

# Two-Group Classification Using Neural Networks\*

Eddy Patuwo, Michael Y. Hu, and Ming S. Hung  
*Graduate School of Management, Kent State University, Kent, OH 44242*

## ABSTRACT

Artificial neural networks are new methods for classification. We investigate two important issues in building neural network models; network architecture and size of training samples. Experiments were designed and carried out on two-group classification problems to find answers to these model building questions. The first experiment deals with selection of architecture and sample size for different classification problems. Results show that choice of architecture and choice of sample size depend on the objective: to maximize the classification rate of training samples, or to maximize the generalizability of neural networks. The second experiment compares neural network models with classical models such as linear discriminant analysis and quadratic discriminant analysis, and nonparametric methods such as  $k$ -nearest-neighbor and linear programming. Results show that neural networks are comparable to, if not better than, these other methods in terms of classification rates in the training samples but not in the test samples.

*Subject Areas: Classification, Neural Networks, Simulation, and Statistical Techniques.*

## INTRODUCTION

Classification has emerged as an important decision making tool. In business, it has been used for credit scoring [4], for prediction of various events including credit card usage [2], and tender offer outcomes [39]. It is used in any decision problem involving the assignment of an object to an appropriate group. Gordon [12] shows such problems exist in fields as diverse as taxonomy, psychology, nosology, archaeology, pattern recognition, linguistics, and market research.

In this paper we present an evaluation of artificial neural networks as tools for classification. The idea of neural computing grew out of a desire to capture the pattern recognition capabilities of a biological brain. McCulloch and Pitts [26] developed the first model of a physiological brain called "McCulloch-Pitts Neuron," which became the basis for almost all the artificial neural networks where nodes are likened to neurons and arcs to dendrites or axons. The neural networks were developed for recognition of ill-defined objects such as handwritten characters [18] [25], finger prints [21], speech [31], electrical or sonar signals [13] [23], and double spirals [17]. They have also been used for detection of faults in a chemical process [14], explosives in airline baggage [32], and prediction of bank failures [38]. Neural network models are nonparametric and are able to adjust the form of the discrimination

---

\*The authors would like to acknowledge the financial support provided by the Research Council of Kent State University.

function to fit the data. For more details on neural networks and their applications, see [5] [27] [40].

An interesting question is how to design a neural network for a classification task at hand. Key issues involve the choice of network architecture and samples. Network architecture refers to the number of nodes and arcs, and the connections between nodes. Another question is whether neural networks can work as well as the more traditional methods such as linear discriminant analysis (LDA) and quadratic discriminant analysis (QDA). Since neural networks are nonparametric, these questions can only be answered by experimental results. In this study, we design two experiments which take into account the various distributions of attributes, the sample sizes, and the different neural network models. To focus on these issues, we restrict the study to two-group classification problems.

In the next section, a fairly complete review of the theory of classification is given. The approach is based on the Bayesian classification rule which is optimal in that it minimizes the misclassification rate over the sample space. It is then shown that the Bayesian rule is exactly the same as the well-known LDA and QDA when attributes are normally distributed. A brief review of neural networks is provided and emphasis is given on how neural networks classify objects. Three research questions are then posed and followed by research design. The experiments are Monte Carlo simulations with clearly defined experimental factors and sufficient replications. Classification problems of different characteristics are used to evaluate the performance of neural networks of various structures in Experiment I. Experiment II entails the comparison of neural network classifiers with the traditional methods provided by SAS procedure DISCRIM [30] and a linear programming (LP) model.

## REVIEW OF CLASSIFICATION THEORY

### Bayesian Classification Theory

Bayesian decision theory is the basis for statistical pattern recognition. This theory provides a unified approach to the well known classification algorithms, including the linear and quadratic discriminant analyses. The following is based on Chapter 2 of Duda and Hart [7].

