Robert E. Dorsey
University of Mississippi
Robert O. Edmister
University of Mississippi
John D. Johnson
University of Mississippi

Bankruptcy Prediction Using Artificial Neural Systems

The Research Foundation of
The Institute of Chartered Financial Analysts

Research Foundation Publications

Time Diversification Revisited
by William Reichenstein, CFA, and Dovalee Dorsett

Analysts' Earnings Forecast Accuracy in Japan and the United States
by Robert M. Conroy, Robert S. Harris, and Young S. Park

Active Currency Management
by Murali Ramaswami

Canadian Stocks, Bonds, Bills, and Inflation: 1950–1987
by James E. Hatch and Robert E. White

Closed-Form Duration Measures and Strategy Applications
by Nelson J. Lacey and Sanjay K. Nawalkha

Corporate Bond Rating Drift: An Examination of Credit Quality Rating Changes Over Time
by Edward I. Altman and Duen Li Kao

Corporate Governance and Firm Performance
by Jonathan M. Karpoff, M. Wayne Marr, Jr., and Morris G. Danielson

Default Risk, Mortality Rates, and the Performance of Corporate Bonds
by Edward I. Altman

Durations of Nondefault-Free Securities
by Gerald O. Bierwag and George G. Kaufman

Earnings Forecasts and Share Price Reversals
by Werner F.M. De Bondt

The Effect of Illiquidity on Bond Price Data: Some Symptoms and Remedies
by Oded Sarig and Arthur Warga

Equity Trading Costs
by Hans R. Stoll

Ethics, Fairness, Efficiency, and Financial Markets
by Hersh Shefrin and Meir Statman

Ethics in the Investment Profession: A Survey
by E. Theodore Veit, CFA, and Michael R. Murphy, CFA

Ethics in the Investment Profession: An International Survey
by H. Kent Baker, CFA, E. Theodore Veit, CFA, and Michael R. Murphy, CFA

The Founders of Modern Finance: Their Prize-Winning Concepts and 1990 Nobel Lectures

Franchise Value and the Price/Earnings Ratio
by Martin L. Leibowitz and Stanley Kogelman

Fundamental Considerations in Cross-Border Investment: The European View
by Bruno Solnik

Global Asset Management and Performance Attribution
by Denis S. Karnosky, and Brian D. Singer, CFA

Initial Public Offerings: The Role of Venture Capitalists
by Joseph T. Lim and Anthony Saunders

Managed Futures and Their Role in Investment Portfolios
by Don M. Chance, CFA

The Modern Role of Bond Covenants
by Ileen B. Malitz

A New Method for Valuing Treasury Bond Futures Options
by Ehud I. Ronn and Robert R. Bliss, Jr.

A New Perspective on Asset Allocation
by Martin L. Leibowitz

Options and Futures: A Tutorial
by Roger G. Clarke

The Poison Pill Anti-Takeover Defense: The Price of Strategic Deterrence
by Robert F. Bruner

A Practitioner's Guide to Factor Models

Predictable Time-Varying Components of International Asset Returns
by Bruno Solnik

Program Trading and Systematic Risk
by A.J. Senchack, Jr., and John D. Martin

The Role of Risk Tolerance in the Asset Allocation Process: A New Perspective
by W.V. Harlow III, CFA, and Keith C. Brown, CFA

Selecting Superior Securities
by Marc R. Reinganum

Stock Market Structure, Volatility, and Volume
by Hans R. Stoll and Robert E. Whaley

Bankruptcy Prediction Using Artificial Neural Systems

This publication is designed to provide accurate and authoritative information in regard to the subject matter covered. It is sold with the understanding that the publisher is not engaged in rendering legal, accounting, or other professional service. If legal advice or other expert assistance is required, the services of a competent professional should be sought.

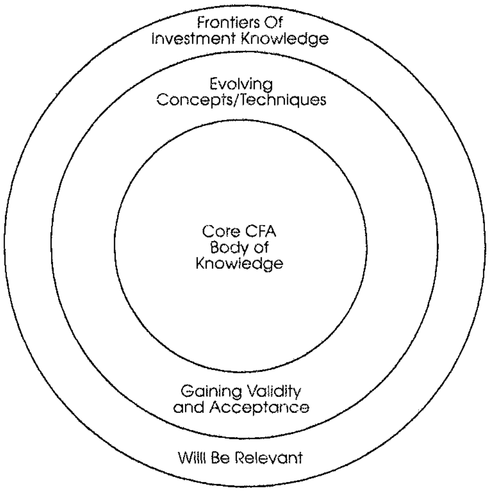
ISBN 978-0-943205-31-1

Printed in the United States of America

June 1995

Mission

The Research Foundation's mission is to identify, fund, and publish research that is relevant to the CFA® Body of Knowledge and useful for AIMR member investment practitioners and investors.



The Research Foundation of
The Institute of Chartered Financial Analysts
P.O. Box 3668
Charlottesville, Virginia 22903
U.S.A.
Telephone: 804/980-3644
Fax: 804/980-3634

Table of Contents

Foreword

Classification models have been a standard part of the financial analyst's tool kit for several decades. Over the years, applications of empirical prediction systems have involved both stocks and bonds (i.e., establishing quality ratings) and derivative securities (i.e., the exercise decision on convertible debt). Nowhere have these models been applied more frequently and successfully, however, than in the evaluation of financial distress. From separating bank loans into default and nondefault probability classes to dividing publicly traded debt issues into investment and noninvestment grades, analysts in the credit markets have had a long association with quantitative techniques for assessing the likelihood of corporate bankruptcy.

The usual approach taken in research of this kind is to collapse firm-specific accounting measures, such as debt-to-equity ratios and times-interest-earned multiples, into statistics that allow companies to be categorized as those that will eventually fail and those that will not. Success is then measured by contrasting these *ex ante* classifications with the actual occurrence (or nonoccurrence) of a default event. Historically, these prediction models have been straightforward linear functions mapping a collection of credit-related independent variables into a single prediction measure. The earliest research along these lines started with straightforward multiple logistic regression equations and quickly progressed through discriminant analytic procedures. These models, however, proved to have both conceptual and practical deficiencies. A recent breakthrough came in the form of the recursive partitioning algorithm (RPA), which is a computer-based, nonparametric technique used to recognize data patterns.

In this monograph, Robert Dorsey, Robert Edmister, and John Johnson take this progression to the next level. In particular, they adapt two state-of-the-art artificial intelligence techniques to the purpose of bankruptcy prediction. The primary weapon they use is the artificial neural network (ANN), which is a computer model that can be trained to mimic the cellular connections in the human brain. That is, by its processing and evaluation of the interactions in a complex set of prior data, a "neural net" attempts to assign the proper weights to the respective inputs so as to allow for the correct deduction of the ultimate outcome. The assignment of these input weights is aided by an optimization procedure known as a genetic algorithm (GA), which simulates the model's predictive power under myriad scenarios and allows the best weighting schemes to "survive and reproduce" from one generation to the next. Although the technological demands of these techniques have inhibited their practical application until recently, the authors have contributed extensively to the embryonic literature in this area and are uniquely qualified to conduct this research.

At its most basic level, this study is a simple comparison of the predictive abilities of the ANN and RPA methodologies. Perhaps not surprisingly, Dorsey, Edmister, and Johnson conclude that their model does a superior job of classifying both bankrupt and nonbankrupt firms during the years prior to the event. Although their evidence does provide strong corroboration for this finding, this comparison is not the most compelling aspect of the study for two reasons. First, the juxtaposition of the ANN and RPA procedures, although conducted using contemporaneous data, masks the fact that the former is considerably more difficult to implement in practice than the latter, even with the computer programs that the authors have made available. Second, the predictive power of the neural net model fluctuates widely across the periods examined; these results are not likely to be directly applicable to the current market environment.

A more important role the authors serve, therefore, is to provide the reader—perhaps for the first time—with an in-depth explanation of the mechanics of the ANN and GA technologies, as well as the way they can be applied to financial/economic problem solving. Dorsey, Edmister, and Johnson do this in two ways. First, in an introductory section of the monograph, they provide a thorough, nontechnical explanation of their methodology using both words and illustrations. Second, in the appendix, they offer a more technically oriented discussion of how their methodology functions. This material is neither easy nor transparent, but it is definitely worthwhile reading on a general approach that is gaining rapid acceptance as a mainstream predictive tool. The Research Foundation is pleased to be able to bring it to your attention.

Keith C. Brown, CFA
Research Director

1. Introduction

In this study, we explored the applicability of a form of artificial intelligence, the artificial neural network (ANN), for predicting bankruptcy of large firms.[1] The emphasis of our study was prediction of firm failure or success based on accounting reports, but the same techniques perhaps could be used to reveal mispriced bonds. We found that neural net models appear to predict which nonbankrupt firms among those priced in the secondary bond market are likely to become bankrupt, thus creating profit opportunities for investors. Substantial risk remains after prediction with the ANN model, although it dominates the recursive partitioning algorithm with respect to predictive accuracy. Other significant findings from this research are the following:

- Single-year models fit the estimation data very well but tend to perform less well in subsequent years.
- Bankrupt firms in 1990 are markedly different from bankrupt firms in 1989 and 1991.
- Bankruptcies occurring in 1990 are far less predictable than those in 1989 and 1991.
- Bankruptcies occurring in 1991 are generally predictable two years in advance with the 1989 and 1989–91 models.

Chapter 2 presents a review of previous bankruptcy-prediction models, with particular emphasis on the Frydman, Altman, and Kao (1985) model, a state-of-the-art model that provides a bench test for the ANN model developed in this research. We discuss the limitations of existing models and propose a solution offered by ANN models. ANN models present a difficult research

[1] The bankruptcy prediction program, all data used in this monograph, and an Adobe Acrobat version of the monograph are available via anonymous ftp to sunset.backbone.olemiss.edu/ in the /pub/business subdirectory. World Wide Web users point to http://www.olemiss.edu, which will put you at the University of Mississippi home page. From the home page, select "Departmental Pages" followed by "School of Business." From the business school home page, go to "Research in Computational Business," where you will find the bankruptcy prediction program page. If you do not have an Acrobat viewer, information on obtaining one at no cost is available. Any questions that cannot be addressed by your local Internet provider can be directed via E-mail to Dr. John D. Johnson at johnson@bus.olemiss.edu.

problem in parameter estimation, the solution to which is described generally in Chapter 2 (and rigorously in the Appendix to this volume).

Chapter 3 contains a discussion of the comparison of the ANN's results against those of the recursive partitioning algorithm from Frydman, Altman, and Kao.

Prediction evaluations for a new data set are presented in Chapter 4. In-sample Type I and II error-rate graphs show that the ANN is excellent for sample reclassification. For classifications of samples not used for estimation, ANN models—like all statistical models—deteriorate as market conditions change. The most accurate models are those estimated with data observed over the longest time frame. We combined data for 1989 to 1991 to create a relatively long observation period. The 1989–91 model is presented at the end of Chapter 4.

Chapter 5 presents an investment application of ANN models. Bankruptcy predictions are combined with yield premiums to provide indications of over- and underpriced bonds. The example bond results show how investors might undertake analyses intended to identify investment opportunities.

2. Prediction Models

The prediction of financial distress through statistical techniques has improved continuously since Beaver (1966) introduced univariate statistics three decades ago. Milestones in the development of models for nonfinancial-firm distress prediction include multivariate analysis (Altman 1968), multivariate discriminant analysis (MDA) with trend factors (Edmister 1972), logit (Ohlson 1980), and the recursive partitioning algorithm (RPA) in Frydman, Altman, and Kao (1985, hereafter denoted FAK).

The advantage of RPA relative to MDA and logit is that RPA incorporates interactions among variables. Combinations of variables and threshold values are evaluated at a large number of potential branching points. In the RPA technique, all the possible combinations are evaluated and those with predictive power are selected. The result is a parsimonious decision model in the form of a tree.

The financial ratios analyzed in FAK encompass the set of ratios previous studies had found to be the best financial distress predictors. These ratios are compared in Table 1. Some of the ratios used in other studies, listed at the bottom of Table 1, were not used directly in the FAK model but have a FAK counterpart that provides similar information. Because FAK was the most comprehensive of the previous models, we focused on its ratio list for the purpose of developing the ANN model of financial distress prediction. This choice also allows direct comparison with the RPA.

Problems with Traditional Methods

Generally, econometricians face a difficult problem when they attempt to identify and estimate an appropriate model. In distress prediction, the only consensus they have reached on the appropriate functional form is that the appropriate model is complex. One true model to predict bankruptcy in firms is impossible or unattainable for some of the following reasons:

Inadequacy of existing estimation techniques. Estimation techniques operate as a constraint that prevents econometricians from identifying and estimating optimal bankruptcy models. The choice of an estimation technique, more often than not, is based on ease of calculation rather than a consideration of

Table 1. Financial Information and Ratios Used in Prior Studies

Ratio	FAK	ROE	EIA	WAB	EBD	JAO	CVZ
Cash flow/total assets	✓				✓		✓
Cash/total sales	✓				✓		✓
Cash flow/total debt	✓			✓	✓	✓	
Current assets/current liabilities	✓			✓	✓	✓	
Current assets/total assets	✓				✓		
Current assets/total sales	✓				✓		
EBIT/total assets	✓		✓				
Retained earnings/total assets	✓		✓				
Net income/total assets	✓			✓	✓	✓	
Total debt/total assets	✓			✓	✓		
Total sales/total assets	✓		✓				
Working capital/sales	✓				✓		
Working capital/total assets	✓		✓	✓	✓	✓	
Quick assets/total assets	✓				✓		
Quick assets/current liabilities	✓	✓			✓		✓
Quick assets/total sales	✓				✓		
Equity mkt. value/total capitalization	✓						
ln(total assets)	✓						
ln(interest+15)	✓						
Cash/current liabilities		✓			✓		
Current liabilities/equity		✓					
Inventory/sales		✓					
Equity/sales		✓					
Equity market value/total debt			✓				
Income/total capitalization							✓

Note: The abbreviations in the column headings refer to the following studies: FAK—Frydman, Altman, and Kao (1985); ROE—Edmister(1972); EIA—Altman (1968); WAB—Beaver (1966); EBD—Deakan (1972); JAO—Ohlson (1980); CVZ—Zavgren (1983).

any "true" functional relationship. For instance, in single-equation models, linearity is often imposed, not because a thorough investigation has been conducted with respect to the true relationship between the dependent and independent variables but because a feasible estimation procedure is readily available.[2]

[2] See Caporaletti, Dorsey, Johnson, and Powell (1994).

Unrealistic restriction of error terms. Standard parametric estimation methods are usually subject to underlying distribution assumptions on the population. Many studies, however, have found that the financial variables used for forecasting insolvency do not conform to the standard distribution assumptions.

Insufficiency of ratio transformations. Accounting ratios do not adequately represent contingent interrelationships. Nonparametric methods, which may not require distribution assumptions, are sometimes difficult to interpret and are often problem specific.

