© *The Journal of Risk and Insurance*, 1994 Vol. 61, No. 3, 402-424

A Neural Network Method for Obtaining an Early Warning of Insurer Insolvency

Patrick L. Brockett
William W. Cooper
Linda L. Golden
Utai Pitaktong

ABSTRACT

This article introduces a neural network artificial intelligence model as an early warning system for predicting insurer insolvency. In order to investigate a firm's propensity toward insolvency, a feed forward, back-propagation methodology is applied to financial data two years prior to insolvency for a sample of U.S. property-liability insurers that became insolvent in 1991 or 1992 and a size-matched sample of solvent insurers. The results of the neural network method are compared with those of discriminant analysis, A. M. Best ratings, and the National Association of Insurance Commissioners' Insurance Regulatory Information System ratings. The neural network results show high predictability and generalizability, suggesting the usefulness of this method for predicting future insurer insolvency.

Introduction

The definition and measurement of business risk has been a central theme of financial and actuarial literature for years. Although the works of Borch (1970), Bierman (1960), Tinsley (1970), and Quirk (1961) dealt with the issue of corporate failure, their models did not lend themselves to empirical testing. Additionally, while the work by Altman (1968), Williams and Goodman

---

Patrick L. Brockett is Joseph H. Blades Centennial Memorial Professor of Risk Management and Professor of Finance, Mathematics, and Management Science and Information Systems. He is also Director of the Center for Cybernetic Studies at the University of Texas at Austin. William W. Cooper is Professor of Management, Accounting, and Management Science and Information Systems, as well as the Nadya Kozmetsky Scott Centennial Fellow at the IC^2 Institute at the University of Texas at Austin. Linda L. Golden is the Zale Corporation Centennial Professor in Business, Professor of Marketing, and Senior Research Fellow at the IC^2 Institute at the University of Texas at Austin. Utai Pitaktong is in the Research Unit of the Texas State Department of Insurance and with the Center for Cybernetic Studies at the University of Texas at Austin. The authors are listed in alphabetic order. This research was supported in part by grants from the University of Texas at Austin College and Graduate School of Business and the Actuarial Education and Research Fund of the Society of Actuaries.

(1971), Sinkey (1975), and Altman, Haldeman, and Narayanan (1977) attempted to predict bankruptcy by using discriminant analysis, their approaches were static in nature. Thus, the dynamics of a firm's operations and the changing economic environment were not included in their analyses. Santomero and Vinso (1977), and Vinso (1979) used actuarial ruin theory models to incorporate the dynamic aspects of the cash flow process for assessing the likelihood of bank insolvency. However, there appear to be mathematical problems with their development, making their results difficult to implement in practice.

Insolvency within the insurance industry has become a major issue of public debate and concern, and the identification of potentially troubled firms has become a major regulatory research objective. Previous research on the topic of insurer insolvency prediction includes Ambrose and Seward (1988), BarNiv and Hershbarger (1990), BarNiv and MacDonald (1992), Barnoff (1993), and Harrington and Nelson (1986). BarNiv and MacDonald provide a particularly good review of the previous research techniques and results on rating and monitoring insolvency risk for insurers and can be consulted for further background on alternative approaches.

The research presented in this article aims to construct an early warning system for regulatory use in insolvency prediction. The approach we utilize is based upon modern methods in artificial intelligence (in particular, a neural network model), and uses financial and other insurer operations data such as those available in the annual statements filed with the National Association of Insurance Commissioners. We also compare the insolvency prediction results using the neural network methodology with those obtained via discriminant analysis, A. M. Best ratings, and the National Association of Insurance Commissioners' Insurance Regulatory Information System ratings.

Situational Overview

In the context of warning of pending insurer insolvency, the regulator has several sources of information. For example, there are reporting and rating services such as the A. M Best Company, which rates 3,000 property-liability and life health insurers. However, many of the insurers of interest to state regulators are not rated by Best's or by other rating services (e.g., Moody's or Standard and Poor's).

In addition, the National Association of Insurance Commissioners (NAIC) has developed a system called the Insurance Regulatory Information System (IRIS). This system was designed to provide an early warning system for insurer insolvency based upon financial ratios derived from the regulatory annual statement. The IRIS system identifies insurers for further regulatory evaluation if four of the eleven (or twelve, in the case of life insurers) computed financial ratios for a particular company lie outside a given "acceptable" range of values. IRIS uses univariate tests, and the acceptable range of values is determined such that, for any given univariate ratio measure, only approximately 15 percent of all firms have results outside of the particular specified "acceptable" range.

The adequacy of IRIS for predicting troubled insurers has been investigated empirically and found not to be strongly predictive. For example, one small-scale comparison study using only five of the IRIS ratio variables from the NAIC data has shown that by using statistical methods it is possible to obtain substantial improvements over the IRIS insolvency prediction rates (cf. Barrese, 1990). More recently, the NAIC has implemented a supplementary system based on a number of additional financial ratios. In addition, the risk-based capital systems recently adopted by the NAIC may enhance regulators' ability to identify problem insurers prior to insolvency. Evaluations of the accuracy of the risk-based capital system for property-liability insurers are provided in Grace, Harrington, and Klein (1993) and Cummins, Harrington, and Klein (1994). BarNiv and MacDonald (1992) review other methodologies for solvency prediction in insurance.

An Artificial Intelligence Approach Using Neural Networks

Inspired by the neurophysical structure of the brain, the collection of mathematical models known as neural networks has developed as an approach to provide algorithmic structures that can interact with the environment in much the same manner as does the human brain. This interaction includes such aspects of artificial intelligence as, for example, learning from experience, generalizing from examples, and abstracting the essence from input data that may contain irrelevant factors. Structurally, the neural network model can be represented as an interconnection of many autonomous individual processing units that behave similarly in certain respects to the interconnections of individual neurons in the brain. Mathematical neural networks function by constantly adjusting the interconnections between individual neural units. The process by which the mathematical network "learns" to change the interconnections, improve performance, recognize patterns, and develop generalizations is called the training rule.

One of the popular algorithms that has been used successfully in many applications is the "back-propagation learning algorithm" based on a "feed forward" network, described below. We use this algorithm for assessing insolvency propensity. Essentially, the feed forward designation indicates that the flow of the network intelligence or information is from input toward output (as, for example, occurs in path models and structural equation or maximum likelihood factor analysis causal models). The back-propagation designation indicates that the particular learning algorithm updates its abilities by starting at the output, determining the error produced with a particular mathematical structure, and then propagating this error backward through the network to determine, in the aggregate, how to efficiently adjust the mathematical structure (in this case, the particular interconnections between the individual neurons) in order to improve the ultimate output behavior of the network. Although this is an iterative and possibly somewhat time-consuming algorithm,

when trained on adequate samples it gives good results in practice.[1] To date, neural network mathematical techniques have been applied in many areas, such as pattern recognition, knowledge data bases for stochastic information, robotic control, and financial decision making.