Let  $\omega$  be the state of nature with  $\omega = \omega_1$  for group 1 and  $\omega = \omega_2$  for group 2. Assume  $P(\omega_1)$  to be the a priori probability for an observation to belong to group 1 and  $P(\omega_2)$  to be the similar a priori probability for group 2. Since we are concerned with two-group classification, it is assumed that  $P(\omega_1) + P(\omega_2) = 1$ . Suppose an observation  $\mathbf{x}$  is given. We shall associate  $\mathbf{x}$  with an  $n$ -vector of attributes (variables). (In other words,  $\mathbf{x}$  will denote both an object and its variables.) Let  $p(\mathbf{x}|\omega_j)$  be the state-conditional probability density function for  $\mathbf{x}$ ; the probability density function for  $\mathbf{x}$  given that the state of nature is  $\omega_j$ . (We use  $P$  to denote a probability and  $p$  to denote a density function.) The a posteriori probability, using the Bayes rule, is:

$$P(\omega_j|\mathbf{x}) = \frac{p(\mathbf{x}|\omega_j)P(\omega_j)}{p(\mathbf{x})}, \quad (1)$$

where

$$p(\mathbf{x}) = \sum_{j=1}^2 p(\mathbf{x}|\omega_j)P(\omega_j).$$

When a random observation  $\mathbf{x}$  is given and a decision is made to declare the group membership of  $\mathbf{x}$ , a probability of error (misclassification) can be determined:

$$P(\text{error}|\mathbf{x}) = \begin{cases} P(\omega_1|\mathbf{x}) & \text{if we decide } \omega_2 \\ P(\omega_2|\mathbf{x}) & \text{if we decide } \omega_1 \end{cases}$$

To minimize the error for observation  $\mathbf{x}$ , the following decision rule, called the Bayesian classification rule, should be used:

$$\text{Decide } \omega_1 \text{ if } P(\omega_1|\mathbf{x}) > P(\omega_2|\mathbf{x}), \omega_2 \text{ if } P(\omega_2|\mathbf{x}) > P(\omega_1|\mathbf{x}).$$

When the two a posteriori probabilities are equal, then we are indifferent as to the membership of  $\mathbf{x}$ . The points  $\mathbf{x}$  where  $P(\omega_1|\mathbf{x})=P(\omega_2|\mathbf{x})$  form the separation curve. The above rule also minimizes the average probability of error (also called misclassification rate), defined as:

$$P(\text{error}) = \int_{-\infty}^{\infty} P(\text{error}|\mathbf{x})p(\mathbf{x})d\mathbf{x}.$$

Note that the integral is vector-valued; in other words, it is a short hand for the  $n$  integrals, each over one dimension of  $\mathbf{x}$ . The classification rate is defined as  $1-P(\text{error})$ .

Let  $g(\mathbf{x})$  denote the discriminant function, defined in either of two forms:

$$g(\mathbf{x}) = P(\omega_1|\mathbf{x}) - P(\omega_2|\mathbf{x}) \tag{2}$$

or

$$g(\mathbf{x}) = \ln P(\omega_1|\mathbf{x}) - \ln P(\omega_2|\mathbf{x}). \tag{3}$$

It is fairly easy to see that  $g(\mathbf{x})$  has the same sign in either form for the same  $\mathbf{x}$ . Then the Bayesian classification rule can be restated:

$$\text{Decide } \omega_1 \text{ if } g(\mathbf{x}) > 0, \omega_2 \text{ if } g(\mathbf{x}) < 0.$$

The average probability of error can be redefined as well:

$$\begin{aligned} P(\text{error}) &= \int_{\mathbf{x}:g(\mathbf{x})<0} P(\omega_1|\mathbf{x})d\mathbf{x} + \int_{\mathbf{x}:g(\mathbf{x})>0} P(\omega_2|\mathbf{x})d\mathbf{x} \\ &= P(\omega_1) \int_{\mathbf{x}:g(\mathbf{x})<0} \frac{p(\mathbf{x}|\omega_1)}{p(\mathbf{x})}d\mathbf{x} + P(\omega_2) \int_{\mathbf{x}:g(\mathbf{x})>0} \frac{p(\mathbf{x}|\omega_2)}{p(\mathbf{x})}d\mathbf{x}. \end{aligned}$$

The separation curve is the set of points  $\mathbf{x}$  where  $g(\mathbf{x})=0$ . Depending on the density function  $p(\mathbf{x}|\omega_j)$ , the separation curve may be a line, a parabola, or a more complicated structure.

