**DS-670-Capestone: Big Data & Business Analytics**

**Assignment 5: Method**

**February 19th 2017, Madhumita D**

**Method**

Neural Networks are valued as one of the foremost techniques in performing nonlinear forecasting tasks and complicated pattern recognition across many applications and domains. Some such interesting applications of neural networks are in decoding protein sequences and deterministic chaos. With these successful applications in varied domains, it is interesting to know whether applying this technique to economic time series will yield in extracting the non-linear regularities of the economic market. Decoding the regularities in asset price movement like day to day or month to month fluctuations of stock prices (which were undetected previously) will be the key to earn benefits from investing.

Efficient market hypothesis states that asset or stock prices follow random walk theory. Apart from the expected risk free returns, the price movement of an asset is totally unpredictable based on the publically available data on the asset such as the daily price and the volume history of the stock itself or that of any other stock. It should be observed that insider information (publically unavailable) cannot be considered for forecasting returns to gain unusual profits. The absence of predictability often opens profit opportunities in stock markets only for a small period of time. For example if we expect a strong increase in the price of an asset, an investor goes a long and invests in this asset which drives the price to the expected level, thereby quickly exploiting the profit opportunity which was available a few moments ago. Understanding the simple form of efficient market hypothesis it is feasible to believe that undetected regularities (high and low price movements) exist in historical price data and may persist to exist. Many researchers (mostly academic) and practitioners made tremendous efforts using various techniques to predict the price movements in stock markets and develop financial strategies to convert these predictions into profits. Among the various econometric forecasting techniques used, neural networks methodology seemed to provide better results in trading and financial forecasting. Hence, using neural networks learning methods to find these non-linear regularities in the pricing information may prove useful to the market participants.

The prediction of expected returns using the stock price information of U.S. market is made up of many neural networks that learn the relationship between technical and economical factors. The aim is to forecast the daily and quarterly returns of the all the stocks. Figure 1 depicts the basic architecture of the prediction system. It basically converts all the factors considered (technical and economic) into a space pattern which form the input to the neural network. The input factors are discussed in detail later.

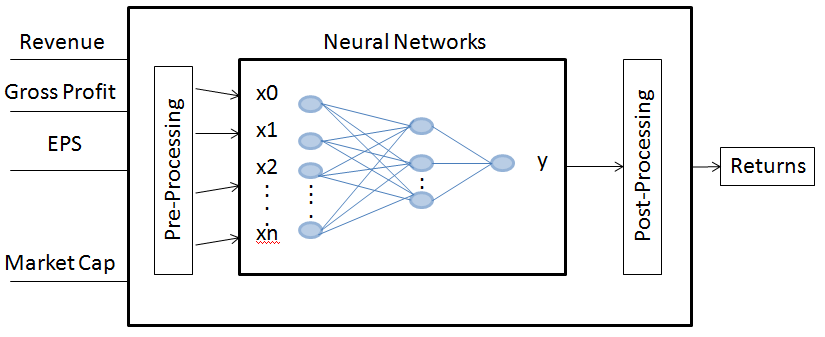


Figure 1: Basic architecture of predicted system

**Neural Networks**

Neural network model is considered as an intelligent data mining technique which is used in pattern recognition problems like stock price predictions. It is a non-linear, non-parametric, data driven model which is trained to map the past values (here past prices and volume information) of a time series. The idea of using a neural network was initially inspired from the nervous system of human beings. The nervous system consists of vast number of processing units known as neurons (Figure 2). Every neuron receives signals from other neurons or from outside and they process these signals using activation function which in turn produces an output. The output from one neuron is sent to other neurons. The impact of each input is different from other inputs. For instance impact of ith neuron on jth neuron is given by a value ‘w’, which is the weight of the connection between both the neurons i and j (Figure 3). That is, each neuron or unit in the network receives input from the lower level neurons and they perform weighted summation to derive the output. We can either use a linear function or sigmoid function as the output function.

