A Fuzzy Neural Network Model for Bankruptcy Prediction

Dr. S.K. Sudarsanam, Professor and Program Chair, VIT Business School Chennai, VIT University, India.

Abstract

Prediction of Bankruptcy has been an important topic for central banks of many countries. Altman’s Z-score model is extremely popular ever since the introduction in 1968. The Altman’s Z-score model is a linear analysis of five ratios, which are all weighted objectively. The summation of the five ratios forms the Z-score. A comparative study of Z-Score model, K-nearest neighbour, ID3, Linear regression model and Neural network model proved that the Neural network model was better than other models. This paper aims to prove that the hybrid fuzzy neural network model works better than other models in bankruptcy prediction.

Keywords: Fuzzy Logic, Neural Network, Neural Fuzzy Design Toolbox, MatLab 8.0, Fuzzy Controller, Rule Viewer, Image Viewer, Rule editor, Bankruptcy, Z-Score, Fuzzy Neural Network Model.

Introduction

Bankruptcy is a legal status of a person or other entity that cannot repay the debts it owes to creditors. In most jurisdictions, bankruptcy is imposed by a court order, often initiated by the debtor. Bankruptcy prediction is very important for central banks of many countries. By having this prediction, Central banks can better advise organizations on their risks and performance. Also, if Organizations also can measure this, it would help them to improve their financial and operational efficiency. The problem of bankruptcy prediction is one of classification. Discriminant analysis, logit and probit models have been typically used for classification. They have restrictive assumptions such as linearity, normality, independence among the criterion variables and the predictor variable is a linear form of criterion variables. (G. Zhang et al.). Altman’s Z-score model is extremely popular ever since the introduction in 1968. The Altman’s Z-score model is a linear analysis of five ratios, which are all weighted objectively. Artificial Neural Networks are powerful tools for pattern recognition and pattern classification due to their non-linear, non-parametric adaptive learning processes. Several researchers (Lacher et al., Sharda and Wilson, Tam and Kiang) have used ANN techniques and concluded that the accuracy of the prediction is much better than the traditional statistical methods. G. Zhang et al. have compared the ANN methods with Logistic regression and have proved that the ANN methods give better results by cross-validating both training and test data. They have also established a process of predicting better using Neural network methods. In this paper we would use Fuzzy neural network methods for bankruptcy prediction. We will be using the MATLAB – Neuro fuzzy design toolbox to design the Neural network, design the fuzzy controller, load the training and testing data, train the neural network and then test the neural network on the testing data to predict the bankruptcy.

Objective of Study

To predict the bankruptcy of Organizations using the hybrid Fuzzy neural network methods.

Methodology of Study

The steps involved are:

a) identify the sample data of Organizations
b) Collect the data for the five financial ratios used in Altman Z-Score
c) Identify the training and test data inputs
d) Load the data into the Neuron-fuzzy design toolbar of Mat lab
e) Generate the Fuzzy Inference System (Sugeno Controller and Sub-clustering option)
f) Train the network using hybrid option
g) Test the test data against the trained network
h) Calculate the prediction percentage for both bankrupt and non-bankrupt organization
i) Compare the results with Altman Z-Score and Logistic Regression

A brief history of Bankruptcy Prediction

A firm can go bankrupt when the total liabilities exceed a fair value of the total assets of that firm. Dun and Bradstreet (1980) identified earlier that lack of experience, unbalanced experience, or just plain incompetence of the management team was the cause of firm failures in more than 44% of the situations. They have also identified that the 50% of all failures occur with firm with ages between two and five.

The Z-score formula for predicting bankruptcy was published in 1968 by Edward I. Altman, Assistant Professor of Finance at New York University. The formula
may be used to predict the probability that a firm will go into bankruptcy within two years. Z-scores are used to predict corporate defaults and an easy-to-calculate control measure for the financial distress status of companies in academic studies. The Z-score uses multiple corporate income and balance sheet values to measure the financial health of a company.

The original data sample consisted of 66 firms, half of which had filed for bankruptcy under Chapter 7. All businesses in the database were manufacturers, and small firms with assets of < $1 million were eliminated.