Limitations of expert systems. Expert systems (ES) attempt to capture the essence and thus mimic the decision-making ability of human experts. Although ES have been successfully applied to many important decision-making tasks, many other tasks are beyond their scope because of their limitations. These limitations include the difficulty of programming and maintaining the system (the Fiegenbaum, or programming, bottleneck), the inability of an ES to use inductive learning and inference to adapt to changing relationships in the decision environment (the learning problem), and the enormous amount of time and effort required to extract the knowledge base from human experts and translate it into the "if–then" rules upon which an ES is based (the knowledge-engineering bottleneck).[3]

Artificial Neural Networks

The functional form used in this study was generated by using a multilayered feedforward ANN. ANNs are simplified models of the interconnections between cells of the brain. Wasserman and Schwartz (1988) defined them as "highly simplified models of the human nervous system, exhibiting abilities such as learning, generalization, and abstraction." Such models were developed in an attempt to examine how the brain processes information. These models have, in concept, been in existence for many years, but the computer hardware requirements of even the most rudimentary systems exceeded existing technology.[4] Recent technological advances, however, have made ANN models a viable alternative for many decision problems, and they have the potential for improving the models of many financial activities such as forecasting financial distress in firms.[5]

The ANN has been shown to approximate any Borel measurable functional

[3] See Hawley, Johnson, and Raina (1990).

[4] See Hawley, Johnson, and Raina (1990).

[5] A general description of neural networks is found in Rummelhart, Hinton, and Williams (1986a and b).

mapping from input to output at any degree of desired accuracy if sufficient hidden layer nodes are used.[6] The Borel measurable functional mapping is sufficiently general to include linear regression, logit, and RPA models as special cases. ANNs are also free of distributional assumptions, avoid problems of colinearity, and are a general model form (or universal approximator).

Consequently, a financial analyst familiar with the structure of a problem selects only the proper inputs and outputs for an ANN model. The weights assigned to each input and the functional form of each of the relationships are determined by the neural network, as opposed to the expert's (e.g., statistician's) explicit a priori assumptions.[7]

With regard to the specification of the functional form, the neural network does not impose restrictions such as linearity because it "learns" the underlying functional relationship from the data themselves, thus minimizing the a priori nonsample information that is required. Indeed, a major justification for the use of a neural network as a completely general estimation device is its function-approximation abilities—that is, its ability to provide a generic functional mapping from inputs to outputs—thus eliminating the need for exact prior specification. With a neural network, a financial analyst has a tool that can aid in function approximation in the same way a spreadsheet aids "what if" analysis.[8] This capability is a major advantage of ANNs in bankruptcy applications.

Because of the function-approximation ability of the ANN, one can compute *any continuous function* using linear summations and a single, properly chosen nonlinearity.[9] In other words, the arrangement of the simple nodes into a multilayer framework produces a mapping between inputs and outputs consistent with any underlying functional relationship regardless of its true functional form. The importance of having a general mapping between the input and output vectors is evident: It eliminates the need for the unjustified a priori restrictions so commonly used to facilitate estimation (e.g., the Gauss–

[6] See Hornik, Stinchcombe, and White (1989).

[7] See Caporaletti et al. (1994).

[8] See Hawley, Johnson, and Raina (1990).

[9] The most commonly cited proof of the function-approximation ability of an ANN is the superposition theorem of Kolmogorov (1957) and its improvements by Sprecher (1965); Lorentz (1976); and Hornik, Stinchcombe, and White (1989). The connection between these results and ANNs has been pointed out by Hecht-Nielsen (1987). Hecht-Nielsen (1990) also discussed several function-approximation results of the ANN.

Markoff assumptions in regression analysis).[10] Also, without the a priori restrictions, decision makers are allowed to involve, to a greater extent, their decision-making expertise (or intuition) in the analysis of problems.

A neural network can approximate arbitrary nonlinear functions to any degree of desired accuracy, given a sufficiently large number of hidden layer nodes. The number of nodes need not be very large, however. Dorsey, Johnson, and Mayer (1994) and Gallant and White (1992), among others, have shown that very complex functions (e.g., chaotic series) can be approximated with a high degree of accuracy by using five or fewer hidden nodes.

The ANN used in this bankruptcy-prediction project is an extension of the perceptron of Rosenblatt (1958), which is a very simple artificial neuron structure, as illustrated in Figure 1. This structured node sums the weighted inputs from its neighbors, compares this sum with its threshold value, θ, and passes the result through a function referred to as an interaction rule. The value of a typical node, Y, is given by

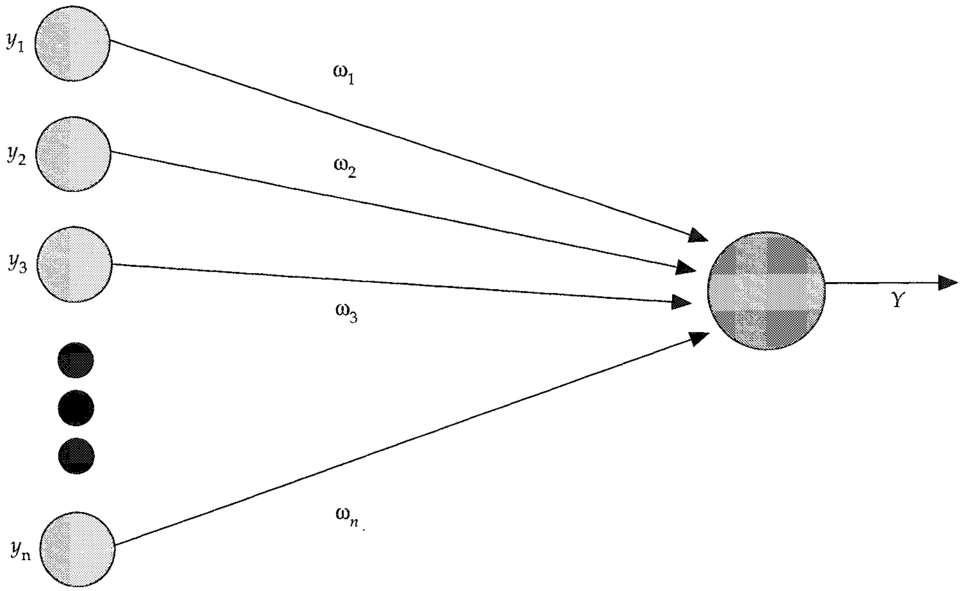
$$Y_k = \frac{1}{\left[1 + e - \left(\sum_{j=1}^{n} w_{jk} y_j - \theta_k\right)\right]}, \tag{1}$$

where θ is the threshold activation level, known as the offset. As is shown in Figure 1, each node, y, can be represented as a function of n weighted inputs. These perceptrons can be arranged in multiple, fully interconnected layers, producing a multilayered perceptron as illustrated in Figure 2. In this network, the input nodes are linked to the output nodes through one or more interconnected hidden layers. The multilayered perceptron is referred to as a feedforward network because inputs are fed into the bottom (or input) layer and propagate forward through the network topology to the output layer.

Although a number of different structures and transfer functions have been proposed, we used a single-hidden-layer feedforward structure similar to the one shown in Figure 2. The bottom layer of nodes represents the observations on the input variables. At each hidden layer node, a weighted sum of the inputs is computed, and the output from the hidden layer node is a nonlinear transformation of this weighted sum. The logistic function we used is de-

[10] Note that if these assumptions hold, the neural network model will yield a similar solution, because the image of any underlying mapping can always be projected into a perfectly flexible mapping. The appropriateness of ordinary least squares is an empirical question that cannot be settled in general for any finite number of observations. Thus, a test of the assumptions must become a routine part of any potential application.

Figure 1. A Typical Artificial Neuron



scribed in Equation 1. Thus, each line in Figure 2 connecting the nodes represents a weight, ω_{jk}. The output from the *k*th hidden layer node is given by Equation 1, where *j* is the number of input variables. The output from the network is the weighted sum of the outputs from the hidden nodes.

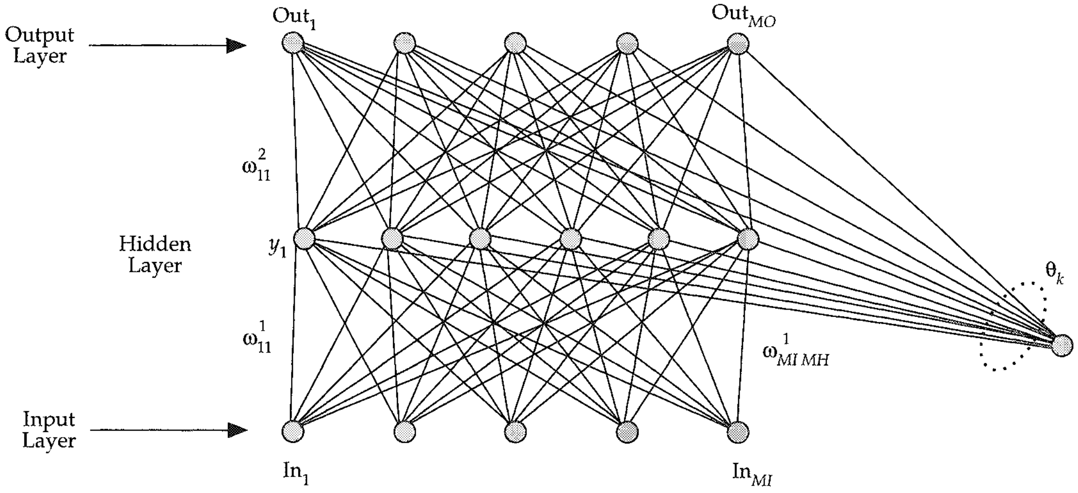
Training Methodologies

The function-approximation ability of the ANN provides a method for making forecasts of future financial events such as financial distress within certain firms. If properly optimized, the ANN should provide a method for making forecasts of future financial events that is more reliable than previous methods.

A primary difficulty with using ANN models has been the lack of a means for correctly optimizing the network. Virtually all researchers are currently using the backpropagation algorithm or a variation of it.[11] Current research at the University of Mississippi has demonstrated that the backpropagation algorithm is highly prone to stopping at a suboptimal location. An alternative

[11] Traditionally, ANNs are trained using the backpropagation training algorithm of Werbos (1974), Parker (1985), LeCun (1986), and Rumelhart, Hinton, and Williams (1986a, 1986b).

Figure 2. The Multilayered Perceptron Net



algorithm, the genetic algorithm, has been adapted for optimizing the ANN, and it achieves the global optimum more consistently than does backpropagation.

Problems with the backpropagation training algorithm have been outlined by Wasserman (1989) and Hecht-Nielsen (1990). These problems include the tendency of the network to become trapped in local optimums, to suffer from network paralysis as the weights move to higher values, and to become temporally unstable—that is, to forget what it has already learned as it learns a new fact. Because the flexibility theorems (mapping and function approximation) depend upon the selection of the proper weights, the utility of backpropagation as a learning rule for producing a flexible mapping is questionable. Therefore, we used a neural network training algorithm based on a modified version of the genetic algorithm.

The genetic algorithm, first proposed by Holland (1975), is a global-search algorithm that continuously samples from the total parameter space while focusing on the best solution so far. It is loosely based on genetics and the concept of survival of the fittest. The optimization process involves determining the set of weights to be used for the interconnections. Dorsey, Johnson, and Mayer (1994) demonstrated that the error surface for the ANN is frequently characterized by a large number of local optimums. Thus, derivative-based search techniques such as the commonly used backpropagation algorithm are subject to becoming trapped at local solutions. Dorsey and

Mayer (1994, 1995) showed that the genetic algorithm can be used as a global search algorithm on a wide variety of complex problems and that it achieves a global solution with a high degree of reliability. We followed the protocol developed by Dorsey, Johnson, and Mayer (1994) and used the genetic algorithm for optimization of the neural network.[12]

Because the genetic algorithm does not use the derivative of the network output to adjust its weight matrixes, as do gradient methods (e.g., the backpropagation training algorithm), the derivative (of the objective function) need not exist, and thus the network can use any objective function.[13] This property also implies that the network paralysis problem can be overcome. The paralysis problem occurs with backpropagation as the node outputs are forced to their extremes, forcing the weight adjustments to become increasingly smaller and thus paralyzing the network. Temporal instability is overcome because the network is trained in a batch mode; that is, weights are only changed at the end of each complete sweep through the data. In addition, the network is less likely to become trapped in a local optimum, because the genetic algorithm provides a global search. Dorsey, Johnson, and Mayer (1994) empirically showed that the genetic algorithm performs very well on a large class of problems with generic network architectures. In fact, they used one hidden layer and six hidden layer neurons for each problem. Thus, they demonstrated that the genetic-algorithm-based training method for the selection of the appropriate weight matrixes overcomes the shortcomings of backpropagation and can achieve the desired flexibility.

The training of the neural network begins when a population of candidate solutions is randomly chosen. Each candidate solution is a vector of all the weights for the neural network. For this study, the population consisted of 20 vectors. The weights constituting each vector are sequentially applied to the neural network, and outputs are generated for each observation of the inputs. Outputs are then compared to known values in the data set, and a sum of squared errors is computed for each vector of weights.

The sum of squared errors represents how well each candidate vector does at modeling the data and is used to compute its fitness value. A probability measure is then computed for each vector based on the vector's fitness value. The smaller the sum of squared errors, the larger the fitness value relative to the other vectors and the larger the probability measure. A new population is created by selecting 20 vectors from the former population. The selection is

[12] For a detailed discussion of the genetic algorithm used for global optimization, see Dorsey and Mayer (1994, 1995).

[13] See Dorsey and Mayer (1994, 1995).

made with replacement, and the probability that any particular vector is selected is based on its probability measure. Thus, those vectors that generate the lowest sum of squared errors will be replicated more often in the next generation. The vectors of the new population are then randomly paired. A point along the vector is randomly chosen for each pair. The pairs are broken at that point, and the upper portion of each pair of vectors is swapped to form two new vectors, each with elements from the original vectors.

The final operation before applying this new set of vectors to the neural network and repeating the above process for another generation is mutation. Each element of each vector of the new population has a small probability of mutating. Should mutation occur, the element is replaced with a random value drawn uniformly from the parameter space. The process of mutation allows the genetic algorithm to escape a local maximum and move to another area of the error surface. After mutation, fitness values are computed for the new population of vectors and the process is repeated. The complete process is repeated for thousands of generations and terminates when improvement in the sum of squared errors diminishes. This process can be summarized in the following steps:

Step 1: Generation of initial population. Values are randomly drawn for the weights to be used in the neural network. Each set of values makes up a single vector. A population of 20 such vectors constitutes the initial population.

Step 2: Calculation of error. For each of the 20 weight vectors (strings), the training input (data) vectors are fed into the network and the ANN's corresponding output vectors (estimates) are compared with the training (or target) output vectors. An error value (sum of squared errors) is calculated for each of the 20 strings.

Step 3: Reproduction. Each of the 20 vectors is assigned a selection probability, which is inversely proportional to its error value calculated in Step 1 above. A new set of 20 weight vectors is selected from the 20 old strings. Each of the 20 old strings has a probability of being selected (with replacement) into the new set.

Step 4: Crossover. The 20 new weight vectors are randomly organized into ten pairs. For each pair, one of the elements of the vector is randomly selected. At this element, each of the vectors of the pair is broken into two fragments. The pair then swaps the vector fragments.

