Management Science

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Managerial Applications of Neural Networks: The Case of Bank Failure Predictions

Kar Yan Tam, Melody Y. Kiang,

To cite this article:
Kar Yan Tam, Melody Y. Kiang, (1992) Managerial Applications of Neural Networks: The Case of Bank Failure Predictions. Management Science 38(7):926-947. http://dx.doi.org/10.1287/mnsc.38.7.926

MANAGEMENT SCIENCE
Vol 38, No 7, July 1992
Printed in U S A

MANAGERIAL APPLICATIONS OF NEURAL NETWORKS: THE CASE OF BANK FAILURE PREDICTIONS*

KAR YAN TAM AND MELODY Y. KIANG
Department of Business Information Systems, School of Business and Management, The Hong Kong University of Science and Technology, Hong Kong
Department of Decision and Information Systems, Arizona State University, Tempe, Arizona 85287-4206

This paper introduces a neural-net approach to perform discriminant analysis in business research. A neural net represents a nonlinear discriminant function as a pattern of connections between its processing units. Using bank default data, the neural-net approach is compared with linear classifier, logistic regression, *k*NN, and ID3. Empirical results show that neural nets is a promising method of evaluating bank conditions in terms of predictive accuracy, adaptability, and robustness. Limitations of using neural nets as a general modeling tool are also discussed.
(NEURAL NETWORKS; ARTIFICIAL INTELLIGENCE; DISCRIMINANT ANALYSIS; BANK FAILURE PREDICTIONS)

1. Introduction

Many managerial decisions involve classifying an observation into one of several groups. A special case of this problem is binary classification in which the number of groups is limited to two. Extensive literature has been devoted to studying this problem under various contexts, including credit scoring, default prediction, merger and acquisition, and bond rating, just to name a few. The solution to this problem is a discriminant function from the variable space in which observations are defined into a binary set.

Since Fisher's seminal work (1936), numerous methods have been developed for classification purposes. They are typically referred to as multivariate discriminant analysis (hereafter referred to as DA). In general, these methods accept a random sample of observations defined by a chosen set of variables and generate a discriminant function that serves as a classifier. They differ in two major aspects: (1) assumption on group distribution, and (2) functional form of the discriminant function. In the current study, we have taken a neural-net approach to the binary classification problem and compared it with popular DA methods. Our goal is to identify the potentials and limitations of neural nets as a tool to do discriminant analysis in business research.

The subject of neural nets, once viewed as the theoretical foundation for building artificial intelligent systems in the 1950s and 1960s, was proven to be too limited by Minsky and Papert (1969). Using simple examples, Minsky and Papert showed that only a few functions are guaranteed to be learned by a neural net. In the case of the well-known exclusive-or (XOR)[1] function, they showed that the function cannot be learned by a two-layer network; however, recent breakthroughs in neural nets research have overcome some of the limitations cited earlier. For example, Rumelhart, Hinton, and Williams (1986) have developed a backpropagation learning algorithm to train a multilayer network that can reproduce the XOR function.

The resurgent interest in neural nets has been manifested in the study of a new class of computation models called the connectionist models which have limited analogy, if any, to their neurophysiology origin (Rumelhart, McClelland and the PDP Research Group 1986). Connectionist systems provide massive parallel processing capabilities that

* Accepted by Vijay Mahajan; received September 1989. This paper has been with the authors 13 months for 2 revisions.

[1] XOR is a binary function which returns true when only one of its inputs is true, and false otherwise.

0025-1909/92/3807/0926$01.25

are essential in many domains, such as pattern recognition, concept classification, speech processing, and real-time process control (Waltz 1987, Tucker and Robertson 1988). Fault-tolerance is another appealing property that has profound implications in the design and fabrication of integrated circuits. A connectionist system can tolerate minor component failures without impairing the entire system. Existing computers are serial machines based on the Von Neumann architecture proposed some 40 years ago. These machines are designed to execute serial threads of instructions and are vulnerable even to minute component failures. Since our main concern is not in the biological isomorphism of connectionist models nor their implications in computer architecture design, we shall focus on the modeling capability of neural nets as inspired by these computation models. In particular, we shall compare the performance of classification models developed by popular DA methods and by neural nets along the following dimensions: robustness, predictive accuracy, adaptability, and explanatory capability.

The testbed used in our comparative study consists of bank-bankruptcy cases reported in the state of Texas. The increasing numbers of commercial bank failures have evolved into an economic crisis that has received much attention in recent years. It is therefore both desirable and warranted to explore new predictive techniques and to provide early warnings to regulatory agencies. Tam and Kiang (1990) introduced a neural-net approach for bank failure prediction. However, there are methodological problems that limit the generalization of the findings. For instance, not all information about a bank was utilized, and the final results may be biased by the hold-out samples chosen. In this study, we have extended our previous work by incorporating misclassification costs and prior probabilities in the neural-net models. We have included additional classification techniques for comparison and have taken a rigorous approach (Lachenbruch 1967) in validating the results.

§2 and §3 briefly review popular DA methods and neural nets respectively. §4 presents the bankruptcy prediction problem. §5 describes the sample data and the set up of each model. §6 reports the results of the comparative study. §7 discusses the potential and limitations of using neural nets in discriminant analysis. This is followed by a conclusion in §8.

2. Multivariate Discriminant Analysis

2.1. *Linear Discriminant Model*

Perhaps the most widely used DA method is the one due to Fisher (1936). The Fisher procedure constructs a discriminant function by maximizing the ratio of between-groups and within-groups variances. Classifiers derived from the Fisher procedure are known to be optimal in minimizing the expected cost of misclassifications, provided the following conditions are satisfied:

(1) each group follows a multivariate normal distribution,

(2) the covariance matrices of each group are identical,

(3) the mean vectors, covariance matrices, prior probabilities, and misclassification costs are known.

In the case of binary classification, the discriminant function is stated as

$$D(X) = X'\Sigma^{-1}(\mu_1 - \mu_2) - \tfrac{1}{2}(\mu_1 - \mu_2)^T\Sigma^{-1}(\mu_1 + \mu_2)$$

where μ_1, μ_2, and Σ^{-1} are the mean vectors and inverse of the common covariance matrix respectively. The threshold value of the decision rule is $\ln(C_{21}\pi_1/C_{12}\pi_2)$ where C_{12}, C_{21}, π_1, and π_2 are the misclassification costs and prior probabilities of each group. This method yields a linear function relating a set of independent variables to a scoring variable. It represents a hyperplane that divides the variable space into two partitions with each assigned to a group.

DA minimizes the expected misclassification cost provided the normality and equal dispersion assumptions are satisfied. Unfortunately, violations of these assumptions occur regularly. It is common that individual variables are not normally distributed (Deakin 1976). Examples can be found where variables are bounded or assume category values. Transformations, such as taking the natural logarithm, are suggested to approximate normal distribution; however, the transformed variables may be difficult to interpret. If the covariance matrices are different, quadratic instead of linear functions should be employed. Quadratic classifiers may be quite accurate in classifying the training sample, but they do not perform as well as linear models in hold-out sample tests (Altman et al. 1981). Lachenbruch, Sneeringer, and Revo (1973) reported a similar conclusion after comparing the two models under various nonmultivariate normal distributions. Whether the function is linear or quadratic, a fundamental condition that must be satisfied is that the two groups are discrete and identifiable. Situations deviating from this condition can be found where observations of each group form clusters in different regions of the variable space. Depending on the number of clusters in each group, the discriminant functions (linear or quadratic) may incur a high error rate for both the training and hold-out sample.

Altman et al. (1981) identified four related problems in the use of DA techniques in classification: (1) relative significance of the individual variables, (2) reduction of dimensionality, (3) elimination of insignificant variables, and (4) existence of time series relationships. Recognizing the limitations of linear classifiers, a common practice is to accept the results as if the assumptions were satisfied.

2.2. *Logistic Regression*

An alternative to the linear DA model is logistic regression. A nonlinear logistic function having the following form is used

$$Y = \frac{1}{1 + e^{y}}, \quad y = c_0 + \sum_{i=1}^{n} c_i X_i$$

where X_i, $1 \leq i \leq n$, represent the set of individual variables, c_i is the coefficient of the ith variable, and Y is the dependent variable. Since Y falls between 0 and 1, it is usually interpreted as the probability of a class outcome. In practice, it has been suggested that the logistic regression approach is often preferred over DA (Press and Wilson 1978). Harrell and Lee (1985) contended that even when all the assumptions of DA hold, a logit model is virtually as efficient as a linear classifier.