The original Z-score formula was as follows:[1]

\[ Z = 1.2X_1 + 1.4X_2 + 3.3X_3 + 0.6X_4 + 0.99X_5. \]

\[ X_1 = \text{Working Capital} / \text{Total Assets}. \text{Measures liquid assets in relation to the size of the company.} \]

\[ X_2 = \text{Retained Earnings} / \text{Total Assets}. \text{Measures profitability that reflects the company's age and earning power.} \]

\[ X_3 = \text{Earnings Before Interest and Taxes} / \text{Total Assets}. \text{Measures operating efficiency apart from tax and leveraging factors. It recognizes operating earnings as being important to long-term viability.} \]

\[ X_4 = \text{Market Value of Equity} / \text{Book Value of Total Liabilities}. \text{Adds market dimension that can show up security price fluctuation as a possible red flag.} \]

\[ X_5 = \text{Sales} / \text{Total Assets}. \text{Standard measure for total asset turnover (varies greatly from industry to industry).} \]

Altman found that the ratio profile for the bankrupt group fell at -0.25 avg, and for the non-bankrupt group at +4.48 avg. An Organization is predicted to be financially sound if the Z-Score is more than 2.9 and likely to go bankrupt if it is below 1.8. Altman's Z-score is a customized version of the discriminant analysis technique of R. A. Fisher (1936).

Ohlson employed logistic regression to predict company failure [10], to overcome the limitations of Altman. This model consists of the following ratios:

\[ Z = \frac{1}{1 + \exp \left( - (a X_{1,4} + b_{1,4} X_{2,4} + b_{2,4} X_{3,4} + b_{3,4} X_{4,4} + b_{4,4} X_{5,4}) \right)} \]

where,

\[ Z = \text{the probability of bankruptcy for a firm} \]

\[ TLTA = \text{Total liabilities divided by total assets.} \]

\[ WCTA = \text{Working capital divided by total assets.} \]

\[ CLCA = \text{Current liabilities divided by current assets.} \]

\[ OENEG = 1 \text{ If total liabilities exceed total assets, 0 otherwise.} \]

\[ NITA = \text{Net income divided by total assets.} \]

\[ FUTL = \text{Funds provided by operations (income from operation after depreciation) divided by total liabilities.} \]

\[ INTWO = 1 \text{ If net income was negative for the last 2 years, 0 otherwise.} \]

\[ CHIN = (NIt - NIt_{-1}) / (|NIt| + |NIt_{-1}|), \text{ where NIt is net income for the most recent period. The denominator acts as a level indicator. The variable is thus intended to measure the relative change in net income} \]
The following table gives the various models developed for Bankruptcy prediction.

<table>
<thead>
<tr>
<th>Model</th>
<th>Variable</th>
<th>Ratios</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multiple – Discriminant analysis</td>
<td>X1</td>
<td>Net working capital / total assets</td>
</tr>
<tr>
<td>Altman (1968)</td>
<td>X2</td>
<td>Retained earnings/total assets</td>
</tr>
<tr>
<td></td>
<td>X3</td>
<td>Earnings before interest and taxes/total assets</td>
</tr>
<tr>
<td></td>
<td>X4</td>
<td>Market value of equity/book value of total liabilities.</td>
</tr>
<tr>
<td></td>
<td>X5</td>
<td>Sales/total assets</td>
</tr>
<tr>
<td>Logit model</td>
<td>TLTA</td>
<td>Total liabilities divided by total assets</td>
</tr>
<tr>
<td>Ohlson (1980)</td>
<td>WCTA</td>
<td>Working capital divided by total assets</td>
</tr>
<tr>
<td></td>
<td>CLCA</td>
<td>Current liabilities divided by current assets</td>
</tr>
<tr>
<td></td>
<td>NITA</td>
<td>Net income divided by total assets</td>
</tr>
<tr>
<td></td>
<td>FULTL</td>
<td>Funds provided by operations (income from operation after depreciation) divided by total liabilities</td>
</tr>
<tr>
<td>Probit model</td>
<td>NITL</td>
<td>Net income divided by total liabilities</td>
</tr>
</tbody>
</table>

| Zmijewski (1984)             | TLTA     | Total liabilities divided by total assets                              |
|                             | CACL     | Current assets divided by current assets                              |
| Hazard model                | NITL     | Net income divided by total liabilities                                 |
| Shumway (2001)              | TLTA     | Total liabilities divided by total assets                              |

The "Classification and Regression Trees" (CART) methodology is introduced by Breiman et al. (1984) and is a classification technique which uses supervised learning to build a decision tree. Decision trees are represented by a set of questions which split a data set into smaller and smaller parts. As a result decision trees partition a data set into mutually exclusive regions. When decision trees are used for classification problems, these trees are called classification trees. If it concerns a regression problem then they are often called regression trees. CART is a technique which combines both of them. The CART algorithm only asks yes or no questions like: “Is the (earnings / total assets) > x?”. Then it would split the input variables into two groups and then proceed to the next input variable and so on.

