Delhi Business Review ✠ Vol. 7, No. 2, July - December 2006

FORECASTING STOCK INDEX RETURNS USING NEURAL NETWORKS

M. Thenmozhi*

MANY studies have shown that artificial neural networks have the capacity to learn the underlying mechanics of stock markets. In fact, artificial neural networks have been widely used for forecasting financial markets. However, such applications to Indian stock markets are scarce. This paper applies neural network models to predict the daily returns of the BSE (Bombay Stock Exchange) Sensex. Multilayer perceptron network is used to build the daily return's model and the network is trained using Error Back Propagation algorithm. It is found that the predictive power of the network model is influenced by the previous day's return than the first three-day's inputs. The study shows that satisfactory results can be achieved when applying neural networks to predict the BSE Sensex.

Key words: Stock market prediction, Neural networks, Financial forecasting, nonlinear time series analysis

Introduction

Predictions in financial engineering have been based on traditional statistical forecasting methods for many decades. Linear models have been the basis of such traditional statistical forecasting. However, traditional methods have seldom proved fruitful owing to the presence of noise and non-linearity in the time series. The successful application of non-linear methods in other areas of research has kindled the hopes of financial researchers. Nonlinear dynamics proposes a new way of viewing financial asset prices, and it suggests new techniques for empirically measuring their nature. This new discipline proposes that past prices help determine future prices, but not in a straightforward way. To begin with, the relation of past prices to future prices will not be linear, but nonlinear. This non-linearity implies that past price change can have wide ranging effects on future prices. Whether past prices generate peaceful or violent movements in future prices will depend on the exact mechanism by which past prices are linked to future prices. With nonlinear dynamics one can gauge when on e state of nature (predictable) or other (unpredictable) is more likely to come into existence. In a sense, nonlinear dynamics tells when a prediction starts becoming unreliable. The best advice one can hope to get from such a prediction is that it is best not to be in a particular market at all. Hence, an attempt is made in this study to understand the use of neural networks in the field of finance.

Neural networks have drawn considerable attention in financial engineering in the recent years because of their interesting learning abilities. They are capable of dealing with problems of structural instability. They are considered to be an effective modeling procedure when the mapping from the input to the output contains both regularities and exceptions. They are in principle capable of solving any non-

* *Associate Professor, Department of Management Studies, Indian Institute of Technology Madras, Chennai – 600 036, India.*

linear classification problems, provided that the network contains a sufficiently large number of free parameters (i.e. hidden units and/or connections).

Neural Networks is a field of research, which has enjoyed a rapid expansion and great popularity in both the academic and industrial research communities. Neural networks are analogous to non-parametric, non-linear regression models. Their novelty lies in their ability to model non-linear processes with few (if any) a priori assumption about the nature of the generating process.

The very power of neural network modeling lies in its ability to adjust itself according to the information given in order to optimize some pre-determined objective. The novelty of neural network lies in their ability to model nonlinear processes with few (if any) assumptions about the nature of generating process. This is particularly useful in financial engineering applications where much is assumed and little is known about the nature of the processes determining asset prices.