### Normal Density Functions

Here, we assume that for each group  $j$ ,  $p(\mathbf{x}|\omega_j)$  is a multivariate normal function with  $\mu_j$  being the vector of  $n$  variable means and  $\Sigma_j$  the variance-covariance matrix among the variables. Then

$$p(\mathbf{x}|\omega_j) = (2\pi)^{-\frac{n}{2}} |\Sigma_j|^{-\frac{1}{2}} \exp\left[-\frac{1}{2}(\mathbf{x} - \mu_j)' \Sigma_j^{-1} (\mathbf{x} - \mu_j)\right],$$

where  $|\Sigma|$  denotes the determinant of matrix  $\Sigma$  and  $t$  denotes vector transpose.

With some manipulation, it can be shown that  $g(\mathbf{x})$  as defined by (3) can be simplified to:

$$g(\mathbf{x}) = \mathbf{x}'(\Sigma_2^{-1} - \Sigma_1^{-1})\mathbf{x} - 2\mathbf{x}'(\Sigma_2^{-1}\mu_2 - \Sigma_1^{-1}\mu_1) + \mu_2' \Sigma_2^{-1} \mu_2 - \mu_1' \Sigma_1^{-1} \mu_1 + \ln \frac{|\Sigma_2|}{|\Sigma_1|} + \ln \frac{P(\omega_1)}{P(\omega_2)}.$$

This function is quadratic in  $\mathbf{x}$ . Indeed, it is exactly the quadratic discriminant function developed by Smith [34] and shown in Anderson [1], when  $P(\omega_1)=P(\omega_2)$ . If the score is computed from a sample, then the population mean  $\mu_j$  is replaced by sample mean  $\bar{x}_j$ , and population variance-covariance matrix  $\Sigma_j$  by a sample matrix  $S_j$ ,  $j = 1, 2$ .

Therefore, it can be concluded that if the variables have multivariate normal density, then the separation curve is determined by a quadratic function. Depending on the specific form of the function, the curve can be an ellipse (which includes a circle), a parabola, or a hyperbola [7].

If the variance-covariance matrices are the same, that is,  $\Sigma_1=\Sigma_2=\Sigma$ , then the discriminant score is reduced to a linear function of  $\mathbf{x}$ :

$$g(\mathbf{x}) = 2\mathbf{x}'\Sigma^{-1}(\mu_1 - \mu_2) - (\mu_1 + \mu_2)' \Sigma^{-1} (\mu_1 - \mu_2) + \ln \frac{P(\omega_1)}{P(\omega_2)}.$$

This score is exactly twice the "Wald-Anderson" statistic (see [1]) for Fisher's linear discriminant analysis [8], again when  $P(\omega_1)=P(\omega_2)$ . It can be seen that the separation curve is the line at the midpoint between the two population means.

These results show that both Fisher's linear discriminant analysis [8] and Smith's quadratic discriminant analysis [34] are optimal, in the sense of minimizing average misclassification rate, when the variables are normally distributed and the population parameters are known. It may be more accurate to say that they are asymptotically optimal: as the sample sizes for both groups approach the population sizes, the sample estimates approach population parameters and the error rate is minimized.

### Nonparametric Models

When the variable distributions are unknown or cannot be specified algebraically, the Bayesian classification rule cannot be applied. For classification of objects with such attributes, nonparametric methods are used. As the optimal separation curve typically cannot be specified either, the performance of a classifier has to be measured by relying on test data which are large samples serving as surrogates for populations.

Prominent nonparametric methods include the  $k$ -nearest-neighbor and mathematical programming models. The  $k$ -nearest-neighbor assigns an object to the group to which most of its  $k$  nearest neighbors belong, as measured by some distance function like the Euclidean distance or Mahalanobis distance. Theoretically speaking the classification rate of this method approaches that of the Bayesian rule as  $k$  approaches infinity [7]. Mathematical programming methods include linear programs [9] [10] [24], mixed integer programs [3] [11], and nonlinear programs [35]. Rubin [28] has a thorough review of these methods.