Properties of Neural Networks

Neural networks are endowed with many special features. Some major characteristics pertinent to the insolvency monitoring problem faced by regulators are pattern recognition, adaptability to changing environments, and resistance to noise in the inputs. To achieve pattern recognition, the neural network takes a given pattern as input (e.g., a digitized picture) and matches this pattern with an associated output (e.g., one of a class of prototypic patterns or images). The ability to keep both the input and output patterns in the associative memory makes the network, to some degree, insensitive to minor variation in its input. Neural networks also have reconstruction ability. When an input pattern is not complete, the network will attempt to identify it with the most closely related pattern in its memory.

It should be noted that, unlike other artificial intelligence methods that train a network deductively by programming in a system of mathematical logic, the neural network model "learns" empirically or inductively by training repeatedly on a given set of sample input data. These training sessions develop an appropriate nonlinear mathematical network model that can essentially reproduce the observed output from the given input. Thus, the method does not start with an *a priori* model of the relationship between input and output (as is the case in causal linear statistical models). Indeed, it is *discovering* the relationship (or logic) between the input and output that is one of the primary thrusts of the training exercise on the network. It follows that, in actual applications, learning can continue even while the network is producing predictions. This, of course, allows the network to adapt to new situations (a feature absent from causal models or other static statistical estimation techniques such as discriminant analysis, logistic regression, linear regression, etc.).

Information in neural networks is distributed throughout the network. When some pieces of information are lost (such as some processing units are destroyed), this may not cause the whole network to collapse. Studies have shown that the network predictions are somewhat insensitive to variations in the network configuration involved (Neti, Schneider, and Young, 1992).

The General Neural Network Model

Although individual models differ, all neural networks possess similar fundamental features. The basic building block of a neural network is the mathematical construct known as the single neural processing unit. This unit takes the multitude of individual *inputs,* determines (through the learning algorithm)

---

[1] There have been, however, some heuristics suggested to improve learning time (cf. Roberts, 1988).

connection weights that are appropriate (or most effective) to apply for these inputs, and applies a combining or *aggregation function* to the derived connection weighted inputs in order to concatenate the multitude of individual inputs into a single value.[2] An *activation function* is then applied, which takes the aggregated weighted values for the individual neural unit and produces an individual output for the neural unit. This process at the individual neural unit is repeated, resulting in the use of as many single neural processing units as desired, connected in whatever fashion is needed in order to produce a well-functioning global neural network.

It is useful to recall that the information in this artificial intelligence technique is incorporated into the functioning of an individual processing unit through the various inputs. In practice, these inputs also can be individually weighted according to the credibility or importance of the source of information. Because of the iterative reweighting scheme for determining the most effective interneuron connection weights in the general neural network, each different input potentially has a different degree of influence on the output of the processing unit.

The combined (weighted aggregate) result $z = \sum w_i x_i$ is then interpreted by the network through the use of an activation function. The logistic function

$$F(z) = \frac{1}{1+\exp(-\eta - z)}$$

is the most usual choice for the activation function because of its desirable properties and its simplicity in analytic representation.[3] Other functions, such as hyperbolic tangent function, step function, or even linear function, are sometimes used as activation functions depending on the situation.

There is only one output for each processing unit, but the value can be zero-one, natural numbers, or continuous, and, because there can be numerous neural processing units, multivariate outputs can be developed at any stage as desired. In general, the neural processing units are grouped together into composite structures called neural layers for ultimate topology of the network. Figure 1 provides a simplified schematic of the described topology.

Neural Network Processing Units and Layers

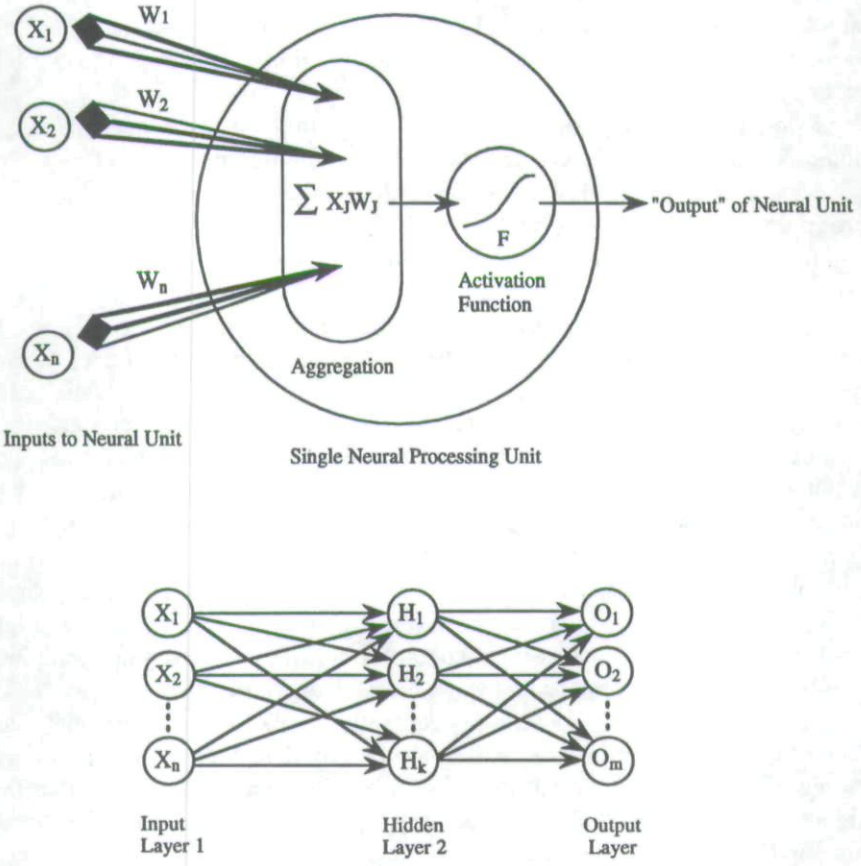
Neural networks emerge from developing interconnections between the individual mathematical constructs known as neural processing units, just as the human brain functions through interconnections between neurons. Mathematically, the processing units at each stage are grouped and each group is called a layer. The layers are then connected. Two-layer feed forward networks

---

[2] The simplest technique (and the one used in this article) for aggregating or combining the various inputs is the summative linear function in which each input is multiplied by its weight, and all the weighted inputs are summed.

[3] Its properties include the fact that, because of its S-shaped form, it has the steepest slope near the threshold (intercept) point η, indicating that the relative impact of inputs on the corresponding output values is most pronounced near the borderline or threshold, and the fact that extreme or outlier values have decreasingly dramatic effects upon predictions.