2.3. *k Nearest Neighbor*

Distribution-free techniques are applicable under less restrictive conditions regarding the underlying population distribution and data measurement scales. kNN is a nonparametric method for classifying observations into one or several groups based on one or more quantitative variables. It not only relaxes the normality assumption, it also eliminates the functional form required in DA and logistic regression. The group assignment of an observation is decided by the group assignments of its first k nearest neighbor. The distance $d(x, y)$ between any two observations x and y is usually defined by the Mahalanobis distance between x and y. Using the nearest neighbor decision rule, an observation is assigned to the group to which the majority of its k nearest neighbors belong. This method has the merits of better approximating the sample distribution by dividing the variable space into any arbitrary number of decision regions, with the maximum bounded by the total number of observations.

2.4. *Decision Tree (ID3)*

Instead of generating a decision rule in the form of a discriminant function, the ID3 method creates a decision tree that properly classifies the training sample (Quinlan 1979, 1983, 1986). This tree induction method has been applied in credit scoring (Carter and Catlett 1987), corporate failures prediction (Messier and Hansen 1988), and stock portfolio construction (Tam et al. 1991). Frydman, Altman and Kao (1985) applied a similar technique, called recursive partitioning to generate a discriminant tree. Both ID3 and recursive partitioning employ a nonbacktracking splitting procedure that recursively partitions a set of examples into disjointed subsets. These methods differ in their splitting criteria. The ID3 method intends to maximize the entropy of the split subsets, while the recursive partitioning technique is designed to minimize the expected cost of misclassifications.

The five techniques compared in this study can be categorized into two groups: machine learning (neural nets and ID3) and statistical techniques (DA, logit, and kNN). While previous research has focused mainly on a single method, the mix of techniques employed in this study allows a more comprehensive comparison of the different approaches to the problem.

3. Neural Networks

A neural net consists of a number of interconnected homogeneous processing units. Each unit is a simple computation device. Its behavior can be modeled by simple mathematical functions. A unit i receives input signals from other units, aggregates these signals based on an input function I_i, and generates an output signal based on an output function O_i (sometimes called a transfer function). The output signal is then routed to other units as directed by the topology of the network. Although no assumption is imposed on the form of input/output functions at each node other than to be continuous and differentiable, we will use the following functions as suggested in Rumelhart et al. (1986):

$$I_i = \sum_j w_{ij} O_j + \phi_i \qquad \text{and} \qquad O_i = \frac{1}{1 + e^{I_i}}, \qquad \text{where}$$

I_i = input of unit i,
O_i = output of unit i,
w_{ij} = connection weight between unit i and j,
ϕ_i = bias of unit i.

3.1. *Feedforward Networks*

The configuration of a neural net is represented by a weighted directed graph (WDG) with nodes representing units and links representing connections. Each link is assigned a numerical value representing the weight of the connection. Variations of the general WDG topology are found in a number of connectionist models (Broomhead and Lowe 1988, Moody and Darken 1989). A special class of neural nets called feedforward networks is used here.

In a feedforward network, there are three types of processing units: input units, output units, and hidden units. Input units accept signals from the environment and reside in the lowest layer of the network. Output units send signals to the environment and reside in the highest layer. Hidden units are units which do not interact directly with the environment, hence are invisible (i.e., hidden from the environment). Connections within a layer or from a higher layer to a lower are prohibited, but they can skip several layers.

The pattern of connectivity of a feedforward network is described by its weight vector W—weights associated with the connections. It is W that constitutes what a neural net

knows and determines how it will respond to any arbitrary input from the environment. A feedforward network with an appropriate W can be used to model the causal relationship between a set of variables. Changing the model is accomplished by modifying the weight associated with each connection.

3.2. *Backpropagation Learning Algorithm*

It is very difficult to assign an appropriate W for a classification task, especially when there is little information about the population distribution. A general solution is to let the network learn the task by training it with examples (Hinton 1989). A typical learning algorithm will search through the space of W for a set of weights offering the best fit with the given examples. Notable learning algorithms are the perceptron convergence procedure (Rosenblatt 1962) and the backpropagation algorithm (Rumelhart, Hinton, and Williams 1986).

The backpropagation learning algorithm, designed to train a feedforward network, overcomes some of perceptron's limitations by making it possible to train a multiple-layer network. It is an effective learning technique that is capable of exploiting the regularities and exceptions in the training sample. A flow chart of the algorithm is shown in Figure 1. The backpropagation algorithm consists of two phases: forward-propagation and backward-propagation. Suppose we have s examples, each described by an input

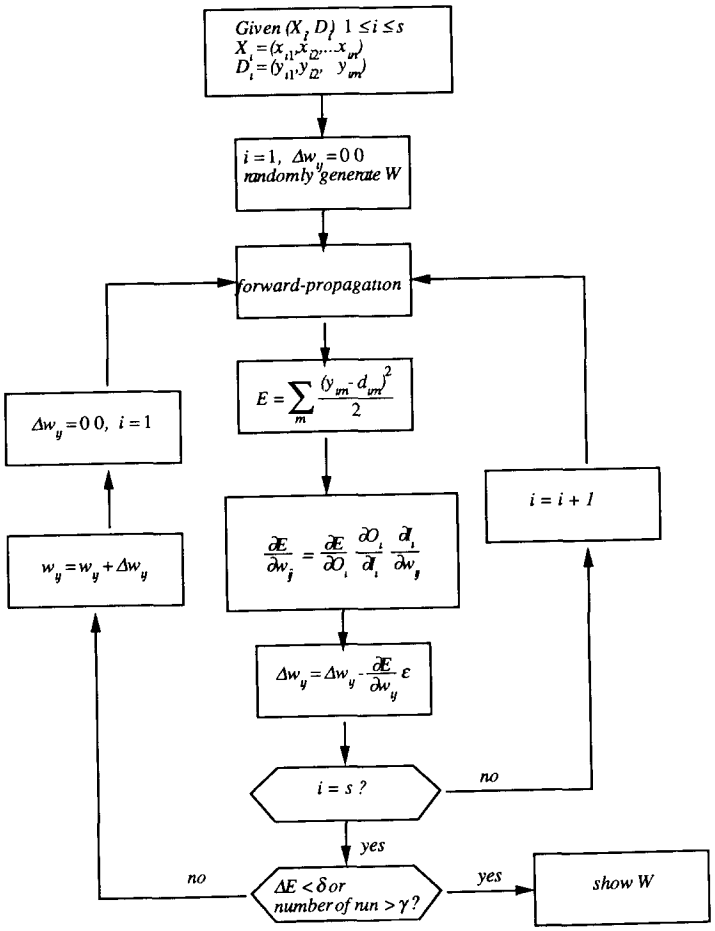

FIGURE 1. Flow Chart of the Backpropagation Algorithm.

vector $X_i = (x_{i1}, x_{i2}, \ldots, x_{im})$ and an output vector $D_i = (d_{i1}, d_{i2}, \ldots, d_{in})$, $1 \le i \le s$. In forward-propagation, X_i is fed into the input layer, and an output $Y_i = (y_{i1}, y_{i2}, \ldots, y_{in})$ is generated on the basis of the current W. The value of Y_i is then compared with the actual (or desired) output D_i by calculating the squared error $(y_{ij} - d_{ij})^2$, $1 \le i \le n$, at each output unit. Output differences are summed up to generate an error function E defined as

$$E = \sum_{i=1}^{s} \sum_{j=1}^{n} \frac{(y_{ij} - d_{ij})^2}{2}.$$

The objective is to minimize E by changing W so that all input vectors are correctly mapped to their corresponding output vectors. Thus, the learning process can be cast as a minimization problem with objective function E defined in the space of W.

The second phase performs a gradient descent in the weight space to locate the optimal solution. The direction and magnitude change Δw_{ij} of each w_{ij} can be calculated as

$$\Delta w_{ij} = -\frac{\partial E}{\partial w_{ij}} \epsilon,$$

where $0 < \epsilon < 1$ is a parameter controlling the convergence rate of the algorithm.