In the recent past, neural networks have proved to be a promising area of research in the field of finance with many applications to its credit. Neural network applications in finance include assessing the risk of mortgage loans (Collins, Ghosh and Scofield, 1988), rating the quality of corporate bonds (Dutta & Shekhar, 1988), predicting financial distress (Salchenberger, Cinar & Lash, 1992, Altman, et.al., 1994), rating of credit card applicants (Suan & Chye, 1997) and predicting bond ratings (Hatfield & White, 1996). Neural neworks are also applied in areas such as credit risk and bankruptcy prediction (Suan & Chye, 1997), pricing of derivatives (Hutchinson, 1994), asset allocation and portfolio management (Refenes & Zapranis), risk management, interest rate modeling, insurance, stock valuation, exchange rate forecasting, predicting stock market prices (Bower, 1988, Teo & Poii, 1995, Refenes, 1997, McCluskey, 1993), generate buy, sell signals (Subrata Kumar Mitra, 2000), design of trading system (Kusturelis, 1998), prediction of option and futures prices (Burgess), etc. Currently, apart from academic researchers many practitioners in the financial market have also realized the potential of neural networks. According to Shandle (1993), companies such as General Electric, American Express and Chase Manhattan Bank are using neural network to screen credit applications, spot stolen credit cards, detect patterns which may indicate fraud and predict commodity and stock prices, bond ratings and currency trading trends. Financial institutions have carried out many projects on financial forecasting with neural networks. For instance, Mahendra Mehta (1994), from Citibank has registered some success on forecasting exchange rates. Besides, Morio Yada (1994), manager of research and trading division at Nikko Securities Company has developed a system for TOPIX (Tokyo Securities Exchange Stock Price Index) forecasting with a high prediction rate using neural networks.

This paper attempts to project the fact that the neural networks are model free estimators and are an ideal launch pad for financial forecasting and analysis. Apart from reviewing the application of neural networks in financial engineering, an attempt is made to predict the daily returns of Bombay Stock Exchange using the back-propagation neural network model.

Section I: Neural Network Architecture

A neural network is a massively parallel distributed processor that has a natural propensity for storing experiential knowledge and making it available for use. Artificial neural networks are computers whose architecture is modeled after the brain. It resembles the brain in two respects: knowledge is acquired by the network through a learning process and inter-neuron connection strengths known as synaptic weights are used to store the knowledge. They typically consist of many hundreds of simple processing units, which are wired, together in a complex communication network. Each unit or *node* is a simplified model of a real neuron, which *fires* (sends off a new signal) if it receives a sufficiently strong input signal from the other nodes to which it is connected. The strength of these connections may be varied to enable the network to perform different tasks corresponding to different patterns of node firing activity.

The basic element of a neural network is the perceptron as shown in Fig.1 below. Frank Rosenblatt first proposed it in 1958 at Cornell University. The perceptron has 5 basic elements: n-vector input, weights, summing function, threshold device and an output. Outputs are in the form of –1 and/or +1. The threshold has a setting, which governs the output based on the summation of input vectors. If the summation falls below the threshold setting, -1 is the output. If the summation exceeds the threshold setting, +1 is the output.