Figure 1
Prototypical Neural Network Configuration with Multiple Inputs, Multiple Hidden Units, and Multiple Outputs



Clusters of processing neural units. Each circle represents a single neural processing unit. The hidden layer is the ensemble of "outputs" of the input layer, and output layer is the output of the hidden layer.

are networks that have a front (or input) layer containing the input data information obtained from an external source and a second output layer providing output information.[4] The two-layer neural network models were initially devel-

[4] There is a lack of consistency in the literature with regard to the counting of the number of layers in a network. Some authors would designate what we have called a two layer network as a single layer structure. They argue that this should be counted as single layer because the front or input layer only acts as an open channel receiving inputs, but not involving in any information processing. Essentially, they would count the number of layers as the number of levels at which the weighted aggregation and activation function application is performed. This is like counting arc groups in a network. Others follow the convention used in this paper where the inputs are considered as providing a layer and the output as providing a layer, and then additionally the

oped and gained much interest because of their abilities to learn to recognize simple static patterns. For example, it follows from the neural network topology given in Figure 1 (without the hidden layer) that in the two-layer neural network with a linear aggregation function, a logistic aggregation function, and a single output node, the mathematical structure developed is isomorphic to the standard logistic regression model. It follows immediately that the neural network models considered here generalize the commonly used logistic regression models.[5]

Geometrically, if we consider the potential outputs in the space spanned by potential inputs, then the two-layer network forms two decision regions separated by a hyperplane (just as discriminant analysis, logistic regression, and probit analysis partition the output space into two halfplanes).

The original neural network was called a perceptron. The convergence procedure for adjusting weights in a perceptron model was developed by Rosenblatt (1959), who proved that if the inputs presented to the network for learning are from two separable classes, then the perceptron model convergence algorithm indeed converges and yields a separating hyperplane as the derived decision hyperplane between the two classes. On the other hand, if the classes of patterns cannot be separated by a hyperplane, the perceptron model is not appropriate. Minsky and Papert (1969) illustrated this weakness by demonstrating that perceptron models cannot classify correctly when attempting to learn to recognize the exclusive "or" problem.[6]

Multilayer networks, where there are one or more hidden processing layers between the input layer and output layer, can overcome many of the limitations possessed by two-layer networks for representing complex nonlinear problems (such as occur with the exclusive "or" problem). In essence, the "hidden" layers in the model are conceptually similar to nonorthogonal latent factors in a factor analysis, providing a mutually dependent summarization of the pertinent commonalties in the input data. (Pertinent in the sense that they are able to be used by the network as, essentially, a learning device for providing input into a best possible logistic regression going between the hidden layer and the output.) There is, of course, dependence (due to the weighting techniques) between the process of arriving at the hidden neural processing units constituting the hidden layer and the process of going from the hidden layer neural units to the final predicted output value.

Originally, these quite complex multilayer networks with hidden layers were not used because efficient training algorithms were not generally available to

---

number of intermediate information processing layers are counted. This is akin to counting the number of node groups in a network.

[5] Other standard statistical models also arise as subsidiary models of the simplest neural networks. For example, the Probit functional arises in the two layer network with a summative aggregation function and an activation function which is the standard normal distribution function.

[6] The exclusive "or" problem occurs when one of two events, but not both, occurs. This is in contrast with the inclusive "or" problem, which is simply the Boolean union of the two events; that is, either one or both of two events occurs.

allow the network to recognize and learn the required complex nonlinear processing. It was not until the development of the back-propagation training algorithms by Rumelhart, Hinton, and Williams (1985) that it became possible to efficiently compute and use multilayer networks.

The two-layer network (such as logistic regression or probit models) can only determine decision regions that are halfplanes. A three-layer neural network can form an arbitrarily complex decision region, and it can be proven that no more than three layers are ever required in a feed forward network. In addition to providing substantially increased complexity in the output region designations for decision making and learning, the nonlinearities used within the processing units and between layers (e.g., the choice of nonlinear activation functionals) provide additional capabilities for multilayer networks.[7] Three-layer neural networks have been shown to "predict" the failure of savings and loan companies (cf. Salchenberger, Cinar, and Lash, 1990) and the shopping behavior of consumers (Golden, Brockett, and Pitaktong, 1993) better than traditional statistical methods, as well as having other desirable characteristics (Brockett, Golden, and Pitaktong, 1993).

Since the topology of the neural network determines its ability to simulate the actual process exhibited by the data for going between the input and output values, the number of processing units must be large enough to form a decision region which is as complex as is required by a given problem. Lippmann (1987) has suggested that, when the decision regions are disconnected, there should be more than one processing unit in the second hidden layer. In the worst case, the number of processing units in this layer must be equal to the number of disconnected regions in input distributions. The number of processing units in the first layer must also be sufficient to provide three or more edges for each convex area generated by every one of the second-layer processing units (cf. Lippmann, 1987). The above discussion pertains to multilayer networks with a single output (as utilized in this article) when nonlinear activation functions are used.

The Back-Propagation Algorithm

The back-propagation algorithm can be viewed as a gradient search technique where the objective function is to minimize mean square error between the computed outputs of the network corresponding to the given set of inputs in a multilayer feed forward network and the actual outputs observed in the data for these same given inputs. The network is trained by presenting an input pattern vector X to the network, performing the calculations sequentially through the network until an output vector O is obtained. The output error is computed by comparing the computed output O with the actual output for the input X. The network attempts to learn by adjusting the weights at each indi-

[7] If linear function is chosen as the activation function, then there is no real advantage in going beyond the two-layer neural network since a two-layer network with appropriate weights can exactly duplicate the calculations of a multilayer network.

vidual neural processing unit in such a fashion as to reduce the observed prediction error. Mathematically, the effects of prediction errors are swept backward though the network, layer by layer, in order to associate a "square error derivative" (delta) with each processing unit, compute a gradient from each (delta), and finally update the weights of each processing unit based upon the corresponding gradient.

The process is repeated starting with another input/output pattern. After the training set is exhausted, the algorithm starts over again on the training set and readjusts the weights throughout the entire network structure until either the objective function (sum of squared prediction errors on the training sample) is sufficiently close to zero or the default number of iterations is reached. The mathematical formulation required for implementing the analysis is provided in the Appendix. The computer algorithm implementing the back-propagation technique used in this study is from Eberhart and Dobbins (1990).