The total squared error calculated in the first phase is propagated back, layer by layer, from the output units to the input units in the second phase. Weight adjustments are determined on the way of propagation at each level. Since I_i, O_i and E are all continuous and differentiable, the value of $\partial E/\partial w_{ij}$ at each level can be calculated by applying the chain rule

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial O_i} \frac{\partial O_i}{\partial I_i} \frac{\partial I_i}{\partial w_{ij}}.$$

W can be updated in two ways. Either W is updated for each (X_i, D_i) pair, or Δw_{ij} are accumulated and updated after a complete run of all examples. The two phases are executed in each iteration of the backpropagation algorithm until E converges. Although the backpropagation algorithm does not guarantee optimal solution, Rumelhart et al. (1986) reported that solutions obtained from the algorithm come close to the optimal ones in their experiments.

4. Bank Bankruptcy Prediction

The number of financial distresses in the banking industry has reached a historic high unparalleled since the great Depression. The number of bankruptcy cases filed under the Federal Deposit Insurance Corporation (FDIC) has increased from less than 50 in 1984 to an estimate of over 400 in 1991. To monitor the member banks and to assure their proper compliance with federal regulations, the FDIC has committed substantial efforts to both on-site examinations and off-site surveillance activities. Since the mid 1970s, the FDIC has been operating an "early warning" system that facilitates the rating process by identifying troubled banks and alerting the agency for early inspection.[2] This is accomplished by statistically evaluating the reports filed by each bank on a regular basis. According to West (1985) "*performance ratios of banks are compared to some standard, or arranged to some statistical cut-off. Banks that 'fail' these ratio tests are singled out for more careful scrutiny.*"

[2] Similar early warning systems have been operated in the Office of Comptroller of the Currency (OCC) and the Federal Reserve System (Fed).

Given the importance of this subject at both the micro and macro level, numerous models have been developed to predict bank failure. These models employ statistical techniques which include regression analysis (Meyer and Pifer 1970), multivariate discriminant analysis (Altman 1968, Sinkey 1975; Santomero and Vinso 1977), multivariate probit or logic analysis (Hanweck 1977; Martin 1977), arctangent regression analysis (Korobow and Stuhr 1985), and Factor-logistic analysis (West 1985). Although widely practiced, these models have been criticized for their problematic methodologies (Eisenbeis, 1977), and a satisfactory model has yet to be developed.

5. Data Sample and Model Construction

5.1. *Data Sample*

The data sample consists of Texas banks that failed in the period 1985–1987 (*Bank of Texas* 1985–1987). Texas banks were selected for two reasons. First, more than one quarter of the failed banks in 1987 were located in Texas (FDIC 1987). Not surprisingly, the high bankruptcy rate coincides with the general economic conditions in the Southwestern region, especially in the energy and real estate sectors of the economy. Thus, it is interesting to develop a prediction model specifically tailored to the economic environment of this region. Second, involving banks from the same region, instead of those from other states, increases the sample's homogeneity.

The data sample consists of bank data one year and two years prior to failure. As a control measure, a failed bank was matched with a nonfailed bank in terms of (1) asset size, (2) number of branches, (3) age, and (4) charter status. In each period, 118 banks (59 failed and 59 nonfailed) were selected as the training sample.

Each bank is described by 19 financial ratios that have been used in previous studies. The list of ratios is shown in Table 1. The selection of variables followed closely the CAMEL criteria used by the FDIC. CAMEL is an acronym for Capital, Asset, Management, Equity, and Liquidity which is generally adopted by all U.S. bank regulatory agencies. An assessor rates a bank according to its scores in each of these five areas, and the composite ratings are taken to reflect the financial conditions of the bank. Rating results

TABLE 1
A List of Financial Variables

Name	Description
capas	capital/assets
agfas	(agricultural production & farm loans + real estate loans secured by farm land)/net loans & leases
comas	(commercial and industrial loans)/net loans & leases
indas	(loans to individuals)/net loan & leases
resas	(real estate loans)/net loan & leases
pasln	(total loans 90 days or more past due)/net loans & leases
nonln	(total nonaccrual loans & leases)/net loans & leases
losln	(provision for loan losses)/average loans
netln	(net charge-offs)/average loans
rtoas	return on average assets
intdp	(total interest paid on deposits)/total deposits
expas	(total expense)/total assets
incas	(net income)/total assets
infln	(interests and fees on loans + income from lease financing rec)/net loans & leases
incex	(total income)/total expense
curas	(cash + U.S. treasury & government agency obligations)/total assets
govas	(federal funds sold + securities)/total assets
llnas	(total loans & leases)/total assets
llndp	(total loans & leases)/total deposits

provide early warnings to an agency, drawing its attention to those banks that have a high likelihood of failure in the coming one or two years.

The 19 ratios can be grouped into four of these five criteria. The first ratio represents the capital adequacy of the bank; ratios 2–10 measure asset quality; the bank's earnings are captured by ratios 11–15, and liquidity is represented by ratios 16–19. No explicit ratio was used for the Management criterion because the quality of management, which is difficult to quantify, will eventually be reflected by the above-mentioned ratios.

5.2. *Linear Discriminant Model*

The Kolmogorov-Smirnov test[3] was performed for each of the 19 financial ratios in the data sample to check if the normal distribution assumption was satisfied. The test indicated that 15 out of 19 ratios were not normally distributed in the one-year period. In the two-year period, only one ratio was shown to be normally distributed. For those ratios that failed the test, the natural logarithm transformation was performed. The Kolmogorov-Smirnov test was then repeated for the transformed ratios. The results showed that 13 out of 19 ratios in the one-year period and 14 out of 19 in the two-year period were still not normally distributed. Since no significant improvement was observed, we decided to use the original ratios to construct the DA models. The DA models were implemented using FORTRAN with embedded IMSL procedure calls.

5.3. *Logistic Model*

Like the DA model, no variable transformation was performed. All 19 variables were used to estimate the logit model for the one- and two-year period.[4] A bank will be classified as a failed bank if the value of the dependent variable is less than 0.5, and as a nonfailed bank otherwise.

5.4. *kNN Models*

Two kNN models, one with $k = 1$ and the other with $k = 3$, were constructed for each period.[5] For $k = 1$, a bank was assigned to the group which contains its nearest neighbor, while, in $k = 3$, the bank was assigned to the group which contained the majority of its three closest neighbors. Even values of k were not included due to the possibility of a tie.

5.5. *ID3*

The ID3 algorithm was implemented in CommonLisp. A chi-square stopping rule with a significance level of 5% was used to reduce the effects of noisy data. The classification trees generated by using ID3 for both periods are shown in Figures 2a and 2b. The F and NF in each leaf of the tree denote the number of failed and nonfailed banks, respectively. The number of variables has been reduced from 19 to 6 in the one-year period and to 8 in the second-year period. The classification tree for the former exhibits a more balanced and less complex structure than the latter, indicating that observations in the two-year period may intermesh more uniformly than in the one-year period. In addition, two-thirds of the variables in the one-year period (nonln, expas, pasln, rtoas) representing mainly asset quality and earnings also appear in the two-year period.

5.6. *Neural Network Models*

We have performed some exploratory experiments to decide on the configuration of the neural nets used in our study. This is a required step because there is yet a formal

[3] Significance level of the test is 5%.

[4] Coefficients of the logit model were estimated using the SAS LOGIT procedure.

[5] The 1NN and 3NN models were constructed using the SAS NEIGHBOR procedure.

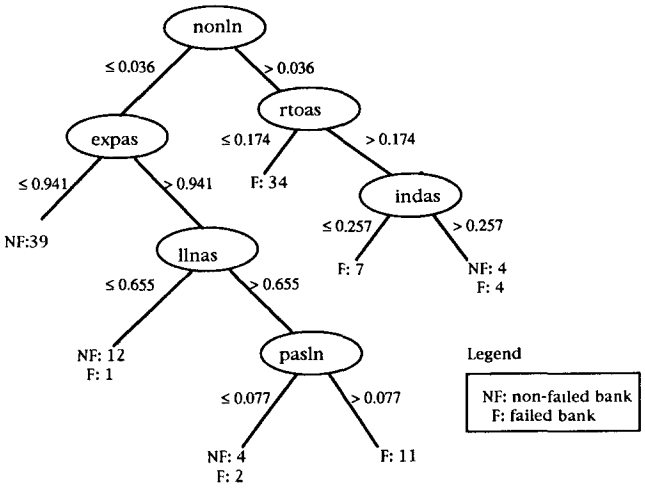

FIGURE 2a. Classification Tree for One-Year Period.

algorithm to be developed to map a task to a configuration. We have constructed 2-layer and 3-layer networks; for 3-layer networks, different numbers of hidden units were tried. Finally two configurations, one with no hidden unit (2-layer) and the other with 10 hidden units (3-layer), were constructed for each period (see Figures 3a and 3b).