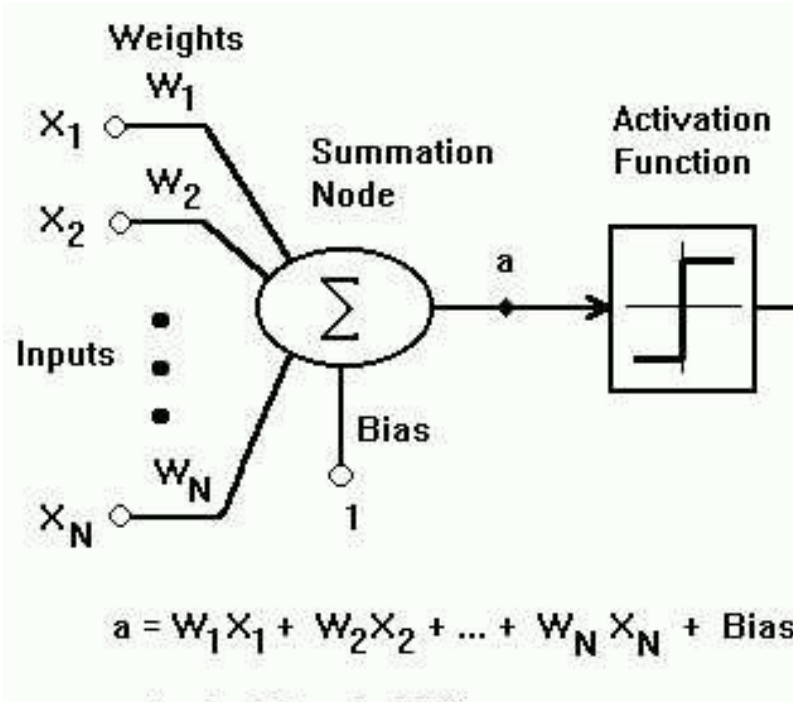

Figure 1: A Simple Perceptron

A more technical investigation of a single neuron perceptron shows that it can have an input vector X of N dimensions. These inputs go through a vector W of Weights of N dimension. Processed by the Summation Node, "a" is generated where "a" is the "dot product" of vectors X and W plus a bias. It is then processed through an activation function, which compares the value of "a" to a predefined threshold. If "a" is below the threshold, the perceptron will not fire. If it is above the threshold, the perceptron will fire one pulse whose amplitude is predefined.

Neural networks itself represents a collection of artificial intelligence models which include, multilayer perceptron neural network, recurrent neural network, modular neural network, radial basis network. Each of these models has its own specific structure, training method and area of application. A thorough understanding on each of them is necessary to make the best choice of network structures for financial forecasting tasks.

The choice of relevant inputs for the networks would essentially be based on the expert knowledge in finance field. However, the best set of inputs is constrained by time factor and market factor thereby, demanding sufficient human expertise to choose the best inputs for a specific market at a specified time. Certain systematic methods give a helping hand to select a best sub-set of inputs from a large collection of possible inputs, including pruning of network, mutual information and embedding dimension. Equally important is the choice of outputs from the network to generate meaningful financial forecasting. This again rests on the expert financial knowledge.

Having chosen a particular network structure, network inputs, network outputs, objective function and historical data set of associated input-output pairs, an optimal set of network parameters with respect to that historical data set is then found by some numerical method. This process is termed "*Training*". It is possible that certain training methods might fail to arrive at an optimal set of parameters with respect to the data set. The possibility of finding optimal parameters varies from problem to problem. In general, if the objective function is highly complicated or if the noise in the data set is too high, it is more difficult to find the optimal parameters. However, there are certain methods to enhance the possibility of finding the optimal solution namely, genetic search, multiple training, hybrid algorithm of random search and back propagation and expanded range approximation.

From a very broad perspective, neural networks can be used for financial decision making in one of the three ways:

1. It can be provided with inputs, which enable it to find rules relating the current state of the system being predicted to future states.
2. It can have a window of inputs describing a fixed set of recent past states and relate those to future states.
3. It can be designed with an internal state to enable it to learn the relationship of an indefinitely large set of past inputs to future states, which can be accomplished via recurrent connections.

It follows that these three methods require decreasing sophistication in the choice of inputs coupled with stringent training methods. In most cases the optimal solution obtained from historical data set can be different from the optimal solution proposed for forecasting methods. For example, to know the returns of certain selected stocks in the past is quite a simple task. However, to forecast the same in the near future is a difficult task. It is interesting to note that the difference between the historical solution and the forecasting solution rests on the strength of the relation between the past and the current events. For financial forecasting tasks, the financial data is bound to contain lot of noise, which is not likely to repeat it in the future. Thus, the relation between the past and the future data needs to be strengthened to even out the differences between the historical solution and the forecasting solution. This would make way for the historical solution to be generalized for forecasting purposes.

Section II: Neural Networks in financial forecasting

Jason E. Kusturelis examines the use of neural networks to predict the future trend of stock market indices. The accuracy of forecasting is compared with traditional multiple linear regression analysis. Besides, the probability of the correctness of the model forecasted is calculated using conditional probabilities. The back propagation algorithm is used in order to minimize the error term between the output of the neural network and the actual desired output value. The error term is calculated by comparing the network's output to the desired output and is then fed back through the network causing the weights to be changed in an effort to minimize error. This process is repeated till the error reaches a minimum value. The study concludes by registering a 93.3% probability in predicting a market rise and 88.07% probability in predicting a market drop in the S&P 500. The author also affirms that linearity assumption in multiple regression analysis may not be true in all cases while neural network can model both linear and curvilinear systems. Besides, neural network models are significantly more accurate than multiple linear regression analysis. The author also cautions not to blindly follow the network's advice but recommends to double check the network using multiple networks incorporating different inputs to predict the same output.