Empirical Methodology

Scope of the Study

The empirical part of our study focuses on property-liability insurers (rather than all forms of insurers). The operations, investment activities, duration of liabilities, and vulnerabilities of life and health insurers differ substantially from those of property-liability insurers. Accordingly, the neural networks and input variables that would be pertinent for the two classes of insurers could be quite different, and attempts to deal with both simultaneously could confound the study. However, the model-building techniques and strategies developed here could also guide subsequent investigations into life/health companies. The study focuses on financial variables available from the NAIC annual statement tapes. It is likely that predictive accuracy could be improved by considering additional explanatory variables.

From the perspective of regulators, potential insolvencies must be detected early if the predictions are to be used to guide state regulators to insurers where deficiencies are developing. Because of the time lag involved in receiving NAIC annual statement data and the time needed to respond to insolvency predictions in a prescriptive (as opposed to merely a liquidation) manner, we built a model that could detect insolvency propensity using annual statement data from two years prior to insolvency.

Variable Selection

The Texas State Board of Insurance was involved in variable selection due to their interest in early warning to help firms prevent insolvency. Their rather large list of several hundred insurer financial health indicator variables was used as the first consideration set in the selection of variables. In cooperation with the Texas State Board of Insurance, this list was culled down using published research identifying variables that failed to identify potential insolvencies in previous research. This resulted in the 24 variables defined in Table 1.

Table 1
Variable Definitions

Variable	*Definition*
V1	Policyholders' Surplus
V2	Potentially Impaired Surplus
V3	Significant Decrease in Policyholders' Surplus from Previous Year
V4	Capitalization Ratio
V5	Excessive Unrealized Capital Gains and Losses to Surplus
V6	Negative Cash Flow
V7	Net Change in Cash and Short-Term Investments
V8	Reinsurance Recoverable as Percentage of Policyholders' Surplus
V9	Change in Invested Assets
V10	Significant Change in Short-Term Investments
V11	Significant Change in Aggregate Write-In Assets
V12	Investment Yield Based on Average Invested Assets
V13	Ratio of Significant Receivables from Parent, Subsidiaries, and Affiliates to Capital and Surplus
V14	Significant Increase in Noninvested Assets
V15	Significant Increase in Current Year Net Underwriting Loss
V16	Premium to Surplus (IRIS 1)
V17	Surplus Aid to Surplus (IRIS 3)
V18	Two-Year Overall Operating Ratio (IRIS 4)
V19	Investment Yield (IRIS 5)
V20	Change in Surplus (IRIS 6)
V21	Liabilities to Liquid Assets (IRIS 7)
V22	Agent's Balance to Surplus (IRIS 8)
V23	Two-Year Reserve Development to Surplus (IRIS 10)
V24	Potential Impaired Surplus/Policyholders' Surplus

For parsimonious model building,[8] the list of variables given in Table 1 was reduced further through a series of statistical analyses using a sample of solvent and insolvent Texas domestic property-liability insurers (using Texas domestic insurer data provided by the Texas State Board of Insurance for insolvencies during the period 1987 through 1990).[9]

The first step in the preliminary analysis was to examine each variable separately to see if a significant difference existed between the solvent and the insolvent insurers that could be detected by using that variable alone. Variables that showed no statistically significant differences between these two groups might provisionally be eliminated in the interests of parsimony and to reduce the number of parameters in the model. Although eliminating the variables that did not individually discriminate did produce a smaller subset of pertinent variables, many of the original variables showed a significant difference be-

[8] While neural network models can be built using all 24 variables, the statistical methods (such as logistic regression and discriminant analysis) against which the neural network method is to be compared require a smaller number of variables due to the sparsity of insolvencies and the consequent degrees of freedom problems.

[9] Using this alternative sample for variable selection avoided the redundancy of building a model using national sample data and then testing the model on the same data (or a subset).

tween the two groups of insurers. Because testing one variable at a time in a univariate manner is known to be subject to various statistical limitations, we also ran discriminant analysis, canonical analysis, collinearity tests, and stepwise logistic regression to check further which sets of variables might be eliminated due to multivariate considerations (e.g., because their particular addition did not significantly contribute to differentiating between the two groups in a multivariate context, perhaps due to collinearity).

Of particular interest in determining an appropriate subset of variables was the stepwise logistic regression procedure. In the context of the neural network application, stepwise logistic regression was deemed appropriate for the following two reasons. First, this was viewed as a preliminary methodology to determine a subset of potential input variables, with the F to enter and F to exit set very high (so that it was conservative in its elimination of potentially discriminating variables). Moreover, the ultimate output of the stepwise logistic regression methodology was not to be used for discrimination purposes, but rather for developing a rough cut in the number of variables to be included in our ultimate artificial intelligence methodology. Essentially, it was used only to screen out "noise" and collinear information from the neural network input variable set.

Second, the stepwise logistic regression methodology was considered a shortcut to the procedure of running the neural network on each possible subset of variables. The stepwise procedure is a computationally efficient approximation to the results that might have been obtained with the "all subsets" methodology. The logistic regression technique was chosen because of the ultimate desired use of the logistic activation function in the feed forward, back-propagation neural network.

In essence, the logistic regression methodology can be considered a two-layer (multiple input and single output only) neural network with a logistic activation function and a summative aggregation function. Using the two-layer network (logistic regression) as an approximation of the ultimately desired three-layer network is a computationally efficient method for preliminary analysis. The final subset of eight variables selected using the above techniques is shown in Table 2.

Table 2
Final Set of Variables

Initial Variable Number	*Description*
V1	Policyholders' Surplus
V4	Capitalization Ratio
V9	Change in Invested Assets
V12	Investment Yield Based on Average Invested Assets
V13	Ratio of Significant Receivables from Parent, Subsidiaries, and Affiliates to Capital and Surplus
V15	Significant Increase in Current Year Net Underwriting Loss
V17	Surplus Aid to Surplus (IRIS 3)
V21	Liabilities to Liquid Assets (IRIS 7)

The Specific Neural Network Used Herein

We denote the set of eight financial statement variables used as neural inputs by $x = (x_1, x_2,..., x_8)$. The ultimate functional relationships between the input units (financial statement variables) and the output prediction units (probability of solvency) are based on a linear aggregation function used in conjunction with a logistic activation function. Thus, the calculated value of the factor corresponding to the ith hidden neural unit is

$$H_i = \frac{1}{1+\exp\{-\eta_i^{(1)} - x' w_i^{(1)}\}}, \tag{1}$$

where $w_i^{(1)} = (w_{i,1}^{(1)}, w_{i,2}^{(1)},..., w_{i,8}^{(1)})$ = the vector of weights (to be subsequently adjusted by learning) bridging between the original set of eight input variables and the ith hidden unit,

$\eta_i^{(1)}$ = the threshold for the ith hidden unit (the intercept term in the logistic regression formulation for H_i), and

H_i = the derived numerical value for the ith hidden neural unit (factor) that results from the vector of input variables $x = (x_1, x_2,..., x_8)'$ and the current best estimate of the weights vector $w_i^{(1)}$.