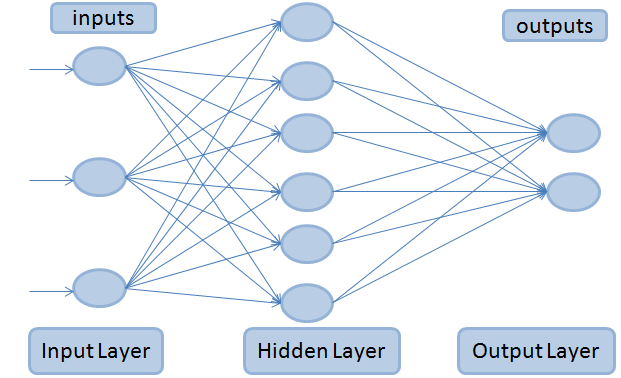


Figure 2: Architecture of Feed Forward MLP network.

The connection is considered to be strong if the weight ‘w’ is high and vice versa. A feed forward multi layer perceptron neural network is made of layers of neurons (Figure 2). The input data is connected to the first layer of the network. After this we can have one or more layers in between which are known as hidden layers. The units or neurons in these hidden layers are called hidden units. These hidden layers are connected to the final output layer which returns the results. Unlike the recurrent neural networks, feedback networks have all the connections towards the output layer. Figure 2 depicts the basic feed forward neural network architecture. It is a three layered network: the input layer, the hidden layer, and the output layer. These three layers are connected completely to form a hierarchical structure.

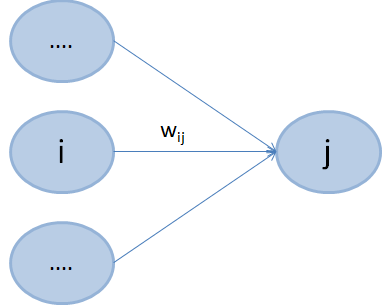


Figure 3: Perceptron neuron’s connections

The learning algorithm used in feed forward multi layer perceptron (MLP) networks is the error back propagation. Error back propagation method is considered as a representative learning rule for hierarchical MLP networks. This learning method allows the network to learn the patterns in the available data and compute the weight of the connection in the inverse direction with respect to the error function. The error function is the regularized sum of squared error. This back propagation methodology selects training vector from the training dataset and processes it from the input layer of the network to the output layer. The error is calculated in the output layer and propagated back to previous layers so that the weights on the connections can be corrected. This process continues until the error reaches a pre defined value. It can be noted that any continuous function can be approximated using a three layered feed forward MLP network with any kind of precision. With any increase in the number of layers and neurons of the neural network, we can see that the learning speed will decrease gradually.

**ARMA models**

The ARMA model or autoregressive moving average model is generally applied to time series data. Most commonly it is used to compute and estimate the Value at Risk (VAR), an estimation model used in simulation forecasting of asset prices. The ARMA model can be used to forecast the future values for a given time series data. Forecasting future stock prices is based on the past prices and volume information which can be converted to a time series data. Many commercial banks employ ARMA methods to predict VAR values.

ARMA has the ability incorporate both the autoregressive models and moving average models. This is usually referred as ARMA ( p, q ) model. The order of autoregressive terms is represented by p and the order of moving average model is represented by q. The autoregressive model of order p is indicated by AR ( p ) notation. The autoregressive model AR ( p ) is written as:

**Xt = c + Xt-i + εt**

where c is a constant, ϕ is the parameter of the model and ε is the error term. The moving average model of order q is indicated by MA ( q ) notation. The moving average model MA ( q ) is written as:

**Xt = εt + εt-i**

where θ is the parameter of the model and ε is the error term. . The autoregressive moving average model of order p, q (p – autoregressive terms and q – moving average terms) is indicated by ARMA ( p, q ) notation. This has two models - AR ( p ) and MA ( q ).The autoregressive moving average model ARMA ( p, q ) is written as:

**Xt = εt + Xt-i + εt-i**

The flexibility and efficiency of the neural networks gives it the ability to deal with robust and uncertain data. This means that to forecast the prices or returns of an asset, neural networks in combination with time series analysis can be efficiently used in stock market analysis. Many previous studies have shown that neural networks outperform linear models, statistical models and classical forecasting. Combining both features of neural networks and ARMA time series model, building a network for prediction of returns is much more complicated. In order to get a robust model all the crucial factors must be considered while designing a prediction model. One of the major factors in networking is the structure of the model which includes number of layers, connections and neurons. Many other factors influence the performance of the model like training algorithm, each neuron’s activation function, dividing the train and test data sets, normalization of data and measurement evaluation (for weights). We have two kinds of neural networks, a feed forward multi layer perceptron and recurrent neural networks, which are used to forecast a company’s asset based on the asset’s share value and historical information. The back propagation learning algorithm is used to train the data in the networks.

