *f* is determined by the size of Γ. In general, the larger the size of Γ, the smaller the approximation error will be. Hence, more complex classifiers reduce the empirical error [24, 25], but increase the generalization error, due to the finite number of training samples. The behavior is consistent with the GMEE criterion, for which generalization error also increases with classifier complexity. Since the error probability of a classifier is bounded from above by the sum of two types of classification error, the classifier which minimizes the GMEE criterion has minimum error probability.

III. Bayes Consistency of the GMEE Criterion

As in estimation theory, the asymptotic optimality of the estimator is of particular interest. In classifier design, we are concerned with the Bayes consistency of the criterion.

Definition [*Bayes Consistency*]: a classifier selection criterion is said to be ***Bayes consistent*** if $P_{f_n}(\text{error}) \to P_{Bayes}(\text{error})$, provided that there exists a sequence of classifiers $f_n \in \Gamma_n$ such that error probability of f_n approaches the Bayes error probability as n goes to infinity. The sequence of classifiers is said to be Bayes optimal.

Lemma 2. if $\frac{\text{VCdim(n)}}{\text{n}} \to 0$, then $\frac{\text{VCdim(n)}}{\text{n}} \ln \frac{\text{n}}{\text{VCdim(n)}} \to 0$

Proof: Let $g(n) = \frac{\text{VCdim(n)}}{\text{n}}$, then

$$\frac{\text{VCdim(n)}}{\text{n}} \ln \frac{\text{n}}{\text{VCdim(n)}}$$

$$= \frac{\ln \frac{1}{g(n)}}{\frac{1}{g(n)}} \qquad (15)$$

By applying l'Hôpital's rule, (15) can be shown to converge to zero if $\frac{\text{VCdim(n)}}{\text{n}}$ converges to 0. Q.E.D.

Theorem 1: The GMEE criterion is Bayes Consistent if $\lim_{n \to \infty} \frac{\text{VCdim(n)}}{\text{n}} = 0$.

Proof: As n approaches infinity, let f^* be the classifier selected by the GMEE criterion, VCdim(f^*) be the VCdim of f^*, and VCdim(n) be the VCdim of f.

For a given bounded constant λ and for every $f \neq f^*$,

$$\nu_{f^*}(\theta) + \lambda\sqrt{\frac{\mathrm{VCdim}(f^*)}{n} ln \frac{n}{\mathrm{VCdim}(f^*)}} \leq \nu_f(\theta) + \lambda\sqrt{\frac{\mathrm{VCdim}(n)}{n} ln \frac{n}{\mathrm{VCdim}(n)}} \quad (16)$$

Since $\sqrt{\frac{\mathrm{VCdim}(n)}{n} ln \frac{n}{\mathrm{VCdim}(n)}}$ -> 0 as $\frac{\mathrm{VCdim}(n)}{n}$ -> 0 (by Lemma 2), it follows that

$$\nu_{f^*}(\theta) \leq \nu_f(\theta). \quad (17)$$

Hence, f^* is the minimum empirical error classifier. Thus, $\nu_{f^*}(\theta)$ converges to $\inf_{f \in \Gamma} P_f(\mathrm{error})$ uniformly by (9). Since the Bayes classifier is included in Γ by assumption, $\nu_{f^*}(\theta)$ converges to $P_{Bayes}(\mathrm{error})$ uniformly. Moreover, $P_{f^*}(\mathrm{error})$ also converges to $P_{Bayes}(\mathrm{error})$ uniformly by (11).

Q.E.D.

IV. Optimal Neural Network Classifier Design

The neural network, in particular, the Multi-Layer Perceptron (MLP), has emerged recently as a solution for difficult perceptual tasks. Successful applications include classification of sonar signals [25] and speech recognition [26]. The MLP has been proven robust [24], and capable of forming the arbitrary complex decision boundaries necessary for pattern recognition [27].

As mentioned above, Lippmann [6] pointed out the advantages of matching classifier complexity to training data. In this section, the GMEE criterion is applied to obtain the optimal neural network classifier, where optimality is in the sense of minimum error probability.

IV.1. Model of Neural Network Classifiers

A neural network consists of an interconnected group of neurons, each a computational unit with several output and input links. Weights associated with each link control the strength of the interconnection between neurons. At each neuron, the sum of the weighted inputs is passed through a nonlinear function, usually a linear threshold or sigmoidal function, giving rise to the output. The mapping relation of the sigmoidal function is defined as follows:

$$\text{sigmoid}\,(x) \equiv \frac{1}{1+e^{-(x\cdot w)}} \tag{18}$$

where x is the input to the neuron, w is the weights associated with the input links, and $(x\cdot w)$ refers the inner product of x and w.

A multi-layer neural network can be characterized by five parameters: the dimension of the input vector (#d), the number of layers (#l), the number of hidden neurons in each hidden layer (#h), the number of classes to be classified (#c), and the decision function computed by each neuron. Figure 1 depicts a homogeneous network, which consists of three layers of neurons: the input, hidden, and output layers, with one output neuron, d input neurons, and h hidden neurons.

To design a neural network classifier, the samples are fed one at a time into the network. The mean square error between the desired outputs and the actual outputs of the

neural network is minimized by adjusting the weights in the negative gradient direction (19).

$$\Delta w_{ij}(t) = -\varepsilon \frac{\partial E}{\partial w_{ij}(t)} + \alpha \Delta w_{ij}(t-1) \tag{19}$$

where w_{ij} is the weight between neuron i and j, ε is the learning rate, and α is the momentum constant. When the last sample is reached, the process is repeated, thus beginning a new epochs. The training is stopped when the terminal condition is satisfied.

To be able to analyze the optimality of neural network classifiers, we need to consider the approximation and generalization capabilities of the network.

IV.2. Analysis of Approximation Capability

The approximation capability of neural networks has been studied from many different points of view [28, 29], and [30, 28] and [29] regard the networks as a basis in function space and justify their use as universal approximating functions, while the latter (Cybenko [30]) discusses the approximating capability of neural networks with sigmoidal nodes in the context of pattern recognition.

Theorem 2 (Cybenko) [30]: Let σ be a continuous sigmoidal function. Let f be the decision function for any finite measurable partition of I_n. For any $\varepsilon > 0$, there is a finite sum of the form

$$G(x) = \sum_{j=1}^{N} \alpha_i \sigma(y_j^T x + \theta_j) \tag{20}$$

and a set $D \subset I_n$, so that $m(D) \geq 1-\varepsilon$ and

$$| G(x) - f(x) | < \varepsilon \quad for\ x \in D \tag{21}$$

Proof: see [30]

Remark:

By Theorem 2, a network with sigmoidal output functions requires only a finite number of hidden layer nodes to approximate any decision function to arbitrary accuracy. The implication is that the complexity of neural network classifiers is also finite.

IV.3. Analysis of Generalization Capability

To evaluate the generalization capability of a neural network, we need a method to compute the VCdim. We have derived two such methods [15], listed below.

Theorem 3: The VCdim of a homogeneous neural network with d inputs, h hidden layer neurons and sigmoidal output functions is

$$0 \leq \text{VCdim} < \sum_{i=1}^{h+1} d = (h+1) * d \tag{22}$$

Proof: see [15].

Theorem 4: For an arbitrary network with sigmoidal output functions and connection graph G, an upper bound on VCdim is given by

$$0 \leq \text{VCdim}(F) \leq 2\log(eN) \sum_{i=1}^{N} \text{VCdim}(F_i) \tag{23}$$

where F is the function computed by the network representation of G, N is the number of nodes in G, and F_i is a function of node i in G.

Proof: see [15].

Remark:

According to the proof given in [15], Theorems 2 and 3 hold for any monotonic decision function, including the sigmoidal function.

IV.4. Bayes Optimality of Neural Network Classifiers

Since networks can approximate any decision function to arbitrary precision, they have the potential to approximate the Bayes classifier if the size of the network is large enough. On the other hand, large networks increase the VCdim, thus increasing the generalization error. A good classifier design criterion must arbitrate the trade-off between

increasing the approximating capability and deteriorating the generalization error. The GMEE criterion can be proven optimal by the following corollary.