Following an exploratory data analysis using non-linear models, Teo Jasic and Hean Lee Poii (1995) use neural networks for the prediction of TOPIX (Tokyo Security Exchange Stock Prices Index) returns using several macro-economic indicators as input time-series. The four input variables correspond to

macro-economic variables such as foreign exchange rate, price/earnings (P/E) ratio, U.S. market performance index and liquidity. These indicators have been chosen deliberately to account for globalization phenomenon of financial markets. The inputs were used to predict the TOPIX monthly returns using back propagation neural networks. While the influence of input variable P/E ratio is decreasing the influence of liquidity is increasing. These findings are associated with increased volatility of Japanese market in the period 1982 – 1990. This study reveals that neural network with four inputs and one lag of output variable as input achieves the best result.0

An attempt is made by Tan Sen Suan and Koh Hian Chye (1997) to construct an ANN model to predict share price movements in the Singapore Stock Exchange (SES) using the Singapore Airlines (SIA) stock as an example. The data for 569 trading days are used in the example from 19th January 1993 to 21st April 1995. The network model was constructed to predict the closing price of SIA for one week into the future, based on the knowledge of the current and historical (in this case, each of the past two days) high, low and closing prices, as well as volume traded. The input vector consists of 12 values (4 data variables per day times 3 days), requiring 12 nodes for the input layer. The output layer has just one node holding the value of the predicted closing share price a week from today. The network's performance is enhanced during training by following rule of thumb with trial and error adjustments. The network is trained using supervised learning via the generalized delta rule. It is seen that the predicted closing prices are pretty close to the actual ones. Out of the 50 test cases, 47 (94%) record absolute errors of less than 5% and 35 predictions (70%) record errors by less than 1%.

A.N. Refenes *et.al* (1997) have developed a model to predict the 30-day stock returns of the Paris Stock Exchange. The stock returns are estimated as a function of long and short-term interest rates, earnings per share, price earning ratio and exchange rate of Franc against the U.S. dollar. The network chosen is the feed-forward, multi-layered and fully connected networks and is trained using the standard back-propagation algorithm. The work concludes by reinforcing the fact that the back propagation networks are more efficient for non-linear data.

Peter C. McCluskey, Peter C. (1993) has used neural network training algorithms to predict the Standard & Poor's 500 Index and then has compared the results with the Genetic programming and hand coding approaches. Feed forward networks are used to predict the Index returns and the network is trained using the back propagation algorithm. The networks are trained to predict the change in S&P 500 closing prices for the next 1, 2 and 4 weeks. The data used is divided into two parts: prior to November 16, 1979 and November 19, 1979 to April 2, 1993. Though the model has worked well based on the historical data set taken as input, the author mentions that the future performance of the model is not warranted as the results are not likely to match the ideal of the historical closing prices.

Tan Sen Suan, Koh Hian Chye (1997) have developed a neural network model to predict bankruptcies. The network consisted of an input layer of six neurons for six financial ratios: quick assets to current liabilities, market value of equity to total assets, total liabilities to total assets, interest payments to earnings before interest and tax, net income to total assets, retained earnings to total earnings. The hidden layer consisted of 13 nodes and the output layer had one node. It was noted that the network's accuracy rate was 100% for bankrupt firms, non-bankrupt firms and overall.

A.N. Burgess describes a study of Eurodollar futures. The data consists of daily, high, low, open and close prices for Eurodollar futures over the time period August 1987 to July 1994, giving 1760 daily observations in all. The futures contract ranged from less than three months at the short end to three years at the long end. The standard multi-layer perceptron network was used with two hidden layers. The out of the sample test showed that the neural network could generate consistent profits resulting in an average return of 47% per year which is sufficient enough to conclude that the Eurodollar yield appears to be predictable by non-linear techniques.