Step 5: Mutation. Whether any element of the 20 vectors should be changed is randomly determined. For each element of the 20 weight vectors, a random number is selected and a Bernoulli trial is conducted. If the Bernoulli trial is successful (with probability equal to the mutation rate) then the element is replaced with the random number; otherwise, the element remains unchanged. This procedure is carried out for every element of every weight vector. With the resultant 20 weight vectors, or new generation, the calculation of error begins again.

As in natural systems, the new offspring inherit a combination of parameters (traits) from their parents. The key to this process is selectivity. Not all population members from the previous generation are given an equal chance of producing progeny to fill the pool of the present or future population of possible solutions. Thus, only a select few population members are likely to contribute; those with the highest probability of surviving to the next generation possess parameters favorable to solving for the optimum of the specific objective function, and those least likely to survive to the next generation possess parameters that yield unfavorable solutions. In this way, a new population of candidate solutions (the second generation) is built from the most desirable parameters of the initial population. As iteration continues from one generation to the next, parameters most favorable in finding an optimal solution for the objective function thrive and grow, while those least favorable die out.

Mutation may also occur at any stage of the progression from one generation to the next. By randomly introducing new parameters into the natural-selection process, mutation tests the robustness of the population of possible solutions. As with parameters included in the vectors of the initial population, if these newly introduced parameters add favorably to the ability of their recipients to optimize the specific objective function, the new parameter will survive and grow. Otherwise, the effect of the mutation will die out. Eventually, the initial population evolves to one that contains an optimal solution and the evolutionary process terminates.

As the decision surface is selected, a trade-off always arises between Type I errors (classifying bankrupt firms as nonbankrupt) and Type II errors (classifying nonbankrupt firms as bankrupt). Traditional methods treat this trade-off by modifying the costs of the types of errors. We will show in Figure 4 that this trade-off can be seen directly on the output of the neural network by adjusting the threshold of the output. As can be seen in the figures in Chapter 4, varying the threshold value above which the firm will be forecast as bankrupt changes the number of Type I and Type II errors directly. Judging the appropriate level of error is discussed in Chapter 4.

3. Prediction Accuracy

Two primary goals of this study were to compare the neural network with a well-known financial distress forecasting tool and to evaluate it from the perspective of a financial analyst. To benchmark the accuracy of the ANN, we compared it with the RPA using data reported in FAK. We found that the ANN model compares favorably with RPA, as measured by a reduced reclassification error rate. The FAK error rates compared very favorably with those of all previously reported studies. FAK reported that the RPA, a nonparametric technique, significantly outperforms multiple discriminant analysis (MDA). In turn, MDA outperforms multiple regression for failure prediction. Thus, RPA provides a powerful benchmark for evaluating the ANN.

The FAK Study

FAK were the first to apply the RPA to bankruptcy prediction. RPA was a significant improvement in methodology relative to parametric methods because it relaxed the restrictions imposed by parametric estimators. FAK used 20 financial ratios for 200 firms. Of those firms, 58 were bankrupt and 142 were not bankrupt.

FAK reported two examples of parsimonious trees that resulted from the RPA analysis. In the smaller tree, they were able to use four financial ratios to classify the firms. These ratios are shown in Table 2. Because the ANN is a completely flexible mapping, it should be able to at least match the classification performance of RPA for any set of variables. We therefore used the four variables identified in the FAK model to train the ANN.

FAK limited reporting classification accuracies in the four- variable model to misclassification cost weights of 50 to 1.[14] For this case, the errors they reported were 5 bankrupt firms misclassified as nonbankrupt and 15 nonbankrupt firms misclassified as bankrupt. Thus, the reclassification error rate for firms predicted nonbankrupt was 3.8 percent (5 of 132), and for firms the model predicted to be bankrupt, the error rate was 22.1 percent (15 of 68). These error rates are remarkably low for a severely restricted (four-ratio) model.

[14] Frydman, Altman, and Kao (1985), Figure 1, p. 272.

Table 2. FAK Ratios in Four- and Six-Variable Models

Ratio Definition	Four-Variable Model	Six-Variable Model
Cash flow/total debt	✓	✓
Retained earnings/total assets	✓	
Cash/total sales	✓	✓
Total debt/total assets	✓	✓
Market value of equity/total capitalization		✓
Log (interest coverage +15)		✓
Quick assets/total assets		✓

Estimation and Validation Data

By using the FAK data and comparing the results for the ANN with the results the RPA achieved, we provided the strongest possible test for the neural network. To evaluate the neural network's ability to generate useful forecasts, we created additional data sets for three successive years, 1989, 1990, and 1991. The financial information was taken from *Compact Disclosures*™. The data sets consist of financial ratios for the year before and an indicator of whether the firm failed in the given year; for example, the 1989 data set consists of firms' financial ratios for the year ending December 31, 1988, and an indicator of whether they failed in 1989. These data replicate the problem financial analysts face: using current financial information to predict financial distress in the coming year. We selected the *Compact Disclosures* data base, in part, because of its ready availability to financial analysts.

We searched for companies that were in financial distress, defined as a firm that has entered bankruptcy under chapters 7 or 11 of the U.S. Bankruptcy Code, and excluded those for which financial data for the previous year were unavailable. We then randomly selected nonbankrupt firms—approximately 50 percent more nonbankrupt firms than bankrupt firms—eliminating those with incomplete data. The final data sets had slightly larger numbers of nonbankrupt firms than bankrupt firms.

Table 3 shows the variables we collected from the data base to construct the data set for this study. These variables were used to generate the financial ratios listed in Table 4.

Compact Disclosures frequently codes information as "NA," or not available. We changed all NA entries to zero because we believed that this designation refers to nonexistent accounts rather than unknown values. For example, preferred stock is NA because the firm does not issue preferred stock; hence, the value of preferred stock is zero.

Table 3. Variable Codes and Names

CODE	*Compact Disclosures* CD Variables	Sample Program Title
CO	Company name	Name
CH	Cash	Cash
IV	Inventories	Inventories
CA	Total current assets	Total current assets
TA	Total assets	Total assets
LI	Total current liabilities	Total current liabilities
TL	Total liabilities	Total liabilities
SA	Net sales	Net sales
IF	Interest expense	Interest expense
IB	Income before tax	Income before tax
NI	Net income	Net income
KA	C/F operating income(loss)	C/F operating income
JW	Net sales/ Working capital	Net sales/ Working capital
RT	Retained earnings	Retained earnings
SE	Shareholder equity	Shareholder equity

The denominator of some ratios is zero because *Compact Disclosures* reports either a zero value or NA. Consequently, some ratios are not finite. In this study, observations with infinite ratio values were infeasible and were therefore deleted. Possible sources of infinite values are ratios with the following variables in the denominator: total assets, total current liabilities, total liabilities, net sales, and working capital. Nonfeasible observations were relatively infrequent.

The FAK–ANN Comparison

We compared the ANN and FAK classification accuracies by controlling for two types of errors. In the first, ANN I, the neural net matches the FAK Type I errors, and in the second, ANN II, the neural net matches the FAK Type II errors. The results of the four-factor comparison are shown in Table 5. In our estimation, the nearest ANN I match is four errors for ANN and five errors for FAK. In this case, the ANN incorrectly classified four bankrupt companies as nonbankrupt and four nonbankrupt companies as bankrupt. Thus, the ANN Type I error rate is 6.9 percent (4 of 58), a favorable comparison with the FAK rate of 22.1 percent.

Table 4. Data Set Ratio Definitions

Ratio	Definition
CASH/TA	Cash/Total assets
CASH/TS	Cash/Net sales
CF/TD	C/F operating income/Total liabilities
CA/CL	Total current assets/Total current liabilities
CA/TA	Total current assets/Total assets
CA/TS	Total current assets/Net sales
EBIT/TA	(Interest expense + Income before tax)/Total assets
LOG(INT+15)	LOG((Interest expense + Income before tax)/Total assets + 15)
LOG(TA)	LOG(Total assets)
MVE/TK	Shareholder equity/(Total assets – Total current liabilities)
NI/TA	Net income/Total assets
QA/CL	(Total current assets – Inventories)/Total current liabilities
QA/TA	(Total current assets – Inventories)/Total assets
QA/TS	(Total current assets – Inventories)/Net sales
RE/TA	Retained earnings/Total assets
TD/TA	Total liabilities/Total assets
TS/TA	Net sales/Total assets
WK/TA	(Total assets/Net sales)/(Working capital/Total assets)
WK/TS	1/(Net sales/ Working capital)

The nearest ANN II match is 9 errors for ANN and 15 errors for FAK. Indeed, the largest number of errors on bankrupt firms is nine for ANN, irrespective of the number of Type I errors. In this case, ANN misclassified one bankrupt firm as nonbankrupt and nine nonbankrupt firms as bankrupt. The ANN Type I error rate is 0.7 percent (1 of 142), much lower than the FAK rate of 3.8 percent.

In summary, after controlling for Type I and II errors, the ANN model generated far fewer errors than the FAK model. The ANN model, restricted to four variables, reduced prediction errors for bankruptcy by 73 percent (11 of 15) relative to the FAK model. Furthermore, the ANN model dominated the FAK model even when Type I or II errors were not controlled. For the four-variable model, by any of the three measures used in the FAK analysis, the ANN model created substantially fewer reclassification errors.

FAK also reported on a more complex model derived with RPA from an

Table 5. Comparison of FAK and ANN Results Using the Same Four Ratios
(percentages in parentheses)

Model	Predicted Bankrupt Correct	Type II Errors	Predicted Nonbankrupt Correct	Type I Errors
FAK	53 (77.9)	15 (22.1)	127 (96.2)	5 (3.8)
ANN I	54 (93.1)	4 (6.9)	138 (97.2)	4 (2.8)
ANN II	49 (84.5)	9 (15.5)	141 (99.3)	1 (0.7)

initial set of 20 variables. This model consisted of the six variables listed in Table 2. With this model, FAK were able to improve their forecasts significantly using a misclassification cost of 20 to 1. Their results are summarized in Table 6. This model misclassified only two bankrupt firms as nonbankrupt and misclassified ten nonbankrupt firms as bankrupt. When the same six variables were used to develop a neural network model, the nearest ANN I match was nine errors compared with the ten errors for FAK. Thus, when the neural network misclassified nine nonbankrupt firms as bankrupt, it correctly identified all of the bankrupt firms. The ANN Type II error rate of 0.0 percent (0 of 50) compares favorably with the FAK error rate of 4.0 percent.

When the ANN II and FAK Type II errors were matched at two, the ANN misclassified six of the nonbankrupt firms as bankrupt. Thus, the error rate of 4.0 percent (6 of 150) compares favorably with the FAK error rate of 7.1 percent.

In summary, after controlling for each type of error, the ANN model generated far fewer errors than the FAK model using the six variables of the

Table 6. Comparison of FAK and ANN Results Using the Same Six Ratios
(percentages in parentheses)

Model	Predicted Bankrupt Correct	Type II Errors	Predicted Nonbankrupt Correct	Type I Errors
FAK	48 (96.0)	2 (4.0)	140 (92.9)	10 (7.1)
ANN I	50 (100.0)	0 (0.0)	141 (94.0)	9 (6.0)
ANN II	48 (96.0)	2 (4.0)	144 (96.0)	6 (4.0)

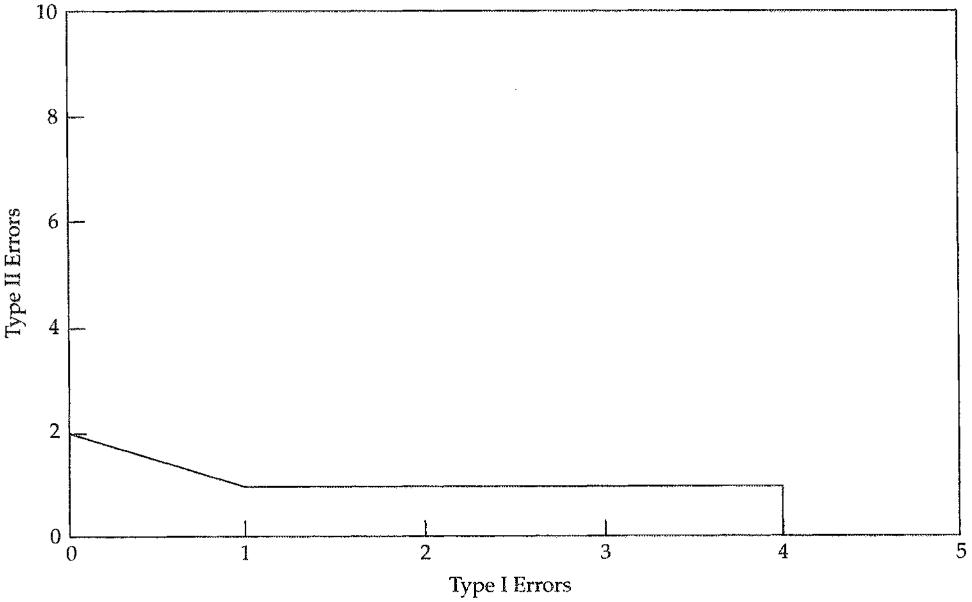
RPA model. Furthermore, the ANN models dominated the FAK model even when Type I and II errors were not controlled. For the six-variable model, the ANN model created substantially fewer reclassification errors than FAK.

FAK reported results for optimal models selected by the RPA from the 20 ratios.[15] They provided reclassification rates relative to naive models for selected error cost ratios. An examination of their results indicates that Type I errors ranged from 19 to 34 and Type II errors were 3. Thus, total errors ranged from 22 to 37 for the reported cost ratios.

To reduce overfitting, FAK selected subsets of the 20 variables. As we have shown, for any specific set of variables, the neural network can provide a more accurate forecast of the relationship between the variables and the likelihood of bankruptcy. Furthermore, the problem of overfitting the data is considerably different for the neural network. The capacity to overfit the data is influenced more by the number of nodes in the hidden layer than the number of variables being used. Thus, an interesting exercise is to examine the degree

[15] Frydman, Altman, and Kao (1985), Table IIIa, p. 284.

Figure 3. ANN Type I and Type II Reclassification Errors for 20-Variable Model

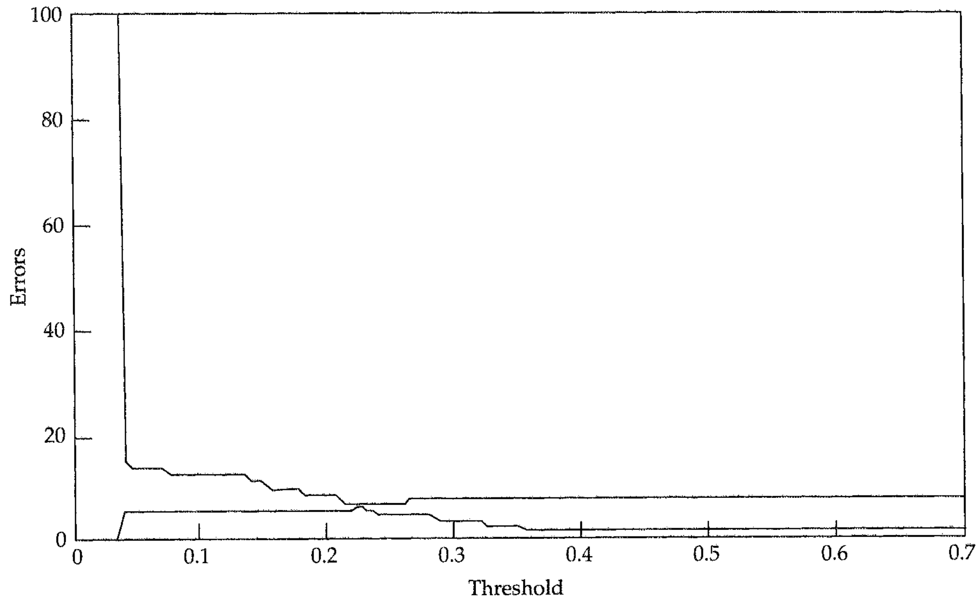


to which the neural network can improve the classifications if all 20 variables are used. To examine this case, the size of the neural network is held constant to minimize the problem of overfitting the data. The results for the resulting 20-variable model are shown in Figure 3.