The original backpropagation algorithm does not take into account the prior probabilities of each group and their misclassification costs. To incorporate them into the learning algorithm, the objective function E is generalized to E_w defined as

$$E_w = \sum_{i=1}^{2} Z_i \sum_{j=1}^{n_i} \frac{(y_{ij} - d_{ij})^2}{2}$$

where $Z_2 = C_{12}\pi_2$ represents type I misclassification error, $Z_1 = C_{21}\pi_1$ represents type II misclassification error and n_i ($i = 1, 2$) is the number of examples in group i. The type I error is defined as the event of assigning an observation to group 1 that should be in

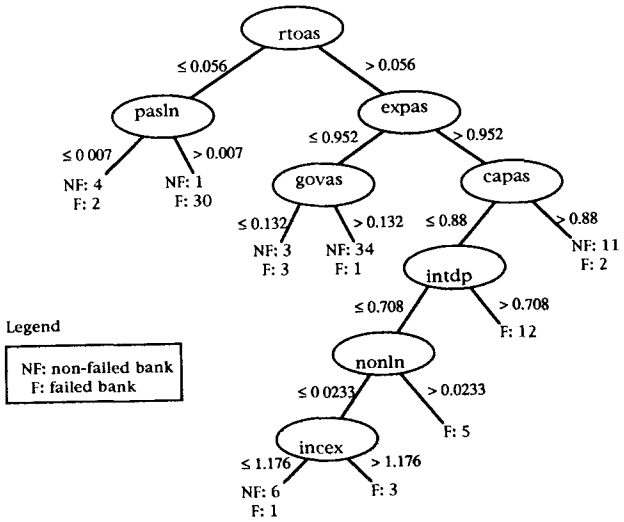

FIGURE 2b. Classification Tree for Two-Year Period.

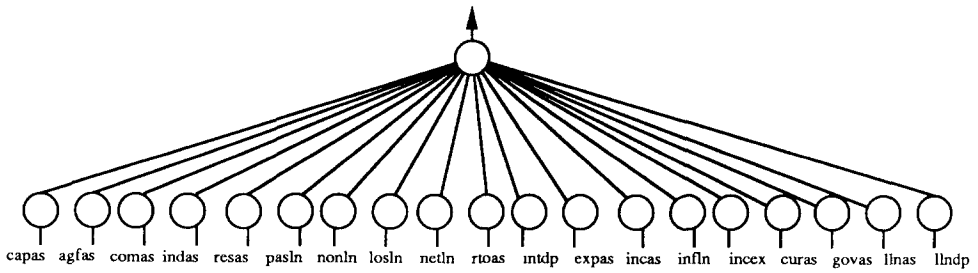

FIGURE 3a. The Network Configuration of Net_0.

group 2, while the type II error involves assigning an observation to group 2 that should be in group 1. The value of Z_i is treated as the weight of the squared error incurred by each observation. It is clear that the initial objective function E is a special case of E_w by setting $C_{12}\pi_2 = C_{21}\pi_1$. In the current study, we will refer to nonfailed banks as group 1 and failed banks as group 2.

Since we are concerned with dichotomous classification (failure vs. nonfailure), only a single output unit is needed. The decision is stated as

$$\text{output unit} > 0.5 \Rightarrow \text{group 1 (nonfailed banks)},$$

$$\text{output unit} \leq 0.5 \Rightarrow \text{group 2 (failed banks)}.$$

Instead of using $\Delta w_{ij} = -\partial E/\partial w_{ij}\epsilon$, which may be slow in terms of convergence, we use an accelerated version as shown below:

$$\Delta w_{ij}(t) = -\epsilon \frac{\partial E}{\partial w(t)} + \alpha \Delta w_{ij}(t-1)$$

where $0 \leq \alpha \leq 1$ is an exponential decay factor determining the contribution of the previous gradient descent.

To smooth out drastic changes to W by some outlining examples, we prefer the accumulated weights updating scheme. Since no prior information is available as to how units should be connected in the 3-layer network, all hidden units are fully connected to the input units.

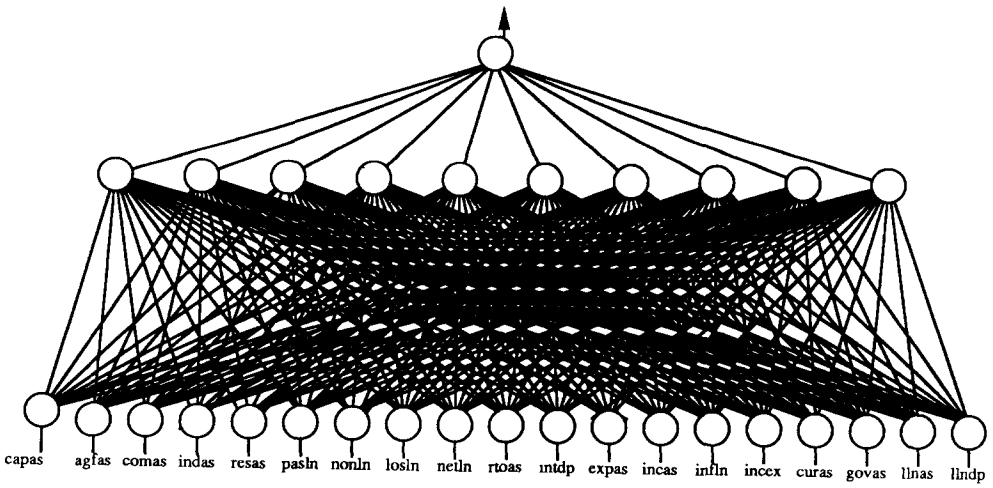

FIGURE 3b. The Network Configuration of Net_{10}.

The backpropagation procedure was implemented in Pascal and run on an EMX machine. Five different sets of weights were generated and five different runs were done for each neural net model. Each run was allocated a maximum of 2000 iterations. The classification accuracy obtained by each run was ranked by total misclassifications, and the median run was taken as the result.

6. Computation Results

The classification accuracy of neural nets is first compared with that of DA using different prior probabilities and misclassification costs. Two prior probabilities for failed banks (π_2), 0.01 and 0.02, and eight misclassification costs are used. We followed the distribution of misclassification costs used previously by Frydman et al. (1985), where the misclassification cost of nonfailed banks (i.e., C_{21}) is kept to 1, while the misclassification costs of failed banks (i.e., C_{12}) are set to 1, 5, 25, 40, 50, 60, 75, and 100. The values of ϵ, α, and δ are set to 0.7, 0.5 and 0.01, respectively.

Results of this comparison are shown in Tables 2–5. In Tables 2 and 3, the type I and type II errors of each model are given for each combination of prior probability and misclassification cost. The resubstitution risks for each combination are also calculated and displayed in Tables 4–5. According to Frydman et al. (1985), the resubstitution risk is the observed expected cost of misclassification defined as

$$C_{12}\pi_2 \frac{n_2}{N_2} + C_{21}\pi_1 \frac{n_1}{N_1},$$

where n_i is the total number of type i misclassifications and N_i is the sample size of the ith group.

The numbers in Tables 2–3 indicate that neural nets transcend smoothly from minimizing type I errors to type II errors as C_{12} increases. In the contrary, there is a sharp transition from $C_{12} = 5$ to $C_{12} = 25$ in DA. The proportion of DA's type I and type II errors remains virtually the same as C_{12} increases beyond 25, and its resubstitution risk starts to level off at this value. As shown, DA models are not sensitive to changes in C_{12} and π_2. This can be explained by the fact that the training samples are not normally distributed. There is evidence that nonfailed banks have a multi-modal distribution which is particularly apparent in the one-year period. For example, looking at the first and fourth rows in Table 2, there is a distinct cluster of 12 to 13 nonfailed banks that fall into the decision region of the failed group. Similarly, a cluster of about 10 nonfailed banks is observed in Table 3.

In Table 2, there are several occasions where Net_0 and Net_{10} generate inconsistent results. For example, the values of Z_i are known to be identical for both $C_{12} = 100$, $\pi_2 = 0.01$ and $C_{12} = 50$, $\pi_2 = 0.02$. One should expect similar results in both cases, but both their number and proportion of errors are quite different. This is indirectly due to the initial weights assigned to Net_0 and Net_{10}. Since the time a net takes to converge varies according to the starting point, the search may have yet to settle on a local optimum when the program halts, resulting in a different number of misclassification errors. We have rerun the algorithm to validate this argument by lifting the iteration bound in some of these inconsistent cases. Similar results were obtained this time. Rumelhart et al. (1986) and many other researchers have mentioned in their work that initial weights are used to break up the symmetry of a net.[6] They do not have a major effect on the final results if sufficient time is allowed for the net to converge. The use of five random weights is designed to average out the discrepancy in convergence periods.