Corollary 1: Neural network-Based classifiers with sigmoidal nodes designed according to the GMEE criterion are Bayes optimal.

proof: Let Γ be the collection of neural network classifiers with sigmoidal nodes. The Bayes classifier is included in Γ by Theorem 2. Since the GMEE criterion is Bayes consistent by Theorem 1, the sequence of neural network classifiers selected by the GMEE criterion is Bayes optimal.

Q.E.D.

V. Experiments

Although the form of the GMEE criterion (14) is known, one quantity, the weighting factor λ applied to the generalization error, is data dependent and must be determined empirically. A simple heuristic is to make λ proportional to the "randomness" of the samples, as more randomly distributed samples tend to increase the generalization error.

Prior to evaluating a particular classifier by the GMEE criterion, we must know its VCdim β and its parameters θ. In the case of neural network classifiers, the VCdim β is directly related to the number of hidden layer nodes, and can be evaluated by (22) or (23), depending on the connectivity of the network.

The parameters θ of the classifier which minimizes the GMEE criterion can be found using a two step procedure. Starting with a small number, h, of hidden layer nodes, find the optimal parameter values using the Back-Propogation training rule (19) [31], then

evaluate the GMEE criterion. Increase h, and repeat the process until a local minimum of the GMEE criterion is achieved, thus determining the optimal classifier.

We now illustrate the behavior of the minimization procedure using two examples. In the first example, samples are drawn from two classes, each represented by a two dimensional Gaussian distribution, and each having a different covariance matrix. In the second example, one class is represented by the mixture of two Gaussians, while the other is represented by a pure Gaussian, as before. In each case, 40 runs of the GMEE criterion optimization are performed, on sample sizes of 50, 150, and 450. Statistics collected include the mean and variance of the empirical error and error probabilities, based on a 10,000 point sample, as well as the success rate of the GMEE criterion.

Example 5.1 - determination of the optimal neural network classifier for two Gaussian distributions.

The covariance matrix for each class is given below.

$$\Sigma_1 = \begin{bmatrix} 1.0 & 0.0 \\ 0.0 & 3.0 \end{bmatrix} \qquad \Sigma_2 = \begin{bmatrix} 1.0 & 0.0 \\ 0.0 & 1.0 \end{bmatrix} \tag{24}$$

The mean of class 1 is -0.9, while that of class 2 is 0.9. Figure 2 shows the samples from the classes and the Bayesian decision boundary, which is quadratic; the Bayes error probability is 0.1711. The minimization of the GMEE criterion is performed over four types of classifier with varying complexity (Figure 3), indexed by the integers 1 through 4, where the index refers to the number of hidden units; thus the complexity increases with the index.

The GMEE criterion is evaluated below for the 450 sample case. Figure 4 shows a plot of the empirical error obtained using the Back-Propagation training rule with parameters listed in Table 1. The estimated generalization error shown in Figure 5 has been computed using a value of .1 for λ and the upper bound on VCdim (22) for the complexity. The resulting GMEE criterion (Figure 6) exhibits a minimum at two, indicating that the optimal classifier has two hidden units (hidden layer nodes). The observed error probability, based on 10,000 samples, is shown in Figure 7. It is consistent with the GMEE criterion in that it also displays a minimum at two.

To evaluate the statistical behavior of the criterion, forty runs of the two step optimization procedure are performed for each sample size. At the start of each run, the network parameters are set to randomly chosen initial values in the range [-1,1]. The average and standard deviation of the empirical error and probability are shown in Figure 8 and 9, respectively, as well as in Table 2. Figure 8 shows that the average empirical error drops as the complexity of the classifier increases, while Figure 9 shows that the standard deviation of the error probability decreases with increasing numbers of training samples, implying that the resulting classifiers are more stable as the number of training samples increases. Table 3 shows that the success rate of the GMEE criterion increases as the number of training samples increases.

Example 5.2 - determination of the optimal neural network classifier for a Gaussian mixture distribution.

The two classes are described by the following density function:

$$p_1(x,y) = N(x,0,\sigma^2)N(y,m_1,1.4\sigma^2) \tag{25}$$

$$p_2(x,y) = \frac{N(x,m_2,\sigma^2)+N(x,-m_2,\sigma^2)}{2}N(y,\ 0,\ 1.5\sigma^2) \tag{26}$$

where m_1 is equal to 1.636σ, m_2 is equal to 3.4σ, and σ is equal to 0.2, and is chosen such that the Bayes error probability is 0.052.

Experiments were performed as per Example 5.1, but now using a value of .15 for λ to account for the increased randomness of the class 2 distribution. The parameters for the Back-Propagation training rule are listed in Table 4. The results for various sample sizes are summarized in Figures 10 through 13. As before, statistics were collected over 40 runs. The effects of sample size and classifier complexity on the average and standard deviation of the empirical error and error probability, shown in Figures 14 and 15, and summarized in Table 5, are as observed previously. That is, the empirical error decreases with increasing classifier complexity, and the standard deviation decreases with increasing numbers of samples.

The success rate of applying the GMEE criterion, shown in Table 6, is lower than for Example 5.1. This can be explained by the increased difficulty of estimating the generalization error with more complex sample distributions.

VI. Discussion and Conclusion

We have developed a Bayes consistent classifier design criterion, the GMEE criterion, from an analysis of classification error. The criterion has been applied to the design of neural network classifiers. The result of two examples indicate that the GMEE criterion can yield optimal neural network classifiers. We have also proven that a neural network classifier is Bayes optimal if it is selected by the GMEE criterion. Hence, our results provide a theoretical foundation for the connectionist approach to classification problems. These results can also be extended to the optimal design of other types of neural network, e.g., radial basis function networks [32]. In our research, the choice of the coefficient λ in the GMEE criterion is done empirically, but it should be readily determined from the data using the cross-validation technique [33].

Appendix :

Theorem 1 (Vapnik) [8,9] : Let Γ_β be the class of decision rules with VCdim= β, $X=\{ x_1, x_2,..., x_n \}$ the set of training samples drawn independently from the distribution D, and $\nu_\beta(\alpha)$ the frequency of incorrectly classified samples. Suppose $n \geq \beta$, then with probability $1-\eta$, the following expression holds for each element in Γ_β.

$$\Pr\{ \sup_{\alpha} \left| P_\beta(\alpha) - \nu_\beta(\alpha) \right| > \gamma \} \leq 9\frac{(2n)^\beta}{\beta!} \exp\{ -\frac{\gamma^2 n}{4}\} \qquad (A.1)$$

REFERENCES

[1] J. Y. Hsiao and A.A. Sawchuk, "Supervised texture image segmentation using texture smoothing and probabilistic relaxation techniques," IEEE Trans. Pattern Analysis and Machine Intel., vol 11, no.12, pp. 1279-1293, Dec. 1989.

[2] R.M. Haralick and L.G. Shapiro, "Survey : image segmentation techniques," Computer Vision, Graphics, and Image Processing 29, pp. 100-132, 1985.

[3] L.R. Rabiner and B. H. Juang, "An introduction to hidden Markov models," IEEE ASSP. Magazine, pp. 4-16, Jan. 1986.

[4] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, Academic Press, New York, 1978.

[5] R. O. Duda and P.E. Hart, Pattern Classification and Scene Analysis, New York : John Wiley and Sons, 1973.

[6] R. P. Lippmann, "pattern classification using neural networks," IEEE Comm. Magazine, pp. 47-64, Nov. 1989.

[7] N. Glick, "Sample-based classification procedure related to empirical distributions," IEEE Trans. Inform. Theory, vol. IT-22, no. 4, pp. 454-461, July 1976.

[8] V. N. Vapnik and A. Ya. Chervonenkis, "On the uniform convergence of relative frequencies of events to their probabilities," Theory of Prob. and its Appl, V17, N2,pp. 264-280, 1971.