Also, researchers have used Logistic regression, Fuzzy inference system, Neural network models and genetic algorithms to predict Bankruptcy.

For our study, we would be using the Altman Z-Score variables and process them through Fuzzy interface system and Neural network models and then predict the Bankruptcy of the organization.

Fuzzy Systems and Neural Networks

FUZZY LOGIC is a situation where you’re unable to say Yes or No because you need more information. You have all the information you need. The situation itself makes either Yes or No inappropriate. If you answered One, or Some, or A Few, or Mostly—all of which are fuzzy answers, somewhere in between Yes and No. They handle the actual ambiguity in descriptions or presentations of reality. With fuzzy logic the answer is Maybe, and its value ranges anywhere from 0 (No) to 1 (Yes).

Crisp value takes - No or Yes only

Fuzzy value - No Slightly Somewhat Sort Of A Few Mostly Yes, Absolutely

Zadeh (1965) proposed the fuzzy set theory which is an important concept to deal with Uncertainty based information.

The main components associated with fuzzy systems are:

- Fuzzification
- Fuzzy rule base
Defuzzification.

**Fuzzification**: Fuzzification refers to transformation of crisp inputs into a membership degree which expresses how well the input belongs to the linguistically defined terms. Experts judgement and experience can be used for defining the degree of membership function for a particular variable. During Fuzzification, a fuzzy logic controller receives input data, also known as the fuzzy variable, and analyzes it according to user-defined charts called membership functions (Klir and Yuan, 1995).

**Fuzzy rule base**: The rule base describes the criticality level of the system for each combination of input variables. Often expressed in ‘If-Then’, they are formulated in linguistic terms using two approaches (i) Expert knowledge and expertise (ii) Fuzzy model of the process (Zimmermann, 1996). Experts judgement and experience can be used for define degree of membership function for a particular variable

**Defuzzification**: The defuzzification process examines all of the rule outcomes after they have been logically added and then computes a value that will be the final output of the fuzzy controller. During defuzzification, the controller converts the fuzzy output into a real-life data value.

A Fuzzy controller is used to convert the five fuzzy inputs into a fuzzy output. The Sugeno controller provides better results than Mamdani Controller. The inputs considered are

\[
\begin{align*}
X_1 &= \text{Working capital / total assets}, \\
X_2 &= \text{Retained earnings / total assets}, \\
X_3 &= \text{EBIT / total assets (where EBIT is earnings before interest and taxes)}, \\
X_4 &= \text{MVE / total debt (where MVE is the market value of equity and total debt is book value of total liabilities)} \\
X_5 &= \text{Sales / total assets. The inputs, controller and output are designed using the neuro-fuzzy design tool of MatLab.} \\
\end{align*}
\]

Fuzzy membership functions for input and output variables and the fuzzy rule engine are generated by the Neuro-fuzzy design toolbox.

Artificial neural networks (ANNs) are simplified models of the interconnections between cells of the brain. They are defined by Wasserman and Schwartz (1987) as "highly simplified models of the human nervous system, exhibiting abilities such as learning, generalization and abstraction.”

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 numerous financial activities such as forecasting financial distress in firms. A general description of neural networks is found in Rummelhart, Hinton and Williams (1986).

Neural networks are typically organized in layers. Layers are made up of a number of interconnected ‘nodes’ which contain an ‘activation function’. Patterns are presented to the network via the ‘input layer’, which communicates to one or more ‘hidden layers’ where the actual processing is done via a system of weighted ‘connections’. The hidden layers then link to an ‘output layer’ where the answer is output as shown in the graphic below.

A Neural Network model
As per Josef Thomas Burger, most ANNs contain some form of 'learning rule' which modifies the weights of the connections according to the input patterns that it is presented with. In a sense, ANNs learn by example as do their biological counterparts. Although there are many different kinds of learning rules used by neural networks, this demonstration is concerned only with one; the delta rule. The delta rule is often utilized by the most common class of ANNs called 'backpropagational neural networks' (BPNNs). Backpropagation is an abbreviation for the backwards propagation of error.