[6] Usually, values slightly different from zero are used as initial weights to break up the symmetry.

TABLE 2

Misclassification Errors in the Training Sample for Different Misclassification Costs and Prior Probabilities (One-Year Period)

	C_{12} =	1			5			25			40			50			60			75			100		
Model		I	II	T	I	II	T	I	II	T	I	II	T	I	II	T	I	II	T	I	II	T	I	II	T
	$\pi_2 = 0.01$																							(*)	
DA		8	7	(15)	4	8	(12)	0	11	(11)	0	12	(12)	0	12	(12)	0	12	(12)	0	13	(13)	0	13	(13)
Net_0		11	1	(12)	9	1	(10)	8	1	(9)	6	2	(8)	6	3	(9)	4	6	(10)	4	7	(11)	0	8	(8)
Net_{10}		16	0	(16)	11	0	(11)	10	0	(10)	8	0	(8)	5	0	(5)	4	0	(4)	3	0	(3)	0	2	(2)
	$\pi_2 = 0.02$														(*)										
DA		6	8	(14)	3	9	(12)	0	12	(12)	0	13	(13)	0	13	(13)	0	14	(14)	0	14	(14)	0	14	(14)
Net_0		17	1	(18)	9	2	(11)	8	3	(11)	8	3	(11)	6	5	(11)	0	7	(7)	0	11	(11)	0	15	(15)
Net_{10}		11	0	(11)	10	0	(10)	10	0	(10)	0	6	(6)	0	7	(7)	0	9	(9)	0	10	(10)	0	13	(13)

Notes:

(*) $\pi_2 C_{12} \cong \pi_1 C_{21}$

I—Number of type I misclassifications

II—Number of type II misclassifications

T—Total number of misclassifications (in parentheses)

TABLE 3

Misclassification Errors in the Training Sample for Different Misclassification Costs and Prior Probabilities (Two-Year Period)

	C_{12} =	1			5			25			40			50			60			75			100		
Model		I	II	T	I	II	T	I	II	T	I	II	T	I	II	T	I	II	T	I	II	T	I	II	T
	$\pi_2 = 0.01$																							(*)	
DA		34	5	(39)	20	6	(26)	5	8	(13)	5	10	(15)	5	10	(15)	5	10	(15)	5	10	(15)	4	10	(14)
Net_0		39	0	(39)	30	0	(30)	17	1	(18)	16	1	(17)	11	2	(13)	12	4	(16)	10	6	(16)	8	6	(14)
Net_{10}		35	0	(35)	19	0	(19)	19	0	(19)	14	2	(16)	12	3	(15)	9	3	(12)	6	5	(11)	5	6	(11)
	$\pi_2 = 0.02$														(*)										
DA		27	5	(32)	13	8	(21)	5	10	(15)	5	10	(15)	4	10	(14)	4	10	(14)	3	10	(13)	3	11	(14)
Net_0		35	0	(35)	20	0	(20)	16	1	(17)	4	8	(12)	4	8	(12)	4	8	(12)	4	10	(14)	3	13	(16)
Net_{10}		29	0	(29)	13	0	(13)	11	3	(14)	10	4	(14)	3	6	(9)	3	6	(9)	3	6	(9)	0	17	(17)

Notes:
(*) $\pi_2 C_{12} \cong \pi_1 C_{21}$
I—Number of type I misclassifications
II—Number of type II misclassifications
T—Total number of misclassifications (in parentheses)

TABLE 4
Resubstitution Risks of the Training Sample (One-Year Period)

	C_{12} =	1	5	25	40	50	60	75	100
Model	$\pi_2 = 0.01$								(*)
DA		0.119	0.138	0.185	0.201	0.201	0.201	0.218	0.218
Net_0		0.019	0.024	0.051	0.074	0.101	0.141	0.168	0.134
Net_{10}		0.003	0.009	0.042	0.054	0.042	0.041	0.038	0.034
	$\pi_2 = 0.02$					(*)			
DA		0.135	0.155	0.199	0.216	0.216	0.233	0.233	0.233
Net_0		0.022	0.019	0.117	0.158	0.185	0.116	0.183	0.249
Net_{10}		0.004	0.017	0.085	0.100	0.116	0.150	0.166	0.216

Notes
(1) Resubstitution risk = $\pi_1 C_{21} n_1/N_1 + \pi_2 C_{12} n_2/N_2$ where n_i = total number of type i misclassifications. N_i = sample size of the ith group.
(2) (*) $\pi_2 C_{12} \cong \pi_1 C_{21}$.

In both periods, Net_0 and Net_{10} dominate DA with lower resubstitution risk across all combinations of π_2 and C_{12}. The results of Tables 4 and 5 also illustrate that Net_{10} outperforms Net_0 in most cases with a few exceptions. These exceptions can be explained by the different running times associated with the initial weights and can be eliminated by allowing a net to run to convergence. The better performance of Net_{10} can be explained by the incorporation of hidden units, which provides a better fit with the training sample distribution. The resubstitution risks of Net_{10} and Net_0 are almost identical in the two-year period for small C_{12}. The dominating performance of Net_{10}, however, starts to degrade in the two-year period. The percentage reduction in resubstitution risk over Net_0 decreases from an average of 44% in the first period to 16.2% in the second period.

Table 6 depicts the misclassification rates of each method in predicting the training samples. Since logit, kNN, and ID3 do not account for prior probabilities and misclassification costs, the comparison is made possible by setting approximately $C_{12}\pi_2 \cong C_{21}\pi_1$ in DA and the neural nets.

In the one-year period, Net_{10} outperforms other methods with lower type I and total misclassification rates. This is followed by logit, ID3, Net_0, DA, 1NN and 3NN. In the two-year period, DA is the best classifier, scoring the lowest type II and total misclassi-

TABLE 5
Resubstitution Risks of the Training Sample (Two-Year Period)

	C_{12} =	1	5	25	40	50	60	75	100
Model	$\pi_2 = 0.01$								(*)
DA		0.089	0.118	0.155	0.202	0.210	0.219	0.231	0.236
Net_0		0.007	0.025	0.089	0.125	0.127	0.189	0.228	0.236
Net_{10}		0.006	0.016	0.081	0.128	0.152	0.142	0.160	0.185
	$\pi_2 = 0.02$					(*)			
DA		0.092	0.155	0.208	0.234	0.234	0.247	0.242	0.284
Net_0		0.012	0.034	0.152	0.187	0.203	0.214	0.268	0.318
Net_{10}		0.010	0.022	0.143	0.202	0.151	0.161	0.176	0.282

Notes.
(1) Resubstitution risk = $\pi_1 C_{21} n_1/N_1 + \pi_2 C_{12} n_2/N_2$ where n_i = total number of type i misclassifications. N_i = sample size of the ith group.
(2) (*) $\pi_2 C_{12} \cong \pi_1 C_{21}$.

TABLE 6

Misclassification Rates of the Various Models Using the Training Sample

	Percentage (%)					
	One-year Prior			Two-year Prior		
Model	I	II	T	I	II	T
DA	0.0	22.0	(11.0)	10.2	1.7	(6.0)
Logit	8.5	6.8	(7.7)	13.6	13.6	(13.6)
1NN	37.3	23.7	(30.5)	32.2	32.2	(32.2)
3NN	35.6	25.4	(30.5)	37.3	32.3	(34.8)
ID3	10.2	5.1	(7.7)	13.5	5.1	(9.3)
Net_0	5.0	11.0	(8.0)	10.2	11.9	(11.0)
Net_{10}	0.0	7.6	(3.8)	6.7	10.2	(8.5)

Note

(1) the type I and type II misclassification rates of DA, Net_0 and Net_{10} are based on the average of (C_{12} = 100, π_2 = 0.01) and (C_{12} = 50, π_2 = 0.02).

fication errors. Net_{10} is the second best which is followed by ID3, Net_0, logit, 1NN and 3NN.