[9] V. N. Vapnik, *Estimation of Dependency Based on Empirical Data*, Springer-Verlag, NY, 1982.

[10] U. Grenander, *Abstract Inference*, New York : Wiely, 1981.

[11] L. Devroye, "Automatic pattern recognition : a study of the probability of error," IEEE Trans. of Pattern Analysis and Machine Intel., Vol. 10, no. 4, pp.530-543, July, 1988.

[12] R. M. Dudley,"Central limit theorems for empirical measures," Ann. Prob. 6960, pp.899-929, 1978.

[13] A. Blumer, A. Ehrenfeucht, D. Haussler, and M. Warmuth, Learning and the Vapnik-Chervonenkis dimension, UC Santz Crutz Tech. Rep. UCSC-CRL-87-20, 1987.

[14] E. B. Baum and D. Haussler, "What Size Net Gives Valid Generalization ?" *Advances in Neural Info. Processing Syst.* 1,D.S. Touretzky, ed., San Mateo, CA : Morgan Kaufmann, 1988.

[15] W.-T. Lee and M. F. Tenorio, "Computation of the Vapnik-Chervonenkis Dimension of Neural Network with Sigmoidal Nodes -Implications for Pattern Recognition Problems," submitted for publication, 1990.

[16] A.R. Barrow, "Statistical properties of artificial neural networks," preprint, 1989.

[17] W.H. Highleyman, "The design and analysis of pattern recognition experiments," Bell System Technical Journal, 41, pp.723-744, March 1962.

[18] D. H. Foley, "Considerations of Sample and Feature Size," IEEE Trans. Inform. Theory, vol. IT-18, vol. 5, pp. 618-626, Sept. 1972.

[19] P.A. Lachenbruch and R.M. Mickey, "Estimation of error rates in discriminant analysis," Technometrics, vol. 10, no. 1,pp.1-11, 1968.

[20] G.T. Toussaint, "Bibliography on estimation of miscalssification," IEEE Trans. Inform. Theory, vol. IT-20, no. 4, pp. 472-429, July 1974.

[21] K. Fukunaga and R.P. Hayes, "Estimation of classifier performance," IEEE Trans. of Pattern Analysis and Machine Intel., vol. 11, no. 10, pp.1087-1101, Oct. 1989.

[22] S. Randys and V. Pikelis, "On dimensionality, sample size, classification error, and complexity of classification algorithm in pattern recognition," IEEE Trans. of Pattern Analysis and Machine Intel., vol. 2, no. 3, pp.242-252, May 1980.

[23] B. Widrow, Plenary Speech, Vol. I: Proc. 1st Int. Conf. on Neural Networks, San Diego, CA, pp. 143-158.

[24] W. Y. Huang and R. P. Lippmann, "Comparisons Between Neural Network and Conventional Classifiers," IEEE conf. on Neural Networks, San Diego, pp. IV.485-494, 1988.

[25] R. D. Gorman and T. J. Sejnowski, "Learned classification of sonar targets using a massively parallel network," IEEE Trans. on Acoustics, Speech, and Signal Processing, vol. 36, no. 7,pp. ,July 1989.

[26] R. L.Watrous and L. Shastri, "Learning phonetic features using connectionist networks: an experiment in speech recognition," IEEE conf. on Neural Networks, San Diego, vol. IV, pp. 381-388, 1988

[27] R. P. Lippmann,"An Introduction to Computing with Neural Nets," IEEE ASSP Magazine, pp. 4-22, April, 1987.

[28] K. Funahashi, "On the approximate of continuous mapping by neural networks," Journal of Int. Neural Net., vol. 2, pp.183-192, 1989.

[29] K. Hornik, M. Stiuchcombe, and H, White, "Multi-layer feedforward networks are universal approximators," Preprint, June 1988.

[30] G. Cybenko, "Approximations by superpositions of a sigmoidal function," CSRD Rpt. No. 856, Univ. of Illion, Urbana, Feb. 1989.

[31] Rumelhart, McClelland, and PDP research Group, *Parallel Distributation Processing*, Vol. 1, MIT press, 1987

[32] S. Renals and R. Rohwer, "Phoneme classification experiments using radial basis functions," Proc. Int. Joint Conf. on Neural Networks, vol. 1, pp. 461-467, IEEE, Washington DC, June 1989.

[33] M. Stone, "Cross-validation choice and assessment of statistical predications," J. of the Royal Statistical Soc., vol. B-36, pp. 111-147, 1974.

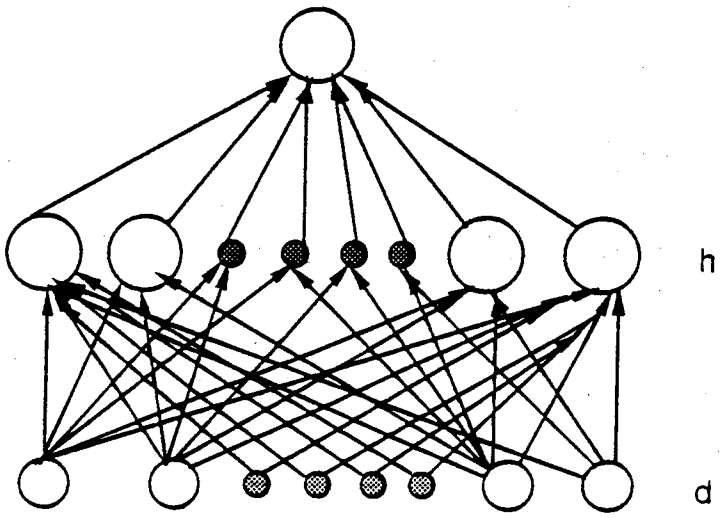

Figure 1. A homogeneous neural network with h hidden and d input nodes.