The inputs to our neural network are various indicators from the financial statements which determine the company’s performance for a fiscal year. There are three forms of financial statements a company releases quarterly and annually and they are income statement, cash flow statement and balance sheet. The model which is proposed in this paper can be observed as a time series prediction model. The inputs considered are 20 indicators (technical and economical) from the financial statements and the output is the expected returns of every stock. Table 1 shows the different types of financial statements.

|  |  |
| --- | --- |
| Balance Sheet | Comprises of Assets, Liability and Equity |
| Income Statement | Reports Company’s Performance |
| Cash Flow Statement | Presents Operating, Investing and Financing Activities |

Table 1: Types of Financial Statements

The proposed model uses a four layered neural network with back propagation learning algorithm to train the network. The four layers are the input layer, two hidden layers and an output layer. A neural network always takes normalized data as input. The data from various indicators is normalized and is given as input to the first layer. Hence we have 20 neurons in the input layer. The choice of the number of hidden units depends on the number of inputs and number of outputs. The ideal choice is to consider two thirds of the size of input layer, plus size of the output layer and it should be less than the number of inputs. Our model has two hidden layers with eight and seven hidden units in each hidden layer. The output layer has only one neuron which gives the predicted value of returns of stocks (computation is shown in Table 2). Figure 4 shows the proposed neural network model with two hidden layers and Table 2 gives the description of new explanatory variables.