Our model uses three hidden neural units (factors) H_i so that the final output O is obtained by using the hidden factors $H = (H_1, H_2, H_3)$ as "inputs" into the third layer and using the logistic activation function with the linear aggregation function.[10] Thus,

$$O = \frac{1}{1+\exp\{-\eta^{(2)} - H' W^{(2)}\}}, \tag{2}$$

where the weights are now given by $w^{(2)} = (w_1^{(2)}, w_2^{(2)}, w_3^{(2)})$. The artificial intelligence technique learns (by iterative feed forward and back-propagation) to simultaneously select the weights $w_i^{(1)} = (w_{i,1}^{(1)}, w_{i,2}^{(1)},.., w_{i,8}^{(1)})$, and $w^{(2)} = (w_1^{(2)}, w_2^{(2)}, w_3^{(2)})$, and the threshold values $\eta_1^{(1)}, \eta_2^{(1)}, \eta_3^{(1)}, \eta^{(2)}$ in such a manner as to minimize the error of predicting output values in the training sample. Of course, a change in any single weight $w_{i,j}^{(1)}$ changes the value of H_i so that the weights going from the hidden layers to the output also change. Thus, the relationship between the initial input variables and the hidden units is that of a dependent logistic activation relationship between the inputs and the hidden units, and then another logistic relationship between the hidden units and the final outputs. The logistic regression weights (which are quite dependent at each stage) are "learned" in such a manner as to provide the best solvency prediction for each company.

[10] Other computations with four and five hidden units have yielded very similar results; hence, only analysis with three hidden units are presented in this article.

Neural Networks Process

The feed forward network is constructed having the eight financial variables selected previously as input units and the insolvent/solvent variable as the output unit.[11] Two hundred forty-three insurers were used in training sessions consisting of 60 insurers that ultimately became insolvent (all U.S. property-liability insurers that became insolvent during 1991 and 1992), and 183 insurers that remained solvent. To avoid bias and to reduce the effect of regulatory differences between states, the solvent companies were matched with the insolvent companies according to size, line of business, and state of domicile (as well as possible).[12] The list of insolvent insurers was obtained by the Texas State Board of Insurance Research Division through written requests to each state insurance department for a list of all domestic insolvent companies during 1991 or 1992. Follow-up telephone calls ensured complete data records for each insolvency. Accordingly, insolvency data were obtained on firms not included in the A. M. Best list of insolvent firms.[13] NAIC annual statement data for each company two years prior to insolvency were used in the analysis.

The neural network methods described above were modified slightly for use with the insurer financial accounting data in order to better classify the resultant financial distress patterns. The financial variables and solvencies of insurers, although exhibiting patterns, yield pattern predictability less than might be expected with physical science data for which there are objectively "true" patterns governed by physical laws. Because we are dealing with diverse business firm characteristics with a multitude of product lines and concentrations, the relationships between the summary input financial variables and the output solvency results can be similar but not identical from business firm to business firm. This would be somewhat analogous to having a "noisy" data set in the physical sciences. With sample sets of limited size, some special care is needed for the training setup. Below we describe some methodological contributions to neural networks we developed to aid our neural network training and to allow us to verify the results in the financial distress application of this article.

---

[11] It is possible, in theory, for the neural network to deal with a continuous output variable rather than the insolvent/solvent dichotomy we have analyzed here. It is, however, much more difficult for the network to "learn" the relationship between the input patterns of financial variables and potential continuous response variables (e.g., relative risk, dollar drain on the states' guarantee funds, etc.) and would require a much larger data set for training than was possible in this study.

[12] A subset of firms was used in order to facilitate computation. In addition, while the neural networks methodology does not have a particular problem with respect to the sample size disparity between the solvent and insolvent firms, the statistical procedures such as discriminant analysis can be greatly effected if one group is very much larger than the other. Since one purpose of this study was to contrast methods, a subsample of firms was used.

[13] In fact, a search of A.M. Best for ratings on the insolvent firms showed 10 of the 60 identified insolvencies not listed in A.M. Best while 37 of the insolvencies were listed as NA (and not rated). For the solvent firms the corresponding figures were 36 and 43 respectively out of the 243 solvent firms in the sample.

Neural Networks Training Sessions

An unknown parameter of neural networks behavior is the ideal length of training. If there are too many iterations (too long a training session), the network can start "over-learning" the idiosyncrasies of the particular training set at the expense of the ability of the network to generalize to other distinct data configurations. If there are too few iterations (too short a training period), the neural network does not adequately learn to discern the intrinsic relationships between the input patterns and the output variables. One criterion that we use to determine when to stop the training process is to terminate training the network when peak "generalizability" is reached in applying the results to a holdout sample.

To accomplish this, and to overcome some other problems associated with training and testing on the same sample data set, we separated the data into three subsets. A large data set (n = 145), called set T1, represents 60 percent of the sample, and is used for training the network (learning the most effective interconnection weights). Two smaller sets, T2 and T3, each comprised 20 percent of the sample (n = 49 each). These sets were used, respectively, for determining when to cease training the neural network, and for testing the resultant trained neural network. The set T2 determined when to stop training the network according to the following rule: Stop training when the predicting ability of the neural network trained on T1 and as tested on T2 begins to drop (indicating a potential over-training on T1 at the expense of generalizability to T2). The subset T3 was then used to assess the trained network's out-of-sample predictability characteristics (i.e., the ability of the learned network patterns as trained on T1 and stopped on T2 to generalize to the new data set T3).

We developed a sample reuse procedure (similar to a bootstrapping technique) to derive the most useful neural network model.[14] Because of the extensive computing needed for each neural network model (which itself includes several thousands of iterations to effectively learn appropriate neural interconnection weights), ten separate data segmentations were performed. In each session, the data set was randomly partitioned into sets T1, T2, and T3, as described above. After all ten training sessions were completed, the weights obtained from the "best" training session were selected. The criteria for best in this case was the ability of the particular network trained on T1, as stopped according to T2, to generalize to and produce the largest percent correctly

[14] A bootstrapping technique in statistical analysis is one in which the original data set is divided into two subsets: One contains a small number of observations (which could even be a single observation) and the other contains the rest of the sample. The large data set is used to estimate parameters which are then applied to the smaller data set. This can be done repeatedly until all data points have been reflected in the subsample to generate a series of parameters which are then averaged and used in the final analysis for the whole sample. The purpose here is to statistically reduce the bias involved in estimating parameters and testing techniques on the same data set. Bootstrapping also provides a measure of sensitivity of the estimation technique to particular data configurations (e.g., the effect of outliers).

classified when applied to the testing subsample T3. The network model so obtained by using the best training sample interconnection weights was then tested for its ability to discriminate between the solvent and insolvent firms in all of the other nine testing subsets T3.