Clearly, using 20 variables substantially improved the classification of the data. Figure 4 shows the effects of varying the cutoff threshold that distinguishes the bankruptcy or nonbankruptcy forecast. ANN Type I errors totaled four and Type II errors did not exceed two, adding to six total errors. The maximum errors for the ANN model were 73 percent fewer than the fewest errors reported in FAK. In addition, the results were remarkably robust to the threshold value. Thus, errors in estimating the unknown prior probability of bankruptcy had only minimal effect on the outcome of the classification.

This comparison of the ANN with RPA demonstrates that when using the exact same data, the ANN consistently outperforms the RPA for classifying bankruptcy in firms. In addition, it demonstrates the improvement potential of incorporating a richer data set for training the neural network.

Figure 4. ANN Six-Variable Model Thresholds and Errors



4. Prediction Evaluation

In the previous chapter, we found that ANN consistently improves bankruptcy forecasts. By using the data and results reported in FAK, our comparisons were not tainted by the possible misapplication of the RPA algorithm. In other studies, we have compared the ANN with other well-known estimation tools. The ANN compared favorably both in sample and out of sample with discriminant analysis; logit; and a nonparametric technique, K-Nearest Neighbor.[16] The ANN predicts better than logit and RPA.[17]

This chapter presents ANN models estimated with financial data for the 1989–91 period obtained from *Compact Disclosures.* The tests in this chapter provide an indication of the information content of *Compact Disclosures* data in predicting bankruptcy among large firms.

We dated models by the year following reported financial information; that is, the model year is the potential bankruptcy year, not the period preceding bankruptcy. The period preceding bankruptcy ranges from one to three years. Initially, we estimated ANN models with data from the prior year. These one-year models are for 1989 (1988 financial ratios with 1989 bankruptcies), 1990 (1989 financial ratios with 1990 bankruptcies), and 1991 (1990 financial ratios with 1991 bankruptcies).

The one-year models fit the data (in sample) better than two- and three-year models. Prediction accuracy beyond the sample period (out of sample), however, decreased as the estimation period of the initial models decreased. We suspect that business conditions causing financial difficulties for certain firms in one year may change in the next year. The new business conditions may now cause difficulties for a totally new set of firms. To examine how well a one-year model applies to another year, the model optimized for the financial data in 1989 predicting 1990 bankruptcies was applied to 1988 financial data to forecast 1989 bankruptcies and to 1990 financial data to forecast bankruptcies in 1991.

Next, we investigated whether the model improved if estimated with more

[16] See Dorsey, Huang, and Boose (1994) and Huang, Dorsey, and Boose (1994).

[17] See Lin, Dorsey, and Boose (1994).

than one year of financial information. We combined the 1989 and 1990 data sets and then optimized the ANN model to forecast 1991 bankruptcies. Then, to develop a general model, we estimated the parameters with three years of data. In our opinion, the three-year model is the most reliable for out-of-sample forecasts for this application.[18]

For each estimate, we graphed Type I and Type II error rates against threshold values. Predictions depend on values ranging from 0 to 1 computed by the model. The threshold values separate bankruptcy and nonbankruptcy predictions. For example, if the model generates values of 0.32, 0.48, and 0.80 for three firms and the threshold value is 0.30, then all firms are predicted as bankrupt; they would all be predicted as nonbankrupt if we select the threshold value as 0.85. As the threshold value varies, the numbers of Type I and Type II errors vary. Typically, the threshold point will be set at the prior probability of bankruptcy. Thus, if 40 percent of the firms in the study are bankrupt, the threshold will be set at 0.40. We have provided a full range of thresholds to reveal the trade-off between error types I and II.

Another common technique used in bankruptcy studies is to assign penalty weights for Type I and Type II errors. The weights are determined relative to misclassification costs, which are typically expressed as ratios, one type of error to the other. For example, if incorrectly forecasting a bankrupt firm as solvent is 50 times more costly than incorrectly forecasting a solvent company as bankrupt, then the misclassification cost for the Type I error would be set at 50. The total cost of misclassification is given by

$$TC = C_1(\text{Number of Type I errors}) + C_2(\text{Number of Type II errors}),$$

where C_1 is the cost of a Type I error and C_2 is the cost of a Type II error. Graphically, this relationship could be seen as a line with slope $-(C_1/C_2)$. A line with this slope tangent to the curve shown on the graph would represent the lowest total cost of misclassification.

In-Sample Models

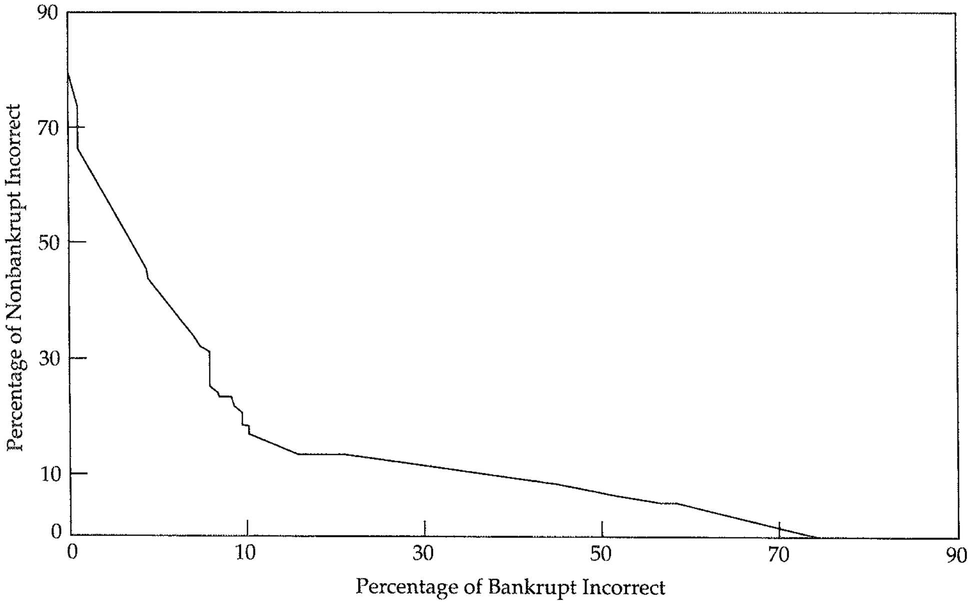
Separate models were estimated for each of the data years 1989, 1990, and 1991. In addition, models were estimated by using data from the paired years 1989–90 and 1989–91. This approach allowed the models to be compared both on the basis of how well they forecast the bankruptcies within this estimation period and also on the years that were not used for the estimation.

[18] The application examples in Chapter 5 are based on the three-year model.

1989 Model. The data for the 1989 model consisted of 130 bankrupt and 242 nonbankrupt firms. The Type I and Type II errors are graphed in Figure 5. Type II errors ranged from approximately 80 percent with no Type I errors to 0 percent when the Type I errors climb to about 75 percent. The Type I and Type II errors were equal at approximately 18 percent. At that point, the complement of the error rate, the accuracy ratio, is 82 percent. As can be seen, the curve relating Types I and II error rates is somewhat U-shaped and thus the minimum total cost for most values of misclassification costs will occur for threshold values near the middle. For a Type I error at the 20 percent level, the Type II error rate is 16 percent, so the chance of investing in bankrupt firms is reasonably low.[19] Thus, the model provides a good fit of the data for the in-sample year 1989.

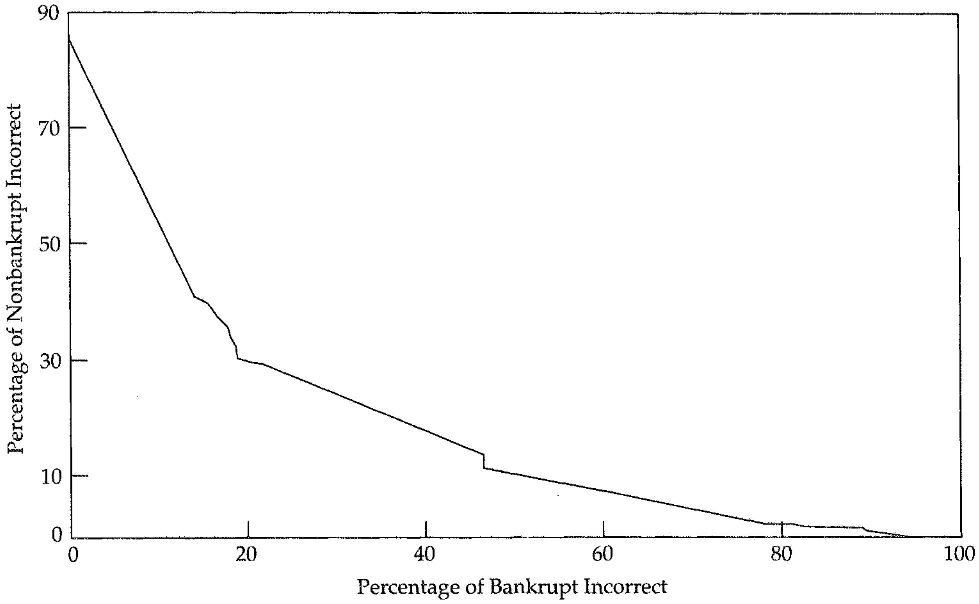
[19] The 20 percent Type I error rate, which implies an accuracy of 80 percent, was selected for exposition across all validation discussions. An accuracy level of 80 percent is a reasonably difficult objective for the prediction of large-firm failure in any given year.

Figure 5. Type I and Type II Errors for In-Sample 1989 Observations



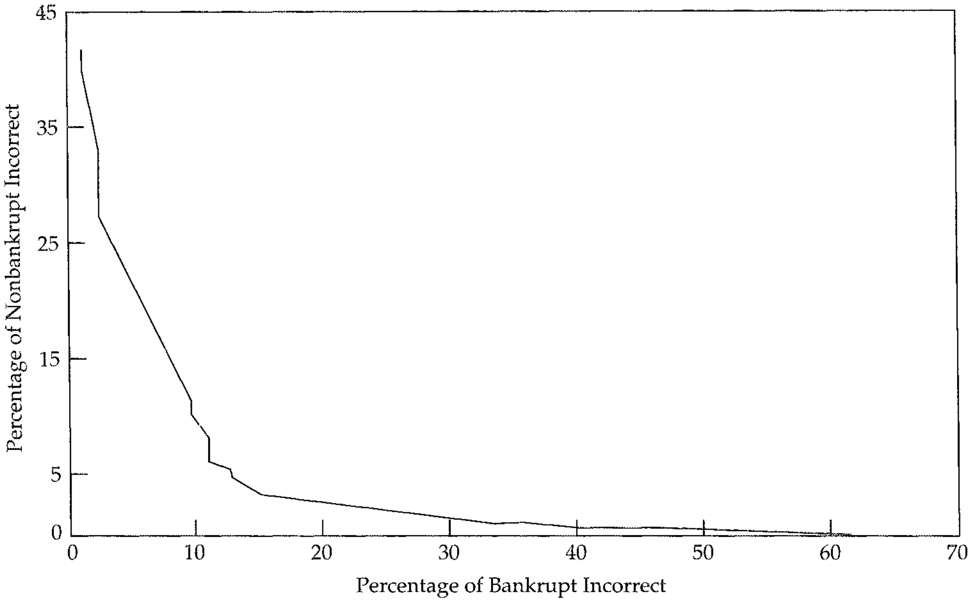
1990 Model. The 1990 data comprise 127 bankrupt and 370 nonbankrupt firms. The trade-off between Type I and Type II errors relative to the threshold is graphed in Figure 6. The Type II errors reach a maximum of nearly 90 percent when the Type I errors are held to 0 percent, and when the Type II errors went to 0 percent, the Type I errors increase to more than 90 percent. Type I and Type II errors are the same at approximately 25 percent. When Type I errors are at the 20 percent level, the Type II error rate is 30 percent.

Figure 6. Type I and Type II Errors for In-Sample 1990 Observations



1991 Model. The 1991 data consisted of 155 bankrupt and 215 nonbankrupt firms. The Type I and Type II errors are graphed in Figure 7. This model clearly performs the best of the three models. Type II errors are below 45 percent when Type I errors are 0 percent, and Type I errors are less than 60 percent when the Type I errors are 0 percent. Type I errors are equal to Type II errors at approximately 10 percent, and when the Type I error is at the 20 percent level, the Type II error rate is 3 percent. The chance of not investing in nonbankrupt firms is extremely low.

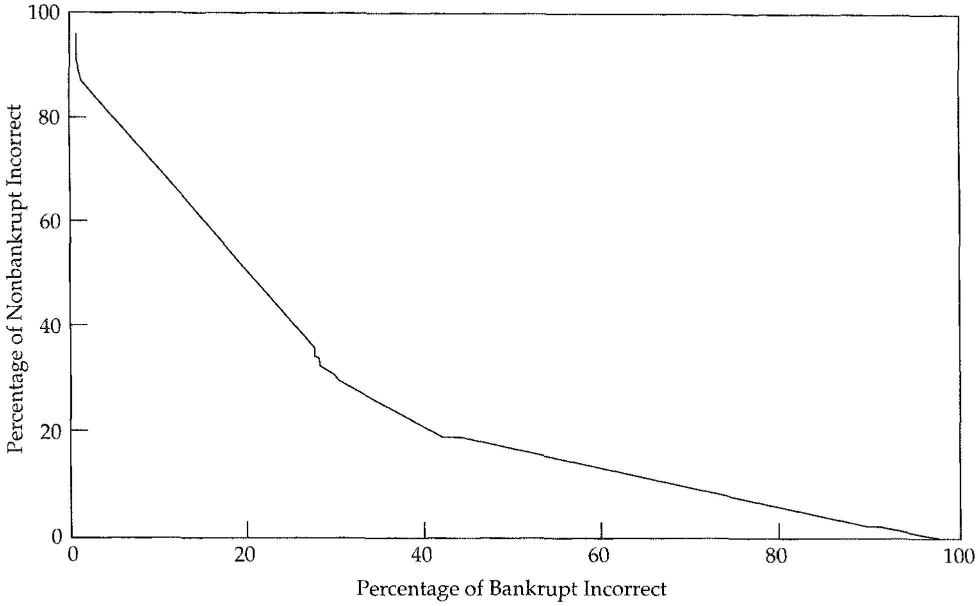
Figure 7. Type I and Type II Errors for In-Sample 1991 Observations



1989–90 Model. When the 1989 and 1990 data sets were combined, the resulting sample totaled 257 bankrupt firms and 656 nonbankrupt firms. The model was reestimated on the combined data. Figure 8 shows the trade-off between the Type I and Type II errors for the combined sample as the threshold is varied. The presence of the 1990 data set appears to affect the predictive accuracy of the ANN model. The Type I and Type II percentages are equal at approximately 30 percent. When the percentage of Type I errors is 20 percent, the percentage of Type II errors is approximately 36 percent.