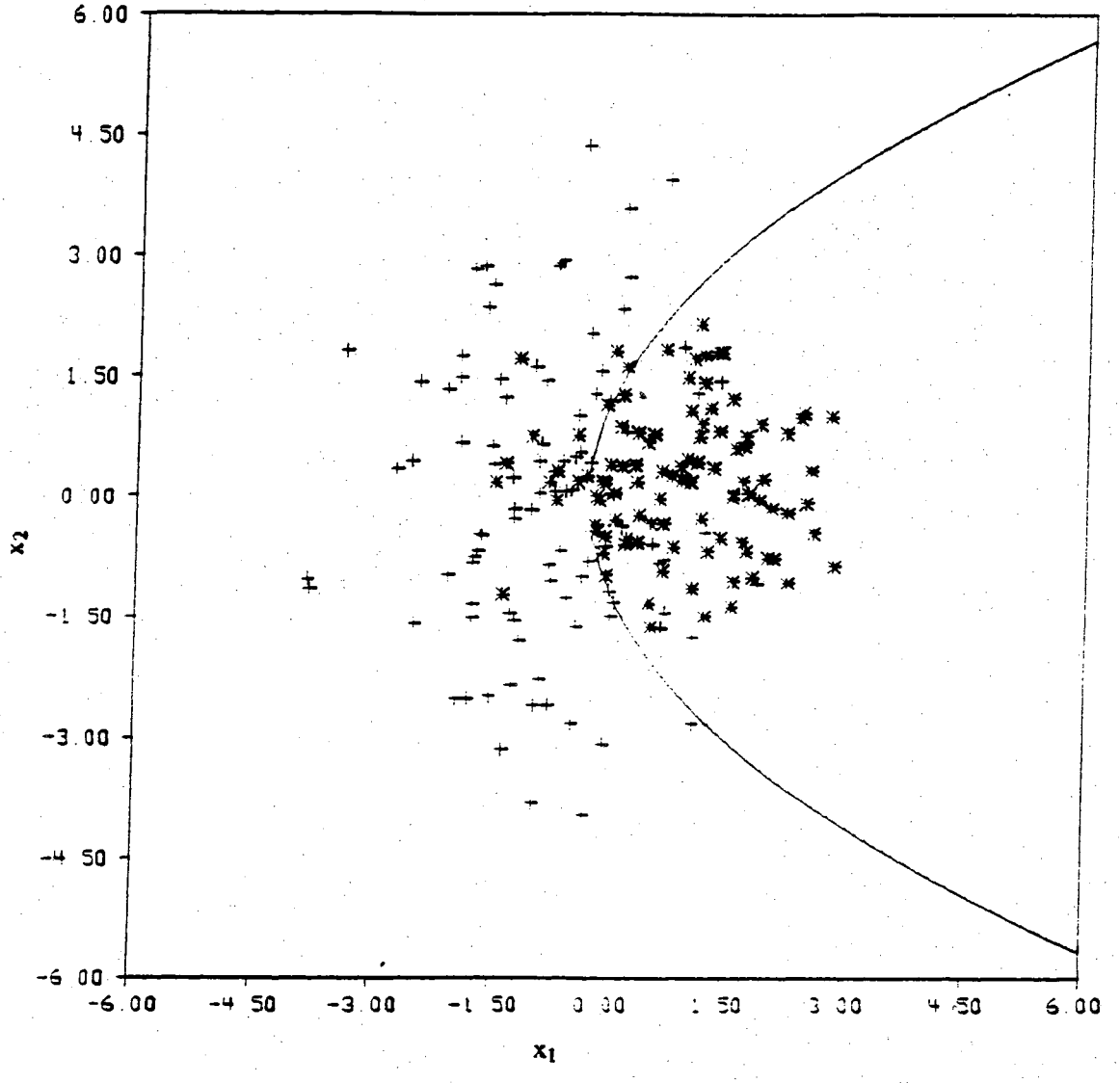

Figure 2. The Bayesian decision boundary, $x_2 = \sqrt{5.41 x_1 + 1.648}$, for two Gaussian distributions with different covariance matrices. Class 1 samples are denoted by "+", class 2 by "*".

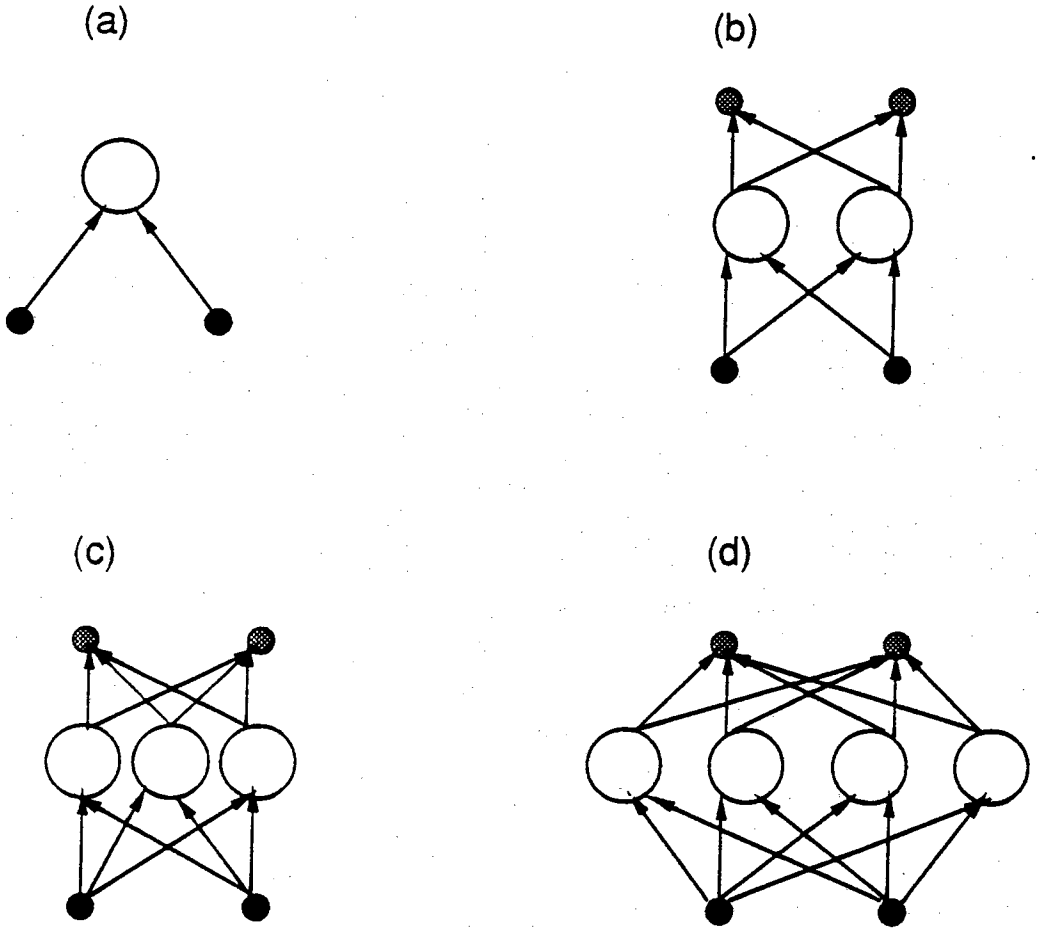

Figure 3. Four types of neural network classifiers. (a) One hidden node.
(b) Two hidden nodes. (c) Three hidden nodes.
(d) Four hidden nodes.

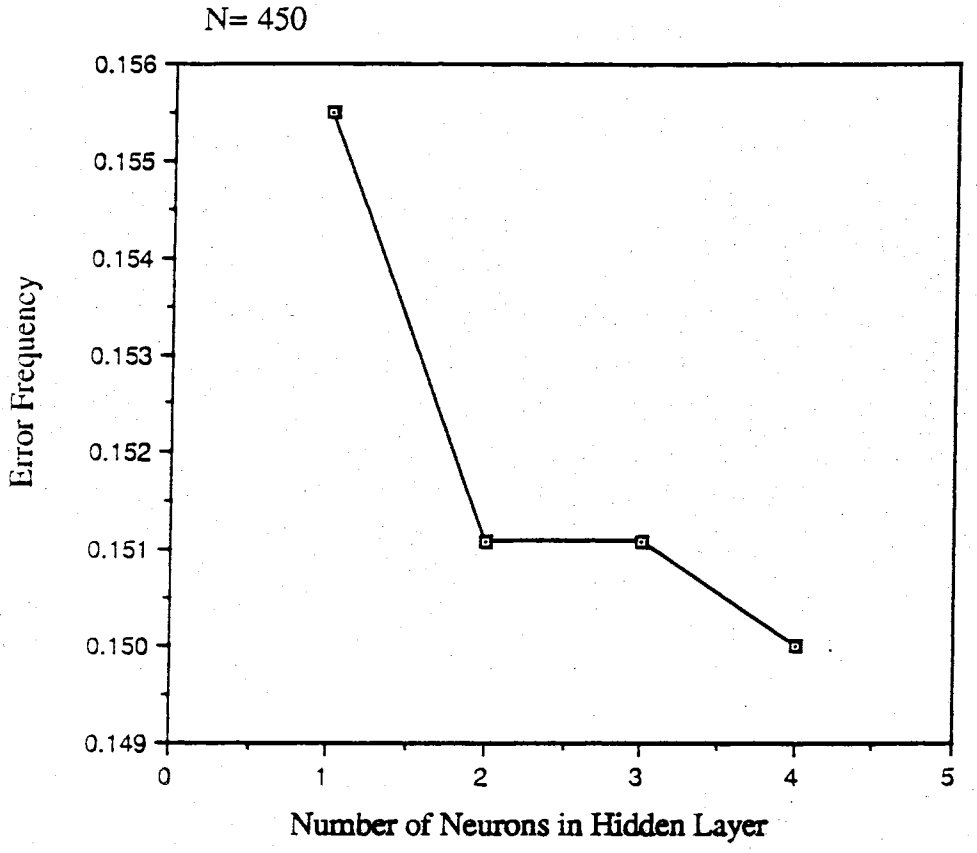

Figure 4. The plot of the error frequency for four types of neural network classifier considered in Example 5.1.