James M.Hutchinson *et.al* (1994) have proposed a non-parametric method for estimating the pricing formula of a derivative using learning networks. The inputs to the network are the primary economic variables that influence the derivative's price namely, current fundamental asset price, strike price, time to maturity, etc. The derivative price is defined to be the output into which the learning network maps the inputs. Once the network is trained, the network becomes the derivative pricing formula. The data used here is the daily closing prices of S&P 500 futures and options for the 5-year period from January 1987 to December 1991. On comparing the results with the parametric derivative pricing formula, the authors have been cautiously optimistic about their general approach with a number of promising directions for future research.

The stock price movements have been basically analysed on the assumption of linearity of the time series in the Indian scenario. The trends in stock prices are estimated using moving averages, regression and other linear methods when the time series seldom moves in a linear fashion. Thenmozhi (2001) has examined the nonlinear nature of the Bombay Stock Exchange time series using chaos theory. The study examines the Sensex returns time series from 16/01/1980 to 26/09/1997 and shows that the daily returns and weekly returns series of BSE sensex is characterised by nonlinearity and the time series is weakly chaotic. The study recommends the use of nonlinear methods to predict the time series rather than using linear methods for prediction.

Subrata Kumar Mitra has used Artificial Neural Networks (ANN) to generate trading signals using five year daily sensex values of Bombay Stock Exchange (BSE) between 1st January 1994 to 31st December 1998. The input parameters included the underlying close to close return, the underlying intra-day return and the intra-day range and the output is the trading signal. Typical non-linear energy transfer function $f(x) = 1/(1+e^{-x})$ is used to determine the output. The weights are adjusted such that the error is minimized using "solver" computer program. Positive valued output signals up-trend while the negative values signaled downtrend. The strategy adopted for up-trend was to "buy" while downtrend confirms, "sell". The network performance is evaluated based on total cumulative return, profit over loss ratio, correlation coefficient of signal, actual price change following signal and t-statistic of daily returns. The results are found to be quite encouraging. The work concludes with suggestions to incorporate other parameters such as interest rate, foreign institutional investments, etc., which influences the sensex values to be incorporated into the structure for better evaluations.

The literature survey shows that:

(i) Neural network models are significantly more accurate than multiple regression models in financial forecasting.

(ii) Neural networks have been used to predict the stock market index, predict bankruptcy, predict the eurodollar futures, estimate derivative price and so on.

(iii) The input used for neural network modeling ranges from 3 to 12 nodes depending on the time series. The output is only one node and the hidden layer varies upto 12 nodes.

(iv) It is observed that the most commonly used neural network model is the multi-layer perceptron. The prediction is done using feed forward network while, the training is predominantly done using error back propagation algorithm.

(v) In the Indian scenario, attempt has been made to examine the nonlinear nature of the BSE Sensex time series and the complexity of the time series recommend the use of nonlinear models for prediction. An effort has been made to generate trading signals for the sensex time series considering only five year data.

The purpose of this study is to examine the feasibility of predicting the movements of the daily returns

of the Bombay Stock Exchange (BSE) index using Neural Network based on the evidence of nonlinear nature of the time series (Thenmozhi 2001).

Section III: Data and Methodology

The data consists of daily index values of the BSE Sensex. The period under consideration is 16/01/1980 to 26/09/1997. The time period chosen for analysis is similar to the time series used by Thenmozhi (2001).The data set consists of 3667 data points. The data has been obtained from Capitaline 2000 database that provides daily stock market data. The entire analysis has been done basically on the daily returns rather than the raw index value as such.