1989–91 Model. The model estimated with a complete data set was created by combining observations for 1989, 1990, and 1991. This resulted in a total of 414 bankrupt firms and 1,026 nonbankrupt firms. Because this sample includes the total data set, out-of-sample cross-validation was not possible; the only error measurement possible was an in-sample analysis. The error rates are a combination of the separate years. The combined-sample error rates are higher than the individual-year error rates in 1989 and 1991 and lower than the individual-year error rates for 1990. As Figure 9 shows, Type I and Type II errors are the same at about 30 percent.

Figure 8. Type I and Type II Errors for In-Sample 1989–90 Observations



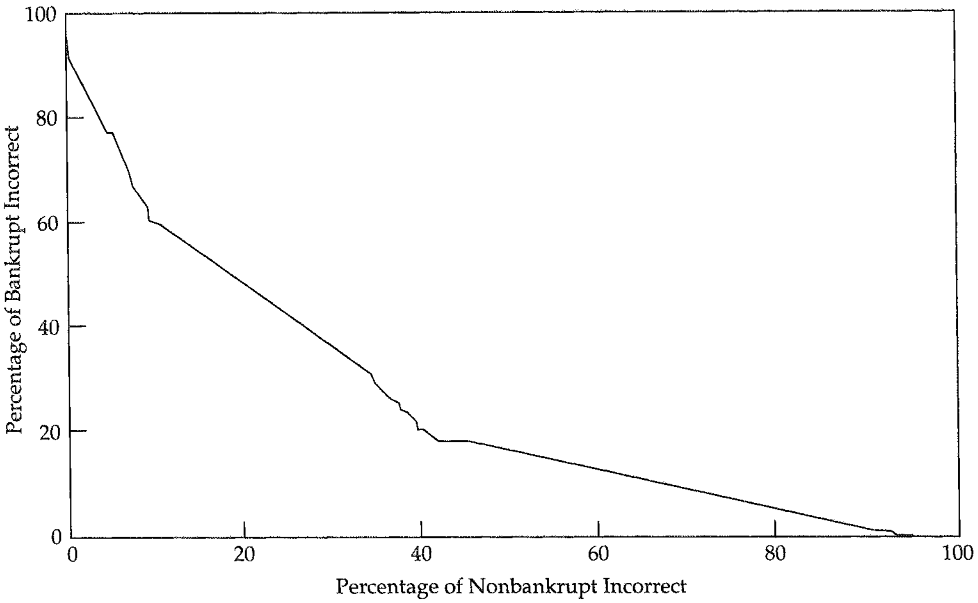
Cross-Validation Predictions

Cross-validations are tests of models using data different from the evaluation period. One set of data is used for estimation and a second set of data for validation. The sets differ in both time and businesses. Thus, the tests are the most robust available.

1989 Prediction Based on 1990 Weights. To explore the effectiveness of the 1989 model, we applied it to the 1990 data set, which included 127 bankrupt and 370 nonbankrupt firms. Results of cross-validation predictions—percentage errors for firms entering bankruptcy one year after reporting financial ratios—are shown in Figure 10.

The 1989 one-year model does not predict well for 1990. As can be seen, the trade-off between Type I and Type II errors is almost linear, in both cases rising to nearly 100 percent when the other type of error falls to 0 percent. The two types of errors are approximately equal at 45 percent, and when the Type I error is 20 percent, the cross-validation Type II error is 70 percent. Compared with the in-sample error of 16 percent, the cross-validation error is very substantial.

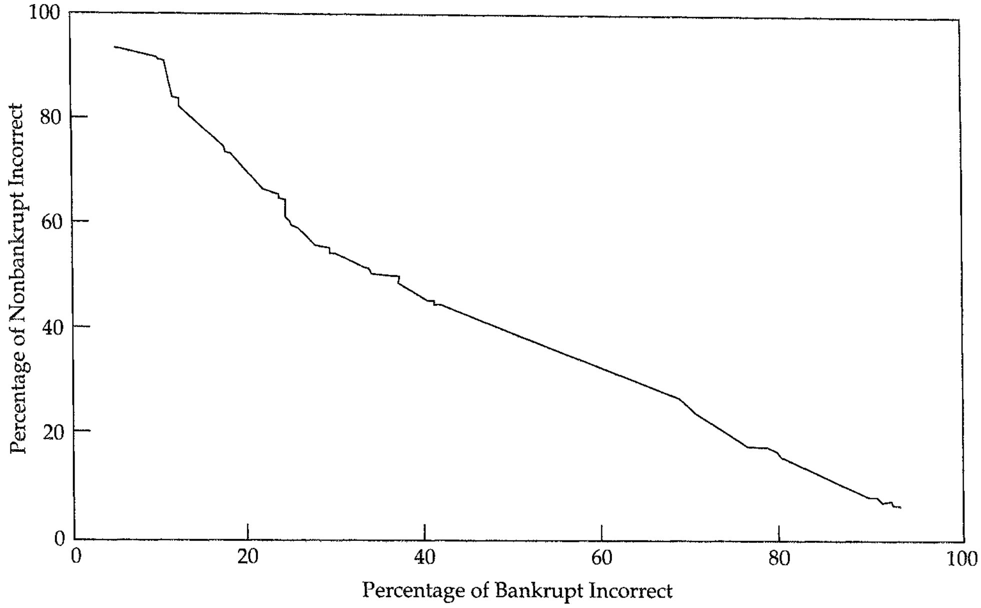
Figure 9. Type I and Type II Errors for In-Sample 1989–91 Observations



1990 Predictions Based on 1991 Weights. We suspect that bankruptcies were less predictable in 1990 than in other years. We evaluated this conclusion by using weights computed with observations in the year after bankruptcy (1991) to predict bankruptcy. If postbankruptcy weights are good predictors, then we can reject the hypothesis that 1990 is not a particularly unusual year. The postbankruptcy weights were not good predictors, however. Figure 11 shows high Type I and Type II errors for a wide range of observations. This finding adds strength to our conclusion that 1990 is less predictable than 1989 or 1991.

1989 Predictions Based on 1991 Weights. A potentially more challenging cross-validation test of the 1989 model is for it to forecast bankruptcies two years after the data used to estimate the model. These results are shown in Figure 12. This graph shows the percentage errors for firms entering bankruptcy in 1989 based on the model estimated with the 1991 data. The 1989 model clearly works well in this case. Although the Type I errors rise to more than 90 percent as the Type II errors go to 0 percent, the Type II errors only reach the mid-40 percent range as the Type I errors go to 0 percent. Both Type

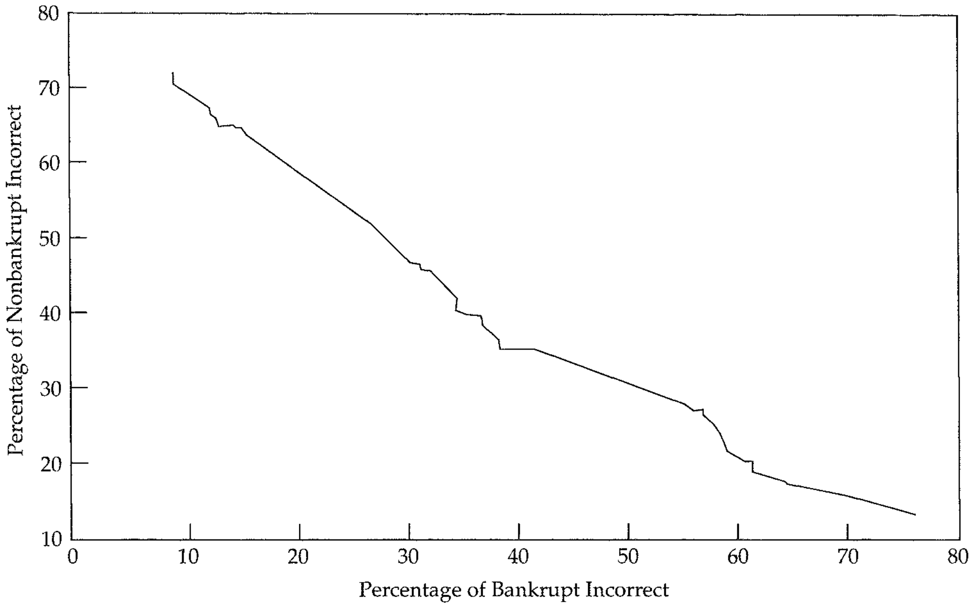
Figure 10. Cross-Validation of 1989 Predictions Based on 1990 Weights



I and Type II errors are approximately the same near 33 percent. When Type I errors are 20 percent, Type II errors are approximately 40 percent. These results provide further evidence that 1990 is inconsistent with the other two years, 1989 and 1991.

1991 Predictions Based on 1990 Weights. The result from using the model developed with the 1990 data set and forecasting the 1991 bankruptcies is shown in Figure 13. Although the model again deteriorates somewhat, reflecting the difference in the 1990 conditions, it still predicts better than the 1989 to 1990 combination. The Type I and Type II errors are approximately equal at the 32 percent level. For our baseline Type I error comparison at 20 percent, the cross-validation Type II error is 58 percent. The cross-validation error is substantially below the comparable error of 70 percent for 1990 predictions based on one-year-earlier weights. The 1990 anomaly is further confirmed by the results shown in Figure 13 for 1991.

Figure 11. Cross-Validation of 1990 Predictions Based on 1991 Weights

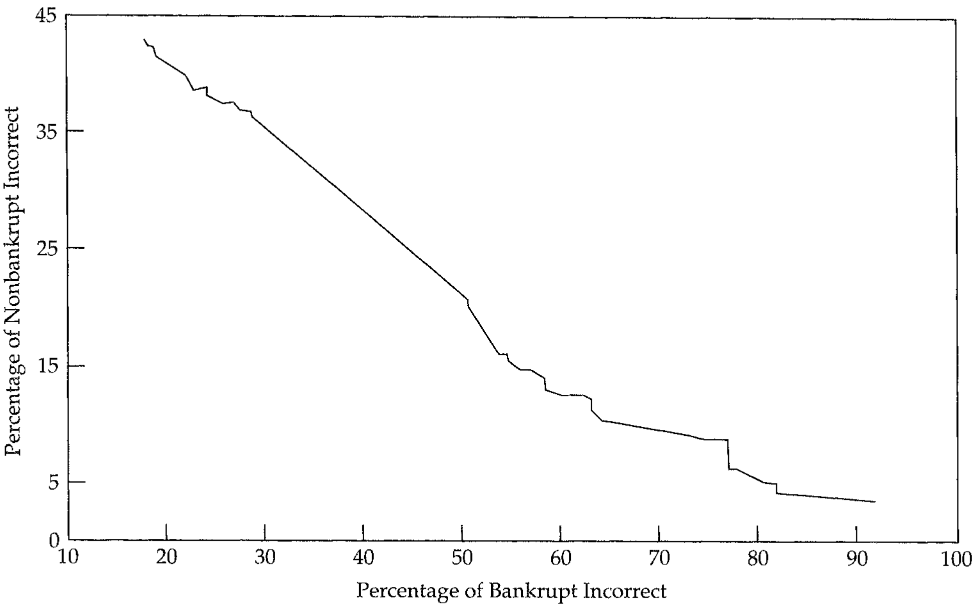


1991 Predictions Based on 1989–90 Weights. The two-year 1991 model (estimated based on the combined two-year data set for 1989 and 1990) was used to forecast 1991 business failures and nonfailures. The purpose of developing the two-year model was to explore whether the model is less sensitive than one-year models to time-varying financial ratios. If this model exhibits less sensitivity, the cross-validation error would be expected to be less for this model than for a single-year model.

The results are shown in Figure 14. As can be seen, the model does not perform particularly well. The baseline Type I error (20 percent) intersects with Type II errors at the 85 percent level. Compared with the single-year estimation period results of 58 percent, the two-year estimates are substantially less reliable. Contrary to our prior expectation, estimates using extended data proved less reliable than only the most recent information. Consequently, we concluded that changing conditions require reestimation of parameters on an ongoing basis.

Extreme Out-of-Sample Observations. Cross-validation tests for the

Figure 12. Cross-Validation of 1989 Predictions Based on 1991 Weights

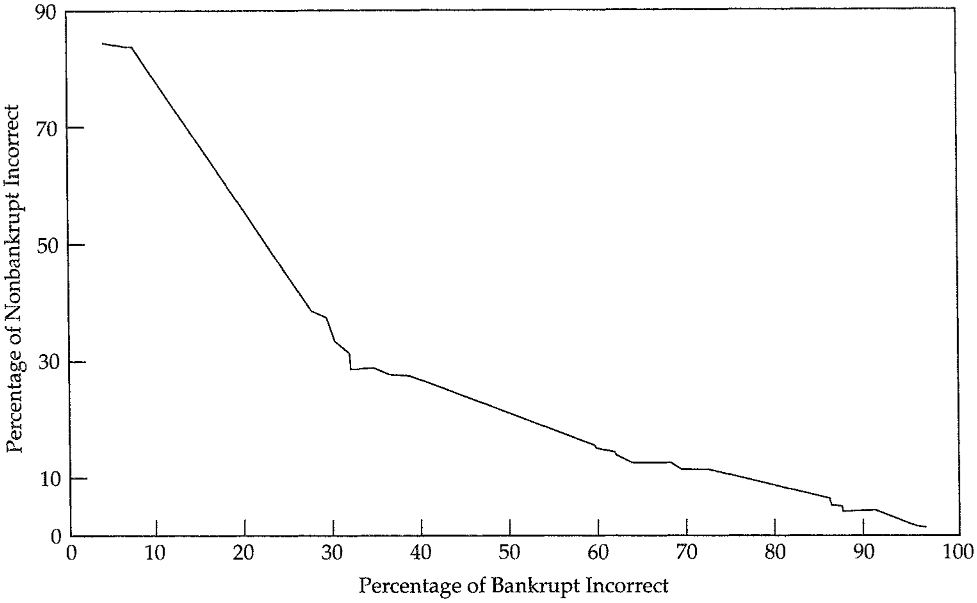


1989–90 period indicated that reclassification with out-of-sample observations results in lower accuracy than with in-sample observations. Furthermore, the passage of time after estimation may cause further degradation in predictive accuracy. The application of the ANN model is likely to be subject to both out-of-sample and delayed-estimation limitations. Consequently, estimation of extreme out-of-sample accuracy is important for validating the model. A second reason for conducting an extreme out-of-sample test arose in the bond evaluation example discussed in Chapter 5. Because none of the firms in the bond example was predicted to be bankrupt, a question arises about the validity of the prediction model.

