Association for Information Systems
AIS Electronic Library (AISeL)

AMCIS 1996 Proceedings

Americas Conference on Information Systems (AMCIS)

8-16-1996

A NEURAL NETWORK APPROACH TO FORECASTING EARNINGS PER SHARE

W. Raghupathi
Department of Accounting and Management Information Systems, California State University, rp@ecst.csuchico.edu

A NEURAL NETWORK APPROACH TO FORECASTING EARNINGS PER SHARE

W. Raghupathi
Department of Accounting and Management Information Systems
California State University, Chico, CA 95929-0011
Phone: (916) 898-4825 Fax: (916) 898-4584
Email: rp@ecst.csuchico.edu

Abstract

This paper explores the potential of neural networks to forecast earnings per share. The neural network would serve as a decision support system for finance managers, stock brokers and investment analysts and investors. Results of experiments with training/testing indicate that neural networks appear to be promising in forecasting EPS. Further investigations are necessary.

Introduction

Within the realm of finance, the earnings per share (EPS) variable is calculated as an indication of the corporate earnings per outstanding share. The EPS serves as an important variable for investment performance evaluation. It is hypothesized that a neural network model provides the end user with concise, timely, and accurate forecasts of EPS. The value of the neural network for the purpose of predicting earnings per share includes pattern recognition, learning, generalization and abstraction, and interpretation of incomplete and noisy data. As a financial model, the neural network possesses an ability to recognize patterns and interdependencies based on specific variables. The purpose of this project is to develop a neural network capable of forecasting earnings per share of Fortune 500 companies.

Literature Review

Neural networks are being utilized by many financial decision makers for various types of forecasting decisions. Drawing trends based on previous data sets is a valuable financial and behavioral prediction mechanism. Neural networks are used by the Swiss Bank to predict investment opportunities based on movement of foreign exchange rates, interest rates, and stock and commodity prices [Stewart, 1994]. Pareto Partners, a mutual fund manager with total funds valued at $12 billion, utilizes a neural network in conjunction with an expert system to track and forecast financial market indicators in several global markets simultaneously [Blubaugh, 1995]. The value of the neural network in these applications is its ability to generate, based on trend recognition, financial performance predictions.

Methodology

The methodology was an iterative process of constructing and developing a neural network capable of predicting earnings per share. It included variable selection, data collection and normalization, model development, selection of implementation tool, training the network, testing the network, performing experiments, and finally, selection of the best model.

Variable Selection: the variable selection phase entailed researching the several variables utilized in the compilation of Fortune Magazine's annual Fortune 500. Input variables were selected to develop interdependencies between each variable and earnings per share.. Revenue, profits, assets stockholders' equity, and market value. EPS, the output variable equal to the dollar earnings divided by the number of outstanding shares. The variables selected are financial indicators hypothesized to be correlated to the performance of EPS.

Data Collection and Normalization: data collection consisted of extracting variable information from the Fortune 500 as published in Fortune Magazine, May 15, 1995. The top 100 firms of the Fortune 500 were to be utilized for the network's training and testing set. However, upon further examination it was discovered that several of the top 100 firms did not report information necessary for the selected variables. For this reason, several data records were extracted from companies rated 101-150. Several companies in the top 100 companies reported negative valued variables such as earnings and profits. These negative values were included in the data set to maintain an accurate portrayal of the problem domain. The data was further normalized.

Model Development: the model development included identifying the five input nodes, the number of hidden layers, and the output node. The five input nodes were set equal to the five variables previously defined above: revenues, profits, assets, stockholder's equity, and market value. These five inputs would establish connections with the neurons in the hidden layer and generate weights to reach the desired output value. The EPS output value will be forecast utilizing backpropagation as our learning algorithm Backpropagation is a widely used method of supervised learning which will feedforward adjusting weighted connections as necessary to reach the desired output value of earnings per share. Brainmaker was used to train and test the neural network models.

Training and Testing Results: Experiments 1-7 held the number of hidden layers and the number of nodes in hidden layers constant. Varying levels of training and testing tolerances were used and the resulting impacts on good/bad ratios were studied. The training and testing tolerances were varied from a low of.05, or +-5%, to a high of.30, or +-30%. It became apparent that the high level training tolerance of +-30% was too high to measure the accuracy level desired. The low level training tolerance of +-5% resulted in 29 good data items and 21 bad data items for a resulting training ratio of 58% good. The +- 10% training tolerance level illustrated in experiment number one resulted in 48 good data items and 2 bad data items for a resulting training ratio of 96% good. The high level testing tolerance of +-30% resulted in a near perfect testing ratio of 98% good. The low level testing tolerance of +-5% resulted in a testing ratio of 46% good. It can be concluded that as the training and testing tolerances were increased the number of good forecasts increased.