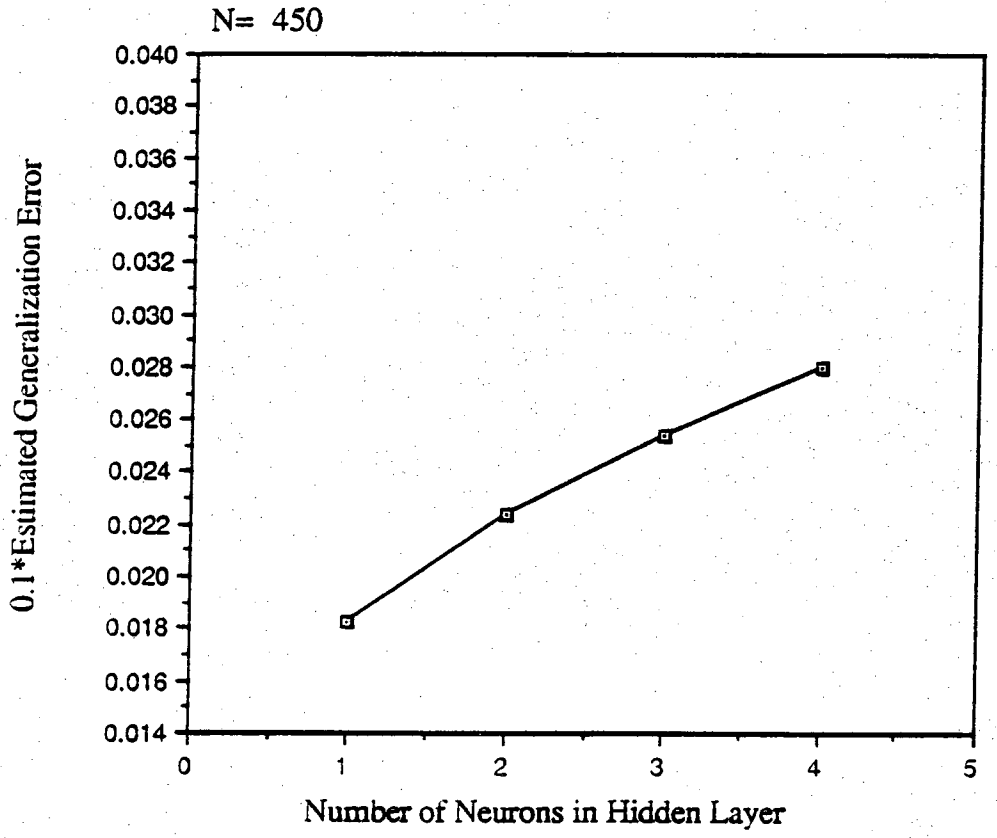

Figure 5. The plot of the estimated generalization error for four types of neural network classifier considered in Example 5.1.

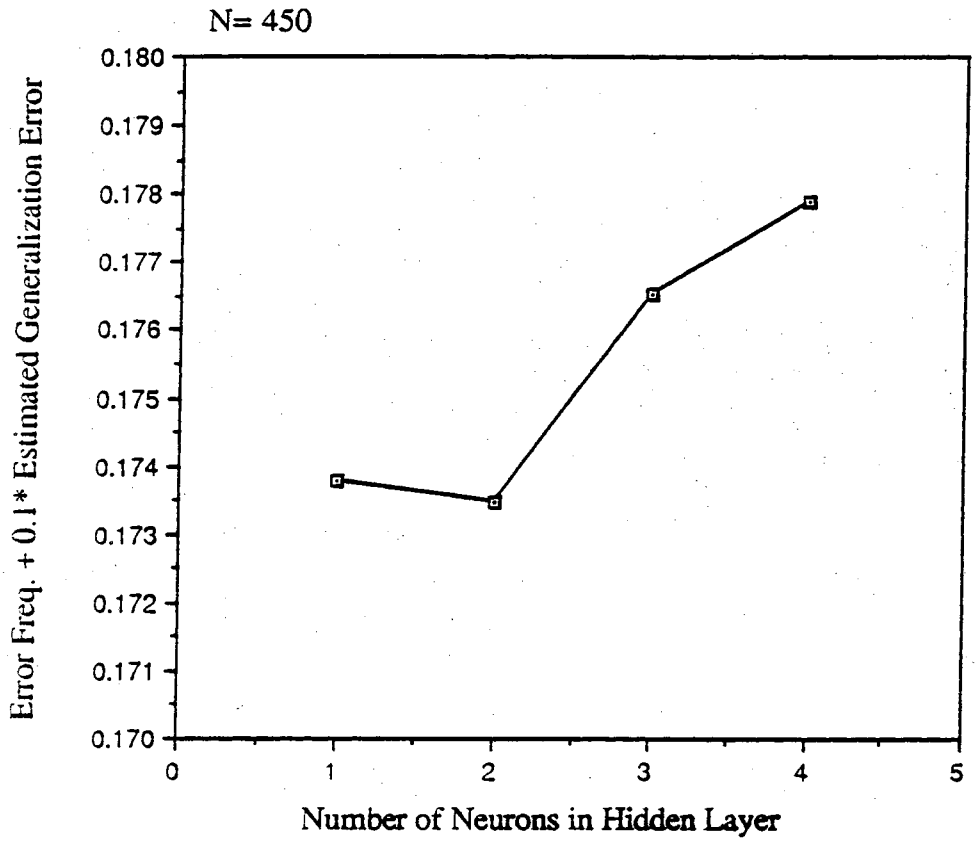

Figure 6. The plot of the Generalized Minimum Empirical Error (GMEE) criterion for four types of neural network classifier considered in Example 5.1.

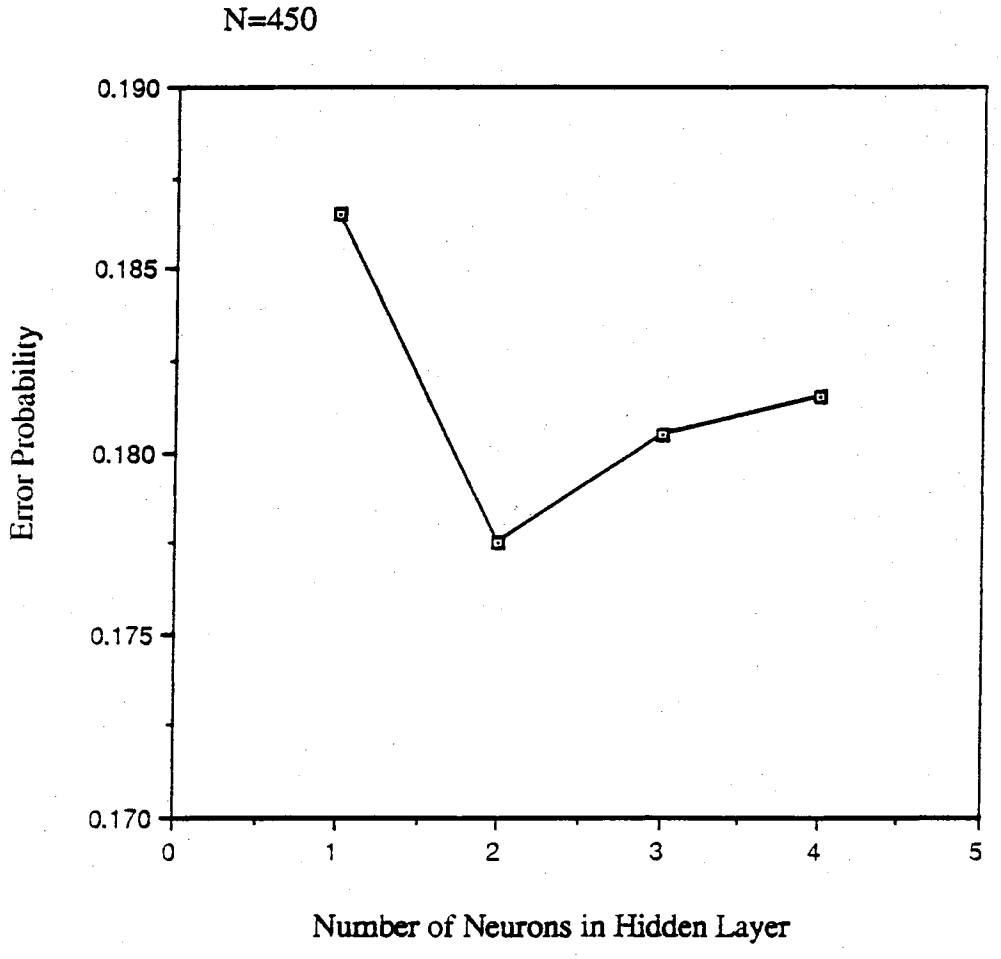

Figure 7. The plot of the error probability for four types of neural network classifier considered in Example 5.1.

