A NEURAL STOCK PRICE PREDICTOR USING QUANTITATIVE DATA

Animesh Chaturvedi, Samanvaya Chandra
B.Tech, Indian Institute of Information Technology,
Allahbad
India
achaturvedi_01@iiita.ac.in, schandra_01@iiita.ac.in

ABSTRACT
Financial forecasting is an example of a signal processing problem which is challenging due to small sample sizes, high noise, non-stationarity, and non-linearity. Neural networks have been very successful in a number of signal processing applications. We discuss fundamental limitations and inherent difficulties when using neural networks for the processing of high noise, small sample size signals. Financial forecasting employing neural networks is a highly significant area for exploratory study. It has long been known that exact prediction is more of an art than a science but attempts can be made to recognize trends and patterns which should help in predicting correctly within an order of magnitude. Our approach uses a typical back propagation neural net and employs various formulas on quantitative data. We picked our training stocks from different categories having varying prices and volumes; this enabled a better analysis while generalizing our findings. While it has not been possible to provide exact predictions, a definite trend is evident in most cases. Statistically-oriented projections of the significance of these findings using standard regression analysis techniques show our approach to be simple yet effective. The historical data was obtained over a time period of 40 days for major stocks on New York Stock Exchange (NYSE). This data was used to train the network while trying out various parameter values and techniques which are discussed below

KEY WORDS
Financial forecasting, stock prediction, artificial intelligence and neural networks

1. Introduction

Forecasting the stock price is a very profitable and active area of research. However the problem becomes complex as stock prices depend on various parameters and there is no clear correlation between them. In economics there are no fixed prediction formulas and this inherently leads to a probabilistic model of prediction. Artificial intelligence especially neural networks are ideally suited for this purpose. The neural networks have surely proved a great investment tool since the introduction of personal computers. The networks can learn and gain experience and can help an analyst in making better decisions. Our model and approach aims to provide projections which can help in better decision making. This paper thus provides a framework for the development of a model using the quantitative publicly available information like the beta value, market capital, earning per share etc. of a stock. The potential combinations of financial market indices or stocks, and neural network type are virtually limitless. For this reason, the research was limited to one stock market index, one neural network type and one statistical forecast tool. This allowed the research to build upon and validate previous research and place boundaries around the vast topic of time series forecasting

The stock market is basically an uncertain beast or an uncertain institution, it's an institution where people trade risk, swap risk, and that's why its there. And so if it were possible to predict it there would be no risk. In individuals, we think there cannot be any publicly available system to predict a financial market. On the other hand, neural networks have been found useful in stock price prediction. Both back propagation and recurrent neural networks have been investigated and good results have been obtained. That means the prediction software would be very useful to assist individuals in reaching a final decision. Assuming that it is possible to predict markets, a prediction system is developed using neural networks with a learning algorithm to predict the future stock values. The system consists of several neural networks modules. These models are all used to learn the relationships between different technical and economical indices and the decision to buy or sell stocks. The inputs to the networks are technical and economic indices. The output of the system is the decision to buy and sell.

2. Objective

The objective of this research was to examine the theory of back propagation neural networks and then develop a model that would accurately predict the future closing price of the stocks. Once this was accomplished, the probability of an accurate forecast would be calculated. Given the accuracy of the forecast, the benefits of the network to the investor would be determined.

Our aim was to find the best topology for the network corresponding to our parameters. The parametric effort varies with:

- Momentum (higher the momentum faster the convergence).
- Learning rate (over a period of time the learning rate was decreased).
- Input noise co-efficient.
- Number of neurons in hidden layer of ANN model (higher number of hidden layers will always add to calculations and are rather cumbersome to use).

With this approach we aim to explore:

- The generalized capability of the ANN by varying the network structure.
- The number of training iterations that are used to obtain best results.
- A comparison of various activation functions like sigmoid, Gaussian, linear etc.
- What are the similarities between the Back propagation neural network and the Biological systems after which they were Designed?
- What is the mathematical theory behind the Back propagation neural network?
- Can neural networks accurately forecast a stock market index?
- Can multiple regression analysis accurately forecast a stock market index?
- Can neural networks be used as a practical forecasting tool by individual Investors?

3. Parameter Selection

Quantitative information is the information which can be expressed in numeric format. Thus this information is to be interpreted before it is used by the investor. The parameters which are publicly available are thus modified for their usage in the neural net. The processing layer (hidden) consists of 3 neurons each of which carries different computation. The formulas for each neuron are composed of the modified parameters are then interpreted through an activation function.

The quantitative parameters used in our implementation were market capitalization value, beta coefficient, earning per share in the previous quarter, 52 week highs and lows and volume of trade. They were then modified and acted upon scaled input stock values as received from the input layer. The scaled inverse of market cap with a weight factor can be designated as a fluctuation rate. This was determined by studying that stocks with large values of market cap do not tend to fluctuate wildly in normal trading. The weighted beta can be used to predict the possible change in stock value corresponding to the change in index value. The trend for fluctuations can be determined using a scaled inverse of earning per share value. This parameter has been determined to be more effective in the beginning of next quarter than later. Thus the new price can be determined by the neural net and changes can be compensated by the average summation of the above modified parameters. In order to predict more accurately the predicted value is measured up in the range of 52 week highs and lows and adjustments are made.