Neural Networks Prediction Results

The results of applying the neural networks methodology to predict financial distress based upon our selected financial variables show very good abilities of the network to learn the patterns corresponding to financial distress of the insurer. In all cases, the percent correctly classified in the training sample by the network technique is above 85 percent, the average percent correctly classified in the training samples is 89.7 percent, and the bootstrap estimate of the percent correctly classified on the holdout samples is an average of 86.3 percent correctly classified (see Table 3). The bootstrap result shows that the calculated ability of the neural network model to predict financial distress is not merely due to upwardly biased assessments (which might occur when training and testing on the same sample as in the discriminant analysis results).

The ability of the uncovered network structure to generalize to new data sets is also a very important component of the learning process for the network and is crucial to the importance of the results for managerial implementation. The bootstrap estimate of the percent correctly classified on the various testing samples T3 shows that generalizability (an average of 86.3 percent correctly predicted) is obtained. Overall, when applied to the entire sample of 243 firms, the neural network method correctly identified the solvency status of 89.3 percent of the firms in the study. Of the firms which become insolvent in the next two years, the neural network correctly predicted 73 percent, and it correctly predicted the solvency of 94.5 percent of the firms that remained solvent for the two-year duration.

Weight Changes During Training Sessions

Training the network by removing unimportant connections is one way to get a smaller network. A very large change from the initial random weight assigned to the interconnection at the onset is indicative of the network viewing this connection as important to the incremental learning process of the network. The interconnection weights resulting from the training sessions are presented in Table 4, and the change in interconnection weights from the initial random assignment to the final trained weights is presented in Table 5.

The results shown in Table 4 are used as follows: For each firm under investigation, taking the weights from Table 4 and the vector of input variables, one constructs

$$H_i = \frac{1}{1+\exp\{-\eta_i^{(1)} - x' w_i^{(1)}\}} \qquad i = 1, 2, 3.$$

Using the values H_i calculated in this manner for this particular firm and the weights from the hidden layer to the final output, one calculates the output

Table 3
Neural Network Training and Predictive Accuracy

Sample	Predictive Accuracy
Training Sample (T1) Results	
Sample Sizes	145
Percentage Correctly Classified	89.7
Stopping Rule Sample (T2) Results	
Sample Sizes	49
Percentage Correctly Classified	87.8
Bootstrapped Test Sample (T3) Prediction Results	
Sample Sizes	49
Mean Percentage Correctly Classified	86.3
Entire Sample Results	
Sample Size	243
Overall Percentage Correctly Classified	89.3
Percentage of Insolvent Firms Correctly Classified	73.3
Percentage of Solvent Firms Correctly Classified	94.5

Table 4
Connection Weights for Insolvency Prediction

	Training Weights		
Financial Variables	*Unit 1*	*Unit 2*	*Unit 3*
Policyholders' Surplus	0.2513	0.5547	0.6171
Capitalization Ratio	0.2675	0.1027	0.2870
Change in Invested Assets	0.1601	0.1748	0.0886
Investment Yield on Invested Assets	-0.0718	-0.0345	-0.0252
Ratio of Amount Received from Parents to Capital	-0.0929	0.1121	0.2332
Increase in Net Underwriting Loss	0.2783	0.5128	0.2925
Surplus Aid to Surplus (IRIS 3)	0.2783	-0.4051	-0.1550
Liability to Liquid Assets (IRIS 7)	0.0285	-0.0288	-0.1795
Hidden Unit Weights to Output	0.3611	-1.7054	-1.6292
Threshold Layer 2	-0.0881	-0.0179	0.0149
Threshold Layer 3	0.6180		

value O for this particular firm, which can be viewed as an assessment of the probability of insolvency. If the probability of insolvency is greater than 0.5, insolvency is predicted; otherwise, solvency is predicted for the next two years. To illustrate, the first firm in the test sample listed in Table 3 has a value of 0.08664 calculated in this manner and is predicted to be solvent for the next two years based upon the annual report data. The firm was, in fact, solvent for this period and the prediction was correct.

Table 5
Weight Change from Initial Random Weight
To Final Optimally Trained Connection Weights

	Weight Change		
Financial Variables	*Unit 1*	*Unit 2*	*Unit 3*
Policyholders' Surplus	0.0000	0.3802	0.3541
Capitalization Ratio	-0.0004	-0.0748	0.1422
Change in Invested Assets	0.0000	-0.0432	-0.0104
Investment Yield on Invested Assets	0.0000	0.0098	0.0040
Ratio of Amount Received from Parents to Capital	0.0000	-0.0072	0.0021
Increase in Net Underwriting Loss	-0.0001	0.5822	0.2459
Surplus Aid to Surplus	0.0000	-0.5683	-0.2537
Liability to Liquid Assets	-0.0039	0.0613	-0.1660
Change in Threshold of Input Layer to Hidden Layer	0.0000	0.0227	0.0252
Change in Hidden Layer to Output Layer Weights	0.4081	-1.711	-1.6379
Change in Threshold of Hidden Layer to Output Layer	0.4075		

Turning to Table 5, we see that there is a potential simplification of the network. Note that there is virtually no change from the initial random weight assigned to connections to Hidden Unit 1, indicating that a two-hidden-unit network might be attempted in order to simplify the network topology for subsequent use.

Comparison with Other Methods

In order to compare the results of the neural network model to other methods, several additional classification techniques were examined. We performed a linear discriminant analysis using the same eight variables obtained from the stepwise logistic regression procedure described above. Such a discriminant analysis approach assumes that the explanatory variables in each group form a multivariate normal distribution with the same covariance matrix for the two groups.

The results of the discriminant analysis are presented in Table 6. As shown, the statistical method, although not doing as well as the neural networks methodology, showed a respectable 85 percent correct classification for insolvent firms and an 89.6 percent correct classification rate for solvent firms.[15]

An additional analysis was performed using the IRIS ratio system promulgated by the NAIC. These results are presented in Table 7. The IRIS system correctly identifies only 26 insolvencies out of the 60 firms that eventually became insolvent within two years, or 43.3 percent of the insolvent companies.

[15] It should be remembered that these rates for the discriminant analysis are upwardly biased, since the model parameters were estimated on the same data set as the prediction. Unbiased estimates of the percent correctly classified would have compared less favorably with the neural network model.