The extreme out-of-sample observations were obtained independently from all the other samples. The sample consists of *Compact Disclosure* firms that reported the words "Chapter 11." The word search returned 208 firms for reports dated on or before the end of 1992. Of the 208 firms, 68 were inactive, 111 were active, and the remaining 19 were unknown.

For each company, the 1989–91 ANN model was used to compute the ANN output value. The criterion value of 0.40 was applied to predict bankrupt and

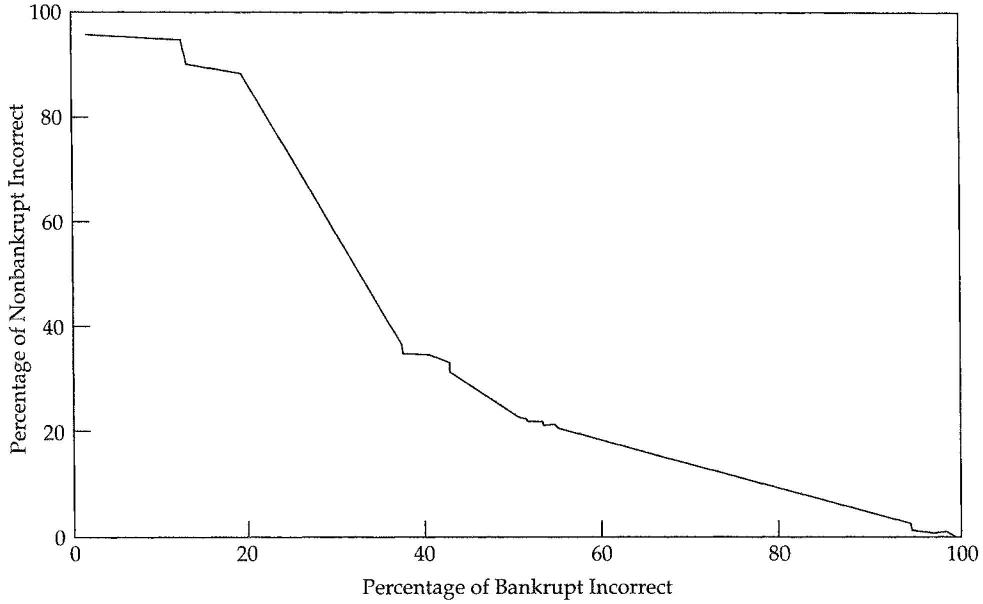
Figure 13. Cross-Validation of 1991 Predictions Based on 1990 Weights



nonbankrupt companies. Year-end 1992 financial statements were used to compute financial ratios.

Of the 208 companies, 51.8 percent were predicted to be bankrupt. That is, the Type II error is 48.2 percent. Predictive accuracy was also measured for the active and inactive subsamples. Bankruptcy predictions were 46.6 for active and 61.4 percent for inactive companies. The difference is significant at the 1 percent level. Because the inactive companies are more severely distressed than the active companies, the predicted failure rates of inactive companies would be expected to be higher than that of the active companies. Thus, the subgroup analysis by company status further confirmed the predictive power of the ANN model. The results again confirm that the ANN model makes reasonably good predictions of Chapter 11 involvement based on the financial characteristics of the firm.

Figure 14. Cross-Validation of 1989–90 Predictions Based on 1991 Weights



5. Investment Applications: Bond Evaluation

Applications for which ANN bankruptcy models might be appropriate include lending to large borrowers, contracting with large firms to supply products and services, risk-arbitraging bonds, renegotiating junk bonds, and risk-rating bonds. This chapter presents an example of one such application: evaluating yields of a nonrandom sample of bonds.

How does the neural network model improve existing bond selection? Bond quality and risk-adjusted yield evaluation depend on estimates of bankruptcy potential. If the bond market over- (under-) estimates default risk, then bond prices will be lower (higher) than the intrinsic value of the bond. Buying (selling) bonds with market prices that are lower (higher) than intrinsic values will lead to superior performance as market prices converge with intrinsic values. The convergence occurs over time as information becomes available to the bond market.

Evaluating new bond issues for rating and pricing purposes depends, in part, on measuring issuer bankruptcy potential. Models like the one presented here reduce measurement error. Consequently, new issues can be evaluated closer to their intrinsic values. Closer pricing benefits issuers by reducing the risk of receiving less than fair value for new bond issues. Moreover, improved pricing contributes to the reputation of the issuer's investment banker and rating agency.

The willingness to extend trade credit is often constrained by minimal information and evaluation tools. Frequently, the amount of credit extended is relatively small, but the profit potential resulting from trade credit is relatively large. The economics of trade credit imply that a relatively modest amount can be spent to evaluate the borrower. The type of model presented here provides a low-cost evaluation tool that potential lenders can apply to publicly available information. Hence, it has the potential to be an ideal evaluation tool for trade credit risk measurement.

The process of bond evaluation involved several steps:

- Selection of companies from the *Wall Street Journal*, New York Stock

Exchange, or American Stock Exchange bond listings

- Calculation of the default risk premium
- Prediction of distress
- Observation of anomalies

The following sections detail how we applied these steps in our example.

Data Acquisition. The data for our example are from *Wall Street Journal* bond listing and pricing data.[20] We selected bonds traded on the New York and American Stock Exchanges. The selection process consisted of selecting one bond for each corporation listing on the selected date. When multiple bonds were listed for a corporation, the bond trading at the lowest price was selected; low prices were a selection criterion to avoid call option effects. Only coupon-bearing bonds were selected. As a result, bonds with low and high current yields, with investment and junk bond ratings, and from both the NYSE and Amex were included. The sample consisted of 17 bonds. The extraction process for *Compact Disclosures* financial data was followed.

Moody's Investors Service credit ratings are described in Table 7. Our data set encompassed both investment and noninvestment grades. The five most frequent ratings were B, 21.2 percent; Bb, 18.6 percent; Baa, 16.81 percent; A, 14.2 percent; and not rated, 15.9 percent. The distribution of the ratings was not stratified but resulted from the selection process described above.

Default Risk Premium. The term "default risk premium" was defined as bond yield minus U.S. Treasury bond yield to maturity. Bond yield to maturity was approximated as the current yield plus amortized discount or premium to maturity. The U.S. Treasury bond yield to maturity was approximated from a loglinear curve fit to the U.S. Treasury bond yields as of the same date. Consequently, the default risk premium is an approximation subject to some computation error and bias from call option rights. The approximations are not significant relative to the risk premiums and do not adversely affect the example.

Bond yields ranged from 5.6 percent to 18.4 percent. After subtracting the upward-sloping U.S. government bond yields, we observed default risk premiums ranging from –0.87 percent to 13.19 percent. The average risk premium is 3.42 percent.

[20] September 29, 1993, p. C16. Quotations are for September 28, 1993.

Table 7. Moody's Credit Quality Ratings

Credit Quality of Securities	Rating
Best quality/smallest credit risk	Aaa
High grade or high quality	Aa
Upper medium grade	A
Medium grade	Baa
Medium grade with some speculative elements	Ba
Lower medium grade	B
Poor standing/may be in default	Caa
Speculative/often in default	Ca
Lowest grade speculative securities/poor prospects	C
Defaulted securities and securities issued by firms that have declared bankruptcy	Not rated

Source: Moody's Investors Service.

Distress Prediction. The ANN models that could be used to predict distress are the 1989, 1990, 1991, 1989–90, and 1989–91 models. The most recent model is 1991, and the most comprehensive is 1989 to 1991. For reasons discussed earlier, both models have advantages and disadvantages. This example computes predictions using the 1989–91 model, which is regarded as the most stable over time.

Observation of Anomalies. Potentially profitable anomalies were defined as being of two types:

Type A	Bonds with high bankruptcy potential and low default risk premiums
Type B	Bonds with low bankruptcy potential and high default risk premiums

Examine Table 8 for profit Types A and B. For Type A, look for ANN values greater than 0.4. For the bonds listed in the table, none of the ANN values is greater than 0.4, suggesting that no Type A profit opportunity exists among these bonds.

Next, look for large yield premiums. Notice that the top three bonds (Stone Container Corporation, Presidio Oil Company, and Westbridge Capital Corporation) are priced to yield large premiums. Thus, they present a Type B profit opportunity. The profit arises from a yield in excess of the ANN-predicted bankruptcy risk. An investor would realize a market value gain if market expectations for the bonds improve or when maturing bonds are repaid at par

Table 8. Actual Yield Premiums and ANN Predictions

Issuer	Yield Premium	ANN Predictions 1989–91
Stone Container Corp.	18.36	0.26
Presidio Oil Co.	12.16	0.26
Westbridge Capital Corp.	11.25	0.26
Maxxam Inc.	10.76	0.26
Chrysler Corp.	10.67	0.26
Wainoco Oil Corp.	10.67	0.26
Chiquita Brands International	10.53	0.26
Bethlehem Steel Corp.	9.67	0.26
Turner Broadcasting System Inc.	9.60	0.26
Borg Warner Corp.	8.58	0.26
Safeway Inc.	8.34	0.33
Pennzoil Co.	8.33	0.24
Ethan Allen Interiors Inc.	8.02	0.26
American Cyanamid Co.	7.90	0.10

value. In the case of Stone Container, market expectations improved after the study date (September 29, 1993) and the market price rose substantially.

This example demonstrates that the ANN model can be applied to a practical investment decision. Similar applications are recommended for accounts receivable and bank loans.

Appendix: Neural Net Structure and Genetic Algorithm Estimation

Parameters of the classification model were estimated with a training algorithm. A number of training algorithms for the neural network have been explored in the literature. The most commonly used algorithms are versions of the backpropagation algorithm developed by Rummelhart, Hinton, and Williams (1986a). The backpropagation algorithm and its many refinements are gradient search techniques. They typically start at a randomly chosen point (set of weights) and then adjust the weights to move in the direction that will cause the errors to decrease most rapidly. These types of algorithms work well when the transition toward the point of minimum error is smooth. Unfortunately, the error surface of the neural network is not smooth; it is characterized by hills and valleys that cause techniques such as backpropagation to become trapped in local minimums. To get around this problem, we used a global search technique, the genetic algorithm. The genetic algorithm samples points uniformly over the total weight space while generally moving in the direction of the minimum value. In this manner, the genetic algorithm is less likely to become trapped in a local minimum. Empirical tests have indicated that the chances of avoiding local minimums are statistically less with the genetic algorithm than with backpropagation.

Genetic Algorithm

The genetic algorithm uses the following definitions:[21]

Ξ = A subset of the k-dimensional Euclidean space.

Ξ_j = The jth set (a subset of the real line) in the Cartesian product $\Xi_1 \times \Xi_2 \times \cdots \times \Xi_k = \Xi$.

ξ = A k-dimensional vector that is an element of Ξ. The first subscript on a ξ indicates a particular vector; superscripts $'$,$''$, and $'''$ are also used for this purpose.

[21] This discussion of the genetic algorithm is taken from Dorsey, Johnson, and Mayer (1994).

ξ_{ij} = The jth component (a scalar) of the vector ξ_i.
$f(\cdot)$ = A scalar-valued function defined on Ξ.
$\Im(\cdot)$ = A strictly increasing function from the range of $f(\cdot)$ into the nonnegative real line.

As indicated in Chapter 2, the genetic algorithm iterates from one generation of candidate solutions to another. For the problem

$$\max f(\xi) \text{ such that } \xi \in \Xi,$$

let G^g denote the set of m candidate solutions (vectors of Ξ) corresponding to the gth generation. The iteration process can be written schematically as follows:

$$G^1 \rightarrow G^2 \rightarrow \cdots \rightarrow G^{c-1} \rightarrow G^c,$$

where convergence is achieved in the cth generation. Iterations are terminated by a stopping rule such as: Stop when

$$\left| \max_{\xi \in G^g} f(\xi) - \max_{\xi \in G^{g-1}} f(\xi) \right| < \delta$$

and

$$\left| \mathit{argmax}_{\xi \in G^g}\, f(\xi) - \mathit{argmax}_{\xi \in G^{g-1}}\, f(\xi) \right| < \varepsilon,$$

which hold for $g = c - p, \cdots, c$: where δ, ε, and p are prespecified numbers.

Each iteration (e.g., $G^1 \rightarrow G^2$) consists of the following eleven basic steps:

Step 1: Select m (an even number) weight vectors $\xi_1, \cdots, \xi_m$ from Ξ, and construct the set $\mathrm{G}^1 = \xi_1, \cdots, \xi_m$. Use these m vectors $\xi_1, \cdots, \xi_m$ from the set G^1 as parameters in the feedforward network.

This step initiates the algorithm and is the only step not repeated in subsequent iterations. The user selects m vectors from Ξ to serve as the first generation of candidate solutions. The choice of G^1 might reflect a priori information on the behavior of $f(\cdot)$. Such information can enhance computational efficiency, but it is not critical for eventually attaining convergence. Alternatively, the initial selection of vectors from Ξ can be purely random. This is in contrast to algorithms based on Newton's method, quadratic hill climbing, or some form of gradient descent, which often break down when their starting values are erroneous. The user also selects m, the number of candidate solutions in the first and each subsequent generation. The number m must be even to accommodate the pairing of vectors in Step 9. Generally, the larger the value of m, the more thorough the search and, therefore, the smaller the probability of convergence at false peaks. As one might suspect, however, computational cost rises as the value of m increases. For bankruptcy prediction, we set m equal to 20.

Step 2: Obtain the first training pair from the collection, $(\text{In}_{1,k}, \text{In}_{2,k}, \cdots, \text{In}_{MI,k}, T_{1,k}, T_{2,k}, \cdots, T_{MO,k})$, of all training pairs (i.e., all observations of input and their respective output [target] vectors), where k is the observation number ($k = 1, 2, \cdots, N$), N is the total number of observations, MI is the number of inputs (In), MO is the number of outputs (targets), MH is the number of hidden nodes, and s is the index of the hidden node.

Step 3: Produce a trial output using the first of the m weight vectors from G^1. This operation begins with the introduction of the input vector $(\text{In}_{1,k}, \text{In}_{2,k}, \cdots, \text{In}_{MI,k})$ to the nodal equations from the first to the hidden layer that are given by

$$y_{s,k} = g\left(\sum_{j=\wp_1}^{\wp_2} \xi_{i,j} \text{In}_{i,j} \right),$$

where

$$\begin{aligned} &\wp_1 = (s-1)(MI+1)+1 \\ &\wp_2 = s(MI+1) \\ &1 \le s \le MH \end{aligned}$$

and where $\xi_{i,MI+1}, \xi_{1,2(MI+1)}, \ldots, \xi_{i,(MH-1)(MI+1)}, \xi_{\text{i},(MH)(MI+1)}$ are equivalent to the node thresholds from the standard backpropagation network and $g(\cdot)$, the iteration rule, is assumed to be the sigmoid logistic function given by

$$g(\alpha) = \frac{1}{1+e^{-\alpha}} .$$

The outputs of the hidden layer (the $y_{s,k}$'s) are then used as inputs to the nodal equations in the output layer, which are given by

$$\text{Out}_{q,k} = g\left(\sum_{j=\wp_3}^{\wp_4} \xi_{i,j} y_{j,k} \right),$$

where

$\wp_3 = MH(MI + 1) + (q - 1)(MH + 1) + 1$
$\wp_4 = MH(MI + 1) + q(MH + 1)$
$1 \leq q \leq MO.$

Step 4: Go to Step 2 and repeat for each of the N inputs.