The approach adopted to train the neural network using daily returns is explained in terms of the following:

a) ***Inputs and Outputs:*** The historical price is used as inputs to the network. The inputs to the neural network are basically the delayed coordinates of the time series. The number of inputs to the network is 4 and they are the 4 consecutive daily returns. The output is the prediction of the return on the fifth day.

b) ***Network Structure:*** The simplest of the neural network architecture – Multi Layer Perceptron (MLP - Fig. 2) neural network is chosen for the purpose of prediction. It essentially consists of three layers of nodes namely, input, hidden and output layers. The first layer consists of the input data. The last layer is the output layer, which consists of the target values. All layers between the input and the output layers are called as the hidden layers.

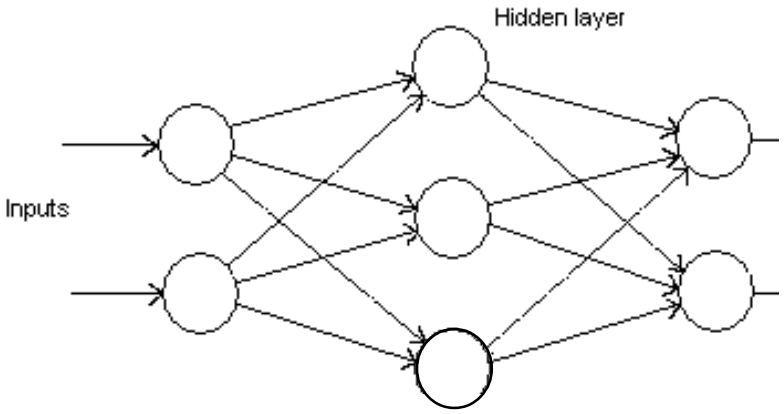

Fig. 2: A Simple MLP Network

c) ***Transfer Function:*** Transfer functions are those, which are used to relate the units of the different layers. They map the neurons input to an output, where the neuron is the link between the layers. The input functions are first multiplied by their respective weights, summed and then mapped to the output via the transfer function. The widely used transfer functions are the *Sigmoid Function* and the *Hyperbolic Function*. Both are very similar except that; the sigmoid function takes values between zero and one while the hyperbolic function takes values between minus one and plus one. A unit in the output layer determines its activity by following a two-step procedure.

(i) First, it computes the total weighted input X_j, using the formula:

$$X_j = \sum_i y_i W_{ij}$$

where yi is the activity level of the jth unit in the previous layer and W_{ij} is the weight of the connection between the i^{th} and the j^{th} unit.

(ii) Next, the unit calculates the activity y_j using some function of the total weighted input. Typically we use the sigmoid function:

$$y_j = \frac{1}{1 - e^{-xj}}$$

(d) ***Training Scheme:*** The most appropriate learning algorithm is the Back-Propagation. The idea behind this algorithm is to adjust the weights in a way such that the error is reduced.

The error E, is defined by the expression:

$$E = \frac{1}{2} \sum_i (y_i - d_i)^2$$

where y_j is the activity level of the jth unit in the top layer and d_j is the desired output of the jth unit.

The back-propagation algorithm consists of four steps:

1. Compute how fast the error changes as the activity of an output unit is changed. This error derivative (EA) is the difference between the actual and the desired activity.

$$EA_j = \frac{\partial E}{\partial y^j} y_j - d_j$$

2. Compute how fast the error changes as the total input received by an output unit is changed. This quantity (EI) is the answer from step 1 multiplied by the rate at which the output of a unit changes as its total input is changed.

$$EI_j = \frac{\partial E}{\partial x_j} = \frac{\partial E}{\partial y_j} \times \frac{\partial y_j}{\partial x_j} = EA_j y_j (1 - y_j)$$

3. Compute how fast the error changes as a weight on the connection into an output unit is changed. This quantity (EW) is the answer from step 2 multiplied by the activity level of the unit from which the connection emanates.

$$EWi_j = \frac{\partial E}{\partial W_{ij}} = \frac{\partial E}{\partial k_j} \times \frac{\partial k_j}{\partial W_{ij}} = EI_j y_i$$