SAS procedure DISCRIM [30] includes LDA, QDA, and nonparametric methods which include the  $k$ -nearest-neighbor method. The default distance measure is Mahalanobis. This procedure is used in Experiment II, along with a linear programming model called minimum sum of deviations (MSD) [9]. This is found to be the most effective among several models proposed by Freed and Glover [9]. Let  $x_i$  denote the (row) vector of the attributes of observation  $i$  and  $w$  be a vector of unknown weights. Model MSD has the following form.

$$\text{Min } \sum_i e_i$$

subject to

$$x_i w \leq b + e_i \quad \text{for all } i \text{ in group 1,}$$

$$x_i w \geq b - e_i \quad \text{for all } i \text{ in group 2,}$$

$$w \text{ unrestricted in sign, } b \neq 0,$$

$$e_i \geq 0 \quad \text{for all } i,$$

where variable  $b$  serves as the boundary between two groups. The idea behind this formulation is that every observation in group 1 is to be projected by  $w$  to the left of  $b$  and every observation in group 2 is to be projected to the right. When observation  $i$  in group 1 is projected to the right of  $b$ , then  $e_i$  is the measure of error. Similarly, observation  $i$  of group 2 has error  $e_i$  when it is projected to the left of  $b$ . Variable  $b$  must be nonzero, otherwise the optimal solution will have all variables equal to zero. The specific value of  $b$  is not important, but whether  $b$  is positive or negative is. For this research, we solve a model with  $b=1$  and another with  $b=-1$ , and select the model with the smaller objective value.

### NEURAL NETWORKS

Let  $G=(N,A)$  denote a neural network where  $N$  is the node set and  $A$  the arc set, and each arc is directed.  $G$  is assumed to be acyclic in that it contains no directed

circuit. The node set  $N$  is partitioned into three subsets:  $N_I$ ,  $N_U$ , and  $N_H$ .  $N_I$  is the set of input nodes,  $N_U$  is that of output nodes, and  $N_H$  that of hidden nodes. In a popular form called the multi-layer perceptron, all the input nodes are in one layer, all the output nodes are in another layer, and the hidden nodes are distributed into several layers in between. The knowledge learned by a network is stored in the arcs and the nodes, in the form of arc weights and node values called biases. We will use the term  $k$ -layered network to mean a layered network with  $k-2$  hidden layers (some authors do not include input nodes as a layer).

When a pattern is presented to the network, the variables of the pattern activate some of the neurons. Let  $a_i^p$  represent the activation value at node  $i$  corresponding to pattern  $p$ .

$$a_i^p = \begin{cases} s_i^p & \text{if } i \in N_I \\ F(y_i^p) & \text{if } i \in N_H \cup N_U, \end{cases}$$

where  $s_i^p$ ,  $i=1, \dots, n$  are the variables of pattern  $p$ . For a hidden or output node  $i$ ,  $y_i^p$  is the input into the node and  $F$  is called the activation function. The input, representing the strength of stimuli reaching a neuron, is defined as a weighted sum of incoming signals:

$$y_i^p = \sum_k w_{ki} a_k^p,$$

where  $w_{ki}$  is the weight of arc  $(k,i)$ . In some models, a variable called bias is added to each node. The activation function is used to activate a neuron when the incoming stimuli are strong enough. Today, it is typically a squashing function that normalizes the input signals so that the activation value is between 0 and 1. The most popular choice for  $F$  is the logistic function [5] [27] [40]. It is given by:

$$F(y) = (1 + e^{-y})^{-1}.$$

So the neural computing process is as follows. The variables of a pattern are entered into the input nodes. The activation values of the input nodes are weighted (with  $w_{ki}s$ ) and accumulated at each node in the first hidden layer. The total is then squashed (by  $F$ ) into the node's activation value. It in turn becomes an input into the nodes in the next layer, until eventually the output activation values are computed.

Before the network can be used for classifying a pattern, the arc weights must be determined. The process for determining these weights is called training. A training sample is used to find the weights which can best fit the patterns in the sample. Each pattern has a target value  $t_i^p$  for output node  $i$ . For a two-group classification problem, only one output node is needed and the target can be  $t^p=0$  for group 1 and 1 for group 2. In order to measure the best fit, a function of errors must be defined. Let  $E^p$  represent a measure of the error for pattern  $p$ :

$$E^p = \sum_{i \in N_U} |a_i^p - t_i^p|,$$