Within each hidden layer node is a sigmoidal activation function which polarizes network activity and helps it to stabilize. Backward propagation performs a gradient descent within the solution's vector space towards a 'global minimum' along the steepest vector of the error surface. The global minimum is that theoretical solution with the lowest possible error. The error surface itself is a hyperparaboloid in shape. In most problems, the solution space is quite irregular with numerous 'pits' and 'hills' which may cause the network to settle down in a 'local minum' which is not the best overall solution.

Since the nature of the error space cannot be known a priori, neural network analysis often requires a large number of individual runs to determine the best solution. Most learning rules have built-in mathematical terms to assist in this process which control the 'speed' (Beta-coefficient) and the 'momentum' of the learning. The speed of learning is actually the rate of convergence between the current solution and the global minimum.

Once the Neural network is trained at sufficient epoch levels with low error tolerance, the network can be used to test and predict for new data. The data is processed through the hybrid network (with fuzzy inference system and rules and trained neural network).

With the delta rule, as with other types of backpropagation, 'learning' is a supervised process that occurs with each cycle or 'epoch' (i.e. each time the network is presented with a new input pattern) through a forward activation flow of outputs, and the backwards error propagation of weight adjustments. More simply, when a neural network is initially presented with a pattern it makes a random 'guess' as to what it might be. It then sees how far its answer was from the actual one and makes an appropriate adjustment to its connection weights. More graphically, the process looks something like

New inputs are presented to the input pattern where they filter into and are processed by the middle layers as though training were taking place, however, at this point the output is retained and no backward propagation occurs. The output of a forward propagation run is the predicted model for the data which can then be used for further analysis and interpretation.

Bankruptcy Prediction using Neuro-Fuzzy Model:

The following are the steps involved in the Bankruptcy prediction using neuro-fuzzy design toolbox of Matlab.

a) Collection of data from CMIE (centre for monitoring Indian economy) database: The data is collected for 125 companies (including some bankrupt organisations) for the input variables. The inputs considered are Working capital / total assets, Retained earnings / total assets, EBIT / total assets, MVE / total debt and Sales / total assets.

b) Partition the collected data into training data and testing data and load both the data into Load data frame of the ANFIS (neuro fuzzy inference system) – using the neuro fuzzy design toolbar of MATLAB

c) Generate the Fuzzy membership functions for the input variables, use the sugeno fuzzy controller and generate the fuzzy rules for using the Generate FIS frame. The Sub Clustering radio button option is chosen as previous study shows Sub-Clustering option generates better fuzzy inference system.
d) Train the FIS model: The two ANFIS parameter optimization method options available for FIS training are backward propagation and Hybrid. Hybrid option is chosen as it involves both backward propagation and mixed least squares method. Error Tolerance is used to create a training stopping criterion, which is related to the error size. The training will stop after the training data error remains within this tolerance. This is best left set to 0 as we don’t know how training error is going to behave. The number of training epochs is set to 30, it can be increased to improve the results. The training will stop after the training data error remains within this tolerance.

e) Finally the test FIS frame is used to test the test data for bankruptcy prediction.

The following provide snapshots of the ANFIS model in action.

Membership function editor:
As we can observe the prediction of non-bankrupt organization is 89% (only one data point not predicted correctly) and the prediction of bankrupt organization is 100%.

G. Zhang et al. have compared the Neural network with logistic regression and the results are shown below.
As a comparison between Logistic, Neural Network and Fuzzy Neural Network we observe that for the data we have chosen from CMIE, Fuzzy Neural Network performs better than the other models.

**Conclusion and Future Research directions:** The Fuzzy Neural Network model is able to predict the Bankruptcy of Organizations (based on the five input variables of Altman) better than the regression models, ANN models and other models for the 125 organizations considered in this paper. The following are the suggestions to be considered for future research:

a) A combination of genetic algorithm and NFIS (neuro fuzzy inference system) can be considered for bankruptcy prediction

b) GENFIS 3 can be considered (instead of GENFIS 2 used in this paper) for the fuzzy membership function generation
c) Neural network methods can be improved
d) Other financial models for bankruptcy prediction can be considered, trained and tested with the NFIS model

**References**