Table 6
Discriminant Analysis Classification

	Predicted Status		
True Status	*Insolvent*	*Solvent*	*Total*
Insolvent	51/60 = 85%	9/60 = 15%	60
Solvent	19/183 = 10.4%	164/183 = 89.6%	183
Total	70	173	243

Table 7
National Association of Insurance Commissioners'
Insurance Regulatory Information System Ratio Classification Table

	Predicted Status		
True Status	*Insolvent*	*Solvent*	*Total*
Insolvent	26/60 = 43.3%	34/60 = 56.7%	60
Solvent	6/183 = 3.3%	177/183 = 96.7%	183
Total	70	173	243

Table 8
A.M. Best Ratings

	A+	*A*	*A-*	*B+*	*B*	*B-*	*C+*	*C*	*C-*	*Not Rated*	*Total*
Insolvent Companies	1	3	1	2	0	1	4	1	0	47	60
Solvent Companies	56	22	17	4	3	1	1	0	0	79	183

On the other hand, a remarkable 96.7 percent of the solvent firms were correctly identified.[16]

Yet another source of frequently used insurer rating information (particularly by insurance purchasers as opposed to regulators) is the A. M. Best Company. Best's ratings for our data set are presented in Table 8.

The A. M. Best rating was not very worthwhile for predicting insolvencies among U.S. property-liability insurers. This is partly because of the very large percentage of nonrated companies (78.3 percent) in the insolvent group. A large subset of solvent firms (43.2 percent) also were not rated by A. M. Best.

Conclusions and Further Directions

Even with the limited data and time frame used in this study, neural network artificial intelligence methods show significant promise for providing useful early warning signals and clearly dominate the current IRIS system for

[16] Unfortunately, because the overwhelming majority of firms in the real competitive marketplace are solvent, a rough calculation using Bayes theorem shows that a positive result on the IRIS ratio insolvency test yields a posterior probability of less than one-third that the firm will actually become insolvent.

solvency monitoring and early warning. In addition, other characteristics positively differentiate neural network models from alternative techniques. For one, the neural network model can be updated (learning supplemented) without completely retraining the network. The current version of the interconnection weights can be used as a starting point for the iterations once new data become available. Accordingly, the system of learning can adapt as different economic influences predominate. In essence, the neural network model can evolve and change as the data, system, or problem changes. Other "nonlearning" or static models do not have these built-in properties.

Despite the positive classification results obtained, we still believe that further study is warranted. Obviously, another study should be conducted using life/health insurers and their particular collection of pertinent input variables. There are also several avenues of research that might potentially enhance the predictability, the robustness, or the interpretability of the neural networks approach. These avenues include the development and inclusion of qualitative data (e.g., complaints received data, managing general agent identification, quality of reinsurance) and trend data (such as changes in product mix offerings over time), which could add significantly to the robustness and accuracy of the model developed. In addition, because of states' different economic climates and regulatory requirements, a comparison of the appropriate models for insolvency prediction in different states (and nationwide) should be investigated to ascertain the impact of certain state-controlled regulatory requirements. (In fact, our investigations with Texas domestic insolvencies during the period 1987 through 1990 showed the neural network model correctly predicting 95.5 percent of the insurers with respect to their ultimate insolvency three years hence). These results would then suggest public policy directives concerning these issues for the purpose of decreasing insolvency propensity.

One further avenue of research relates to minimizing the financial effects of insurer insolvency on the state guarantee funds rather than minimizing the number of incorrectly identified firms. Since not all insolvencies are equally costly, one may introduce a "cost of misclassification" variable into the analysis for each firm. The numerical value used for the error of misclassification for the solvent firms could be the audit costs, while the numerical value used for the error of misclassification for the insolvent firms could be an estimated charge to the guarantee fund of the insolvency.[17] This numerical value for the error could be used in the feed forward, back-propagation algorithm in order to minimize total costs of regulation in a manner similar to that outlined in this article.

[17] Actuarial methods for predicting the size of the first negative surplus might also be used if appropriate data for guarantee fund demand is not available at the firm level.

Appendix

The Feed Forward, Back-Propagation Algorithm

We summarize here the feed forward, back-propagation algorithm for a neural network with three layers as used in this article. Moreover, as in this article, we shall consider only a single output node O. The more general situation of a network with M layers with multiple output nodes can be found in Hertz, Krogh, and Palmer (1991).

Let $w_{i,j}^{(1)}$ be the interconnection weight between unit i of the hidden layer and the input variable x_j from layer 1, and $w_i^{(2)}$ denote the interconnection weight between $H_i = F(\eta_i^{(1)} + x'w_i^{(1)})$, the ith neural unit in layer 2 (the hidden layer which is the output from layer 1), and the output variable O. The threshold values $\eta_i^{(1)}$ and $\eta^{(2)}$ are defined for each layer, as above, with the activation function F defined via the logistic function

$$F(z) = \frac{1}{1+\exp(-\eta - z)}.$$

The actual back-propagation algorithm proceeds by taking each insurer's response pattern, u, one at a time and incrementally updating the interconnection weights. This is formalized as follows:

Step 1: Initialization

Set the weights $w_i^{(2)}$, $w_{i,j}^{(1)}$, $\eta_i^{(1)}$, and $\eta^{(2)}$ all equal to small random values.

Step 2: Feed Forward

At the input layer, let x^u be a vector of inputs corresponding to a pattern (a response vector for the respondent) u. For each unit i of the hidden layer, compute $H_i = F(\eta_i^{(1)} + x^{u\prime}w_i^{(1)})$, and then compute the predicted output $\hat{O}^u = F(\eta^{(2)} + H'w^{(2)})$ at the output layer.

Step 3: Propagate Backward

Let O^u be the "target" or actual output of the respondent with input pattern u.

At output layer, compute $\delta_i^{(2)} = F'(\eta^{(2)} + H'w^{(2)})\,(O^u - \hat{O}^u)$.

At hidden layer, calculate $\delta_i^{(1)} = F'(\eta_i^{(1)} + x^{u\prime}w_i^{(1)})w_i^{(2)}\delta_i^{(2)}$, for all i.

These computations are facilitated by the formula $F' = F\cdot(1\text{-}F)$, which holds for the logistic activation function.

Step 4: Update Weights

First compute $\Delta w_i^{(2)} = \alpha\delta_i^{(2)}H_i$. Then, in order to update the connection weights between hidden layer and output layer, use the formula $w_i^{(2)new} = w_i^{(2)old} + \Delta w_i^{(2)}$.

Similarly, to update all connection weights between the input layer and hidden layer, compute $\Delta w_{i,j}^{(1)} = \alpha\delta_i^{(1)}x_j^u$ and update the weights via the equation $w_{i,j}^{(1)new} = w_{i,j}^{(1)old} + \Delta w_{i,j}^{(1)}$.

The parameter α in the above expressions is a learning rate parameter.