Misclassification rates based on the training sample are often overestimated and need further validation. The predictive accuracy of each method is validated by a hold-out sample. The sample consists of 44 banks (22 failed and 22 nonfailed) and 40 banks (20 failed and 20 nonfailed) in the one- and two-year periods respectively. Selection is made according to a similar matching procedure for the training sample. To facilitate comparison, the expected costs of misclassification for both type I and type II errors are approximately identical (i.e., $C_{12}\pi_2 \cong C_{21}\pi_1$). The validation results of the hold-out sample are reported in Table 7.

The performance ranking in Table 7 is different from that of Table 6. In the one-year period, Net_{10} remains the best classifier in terms of fewer type II and total errors. This is followed by DA, Net_0, logit, ID3, 1NN and 3NN. To our surprise, logit scores the lowest type II and total errors in the two-year period. Net_{10} comes next and is followed by Net_0, DA, 3NN, ID3, and 1NN.

TABLE 7

Misclassification Rates of the Various Models Using the Hold-Out Sample

	Percentage (%)					
	One-year Prior			Two-year Prior		
Model	I	II	T	I	II	T
DA	18.2	13.6	(15.9)	30.0	5.0	(17.5)
Logit	31.8	4.5	(18.2)	15.0	0.0	(7.5)
1NN	40.9	4.6	(22.8)	20.0	25.0	(22.5)
3NN	36.4	9.1	(22.8)	30.0	10.0	(20.0)
ID3	22.7	18.2	(20.5)	40.0	5.0	(22.5)
Net_0	31.8	4.5	(18.2)	20.0	12.6	(16.3)
Net_{10}	18.2	11.4	(14.8)	2.5	20.0	(11.3)

Note.

(1) the type I and type II misclassification rates of DA, Net_0 and Net_{10} are based on the average of (C_{12} = 100, π_2 = 0.01) and (C_{12} = 50, π_2 = 0.02).

The performances of DA and logit are not stable in both tests. In the first test, DA ranks the fifth in the one-year period and becomes the best classifier in the two-year period. In the hold-out sample test, DA scores the second lowest misclassification rates but degrades to the fourth place in the two-year period. Logit behaves in a similar way. In the first test, it ranks second to Net_{10} in the first period, and drops to the fifth place in second period. In the validation test, it jumps from the fourth place in the one-year period to the first in the two-year period. The performances of other methods are relatively stable. While Net_{10} remains high in the ranking list, *k*NN performs the worst in both tests. Net_0 and ID3 reside in the middle of the list with their positions interchanged in Tables 6 and 7.

Five out of seven methods in Table 7 have lower misclassification rates in predicting bank failures two years ahead. This is contrary to our intuition, since the earlier the prediction, the more uncertain it is, and one should expect a higher misclassification rate. It is difficult to conceive that the logit model can reduce more than half of its misclassification errors (18.2% in the one-year and 7.5% in two-year period) one year earlier. Furthermore, the ratios of type I to type II errors in the hold-out sample test are not consistent with those in Table 6. The number and ratio of type I and type II errors vary widely in both tables. The only explanation is that the hold-out sample (probably both training and hold-out samples) is not a representative sample of the group distributions.

Depending on the samples chosen, error rates estimated by the hold-out sample may be biased. An alternative estimation method is the jackknife method that Lachenbruch (1967) has shown to produce unbiased estimates for the probability of misclassification. The method involves holding one example out of the training set and using the estimated discriminant function to predict the extracted example. This is repeated for each example in the training set, and the proportion of misclassifications in each class is reported as its misclassification rate. The training and hold-out samples are pooled to form one single training sample which consists of 162 (81 failed and 81 nonfailed) and 158 (79 failed, 79 nonfailed) examples in the one- and two-year periods, respectively. Descriptive statistics of the training samples are shown in Appendices I and II. The neural nets are allowed to run until convergence this time.

As shown in Table 8, Net_{10} scores the lowest total misclassification rates in both periods. This is followed by Net_0, DA, logit, ID3, 3NN and 1NN. The relative ranking remains virtually the same in both periods. The only difference is between DA and Net_0 which

TABLE 8

Misclassification Rates Estimated Using the Jackknife Method

	Percentage (%)					
	One-year Prior			Two-year Prior		
Model	I	II	T	I	II	T
DA	17.3	11.1	(14.2)	17.3	13.9	(15.6)
Logit	12.3	17.3	(14.8)	15.2	20.3	(17.7)
1NN	17.3	38.3	(27.8)	31.6	29.1	(30.4)
3NN	18.5	30.9	(24.7)	19.0	26.6	(22.8)
ID3	21.0	17.3	(19.2)	20.3	25.3	(22.8)
Net_0	8.6	13.5	(11.1)	8.9	25.3	(17.1)
Net_{10}	8.6	12.3	(10.5)	8.9	12.7	(10.8)

have their positions interchanged. The performance of logit is not as superior as in the hold-out sample test. It ranks in the fourth place behind Net_{10}, Net_0, and DA. 1NN and 3NN remain the worst classifiers after ID3 in both periods. When compared between predictive periods, all methods except 3NN have lower total misclassification rates in the one-year period than in the two-year period. This is consistent with our intuition.

Misclassification rates estimated from both validation tests are compared. The proportions between type I and type II errors are less extreme in the jackknife test. For example, in the two-year period, the ratio of type I to type II errors changed from 6 to 1.24 in DA and from 0.13 to 1 in Net_{10}. Hold-out test overestimates the total misclassification rates of 1NN, 3NN, logit (one-year) and Net_0 (two-year) and underestimate that of DA, Net_{10}, logit (two-year), and Net_0 (one-year). The results of ID3 are quite consistent, although its type I and type II error compositions are very different in the two tests. Furthermore, the absolute difference between total misclassification rates estimated by the two tests are relatively small for most methods.

Since the misclassification rates estimated by the jackknife method have been shown to be unbiased, there is evidence that the neural-net approach provides better predictive accuracy than DA methods. We have also eliminated the effects of premature termination of the learning procedure by allowing each net in the jackknife method to run until convergence. As shown in both tests, a net with no hidden unit has a performance similar to a DA, but the incorporation of a layer of hidden units improves considerably its predictive accuracy. This can best be explained by viewing the partitioning structure induced by a DA method. Each method divides the variable space into disjointed partitions in very different ways. For instance, a DA model cuts the space into two partitions with a hyperplane, while an ID3 model divides the space into a number of recursive rectangular partitions. The sensitivity of a partitioning structure to the distribution of the training sample varies among methods, resulting in very different misclassification rates. Net_{10}, with the lowest total misclassification rates, offers a structure that best matches the training examples. In fact, it has been shown that a three-layer net can be used as a universal approximate for any continuous function in a multi-dimensional space. Geometrically, a network is capable of generating nonlinear partitioning structures that very often fit better a given training sample than other DA methods.

7. Discussion

The neural-net approach presented in this paper offers an alternative to existing bankruptcy prediction models. Empirical results show that neural nets offer better predictive accuracy than DA, logit, kNN and ID3. The original backpropagation algorithm is modified to include prior probabilities and misclassification costs. Depending on the classification tasks, the tradeoff between type I and type II errors may be very different and needs to be accounted for. It is essential to allow an assessor to state his own preference in deciding such a tradeoff. For example, the error of misclassifying a failed bank to the nonfailed group (type I error) is generally accepted to be more severe than the other way. The original function E is generalized to E_W by multiplying each error term by Z_t. It is worthwhile to note that minimizing E_W is not equivalent to minimizing the expected misclassification cost. Although the results in Tables 2–5 show that the nets do behave in this direction and outperform linear classifiers in minimizing resubstitution risks, more empirical studies are needed to validate this result.

Our comparison is based on a training set with an equal proportion of failed and nonfailed banks. In many cases, the number of defaults constitutes only a small portion of the whole population. The matching process may introduce biases to the model. To avoid this, the entire population should be used as the training set. There are many

application domains (e.g., handwritten character recognition) for which a neural net is an appropriate choice for identifying a single group from a large set of alternatives. It has been proved that a net with a hidden layer can compute any Boolean function with k variables (Denker et al. 1987). It is therefore possible to identify a group out of a total of 2^k cases. As illustrated in the XOR example, this is not possible for a linear DA model.

In terms of explanatory capability, it has been shown that the coefficients of a linear discriminant function convey little information about the relative importance of individual variables. Unlike logit analysis, there is no rigorous statistical test on the significance of individual coefficients. The same criticism is also applicable to kNN and neural nets, the results of which are difficult to interpret. On the other hand, the symbolic approach of ID3 sheds some light on the importance of individual variables. A variable is selected as the splitting variable when it can partition a set of examples into the most homogeneous subgroups. Homogeneity is measured by the weighted entropy of the result subgroups. For example, in Figure 2a, the root node nonln (nonln > 0.036 ?) correctly identifies 93.22% of the nonfailed banks and 76.27% of the failed banks, and, in Figure 2b, the root node rtoas (rtoas > 0.056 ?) accounts for 91.5% of the nonfailed banks and 54.32% of the failed banks.