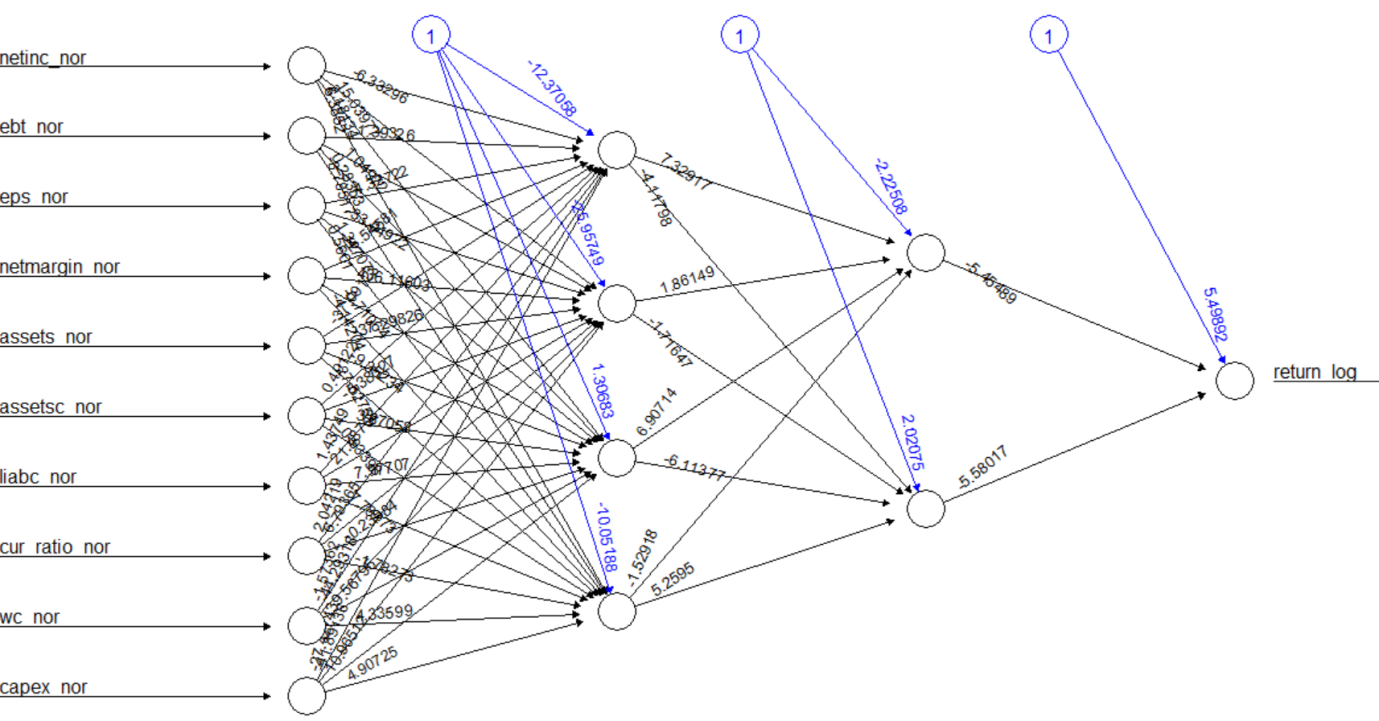


Figure 4: Neural network sample.

|  |  |
| --- | --- |
| Notation | Description |
| Pt | Stock Price |
| pt | log Pt |
| Rt | Return = (Pt /Pt-1) – 1 |
| rt | Return from ‘risk free’ investment |
| Rt – rt | Excess return |

Table 2: Explanatory variables

**Pseudo Code for the proposed model**

*Read quarterly data file into arq\_data*

*Assign ((Pt /Pt-1) – 1) to returns*

*Initialize arq\_data\_factors to required factors from arq\_data*

*Generate 15 subsets for each date from arq\_data\_factors*

*Normalize the data in the subsets*

*Initialize var to 1*

*While var less than or equal to 15*

*Generate neural network model for date subset*

*Process expected returns using predict()*

*Add expected returns to the subset*

*Plot the neural network model for each date subset*

The hybrid approach of using neural networks with time series analysis addresses the fundamental problems by predicting a noisy time series. Predicting stock prices in an economic system is subject to change vigorously. For such forecasting the prediction rules continuously change and hence learning and prediction must adhere to these changing rules. A prediction system known as moving simulation was introduced for this kind of changes. In moving simulation method the prediction is carried out by simulation while moving the objective learning and prediction time intervals. Observing figure 5, the learning of data is done for M months in the past and then prediction is done for the next L months. The system continues this process until we can forecast the returns of the stocks.

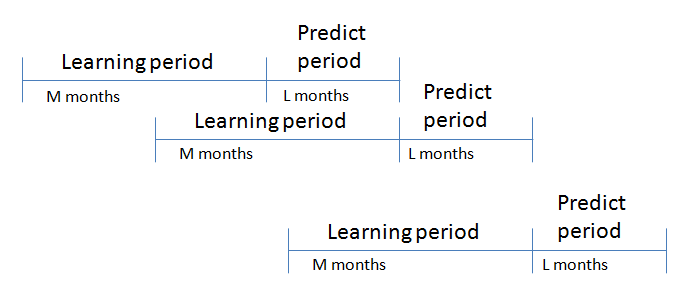


Figure 5: Moving Simulation method

The proposed neural network model in this paper follows this simulation method. For instance we train the neural network for stock market data from September 2014 quarter using the high speed learning algorithm (error back propagation algorithm). We use this model to predict the expected returns of December 2014 quarter. This is learning the data for M past months – September quarter and predicting the values for next L months – December quarter.

Neural Networks are considered as highly dynamic, non-linear, large scale, continuous and time based systems. The knowledge from a neural network is stored on the numerous nodes in the form of a weight matrix where each weight depicts the relationship between the nodes. Neural network is a data driven approach where learning and training of the model is based on the available data. This feature makes it very robust to deal with the uncertainty of data. Neural networks are highly capable in pattern recognition and dealing with machine learning challenges. It is believed that data driven methods are best ways to handle the real time prediction problems. The next section explains in detail the results from our proposed neural networks model with time series data analysis to forecast the returns of stock market data.