A "Go/No' Perceptron Stock Classification Model

Probir Roy

University of Missouri, Kansas City

Follow this and additional works at: http://aisel.aisnet.org/amcis1998

Recommended Citation
http://aisel.aisnet.org/amcis1998/72
A “Go/No” Perceptron Stock Classification Model

Probir Roy
Henry W. Bloch School of Business and Public Affairs
University of Missouri, Kansas City

Abstract

Using MATLAB’s Perceptron model, this paper presents an attempt to train a neural network to distinguish between acceptable and unacceptable purchases of publicly traded stock. In the past, Perceptron models have been used, quite successfully, in similar classification exercises. The input vectors used in training the network and in making the classifications in our model, involve readily available financial data like current ratio; quick ratio; gross margin as a percentage; sales/asset turnover; and earnings per share. The initial results of our analysis were quite encouraging insofar as the model had a ninety percent prediction accuracy using held-back test data. On the basis of our initial success, we are currently trying to extend this model to a “forward-looking” investment decision process model.

Introduction

Neural networks have been relatively widely used in various financial screening processes. There are a number of vendor supplied neural network based packages that provide decision-making support in evaluating credit and mortgage applications. In these applications the decision process is essentially a "go/no go" decision. In other words the decision is simplified into a yes or no dichotomy. It appears that neural network approaches offer significant advantages over credit scoring models based on statistical approaches like discriminant analysis. Thus we find that major financial institutions like American Express and Citibank and others have adopted neural network based models in their loan evaluation process. A couple of Web-based articles have been provided in the references to substantiate some of the statements that have been made above. Given this perceived relative superiority of neural network approaches, we decided to develop a neural network model of a dichotomous investment decision process. We decided to use a Perceptron model because of the discrete nature of both the inputs and the target outputs of such models. We strongly feel that in many decision making situations, especially those involving a variety of disparate inputs, discrete inputs and their processing provide a clearer representation of human behavior compared with models based on the processing of continuous inputs. Human behavior appears to be more in tune with herky-jerky reactions to perceptions of "lumpy" changes rather than to a continuous fine tuning of behavior to incremental changes in inputs.

Perceptron Model

The analysis, in this paper, was based on a single-layer Perceptron model. Single-layer Perceptrons consist of an input layer and an output layer. The elements in the input and output layers are usually binomial (0,1) in nature. The activation function is a hard-limiting function. In such functions, the output measure will assume the value of 1 if the weighted sum of inputs is greater than a threshold value. Perceptrons work well if the outputs are linearly separable.

For this study we asked a group of five investment advisors to name 20 publicly traded companies that they would rate as "buys" and 20 companies that they would rate as "sells". From the list of companies we thus obtained, we randomly chose 10 buys and 10 sells. We then randomly split this group into two equal groups of buys and sells. One group of buys and sells was used to train the single-layered Perceptron, whereas the other group was used as a hold out sample to test the calibrated model of final weights. In this phase of our study we kept our sample size small with the intention of using a much larger sample size if our results were encouraging. We are currently in the phase of increasing the dimensionality of our study and hope to have some initial results by the end of this year.

In order to establish a vector of binary input elements, we dichotomized the five input variables (current ratio; quick ratio; gross margin; sales/asset turnover; and earnings per share) as follows. We compared each company with its corresponding industry average. If the company’s value was higher than the corresponding industry average, then the corresponding input element was given a value of 1. Otherwise the corresponding input element was 0. If the company was deemed a buy, then the target output value was 1; otherwise the target output value was 0 if it was deemed a sell.

Entering these modified input element vectors and the corresponding 0,1 target outputs, we obtained the following weights:

Intercept = -4.6795
Current ratio = 0.993
Quick Ratio = 5.7995
Gross Margin = 8.6433
Sales Turnover = 1.2898
Earnings per share = 1.6359
These weights were then used to test against the hold out sample. We obtained a very encouraging 90 percent match between the model’s predicted buy and sell and the investment advisors recommendations.

A key question is, how does the Perceptron approach fare in comparison with classical multivariate techniques like Discriminant analysis. To start with, the Perceptron model was calibrated, in this paper, using only 20 data points. The validity of a statistical model with 5 independent variables and only 20 data points, on the other hand, would be suspect. Thus on the basis of the Law of Parsimony or Occam’s Razor we find that the Perceptron model offers an initial advantage. However, we are gathering more data with the intention of testing whether there is a significant difference between the Perceptron approach and Discriminant Analysis if paucity of data is not a major concern. We hope to present our updated results in the near future.

References
Torsun, I.S. "A Neural Network for a Loan Application Scoring System", (i.s.torsun@bradford.computing.ac.uk)