Step 5: Compare each of the k target vectors, $(T_{1,1}, T_{2,1}, \ldots, T_{MO,1}), (T_{1,2}, T_{2,2}, \ldots, T_{MO,2}), \ldots, (T_{1,k}, T_{2,k}, \cdots, \mathrm{T}_{MO,k})$, to its respective trial output vector, $(\mathrm{Out}_{1,1}, \mathrm{Out}_{2,1}, \ldots, \mathrm{Out}_{MO,1}), (\mathrm{Out}_{1,2}, \mathrm{Out}_{2,2}, \ldots, \mathrm{Out}_{MO,2}), \ldots, (\mathrm{Out}_{1,k}, \mathrm{Out}_{2,k}, \ldots, \mathrm{Out}_{MO,k})$, and calculate the value of the objective function $f(\xi_i)$. As an example, $f(\xi_i)$ could be the sum of squared errors given by

$$f(\xi_i) = -\sum_{k=1}^{N} \sum_{q=1}^{MO} (T_{q,k} - \mathrm{Out}_{q,k})^2 .$$

As was mentioned earlier, because the genetic algorithm does not use the derivative of the network output to adjust its weight matrixes proportionately, as with gradient descent methods, the derivative (of the objective function) need not exist and thus the network can use any objective function, $f(\cdot)$, as long as its value can be readily computed.

Step 6: Go to Step 2 and repeat for each of the m weight vectors from G^1.

Step 7: Compute the selection probabilities:

$$prob_i = \frac{\Im\,[f(\xi_i)]}{\sum_{i=1}^{m} \Im\,[f(\xi_i)]} .$$

This step gives direction to the search. In particular, the selection probabilities determine which members of G^1 contribute offspring to the second generation, G^2, through Steps 8, 9, and 10. The ξ_i most likely to contribute are those corresponding to the largest values of $prob_i$. Recall that $\Im(\cdot)$ is required to be strictly increasing and nonnegative. The nonnegativity requirement ensures that the probabilities are well defined. The requirement that $\Im(\cdot)$ is strictly increasing ensures that the most promising members of G^1 [the largest $f(\xi_i)$] are given the best chance of contributing to G^2.[22]

Obviously, many different specifications of $\Im(\cdot)$ satisfy both requirements.

[22] Note that if the sum of squared errors is the appropriate objective function, we are actually maximizing the negative of the sum of squared errors as shown in Step 5 above.

A fairly simple but often productive specification is

$$\Im\ [f(\xi_i)] \equiv f(\xi_i) - \left| \min_{\xi \in G^1} f(\xi) \right| .$$

In practice, this specification is sometimes modified to enhance computational efficiency. We will discuss these modifications after presenting the remaining four steps.

Step 8: Select a vector from G^1 with the probability of drawing ξ_i equal to $prob_i$, $i = 1, \ldots, m$. Repeat this selection process m times. Let $\xi_1', \ldots, \xi_m'$ denote the resulting vectors; construct the set $H^1 = \{\xi_1', \cdots, \xi_m'\}$.

Step 9: Draw two vectors ξ_r', ξ_s' at random from H^1. Select an integer, I, from 0 to k at random. Create a third and forth vector by crossing over ξ_r' and ξ_s' at the *Ith* position as follows:[23]

$$\xi_1'' = (\xi_{r,1}', \ldots, \xi_{r,I}', \xi_{s,I+1}', \ldots, \xi_{s,k}')$$
$$\xi_2'' = (\xi_{s,1}', \ldots, \xi_{s,I}', \xi_{r,I+1}', \ldots, \xi_{r,k}').$$

Do not replace ξ_r' and ξ_s' in H^1.

Step 10: Repeat Step 9 until H_1 is empty ($m/2$ times) and, thereby, generate the m vectors ξ_i'', $i = 1, \ldots, m$. Construct the set $G^2 = \{\xi_1'', \ldots, \xi_m''\}$.

Steps 8, 9, and 10 are commonly referred to as "reproduction and crossover" in the genetic algorithm literature. Through these steps, the desirable traits of G^1 are passed on to G^2. Step 8 selects the members of G^1 to contribute offspring to G^2. The set H^1 is called the "reproduction pool." The probability that ξ_i is selected, $prob_i$, varies directly with its value, $f(\xi_i)$, relative to the values generated by the other members of G^1 (see Step 7). This bias gives direction to the search for a solution. The draws in Step 8 are with replacement, and therefore, a given ξ_i can be listed more than once in H^1.

In Steps 9 and 10, the members of the reproduction pool H^1 are paired and then mated through the "crossover operation." Crossover combines the traits of each pair to create two offspring solutions. In Step 10, the offspring are collected in the set G^2, which is subject to further modification in Step 11.

Step 11: For each of the mk vector components ξ_{ij}'' of G^2, pick a scalar ξ_{ij}''' at random from Ξ_j. Let Y be the outcome of a Bernoulli trial, and specify $\gamma = Prob(Y_i = 1)$ and $1 - \gamma = Prob(Y_i = 0)$. Generate mk observations on Y. Replace ξ_{ij}'' in G^2 with ξ_{ij}''' if and only if $Y = 1$ on the corresponding trial.

Step 11 is called "mutation." Reproduction and crossover determine the

[23] As it stands, this operation obviously breaks down if one dimensional. One-dimensional problems are typically handled by transforming into an equivalent multidimensional space of binary vectors (Base-2 numbers).

path taken through the parameter space in the search for a solution. The purpose of mutation is to check randomly the appropriateness of that path, and if necessary, to redirect it. By randomly introducing new information into the search, mutation tests the robustness of what has evolved. The mutation probability, γ, is typically set to 10 percent or less. Larger values result in slower convergence rates; as γ approaches 100 percent, the search loses direction altogether and becomes purely random. An important consideration for mutation is the dimension of the weight vector, $\xi_{i,k}$. In particular, the probability that a given vector is altered, $1 - (1 - \gamma)^k$, is an increasing function of k (holding γ constant). Therefore, for given choices of γ and the population size m, a search on a large-dimensional parameter space will be subject to more mutation and thus less guidance than a search on a smaller parameter space. For this reason, as the number of nodes in the network increases, γ must be decreased to maintain a given level of mutation.

Enhancing Computational Efficiency

Two potential convergence problems are associated with the genetic algorithm. First, premature convergence can occur if an early generation has a small number of members that give much larger values of $f(\cdot)$ than the other members of the generation. The danger here is that the few exceptional members might dominate all subsequent reproduction pools before the parameter space has been adequately searched.

The second problem arises during later generations. As the algorithm approaches convergence, within each generation, the average value of $f(\cdot)$ will tend to be close to the best values of $f(\cdot)$. Therefore, if the suggested specification of $\Im(\cdot)$ in Step 7 is used, the selection probabilities for the best members will differ little from the average. If this result occurs, the algorithm could take a long time to converge.

Appropriately modifying $\Im(\cdot)$ at various stages of the search can help mitigate these problems and, thereby, enhance computational efficiency. The basic idea is to make the selection probabilities more homogeneous relative to the computed values of $f(\cdot)$ during early generations and relatively more heterogenous during later generations, which can be accomplished by scaling down the largest values of $f(\cdot)$ (relative to the average value) during early generations and scaling up the largest values during later generations.

Backpropagation

The backpropagation learning mechanism involves the continuous adjustment of the nodal weights as the system is repeatedly exposed to inputs and desired outputs. This learning mechanism is a supervised, iterative, gradient search technique.[24] To describe the backpropagation learning algorithm formally, let

$$\begin{aligned}\Omega = [&\omega^1_{1,1},\omega^1_{2,1},\cdots,\omega^1_{MI,1},\theta^1_1,\omega^1_{1,2},\omega^2_{2,2},\cdots,\omega^2_{MI,2},\theta^{1,}_2\cdots,\omega^1_{1,MH},\omega^1_{2,MH},\cdots,\omega^1_{MI,MH},\theta^1_{MH},\\ &\omega^2_{1,1},\omega^2_{2,1}\cdots,\omega^2_{MH,1},\theta^2_1,\omega^2_{1,2},\omega^2_{2,2},\cdots,\omega^2_{MH,2},\theta^2_{2,}\cdots\omega^2_{1,MO},\omega^2_{2,MO},\cdots,\omega^2_{MH,MO},\theta^2_{MO}]\\ = [&\xi_{i,1},\xi_{i,2},\cdots\xi_{i,MI},\xi_{i,MI+1},\xi_{i,MI+2},\xi_{i,MI+3},\cdots,\xi_{i,2(MI+1)},\cdots,\xi_{i,(MH-1)MI+MH},\\ &\xi_{i,(MH-1)MI+MH+1},\cdots,\\ &\xi_{i,MH(MI+1)},\xi_{i,MH(MI+1)+1},\xi_{i,MH(MI+1)+2},\cdots,\xi_{i,MH(MI+1)+MH+1},\xi_{i,MH(MI+1)+(MO-1)MH+MO},\\ &\xi_{i,MH(MI+1)+(MO-1)MH+MO+1},\cdots,\xi_{i,MH(MI+1)+MO\,(MH+1)}]\\ = \;&\xi_i\,.\end{aligned}$$

The system would "learn" by repeating the following steps:

Step 1: Initialize the connection weights (ω_{ij}) and node thresholds (θ_i) to small random values.

Step 2: Obtain the first training pair from the collection ($\mathrm{In}_{1,k},\mathrm{In}_{2,k},\ldots,\mathrm{In}_{MI,k},T_{1,k},T_{2,k},\ldots,T_{MO,k}$) of all training pairs (i.e., all observations of input and their respective output [target] vectors), where k is the observation number ($k = 1, 2, \ldots, N$), N is the total number of observations, MI is the number of inputs, In, and MO is the number of outputs (targets).

Step 3: Produce a trial output using the initialized connection weights, ω_{ij}, and node thresholds, θ_i. This operation begins with the introduction of the input vector, $\mathrm{In}_{1,k}$, $\mathrm{In}_{2,k}$, . . . , $\mathrm{In}_{MI,k}$, to the nodal equations from the first to the hidden layer that are given by

$$y_{s,k} = g\left(\sum_{i=1}^{MI} \omega^1_{i,s}\,\mathrm{In}_{i,k} - \theta^1_s\right) \qquad 1 \le s \le MI\,,$$

where $g(\cdot)$, the interaction rule, is assumed to be the sigmoid logistic function given by

$$g(\alpha) = \frac{1}{1+e^{-\alpha}}\,.$$

The outputs of the first hidden layer, the $y_{j,k}$, are then used as inputs to the

[24] Backpropagation is very similar to the "stochastic approximation method" of Robbins and Monroe (1951). For an interesting comparison of neural network learning and statistics, see White (1989).

nodal equations in the output layer that are given by

$$\mathrm{Out}_{g,k} = g\left(\sum_{i=1}^{MO} \omega_{i,q}^{2}\; y_{i,k} - \theta_q^2\right) \qquad 1 \le q \le MO\,,$$

which yields the trial output vector, $\mathrm{Out}_{1,k}, \mathrm{Out}_{2,k}, \ldots, \mathrm{Out}_{MO,k}$.

Step 4: Compare the trial output vector, $\mathrm{Out}_{1,k}, \mathrm{Out}_{2,k}, \ldots. \mathrm{Out}_{MO,k}$, to the target vector, $T_{1,k}, T_{2,k}, \ldots, T_{MO,k}$, and compute the value of the objective function $f(\Omega)$ for this observation, k, of N. As an example, $f(\Omega)$ could be the sum of squared errors given by

$$f(\Omega) = \frac{1}{2} \sum_{q=1}^{MO} (T_{q,k} - \mathrm{Out}_{q,k})^2\,.$$

As will be shown, because the backpropagation algorithm uses the derivative of the network output to adjust its weight matrixes proportionately, the derivative (of the objective function) must exist and thus the objective function, $f(\cdot)$, as well as the network output, must be differentiable.

Step 5: Determine the magnitude of δ. If node q is in the output layer, then δ_q is determined by:

$$\delta_q = -\frac{\partial f(\Omega)}{\partial \alpha_{q,k}}$$

$$= -\frac{\partial f(\Omega)}{\partial \mathrm{Out}_{q,k}}\;\frac{\partial \mathrm{Out}_{q,k}}{\partial \alpha_{q,k}}$$

$$= -\frac{1}{2}(2)(T_{q,k} - \mathrm{Out}_{q,k})(-1)(-1)(1 + e^{-\alpha_{q,k}})(-e^{-\alpha_{q,k}})$$

$$= (T_{q,k} - \mathrm{Out}_{q,k})\left(\frac{e^{-\alpha_{q,k}}}{1 + e^{-\alpha_{q,k}}}\right)$$

$$= (T_{q,k} - \mathrm{Out}_{q,k})\mathrm{Out}_{q,k}(1 - \mathrm{Out}_{q,k}).$$

This means that we know how a change in the total input, $\alpha_{q,k}$, to the output, $\mathrm{Out}_{q,k}$, will affect the objective function $f(\Omega)$. If, however, node s is in an internal hidden layer, then the effect is

$$\delta_s = -\left(\sum_{q=1}^{MO} \frac{\partial f(\Omega)}{\partial y_{s,k}}\right)\frac{\partial y_{s,k}}{\partial \alpha_{s,k}}$$

$$= -\left(\sum_{q=1}^{MO} \frac{\partial f(\Omega)}{\partial \mathrm{Out}_{q,k}}\frac{\partial \mathrm{Out}_{q,k}\partial \alpha_{q,k}}{\partial \alpha_{q,k}\;\;\partial y_{s,k}}\right)\frac{\partial y_{s,k}}{\partial \alpha_{s,k}}$$

$$= \left(\sum_{q=1}^{MO} \delta_q \omega^2_{s,q}\right) y_{s,k}(1 - y_{s,k}).$$

Step 6: Adjust the connection weights in proportion to the derivative of the objective function to the weight in question. That is, calculate

$$-\frac{\partial f(\Omega)}{\partial \alpha_{q,k}}\frac{\partial \alpha_{q,k}}{\partial \omega^2_{s,q}} = \delta_q y_{s,k}$$

and

$$-\left(\sum_{q=1}^{MO} \frac{\partial f(\Omega)}{\partial \alpha_{s,k}}\right)\frac{\partial \alpha_{s,k}}{\partial \omega_{i,s}} = \delta_s \; In_{i,k}\,.$$

The weights are then adjusted in proportion to this derivative as with other gradient descent methods, which yields:

$$\omega^2_{s,q}(t+1) = \omega^2_{s,q}(t) + \eta\delta_q y_{s,k} \;\; ,$$

where $\omega_{i,s}(t + 1)$ represents the weight from the node i (in the input layer) to a hidden layer, node s, at time period $t + 1$; δ_s is the error term from the nodes produced in Step 5; and η is a gain term. In other words, the new weight between the node from the current level and its higher level neighbor is equal to the old weight plus the error from said neighbor times a gain term times the node's current activation level. If the current node is inhibitory (assuming $\eta > 0$), the weight between it and its higher level neighbor decreases if $\delta_j > 0$, increases if $\delta_j < 0$, and is unchanged if $\delta_j = 0$. The opposite is true if the current node is excitatory. The node threshold values are updated using a similar method by "assuming they are connection weights on links from auxiliary constant-valued inputs" (Lippmann 1987).