Dimension reduction is another problem associated with existing DA techniques. West (1985) extended the logit approach by augmenting it with factor analysis. The factor-logistic method reduces the number of dimensions by transforming the space of initial variables into one comprised of important factors that account for a large portion of the variance (e.g., 90%). Observations are described by their factor scores in the new factor space. The factor scores are then put into a logistic regression model with a dichotomy dependent variable. This combined factor-logistic approach has proven effective in predicting bank bankruptcy; however, the meaning of each factor is subject to interpretation, and the actual number of variables for describing each observation remains the same. In the ID3 approach, Quinlan (1986) showed that minimizing the entropy of a decision tree is equivalent to minimizing the expected number of tests to make a decision. Thus, the ID3 method has a built-in mechanism to reduce the dimensions of the variable space. For example, in Figures 2a and 2b, the number of variables is reduced by 66.67% and 57.89% in the one-year and two-year periods respectively. In feedfoward nets, the number of dimensions equals the number of input units.

A neural net allows adaptive adjustment to the predictive model as new examples become available. This is an attractive property especially when the underlying group distributions are changing. Statistical methods assume old and new examples are equally valid, and the entire training set is used to construct a new model. The batch update is necessary if the distributions do not change. However, in situations where the new sample is drawn from a new distribution, retaining the old examples may result in a predictive model with low accuracy. An important feature of a neural net is that past information is not ignored; instead, its importance will be reduced (or strengthened) incrementally as new examples are fed into the network. In actual implementation, a sliding window scheme is needed to retain part of the old sample and combine it with the new sample to create a new training set. The exact proportion of old sample to be retained depends on the stability of the distribution and the level of noise in the sample.

Although the study reported here is far from sufficient to generate any conclusive statements about the applicability of neural nets in general, it does provide some insights into their potentials and limitations. Based on the comparison reported above, the neural-net approach offers a comparative alternative to classification techniques, especially under the following conditions:

(1) multi-modal distribution: the nonlinear discriminant function represented by a neural net provides a better approximation of the sample distribution, especially when

the latter is multi-modal. Many classification tasks have been reported to have a nonlinear relationship between variables. Whitred and Zimmer (1985) suggest that loan officers may have a higher prediction accuracy than linear DA models because of their ability to relate variables and loan outcome in a nonlinear manner. In another experiment conducted by Shepanski (1983), it was reported that human judgements are better approximated by a nonlinear function.

(2) Adaptive model adjustment: the ability to adaptively adjust the model is a virtue of a neural net. This allows the model to respond swiftly to changes in the real world.

(3) Robustness: the network does not assume any probability distribution or equal dispersion. There is also no rigid restriction on the use of input/output functions other than that they be continuous and differentiable.

Despite the successful applications of neural nets reported recently, their usage is still rather ad hoc. Some of their limitations are summarized below.

Network Topology. There is no formal method to derive a network configuration for a given classification task. Although it was shown that only one hidden layer is enough to approximate any continuous functions (Cybenko 1989, Funahashi 1989, and Hornik et al. 1989), the number of hidden units can be arbitrarily large. In addition, there is a possibility of overfitting the network. This problem arises when the number of hidden units is relatively large with respect to the size of the training sample (Baum and Haussler 1989). Unless the whole population is used for training, one has to be cautious in selecting the number of hidden units in order to avoid this problem. Currently, deciding how many hidden units to use is part of the modeling process itself.

Computational Efficiency. Training a neural net demands more computation time than the other methods. In the current study, computation time ranges from a few minutes to 3 hours on an EMX minicomputer. One strategy we have employed to reduce computation time is to allocate five different sets of weights and to restrict each run to an acceptable number of iterations. It seems to be an effective strategy in most cases, but inconsistent results are generated occasionally. All statistical methods took at most half a minute on an IBM 3081 mainframe. ID3 required on average 8 minutes on a Mac II microcomputer.

Explanatory Capability. The discriminant capability of a neural net is difficult to express in symbolic form. This may not be a serious drawback if one is concerned primarily with predictive accuracy. However, a neural net is limited if one wants to test the significance of individual inputs. There is no formal method to derive the relative importance of an input from the weights of a neural net.

Continuous improvements are being made in all these directions. Recently, Miller, Todd, and Hedge (1989) suggested applying genetic algorithms to the design of network configurations. Genetic algorithms adopt an evolutionary approach in which a pool of networks, called the population, is continuously being modified by using genetic operators such as crossover and mutation (Goldberg 1989). Each synthesized network, which corresponds to a possible configuration, is evaluated using the backpropagation algorithm. Genetic algorithms have a built-in bias towards retaining and combining good configurations in the next generation. The evolutionary nature of the algorithm enables the search for good configurations to proceed in a parallel fashion, thus reducing the possibility of trapping in local optimal configuration.

The problem of lengthy computation time is attributed to our implementation which is basically a simulation that runs on a serial machine. The intrinsic parallel processing capability of a network is not exploited in our study. Progress is underway to implement neural nets on silicon which will significantly reduce computation time (Mead 1988).

Heuristics have been developed to give the modeler some insights into the relative importance of variables with respect to a single example. A method using the partial

derivative of the error function has been suggested in Tam and Kiang (1990). In addition, the limited explanatory capability of neural nets can be explained by linking them with fuzzy logic. The latter provides a means of combining symbolic and numeric computations in inference processing. The linkage between neural nets and symbolic reasoning can be established through the membership function of fuzzy logic. The function measures the degree of possibility of a concept as related to a numeric quantity. A neural net can be used to synthesize a membership function by training it with instances of the relation. In our present case, a neural net may represent the membership function of the concept "high likelihood of failure." Such a representation can be easily combined with other symbolic conditions appearing in the rules of an expert system. By utilizing neural nets as frontends in rules definition, one can take advantage of the explanatory capability of expert systems as well as the subsymbolic computational capability offered by neural nets (Kosko 1990).

8. Conclusion

We have presented a new approach to bank bankruptcy prediction using neural nets. We believe neural nets can be extended to other managerial applications, particularly those involving classification. Furthermore, a neural net may supplement a rule-based expert system in real-time applications. While rule-based expert systems are satisfactory for off-line processing, a neural net-based system offers on-line capabilities. More work needs to be done in prototype development, and actual applications need to be empirically tested before the full potential of neural nets can be asserted.[7]

[7] The authors are grateful for the comments and suggestions of the Associate Editor and the four anonymous reviewers. They would like to thank William Cooper and Patrick Brockett for their valuable comments on an early draft of this paper.

APPENDIX I. Descriptive Statistics of Variables (One-Year Ahead)

	Nonfailed		Failed	
Name	Mean	St. Dev.	Mean	St. Dev.
capas	10.76	5.45	5.53	2.98
agfas	0.09	0.15	0.09	0.15
comas	0.29	0.16	0.35	0.16
indas	0.28	0.16	0.26	0.16
resas	0.36	0.16	0.34	0.16
pasln	0.01	0.01	0.05	0.07
nonln	0.01	0.02	0.08	0.06
losln	1.80	1.92	5.31	3.71
netln	1.45	1.72	4.18	3.51
rtoas	−0.14	2.10	−3.19	2.79
intdp	0.06	0.01	0.07	0.01
expas	0.09	0.02	0.12	0.02
incas	0.00	0.02	−0.04	0.03
infln	0.12	0.03	0.13	0.02
incex	1.11	0.19	1.00	0.17
curas	0.27	0.12	0.20	0.11
govas	0.28	0.15	0.17	0.10
llnas	0.53	0.14	0.67	0.11
llndp	0.61	0.16	0.71	0.18

APPENDIX II. Descriptive Statistics of Variables (Two-Years Ahead)

	Nonfailed		Failed	
Name	Mean	St. Dev.	Mean	St. Dev.
capas	9.87	4.03	7.87	2.54
agfas	0.08	0.14	0.10	0.16
comas	0.30	0.14	0.36	0.14
indas	0.28	0.15	0.25	0.14
resas	0.35	0.15	0.31	0.15
pasln	0.01	0.01	0.02	0.03
nonln	0.01	0.01	0.04	0.05
losln	1.16	1.15	3.01	2.99
netln	0.81	1.04	2.51	2.78
rtoas	0.69	1.44	−1.44	2.51
intdp	0.06	0.01	0.07	0.01
expas	0.09	0.01	0.11	0.02
incas	0.01	0.01	−0.01	0.03
infln	0.12	0.02	0.13	0.02
incex	1.16	0.12	1.05	0.15
curas	0.25	0.13	0.21	0.09
govas	0.28	0.15	0.17	0.09
llnas	0.55	0.14	0.66	0.10
llndp	0.62	0.17	0.73	0.11

References

ALTMAN, E. I., "Financial Ratios, Discriminant Analysis and the Prediction of Corporate Bankruptcy," *J. Finance*, 23, 3 (1968), 589–609.