Step 5: Repeat

Go back to Step 2 and repeat the above process for the next respondent's data pattern. When all respondents have been analyzed in the above manner, start over with the first respondent, again updating the weights as necessary to reduce the average disparity between O^u and $\hat{O}^u$. The iterations cease according to the stopping rules outlined in the article.

References

Altman, E. I., 1968, Financial Ratios, Discriminant Analysis and the Prediction of Corporate Bankruptcy, *Journal of Finance,* 23(4): 589-609.

Altman, E. I., R. G. Haldeman, and P. Narayanan, 1977, ZETA Analysis: A New Model to Identify Bankruptcy Risk of Corporations, *Journal of Banking and Finance,* 1(1): 29-54.

Ambrose, Jan Mills and J. Allen Seward, 1988, Best's Ratings, Financial Ratios and Prior Probabilities in Insolvency Prediction, *Journal of Risk and Insurance,* 55: 229-244.

BarNiv, Ran and Robert A. Hershbarger, 1990, Classifying Financial Distress in the Life Insurance Industry, *Journal of Risk and Insurance,* 57: 110-136.

BarNiv, Ran and James B. MacDonald, 1992, Identifying Financial Distress in the Insurance Industry: A Synthesis of Methodological and Empirical Issues, *Journal of Risk and Insurance,* 59: 543-573.

Barnoff, Etti, 1993, Financial Analysis of the Life and Health Insurance Industry: A Disaggregated and Clustered Approach, Unpublished Ph.D. dissertation, University of Texas at Austin, Department of Finance.

Barrese, James, 1990, Assessing the Financial Condition of Insurers, *CPCU Journal,* 43(1): 37-46.

Bierman, H. Jr., 1960, Measuring Financial Liquidity, *Accounting Review,* 35: 628-632.

Borch, Karl, 1970, The Rescue of an Insurance Company After Ruin, *ASTIN Bulletin,* 6(Part 1): 66-69.

Brockett, Patrick L., Linda L. Golden, and Utai Pitaktong, 1993, Methodological Contribution to Neural Network Interpretation, Working Paper, University of Texas at Austin, Graduate School of Business.

Charnes, A., 1963, The Geometry of Convergence of Simple Perceptions, *Journal of Mathematical Analysis and Applications,* 7(3): 475-481.

Cummins, J. David, Scott E. Harrington, and Robert Klein, 1994, Insolvency Experience, Risk-Based Capital, and Prompt Corrective Action in Property-Liability Insurance, *Journal of Banking and Finance,* forthcoming.

Eberhart, Russell C. and Roy W. Dobbins, 1990, *Neural Network PC Tools: A Practical Guide* (New York: Academic Press).

Golden, Linda L., Patrick L. Brockett, and Utai Pitaktong, 1993, Using Artificial Intelligence Methods to Predict Patronage Behavior from Retail Store

Image Variables, Working Paper, University of Texas at Austin, Graduate School of Business.

Grace, Martin, Scott Harrington, and Robert Klein, 1993, Risk-Based Capital Standards and Insurer Insolvency Risk: An Empirical Analysis, Paper presented to the 1993 Annual Meeting of the American Risk and Insurance Association.

Harrington, Scott and Jack M. Nelson, 1986, A Regression-Based Methodology for Solvency Surveillance in the Property-Liability Insurance Industry, *Journal of Risk and Insurance,* 53: 583-605.

Hebb, D. O., 1949, *The Organization of Behavior* (New York: John Wiley).

Hecht-Nielsen, R., 1988, Neurocomputing: Picking the Human Brain, *IEEE Spectrum,* 25: 36-41.

Hertz, John, Anders Krogh, and Richard G. Palmer, 1991, *Introduction to Theory of Neural Computation* (Redwood City, Cal.: Addison-Wesley).

Hopfield, J. J., 1982, Neural Networks and Physical Systems with Emergent Collective Computational Abilities, *Proceedings of the National Academy of Science,* 79: 2554-2558.

Lippmann, R. P., 1987, An Introduction to Computing with Neural Nets, *IEEE ASSP Magazine,* 1: 4-22.

Lorentz, G. G., 1976, The 13th Problem of Hilbert, in: F. E. Browder, ed., *Mathematical Developments Arising from Hilbert's Problems* (Providence, R.I.: American Mathematical Society).

Minsky, M. and S. Papert, 1969, *Perceptrons: An Introduction to Computational Geometry* (Boston: MIT Press).

Neti, Chalapathy, Michael H. Schneider, and Eric D. Young, 1992, Maximally Fault Tolerant Neural Networks, *IEEE Transactions on Neural Networks,* 30(1): 12-23.

Quirk, J. P., 1961, The Capital Structure of Firms and the Risk of Failure, *International Economic Review,* 2(2): 210-228.

Roberts, J. A., 1988, Increase Rates of Convergence Through Learning Rate Adaptation, *Neural Networks,* 1: 295-307.

Rosenblatt, F., 1959, Two Theorems of Statistical Separability in the Perceptron, *Proceedings of a Symposium on the Mechanization of Thought Processes* (London: Her Majesty's Stationery Office), 421-456.

Rosenblatt, F., 1962, *Principles of Neurodynamics: Perceptron and the Theory of Brain Mechanisms* (Washington, D.C.: Spartan Books).

Rumelhart, D. E., G. E. Hinton, and R. J. Williams, 1985, Learning Internal Representations by Error Propagations by Error Propagation, *ICS Report 8506* (La Jolla: Institute for Cognitive Science, University of California at San Diego).

Salchenberger, Linda M., E. Mime Cinar, and Nicholas A. Lash, 1990, Using a Neural Network to Predict Savings and Loan Failures, Working Paper 90-09, Loyola University School of Business, Chicago.

Santomero, A. M. and J. D. Vinso, 1977, Estimating the Probability of Failure for Commercial Banks and the Banking System, *Journal of Banking and Finance,* 1(2): 185-205.

Sinkey, J., 1975, A Multivariate Statistical Analysis of the Characteristics of Problem Banks, *Journal of Finance*, 30(1): 21-36.

Tinsley, P. A., 1970, Capital Structure, Precautionary Balances and the Valuation of the Firm: The Problem of Financial Risk, *Journal of Financial and Quantitative Analysis*, 5(1): 33-62.

Vinso, J. D., 1979, A Determination of the Risk of Ruin, *Journal of Financial and Quantitative Analysis*, 14(1): 77-100.

Widrow, B., 1962, Generalization and Information Storage in Networks of Madaline Neuron, in: M. Yovitz, G. Jacobi, and G. Goldstein, eds., *Self-Organizing Systems* (Washington, D.C.: Spartan Books), 435-461.

Williams, W. H. and M. L. Goodman, 1971, A Statistical Grouping of Corporations by Their Financial Characteristics, *Journal of Financial and Quantitative Analysis*, 6(4): 1095-1104.