Experiments 8-11 maintained the training and testing tolerances both constant at .10, or +10%. The objective of these experiments was to determine the effect of the number of hidden layers and resulting number of neurons involved with the network. In theory, the number of hidden layers and resulting number of neurons increases the number of weighted connections of the network. Too many hidden layers may generate such a great number of interactions that the network is incapable of learning. Similarly, too few hidden layers may not allow the network to learn based on a lack of weighted interactions. The low level of hidden layers included two hidden layers with the default number of neurons at ten per layer. The number of layers were cumulatively increased by one in each of experiments 9, 10, and 11. The impact of the number of hidden layers cannot be accurately determined, although it is interesting to note the testing results do decrease as the number of layers increase.

Experiments 12-16 maintain the training and testing tolerances both constant at. +- 10%. The objective of these experiments was to maintain one hidden layer but vary the number of neurons with in that hidden layer. The intent was to determine the if an optimal number of neurons within the hidden layer existed. The high number of 8 neurons resulted in a training ratio of 45 good dam items and 5 bad data items for a resulting training ratio of 90% good. The low number of 1 neuron resulted in a training ratio of 39 good data items and 11 bad data items for a resulting training ratio of 78% good. The testing ratio results were 80% good for a level of 8 neurons and 76% good for a level of 1 neuron. The results indicates a peak, or maximum, level of good data items for both training and testing at a level of four neurons. However, reviewing our base experiment number one which also contains one hidden layer but with 10 neurons, it is concluded that the neuron interactions and number of neurons cannot be completely "optimized" within the scope of this experiment.

Experiments 17-22 maintain parameters as defined in base experiment number one. The objective of these experiments was to remove input variables sequentially and record results. It can be determined from these results that, based on the set parameters, excluding the profit variable negatively affects the accuracy of the network. Intuitively, without profit, predicting earnings per share would be quite difficult.

Selection of the Best Neural Network: the neural network for the prediction of earnings per share was selected based on our limited experimentation with various learning and testing parameters, the number of hidden layers and the resulting neurons, and the best utilization of the variables selected. The selected neural network consists of three layers including the following: an input layer consisting of all five variables, one hidden layer with four neurons, a training tolerance of. 10 or +-10%, a testing tolerance of. 10 or +-10%, and a learning rate of 1.00. The network was selected based on performance characteristics of the several components of the network generated by the experimentation. The resulting neural network will perform at a training level of 94% and a testing level of 84%. The resulting lesser testing level may be attributed to either the random of the data sets or the relatively stringent testing tolerance of +- 10%.

Scope, Limitations and Assumptions

The scope of this neural network model is intended to accurately forecast earnings per share. The variables utilized for modeling included each of the accepted financial performance indicators used to measure and evaluate the Fortune 500. One assumption is that these financial indicators correlate as a group to influence the behavior of earnings per share. This assumption is necessary to realize that the neural network is not simply random in nature but does develop intricate logical relationships among the input variables and the desired output. The next assumption is that the secondary information obtained from Fortune Magazine is consistently calculated and accurate.

Conclusions

Based on the experimental results it can be concluded that the neural network's capability for prediction varies greatly on the levels of constraint placed on the learning and testing parameters. Varying parameter levels such as training tolerance alter the ability of the network to meet the desired output level. Achieving an optimal network for this application, or any other application, requires a continuous and thorough training/testing operation exploring many combinations of variables, training parameters, testing parameters, learning rates, and extensive dam sets. Prior to actual application for financial forecasting and analysis, it would be beneficial to explore results utilizing other learning algorithms such as the Hopfield network or possibly some type of algorithm utilizing unsupervised learning such as Adaptive Resonance Theory.

References

Stewart, George. "Forecasting the Future", (June 1994), pp. 237-239. Johnson, George, "Sifting Hidden Market Patterns for Profit", New York Times.

Raghupathi, Wullianallur, "A Neural Network Approach to Bankruptcy Prediction", Journal of Information Science Technology, (July 1994), pp. 351-360.

Turban, Efraim, Decision Support and Expert Systems, 1995, Chapter 18-19.

Bluhaugh, William, "Artificial Intelligence: The Model Fund Manager," Economist, (September 1995), pp. 75-76.