———, R. A. EISENBEIS AND J. SINKEY, *Applications of Classification Techniques in Business, Banking, and Finance*, JAI Press, Greenwich, CT, 1981.

Bank of Texas, Vols. 1–3, Sheshunoff Information Services Inc, 1987.

BAUM, E. B. AND D. HAUSSLER, "What Size Net Gives Valid Generalization?," *Neural Comput.*, 1 (1989), 151–160.

BROOMHEAD, D. S. AND D. LOWE, "Multivariate Functional Interpolation and Adaptive Networks." *Complex Systems*, 2 (1988), 321–355.

CARTER, C. AND J. CATLETT, "Assessing Credit Card Applications Using Machine Learning," *IEEE Expert*, (Fall 1987), 71–79.

CYBENKO, G., "Approximation by Superpositions of a Sigmoidal Function," *Math. Control, Signals, and Systems*, 2 (1989), 303–314.

DEAKIN, E. B., "Distributions of Financial Accounting Ratios: Some Empirical Evidence," *Accounting Rev.*, (January 1976), 90–96.

DENKER, J., D. SCHWARTZ, B. WITTNER, S. SOLLA, R. HOWARD, L. JACKAL AND J. HOPFIELD, "Large Automatic Learning, Rule Extraction and Generalization," *Complex Systems*, 1 (1987), 877–922.

EISENBEIS, R. A., "Pitfalls in the Application of Discriminant Analysis in Business, Finance, and Economics," *J. Finance*, 32, 3 (1977), 875–900.

Federal Deposit Insurance Corporation, 1987 Annual Report.

FISHER, R. A. "The Use of Multiple Measurements in Taxonomic Problems," *Ann. Eugenics*, 7 (1936), 179–188.

FRYDMAN, H., E. ALTMAN AND D. KAO, "Introducing Recursive Partitioning for Financial Classification: The Case of Financial Distress," *J. Finance*, 40, 1 (1985), 269–291.

FUNAHASHI, K., "On the Approximate Realization of Continuous Mappings by Neural Networks," *Neural Networks*, 2 (1989) 183–192.

GOLDBERG, D. E., *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Reading, MA, 1989.

HANWECK, G. A., "Predicting Bank Failures," *Research Papers in Banking and Financial Economics*, Financial Studies Section, Board of Governors of the Federal Reserve System, Washington, DC, 1977.

HARRELL, F. E. AND K. L. LEE, "A Comparison of the Discrimination of Discriminant Analysis and Logistic

Regression under Multivariate Normality," in *Biostatistics: Statistics in Biomedical, Public Health, and Environmental Sciences*, (P. K. Sen, Ed.), North Holland, Amsterdam, 1985.

HINTON, G. E., "Connectionist Learning Procedures," *Artificial Intelligence*, 40 (1989), 185–234.

HORNIK, K., M. STINCHCOMBE AND H. WHITE, "Multilayer Feedforward Networks are Universal Approximators," *Neural Networks*, 2 (1989), 359–366.

KOROBOW, L. AND D. STUHR, "Performance Measurement of Early Warning Models," *J. Banking and Finance*, 9 (1985), 267–273.

KOSKO, B., *Neural Networks and Fuzzy Systems*, Prentice-Hall, Englewood Cliffs, NJ; 1990.

LACHENBRUCH, P. A., "An Almost Unbiased Method of Obtaining Confidence Intervals for the Probability of Misclassification in Discriminant Analysis," *Biometrics*, (December 1967), 639–645.

———, C. SNEERINGER AND L. REVO, "Robustness of the Linear and Quadratic Discriminant Function to Certain Type of Non-Normality," *Commun. Statist.*, 1, 1 (1973), 39–56.

MARTIN, D., "Early Warning of Bank Failure, A Logit Regression Approach," *J. Banking and Finance*, 1, 3 (1977), 249–276.

MEAD, C. A., *Analog VLSI and Neural Systems*, Addison-Wesley, Reading, MA, 1988.

MESSIER, W. F. AND J. HANSEN, "Inducing Rules for Expert System Development: An Example Using Default and Bankruptcy Data," *Management Sci.*, 34, 12 (1988), 1403–1415.

MEYER, P. A. AND H. PIFER, "Prediction of Bank Failures," *J. Finance*, 25 (September 1970), 853–868.

MILLER, G. F., P. TODD AND S. HEGDE, "Designing Neural Networks using Genetic Algorithms," *Proc. Third Internat. Conf. Genetic Algorithms*, Morgan Kaufmann, Palo Alto, CA, (1989), 379–384.

MINSKY, M. AND S. PAPERT, *Perceptrons*, MIT Press, Cambridge, MA, 1969.

MOODY, J. AND C. DARKEN, "Fast Learning in Networks of Locally-Tuned Processing Units," *Neural Comput.*, 1, 2 (1989), 281–294.

PAO, Y. H., *Adaptive Pattern Recognition and Neural Networks*, Addison-Wesley, Reading, MA, 1989.

PETTWAY, R. H. AND J. SINKEY, "Establishing On-Site Banking Examination Priorities: An Early Warning System Using Accounting and Market Information," *J. Finance*, 35, 1 (1980), 137–150.

PRESS, S. J. AND S. WILSON, "Choosing between Logistic Regression and Discriminant Analysis," *J. Amer. Statist. Assoc.*, 73 (1978), 699–705.

QUINLAN, J. R., "Discovering Rules by Induction from Large Collection of Examples," in *Expert Systems in the Micro Electronic Age*, (D. Michie, Ed.), Edinburgh University Press, Edinburgh, 1979.

———, "Learning Efficient Classification Procedures and Their Applications to Chess End Games," in *Machine Learning: An Artificial Intelligence Approach. Vol.* I, (R. S. Michalski, J. Carbonell, and T. Mitchell, Eds.), Tioga Publishing Company, Palo Alto, CA, 1983.

———, "Induction of Decision Trees," *Machine Learning*, 1 (1986), 81–106.

ROSENBLATT, F., *Principle of Neurodynamics*, Spartan, New York, 1962.

RUMELHART, D. E., G. HINTON AND R. WILLIAMS, "Learning Representation by Back-Propagating Errors," *Nature*, 323, 9 (1986), 533–536.

———, J. MCCLELLAND AND THE PDP RESEARCH GROUP (Eds.), *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, Bradford Books, MA, 1986.

SANTOMERO, A. M. AND J. VINSO, "Estimating the Probability of Failure for Commercial Banks and the Banking System," *J. Banking and Finance*, 1, 2 (1977), 185–205.

SHEPANSKI, A., "Tests of Theories of Information Processing Behaviour in Credit Judgement," *Accounting Rev.*, 58 (1983), 581–599.

SINKEY, J. F., "A Multivariate Statistical Analysis of the Characteristics of Problem Banks," *J. Finance*, 30, 1 (1975), 21–36.

TAM, K. Y. AND M. KIANG, "Predicting Bank Failures: A Neural Network Approach," *Appl. Artificial Intelligence*, 4, 4 (1990), 265–282.

———, ——— AND R. CHI, "Inducing Stock Screening Rules for Portfolio Construction," *J. Oper. Res. Soc.* (to appear).

TUCKER, L. W. AND G. ROBERTSON, "Architecture and Applications of the Connection Machine," *IEEE Computer*, (August 1988), 26–38.

WALTZ, D. L., "Applications of the Connection Machine," *IEEE Computer*, (January 1987), 85–97.

WEST, R. G., "A Factor-Analytic Approach to Bank Condition," *J. Banking and Finance*, 9, 2 (1985), 253–266.

WHITRED, G. AND I. ZIMMER, "The Implications of Distress Prediction Models for Corporate Lending," *Accounting and Finance*, 25 (1985), 1–13.