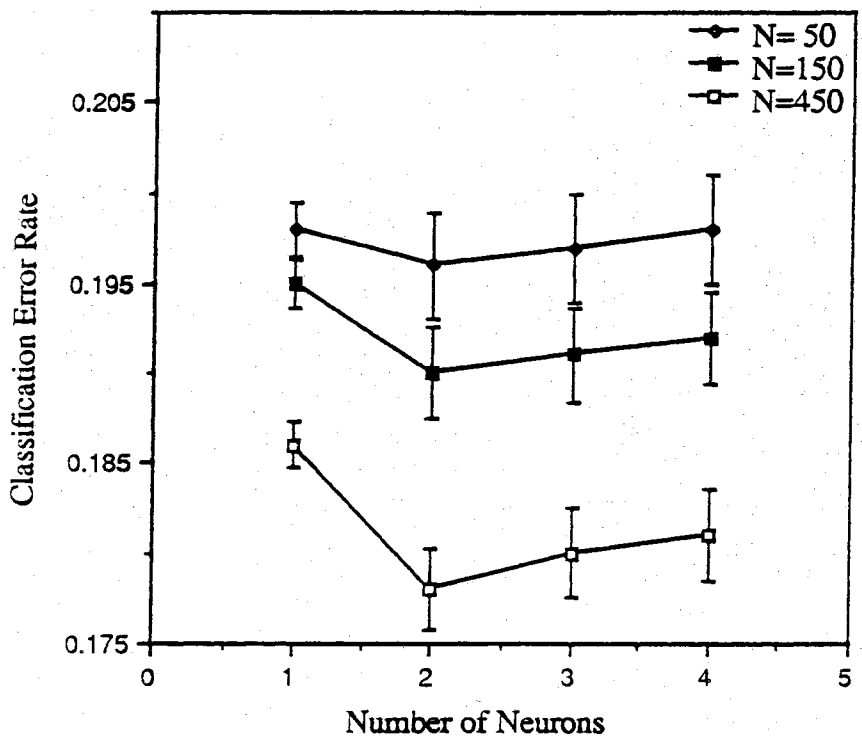

Figure 9. The plot of the mean and variance of the error probability for four types of neural network classifier considered in Example 5.1.

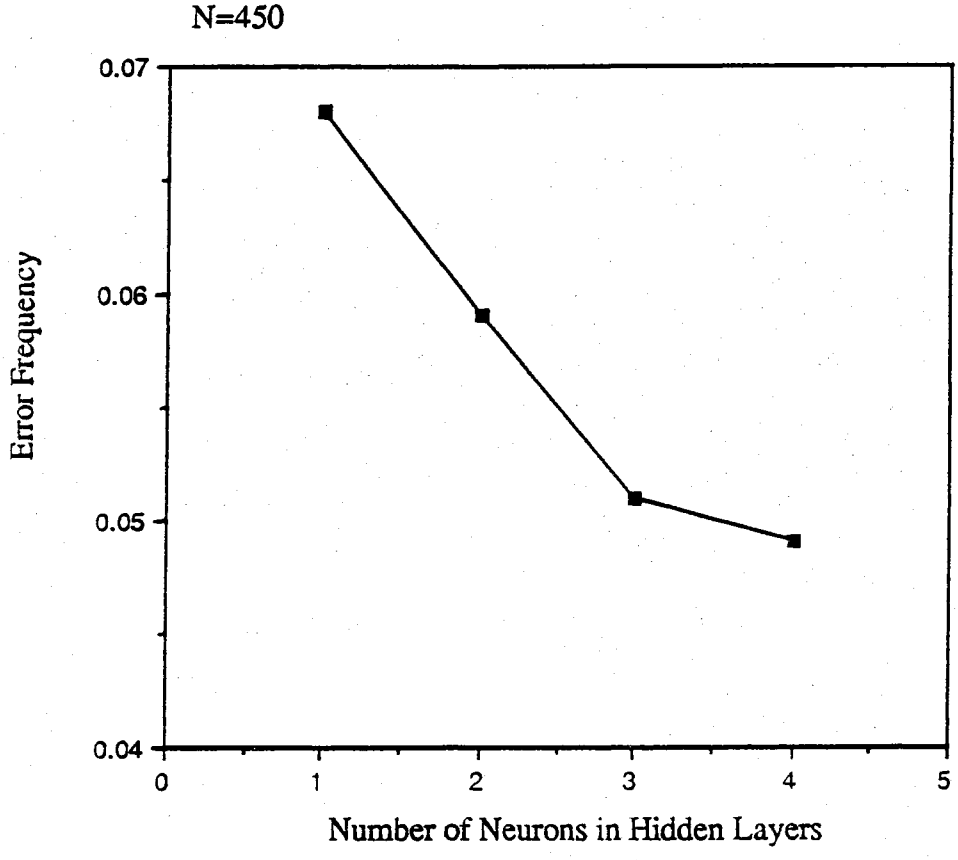

Figure 10. The plot of the error frequency for four types of neural network classifier considered in Example 5.2.

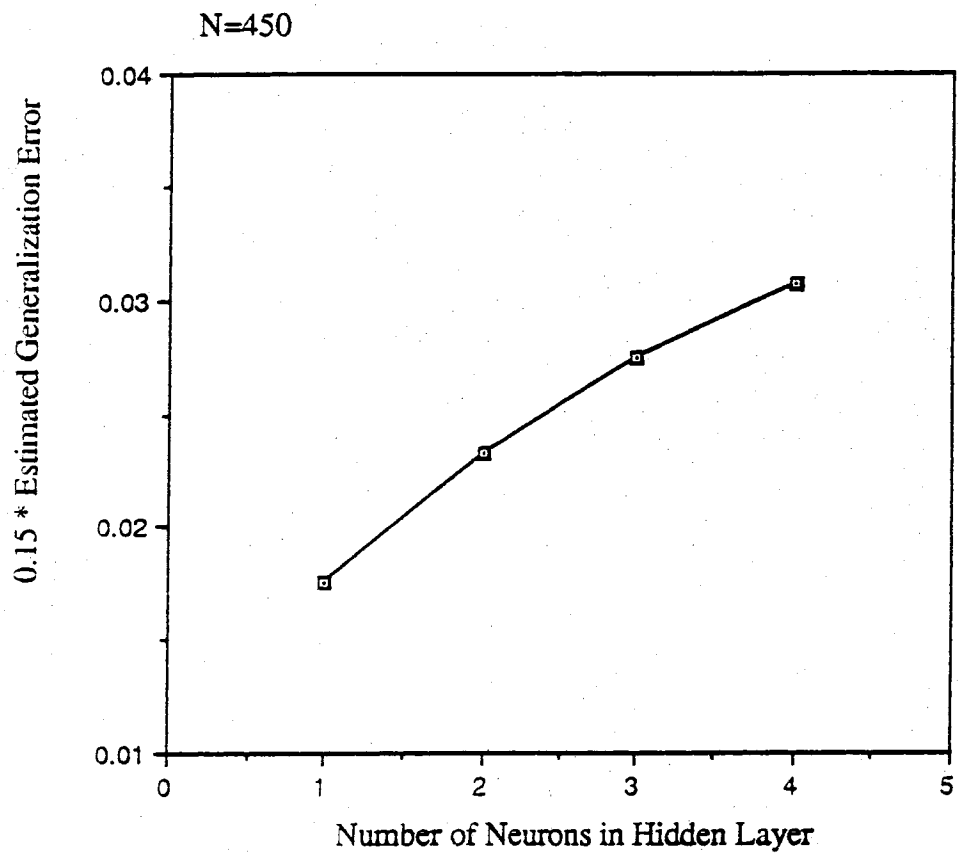

Figure 11. The plot of the estimated generalization error for four types of neural network classifier considered in Example 5.2.