4. Compute how fast the error changes as the activity of a unit in the previous layer is changed. This crucial step allows back propagation to be applied to multilayer networks. When the activity of a unit in the previous layer changes, it affects the activities of all the output units to which it is connected. So to compute the overall effect on the error, we add together all these separate effects on output units. But each effect is simple to calculate. It is the answer in step 2 multiplied by the weight on the connection to that output unit.

$$EA_j = \frac{\partial E}{\partial y_i} = \sum_i \frac{\partial E}{\partial x_j} \times \frac{\partial x_j}{\partial y_i} = \sum_j EI_j w_{ij}$$

By using steps 2 and 4, we can convert the EAs of one layer of units into EAs for the previous layer. This procedure can be repeated to get the EAs for as many previous layers as desired. Once we know the EA of a unit, we can use steps 2 and 3 to compute the EWs on its incoming connections.

The Sensex returns time series has been trained and evaluated using the software package QNET which supports Multi-Layered Perceptron structure and Back-Propagation training method.

Section IV: Results

Information about the neural network used for predicting the daily returns is indicated in Table 1 and Table 2. Three layers were used with only one hidden layer. The input and the hidden layers have four nodes while the output layer has only one node. The network is fully connected. Sigmoid transfer functions have been used in the network. There were 3612 patterns, of which the last 1200 patterns were reserved for testing, which meant that the first 2412 patterns formed the training set.

Table 1: Network Information for Daily Returns

Network Information	Input Layer	Hidden Layer	Output Layer
Nodes	4	4	1
Transfer Function	Linear	Sigmoid	Sigmoid

Table 2: Training Information

Training Information	Daily Returns
Iterations	5000
Training Error	0.037940
Test Set Error	0.034655
Learn Rate	0.014071
Training Patterns	2412
Test Patterns	1200

Prediction Accuracy: The network is trained by repeatedly presenting it with both the training and test data sets. To identify when to stop training, four parameters, namely the RMS error and the correlation coefficients of both the training and test sets were used. After 5000 iterations, the RMS error and correlation coefficient of the training set was 0.037940 and 0.08231 respectively, while those of test set was 0.034655 and 0.18169. The training was stopped after 5000 iterations as there was no significant decrease in either of the RMS errors (Fig.3 & 5) and the correlation coefficient of the training set (Fig.4) was also starting to form a plateau. The correlation coefficient of the test set (Fig.6) was also oscillating within a small band. Thus, any further training was not going to be productive or cause any significant changes. The prediction accuracy of the training data is 96.3% and that of test data is 96.6% and are significant at .0.05 level of significance. The correlation between the input and predicted output has improved for the test data. As the values of RMS and correlation coefficient of the test data set are better than those of the training set, the model fits the test pattern better than the training pattern.

Importance of Inputs: A sensitivity analysis is done in order to determine the relative importance of each input on the output once the network is fully trained. Sensitivities are determined by cycling each input for all training patterns and computing the effect on the network's output response. A plot (Fig.7) shows the result of such a sensitivity analysis on the model. It indicates that the most recent return (Input Node 4) contributes most to the output while all the other three inputs have roughly the same

impact on the output. Thus the immediate previous day return contributes significantly in predicting the returns compared to the first three-day returns used for prediction.

Utilisation of Hidden Nodes: Finally, an analysis is done to assess the efficient utilization of the hidden nodes. This is done to prevent the over-designing of the hidden layer in the network which may otherwise result in many nodes contributing little or nothing to the output response. For each hidden layer in the network, a plot is generated to compare the relative strengths of all output connections for that layer. The plot shows the percent contribution to that layer's output signals over all training patterns. If there are many nodes in a layer that is showing too many contributions, then that layer may be specified with too many nodes. Likewise, if all nodes show strong contributions, it is possible that adding extra nodes might as well help the model. The plot (Fig.8) shows that the hidden node 2 has got the weakest connection strength while the other three nodes have better strength and are roughly equal. This indicates that the network uses all the hidden nodes roughly equally and the hidden layer is saturated, meaning that it has attained the maximum number of nodes that can be useful to the network.