Step 7: Repeat the process from Step 2 with another training pair.

Step 8: Go to Step 2 and iterate until convergence.

The training process can be slow for multilayer perceptron nets. Many cases must be processed to train the net completely. The object of the network training process is to minimize the error surface $f(\Omega)$. Each value of the weight vector Ω results in a different value of the objective function $f(\Omega)$. The problem with backpropagation and other gradient-descent methods is that they can become stuck in local optima of the error surface. It is, therefore, desirable to make separate training runs with different sets of initial random weights. This process minimizes the possibility of the algorithm finding a local minimum rather than the global minimum.

Optimal Decision Rules

The general statistical framework of pattern recognition is based on the search for the optimal decision rule, the one that best discriminates between the groups in the sample. Formally, a decision rule is defined as a test condition that partitions the sample space into distinct regions, Ω_i, $i = 1, 2, \ldots, G$, where G is the number of groups. A sample point, x, is classified as coming from group ω_i if x is in the region Ω_i. The boundaries between regions are called decision surfaces. For the financial distress case, we restricted ourselves to two groups: bankrupt and nonbankrupt.

We also confined ourselves to the Bayes minimal-risk decision rule, which is sometimes called the Bayes optimal decision rule.[25] Following the discussion in Hand (1981), a Bayes minimal decision rule was established to minimize the total expected misclassification cost, which is

$$r = r_1\, p(\omega_1) + r_2\, p(\omega_2), \tag{2}$$

where $p(\omega_i)$ is the prior probability of group i and

$$r_i = \sum_{j=1}^{2} C_{ij} \int_{\Omega_j} p(x|\omega_i)dx \quad i = 1,2\,;\, i{\neq}j\,, \tag{3}$$

where $p(x|\omega_i)$ is the class-conditional probability density function of group ω_i and C_{ij} is the cost of misclassifying a sample point, x, from group i into region Ω_j. By assuming that $C_{11} = C_{22} = 0$, the Bayes minimum-risk rule can then be expressed as

[25] Several special cases of this rule have been used in the literature. For a detailed discussion of these decision rules, see for example, Hand (1981).

$$\begin{aligned} p(x|\omega_1)/p\ (x|\omega_2) > C_{21}p(\omega_2)/C_{12}\,p(\omega_1) \rightarrow x\varepsilon\Omega_1 \\ p(x|\omega_1)/p\ (x|\omega_2) < C_{21}p(\omega_2)/C_{12}\,p(\omega_1) \rightarrow x\varepsilon\Omega_2\ , \end{aligned} \tag{4}$$

where $p(\omega_i)$ is the known prior probability.

Because C_{21}, C_{12}, $p(\omega_2)$, and $p(\omega_1)$ are predetermined, Equation (2) can be generalized as

$$h(x) \begin{smallmatrix} > \\ < \end{smallmatrix} C \rightarrow \begin{matrix} x\ \varepsilon\ \Omega_1 \\ x\ \varepsilon\ \Omega_2 \end{matrix}\ , \tag{5}$$

where C is a constant representing relative costs and $h(x)$ is a general function of attributes of x. This formulation is a very general structure of discriminant functions.

Bibliography

Aigner, D.J., and C.M. Sprenkle. 1968. "A Simple Model of Information and Lending Behavior." *The Journal of Finance,* vol. 23, no. 1 (March):151–66.

Altman, Edward I. 1968. "Financial Ratios, Discriminant Analysis and the Prediction of Corporate Bankruptcy." *The Journal of Finance,* vol. 23, no. 4 (September):589–609.

———. 1980. "Commercial Bank Lending: Process, Credit Scoring, and Costs of Errors in Lending." *Journal of Financial and Quantitative Analysis,* vol. 15, no. 4 (November):813–32.

Altman, Edward I., Michel Margaine, Michel Schlosser, and Pierre Vernimmen. 1974. "Financial and Statistical Analysis for Commercial Loan Evaluation: A French Experience." *Journal of Financial and Quantitative Analysis,* vol. 9, no. 2 (June):195–211.

BarNiv, Ran, and James B. McDonald. 1992. "Identifying Financial Distress in the Insurance Industry: A Synthesis of Methodological and Empirical Issues." *The Journal of Risk and Insurance,* vol. 59, no. 4 (December):543–74.

Beaver, William H. 1966. "Financial Ratios as Predictors of Failure." *Empirical Research in Accounting: Selected Studies, Supplement to Journal of Accounting Research,* vol. 4 (Autumn):71–111.

Benishay, Haskel. 1971. "Economic Information in Financial Ratio Analysis." *Accounting and Business Research,* no. 2 (Spring):174–79.

Benston, George J. 1967. "Substandard Loans." *The National Banking Review,* vol. 4, no. 3 (March):271–81.

———. 1977. "Risk on Consumer Finance Company Personal Loans." *The Journal of Finance,* vol. 32, no. 2 (May):593–607.

Caporaletti, Louis E., Robert E. Dorsey, John D. Johnson, and William A. Powell. 1994. "A Decision Support System for In-Sample Simultaneous Equation Systems Forecasting Using Artificial Neural Systems." *Decision Support Systems,* vol. 11:481–95.

Chatterjee, Samprit, and Seymour Barcun. 1970. "A Nonparametric Approach to Credit Screening." *Journal of the American Statistical Association*, vol. 65 (March):150–54.

Deakin, E.B. 1972. "A Discriminant Analysis of Predictors of Business Failure." *Journal of Accounting Research*, vol. 10, no. 1 (Spring):167–79.

Dorsey, Robert E., Chin-Sheng Huang, and Mary Ann Boose. 1994. "A Comparison of Traditional Statistical Techniques to Neural Networks for Solvency Surveilance of Life Insurers." In *Papers and Proceedings of the Midsouth Academy of Economics and Finance*, no. 17:124–33.

Dorsey, Robert E., John D. Johnson, and Walter J. Mayer. 1994. "A Genetic Algorithm for the Training of Feedforward Neural Networks." In *Advances in Artificial Intelligence in Economics, Finance and Management*, vol. 1, edited by Andrew Whinston and John D. Johnson, 93–111. Greenwich, Conn: JAI Press.

Dorsey, Robert E., and Walter J. Mayer. 1994. "Optimization Utilizing Genetic Algorithms." In *Advances in Artificial Intelligence in Economics, Finance and Management*, vol. 1, edited by Andrew Whinston and John D. Johnson, 69–91. Greenwich, Conn.: JAI Press.

______. 1995. "Genetic Algorithms for Estimation Problems with Multiple Optima, Non Differentiability, and Other Irregular Features." *Journal of Business and Economics Statistics*, vol. 13, no. 1:53–66.

Edmister, R.O. 1972. "An Empirical Test of Financial Ratio Analysis for Small Business Failure Prediction." *Journal of Financial and Quantitative Analysis*, vol. 7, no. 2 (March):1477–93.

Frydman, Halina, Edward I. Altman, and Duen-Li Kao. 1985. "Introducing Recursive Partitioning for Financial Classification: The Case of Financial Distress." *The Journal of Finance*, vol. 40, no. 1 (March):269–91.

Gallant, A.R., and H. White. 1992. "On Learning the Derivatives of an Unknown Mapping with Multilayer Feedforward Networks." *Neural Networks*, vol. 5, no. 1:129–39.

Gardner, Mona J., and Dixie L. Mills. 1989. "Evaluating the Likelihood of Default on Delinquent Loans." *Financial Management*, vol. 18, no. 4 (Autumn):55–63.

Gau, George W. 1978. "A Taxonomic Model for the Risk-Rating of Residential Mortgages." *The Journal of Business,* vol. 51, no. 4 (October):687–706.

Gehrlein, William V., and Thomas H. McInish. 1985. "Cyclical Variability of Bond Risk Premia: A Note." *Journal of Banking and Finance,* vol. 9, no. 1 (March):157–66.

Gombola, Michael J., Mark E. Haskins, J. Edward Ketz, and David D. Williams. 1987. "Cash Flow in Bankruptcy Prediction." *Financial Management,* vol. 16, no. 4 (Autumn):55–65.

Grablowsky, Bernie J., and Wayne K. Talley. 1981. "Probit and Discriminant Functions for Classifying Credit Applications: A Comparison." *Journal of Economics and Business,* vol. 33, no. 3 (Spring):254–61.

Hand, D.J. 1981. *Discrimination and Classification.* New York: John Wiley & Sons Ltd.

Hawley, Delvin D., and John D. Johnson. 1994. "Artificial Neural Networks: Past, Present, and Future: An Overview of the Structure and Training of Artificial Learning Systems." In *Advances in Artificial Intelligence in Economics, Finance and Management,* vol. 1, edited by Andrew Whinston and John D. Johnson, 1–22. Greenwich, Conn.: JAI Press.

Hawley, Delvin D., John D. Johnson, and Dijjotam Raina. 1990. "Artificial Neural Systems: A New Tool For Financial Decision-Making." *Financial Analysts Journal,* vol. 46, no. 6 (November/December):63–72.

Hecht-Nielson, Robert. 1987. "Kolmogorov's Mapping Neural Network Existence Theorem." *Proceedings of the IEEE First International Conference on Neural Networks III* (June 21–24):11–14.

———. 1990. *Neurocomputing.* Reading, Mass.: Addison-Wesley.

Hempel, George H. 1973. "Quantitative Borrower Characteristics Associated With Defaults on Municipal General Obligations." *The Journal of Finance,* vol. 28, no. 2 (May):523–30.

Holland, J. 1975. *Adaption in Natural and Artificial Systems.* Ann Arbor, Mich.: University of Michigan Press.

Hornik, K., M. Stinchcombe, and H. White. 1989. "Multilayer Feedforward Networks are Universal Approximators." *Neural Networks,* vol. 2, no. 5:359–66.

Huang, Chen-Sheng, Robert E. Dorsey, and Mary Ann Boose. 1994. "Life Insurer Financial Distress Prediction: A Neural Network Model," *Journal of Insurance Regulation*, vol. 3, no. 2 (Winter):131–67.

Kolmogorov, A.N. 1957. "On the Representation of Continuous Functions of Many Variables by Superposition of Continuous Functions of One Variable and Addition." *Dokl. Akad. Nauk USSR*, vol. 114:953–56.

LeCun, Y. 1986. "Learning Processes in an Asymmetric Threshold Network." In *Disordered Systems and Biological Organization*, edited by E. Bienenstock, F. Fogelman Souli, and G. Weisbuch, 233–40. Berlin: Springer.

Lin, S., R.E. Dorsey, and M.A. Boose. 1994. "A Comparative Study of Solvency Prediction in the Life Insurance Industry." *Papers and Proceedings of the Midsouth Academy of Economics and Finance*, vol. 18:464–73.

Lorentz, G.G. 1976. "The 13th Problem of Hilbert." In *Proceedings of Symposia in Pure Math* 28, American Mathematical Society.

Ohlson, J. A. 1980. "Financial Ratios and the Probabilistic Prediction of Bankruptcy." *Journal of Accounting Research*, vol. 18:109–31.

Orgler, Yair E. 1971. "Evaluation of Bank Consumer Loans with Credit Scoring Models." *Journal of Bank Research*, vol. 2, no. 1 (Spring):31–37.

Parker, D. 1985. "Learning Logic." Technical Report TR-87, Center for Computational Research in Economics and Management Science, MIT, Cambridge, Mass.

Pinches, George E., Arthur A. Eubank, Kent A. Mingo, and Kent J. Caruthers. 1975. "The Hierarchical Classification of Financial Ratios." *Journal of Business Research*, vol. 3, no. 4 (October): 295–310.

Pogue, Thomas F., and Robert M. Soldofsky. 1969. "What's in a Bond Rating?" *Journal of Financial and Quantitative Analysis,* vol. 4, no. 2 (June):201–28.

Robbins, H., and S. Monro. 1951. "A Stochastic Approximation Method." *Annals of Mathematical Statistics*, vol. 22:400–07.

Rosenblatt, F. 1958. "The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain." *Psychological Review*, vol. 65:386–408.

Rumelhart, D.E., G.E. Hinton, and R.J. Williams. 1986a. "Learning Internal Representation by Error Propagation." *Parallel Distributed Processing: Exploration in the Microstructures of Cognition*, vol. 1, edited by D.E. Rumelhart and J.L. McClelland, 318–62. Cambridge, Mass.: MIT Press.

———. 1986b. "Learning Representations by Back-Propagating Errors." *Nature,* vol. 323, no. 9 (October):533–36.

Salchenberger, Linda, M.E. Mine Cinar, and Nicholas A. Lash. 1992. "Neural Networks: A New Tool for Predicting Thrift Failures." *Decision Sciences,* vol. 23, no. 4 (July/August):899–916.

Scott, James. 1981. "The Probability of Bankruptcy: A Comparison of Empirical Predictions and Theoretical Models." *Journal of Banking and Finance*, vol. 5, no. 3 (September):317–44.

Singleton, J. Clay, and A.J. Surka. 1991. "Modelling the Judgement of Bank Rating Agencies: Artificial Intelligence Applied to Finance." *Journal of the Midwest Finance Association*, vol. 20:72–80.

Spiro, Leah Nathans. 1993. "The Arb Boys Ride Again." *Business Week* (September):80–81.

Sprecher, D.A. 1965. "On the Structure of Continuous Functions of Several Variables." *Transactions of the American Mathematical Society*, vol. 115 (March):340–55.

Srinivasan, Venkat, and Yong H. Kim. 1987. "Credit Granting: A Comparative Analysis of Classification Procedures." *The Journal of Finance*, vol. 42, no. 3 (July):665–81.

Swan, Craig. 1982. "Pricing Private Mortgage Insurance." *Journal of the American Real Estate and Urban Economics Association,* vol. 10, no. 3 (Fall):276–96.

Vandell, Kerry D., and Thomas Thibodeau. 1985. "Estimation of Mortgage Defaults Using Disaggregate Loan History Data." *Journal of the American Real Estate and Urban Economics Association*, vol. 13, no. 3 (Fall):292–315.

von Furstenberg, George M. 1969. "Default Risk on FHA-Insured Home Mortgages as a Function of the Terms of Financing: A Quantitative Analysis." *The Journal of Finance*, vol. 24, no. 3 (June):459–77.

Wasserman, P.D. 1989. *Neural Computing: Theory and Practice,* chapters 2 and 3. New York: Van Nostrand Reinhold.

Wasserman, P.D., and T. Schwartz. 1988. "Neural Networks, Part 2." *IEEE Expert,* vol. 3, no.1 (Spring):10–15.

Werbos, P.J. 1974. "Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences." Ph.D diss., Harvard University.

White, H. 1989. "Neural Network Learning and Statistics." *AI Expert* (December): 48–52.

Wilcox, J.W. 1973. "A Prediction of Business Failure Using Accounting Data." *Journal of Accounting Research,* Supplement to vol. 11 (Autumn):163–71.

Zavgren, C.V. 1983. "The Prediction of Corporate Failure: The State of the Art." *Journal of Accounting Literature,* vol. 2 (Spring):1–38.