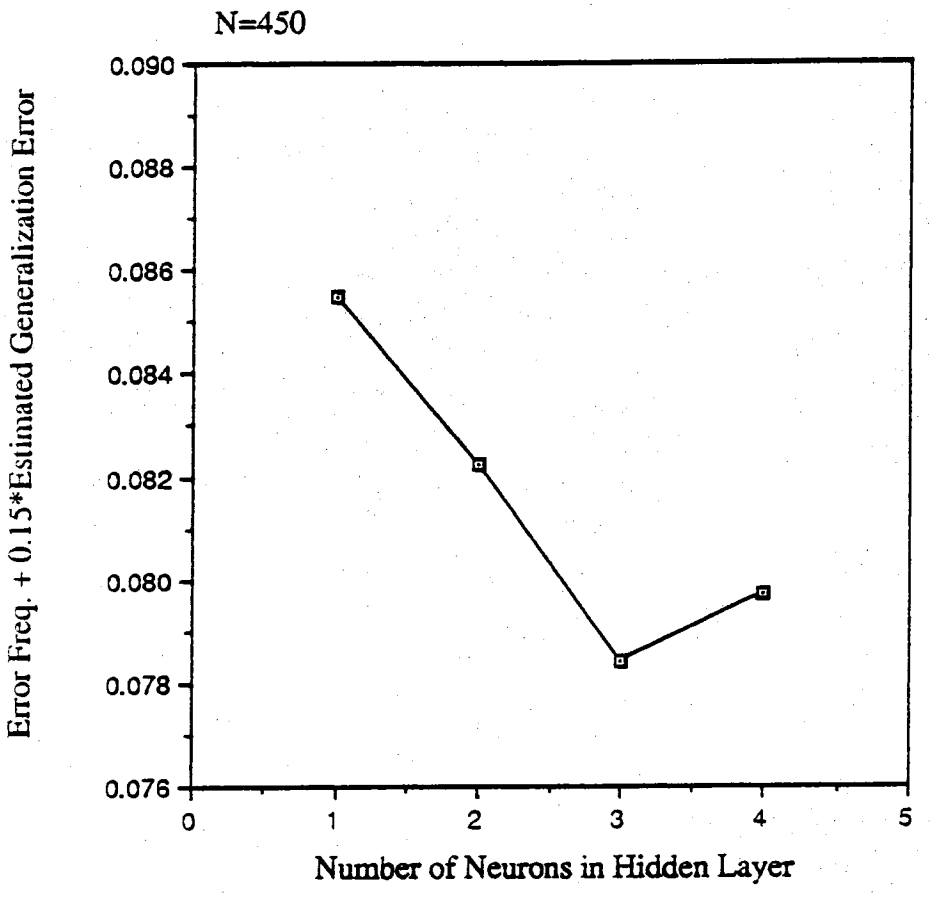

Figure 12. The plot of the Generalized Minimum Empirical Error (GMEE) criterion for four types of neural network classifier considered in Example 5.2.

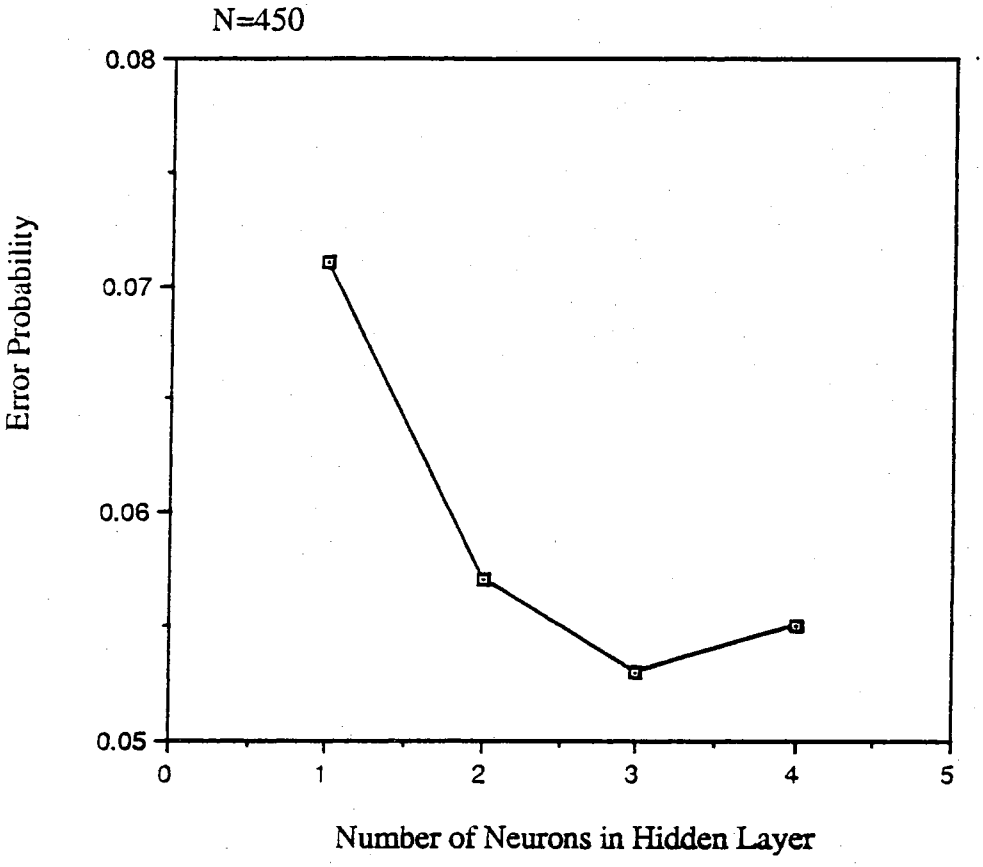

Figure 13. The plot of the error probability for four types of neural network classifier considered in Example 5.2.