Conclusion

In this paper, an attempt is made to explore the use of neural network to predict the daily returns of BSE sensex. Multi layer perceptron network is used to build the daily returns model and the network is trained using Error backpropagation algorithm. The results show that the predictive power of the immediate previous day's return is higher than the first three-day's inputs. The network uses all the hidden nodes and the hidden layer is saturated indicating that the maximum number of hidden nodes useful for the network is four. The prediction accuracy of the model is high both for the training data and test data and the model fits the test data better than the training data set.

Neural networks are basically experimental methods where lot of trial and error is involved. Neural network with different structures can be used to predict the stock market behaviour and a comparison can be made of the predictive power of the different architectures. Global financial market players, institutional investors and generic software developers should consider developing stock market trading strategies using neural networks. Probably, further experimentation is required for producing better prediction of stock prices and further work has to be done by testing it for weekly or monthly returns, as well as by including other micro and macro-economic variables as inputs. Further, the influence of macro economic variables such as interest rates, stability of the government, GDP, trend in global stock market, etc., have to be considered to obtain a better network structure. The network may also be developed using technical indicators apart from fundamental data. Better network structure could be obtained by varying the parameters of the training algorithm and different trading strategies can be developed for different market players. Neural networks have the capability to forecast financial markets and if properly trained, the individual and institutional investors could benefit from the use of this forecasting tool.

References

Altman, E., M.Giancarlo, and F.Varentto (1994), "Corporate Distress Diagnosis: Comparisons using Linear Discriminant Analysis and Neural Networks (the Italian Experience), Journal of Banking and Finance, Vol.18, pp.505-29.

Burgess, A.N. "Statistical Yield Curve Arbitrage in Eurodollar Futures using Neural Networks", Department of Computer Science, London Business School, London, U.K.

Burgess, A.N. & Refenes, A.N. (1996), "The use of Error feedback terms in Neural Network Modeling of Financial Time Series", in Dunic, C.(ed.) Quantitative Methods in Trading and Finance, Wiley & Sons.

Collins, E., Ghosh, S. and Svofield. C. (1998), "An Application of a multiple Neural Network Learning System to Emulation of Mortgage Underwriting Judgements", Proceedings of the IEEE International Conference on Neural Networks, Vol.2, pp.459-466.

Dutta, S., and Shekhar, S. (1998), "Bond Rating: A Non-conservative Application of Neural Networks", Proceedings of the IEEE International Conference on Neural networks, Vol.2, pp.443-450.

Hatfield, Gay B., and White, A. Jay (1996), "Bond Rating Changes: Neural Net Estimates for Bank Holding Companies", Working Paper.

Hutchinson, James M. Andrew W.Lo and Tomaso Poggio (1994), "A Non-Parametric Approach to Pricing and Hedging Derivatives Securities Via Learning Networks" – The Journal of Finance, Vol.XLIX, No.3, July.

Kusturelis, Jason E. (1998) "Forecasting Financial Markets using Neural Networks: An Analysis of Methods and Accuracy", Master of Science in Management Thesis, Department of System Management, http://www.nps.navy.mil/Code36/KUTSURE.html

McCluskey, Peter C. (1993) "Feed-forward and Recurrent Neural Networks and Genetic Programs for Stock Market and Time Series Forecasting", Department of Computer Science, Brown University, September.

Refenes, A.N. and Zapranis, A.D. "Neural Networks in Tactical Asset Allocation: A Comparative Study with Regression Models", Department of Decision Science, London Business School, London.

Refenes , A.N., Bentz, Y. Bunn, D.W., Burgess, A.N. and Zapranis, A.D. (1997) "Financial Time Series Modeling with Discounted Least Squares back Propagation", Neuro Computing Vol. 14, pp.123-138.

Suan, Tan Sen and Koh Hian Chye (1997), "Neural Network Applications in Accounting and Business", Accounting and Business Review, Vol.4, No.2, July, pp.297-317.