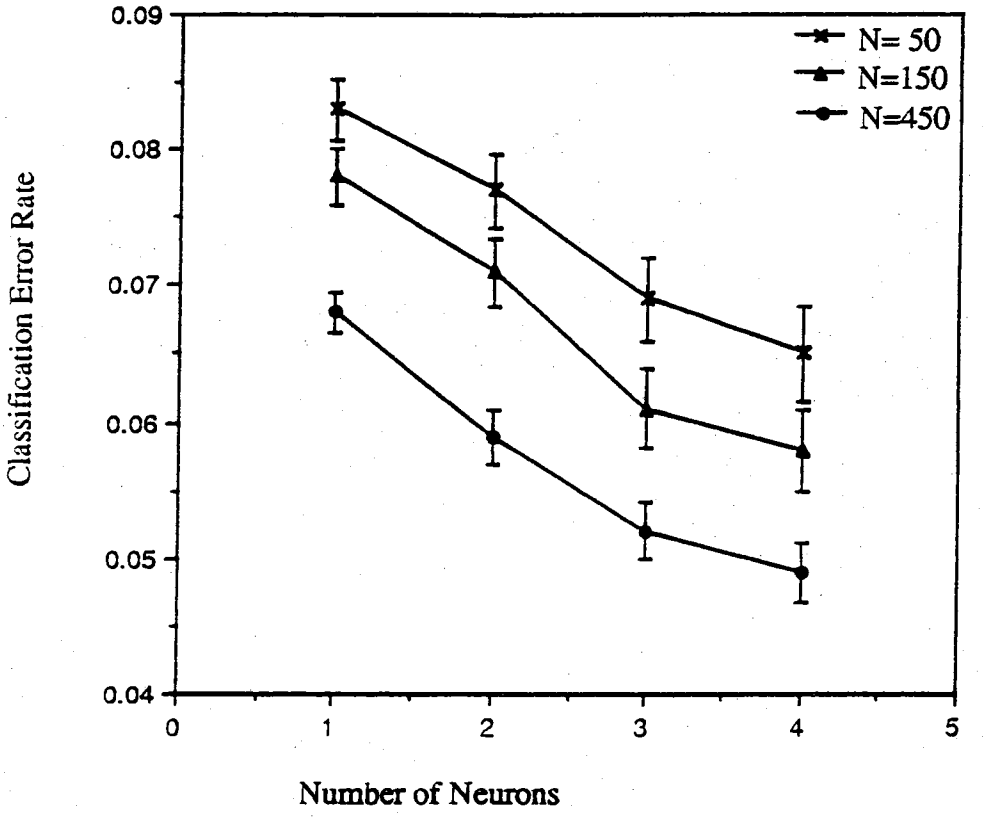

Figure 14. The plot of the mean and variance of the empirical error of four types of neural network classifier considered in Example 5.2.

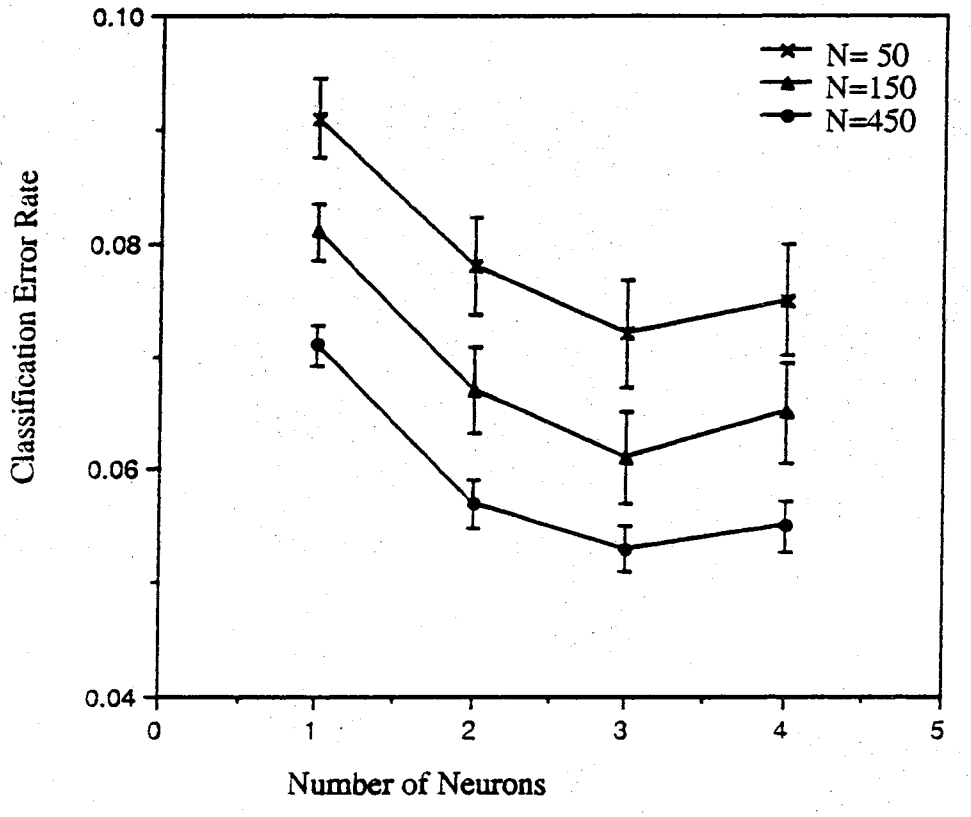

Figure 15. The plot of the mean and variance of the error probability of four types of neural network classifier considered in Example 5.2.

(a)

No. of Neurons in Hidden Layer	1	2	3	4
No. of Iterations	600	600	600	600
Learning Rate	0.001	0.001	0.001	0.001
Momentum Constant	0.5	0.5	0.5	0.5

(b)

No. of Neurons in Hidden Layer	1	2	3	4
No. of Iterations	500	500	500	500
Learning Rate	0.0001	0.0001	0.0001	0.0001
Momentum Constant	0.4	0.4	0.4	0.4

(c)

No. of Neurons in Hidden Layer	1	2	3	4
No. of Iterations	500	500	500	500
Learning Rate	0.0001	0.0001	0.0001	0.0001
Momentum Constant	0.3	0.3	0.3	0.3

Table 1. The Back-Propogation training rule parameters for each sample size in Example 5.1. (a) 50 samples. (b) 150 samples. (c) 450 samples.

(a)

No. of Neurons in Hidden Layer		1	2	3	4
N	50	0.222±0.0050	0.212±0.0052	0.211±0.0053	0.210±0.0053
(Number of samples)	150	0.193±0.0033	0.180±0.0033	0.180±0.0034	0.178±0.0034
	450	0.156±0.0021	0.151±0.0023	0.151±0.0022	0.150±0.0023

(b)

No. of Neurons in Hidden Layer		1	2	3	4
N	50	0.198±0.0015	0.196±0.0029	0.197±0.0030	0.198±0.0030
(Number of samples)	150	0.195±0.0013	0.190±0.0025	0.191±0.0026	0.192±0.0027
	450	0.186±0.0013	0.178±0.0023	0.180±0.0025	0.181±0.0025

Table 2. The training and testing set error rates for four types of neural network classifier of Example 5.1. (a) Training set error rates. (b) Testing set error rates.

N	#success(#total)	Success Rate of GMEE(%)
50	27(40)	67.5
150	29(40)	72.5
450	33(40)	82.5

Table 3. The success rates of the Generalized Minimum Empirical Error (GMEE) criterion for Example 5.1.

(a)

No. of Neurons in Hidden Layer	1	2	3	4
No. of Iterations	600	600	600	600
Learning Rate	0.001	0.001	0.001	0.001
Momentum Constant	0.8	0.8	0.8	0.8

(b)

No. of Neurons in Hidden Layer	1	2	3	4
No. of Iterations	600	600	600	600
Learning Rate	0.0005	0.0005	0.0005	0.0005
Momentum Constant	0.75	0.75	0.75	0.75

(c)

No. of Neurons in Hidden Layer	1	2	3	4
No. of Iterations	500	500	500	500
Learning Rate	0.0005	0.0005	0.0005	0.0005
Momentum Constant	0.7	0.7	0.7	0.7

Table 4. The Back-Propogation training rule parameters for each sample size in Example 5.2. (a) 50 samples. (b) 150 samples. (c) 450 samples.

(a)

No. of Neurons in Hidden Layer		1	2	3	4
N	50	0.086±0.0032	0.077±0.0036	0.068±0.0041	0.065±0.0043
(Number of samples)	150	0.079±0.0021	0.071±0.0028	0.056±0.0035	0.053±0.0042
	450	0.068±0.0015	0.059±0.0027	0.052±0.0021	0.049±0.0025

(b)

No. of Neurons in Hidden Layer		1	2	3	4
N	50	0.091±0.0034	0.078±0.0042	0.072±0.0047	0.074±0.0049
(Number of samples)	150	0.081±0.0025	0.067±0.0039	0.061±0.0041	0.065±0.0045
	450	0.071±0.0018	0.057±0.0021	0.053±0.0021	0.055±0.0023

Table 5. The training and testing set error rates for four types of neural network classifier of Example 5.2. (a) Training set error rates. (b) Testing set error rates.

N	#success(#total)	Success Rate of GMEE(%)
50	25(40)	62.5
150	28(40)	70.0
450	32(40)	80.0

Table 6. The success rates of the Generalized Minimum Empirical Error (GMEE) criterion for Example 5.2.