4. Network Parameter Issues

After comprehensive testing over both static and dynamic topologies, it was decided to go for a three layered approach as this was the most computationally efficient. Since we plan to upgrade this framework for real time prediction in the future, a 3 layered architecture was deemed to be the best fit. The first layer was designated the input layer where 3 input neurons would take 3 scaled input stock values of the series in the historic data. After testing, it was found that initialization with random weights was most effective.

The second layer is the hidden layer which does all the computations and again comprises of three neurons. The first neuron is a statistically oriented neuron whose function computes the weighted mean average by the formula

$$A_t = sC_t + (1-s)A_{t-1}$$

Where s=2/13, a well known constant in securities market and C is the closing value of the stock.

For better assessment the neuron can optionally include the moving average and weighted differences between the inputs in the series.

The second neuron uses the scaled addition of the various parameters discussed in parameter selection. This is a more comprehensive and exact assessment of the series as the next value to be predicted not only carries the momentum associated with the statistical series, but also incorporates the affecting parameters.

The third neuron takes its computation function as the scaled average of the statistical prediction multiplied by 2 parameters. These two parameters are initialised by the investor and during the course of training they can be changed. These are in essence the quantitative counterparts of qualitative information which is available in form of news, information on discussion boards and ratings from agencies like Standard and Poor of Moody's. The first parameter can be designated as the short term outlook on the stock and will have more effect on the next value and hence a higher weight. The second parameter can be thought to be as a long term outlook and will serve as a check for unexpected rise and fall in the stock value. For the immediate prediction this parameter will have less weight.

The output layer consists of 3 neurons each with combinations of different weights on the inputs received from the hidden layer. The weighted mean of these outputs is the next predicted value and is scaled back to give the exact value to the investor.

5. Data Series Partitioning

One typical method for training a network is to first partition the data series into three disjoint sets: the training set, the validation set, and the test set. The network is trained (e.g., with back propagation) directly on the training set, its generalization ability is monitored on the validation set, and its ability to forecast is measured on the test set. A network's generalization ability indirectly measures how well the network can deal with unforeseen inputs, in other words, inputs on which it was not trained. A network that produces high forecasting error on unforeseen inputs, but low error on training inputs, is said to have over fit the training data. Over fitting occurs when the network is blindly trained to a minimum in the total squared error based on the training set. A network that has over fit the training data is said to have poor generalization ability.

To control over fitting, the following procedure to train the network was used:

1. After every n epochs, sum the total squared error for all examples from the training set.
2. Also, sum the total squared error for all examples from the validation set. This error is the validation error.
3. Stop training when the trend in the error from step 1 is downward and the trend in the validation error from step 2 is upward.

When consistently the error in step 2 increased, while the error in step 1 decreased, this indicated the network had over-learned or over fitted the data and training should stop. When using real-world data that is observed (instead of artificially generated), it may be difficult or impossible to partition the data series into three disjoint sets. The reason for this is the data series may be limited in length and/or may exhibit non stationary behaviour for data too far in the past (i.e., data previous to a certain point are not representative of the current trend). We tried our best to overcome this real situation.

6. Learning and Training

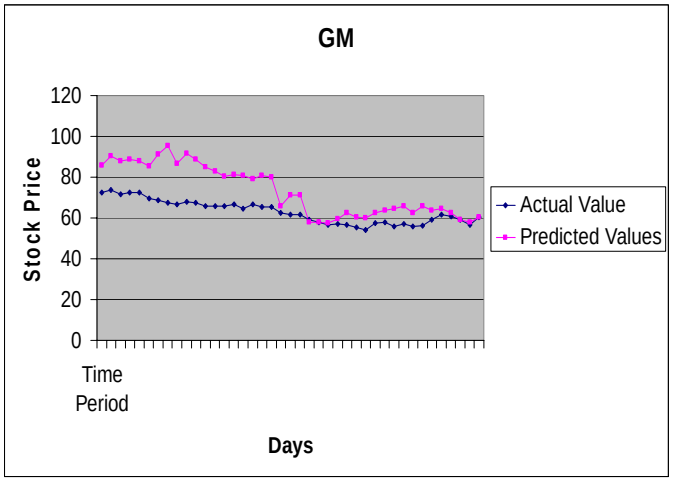
By observing neural network training characteristics, a heuristic algorithm was developed by us. The heuristic requires the user to set the learning rate and epochs limit to higher-than-normal values and the error limit to a lower-than-normal value. Finally, the heuristic requires the data series to be partitioned into a training set and validation set. Given these user set parameters, the heuristic algorithm is:

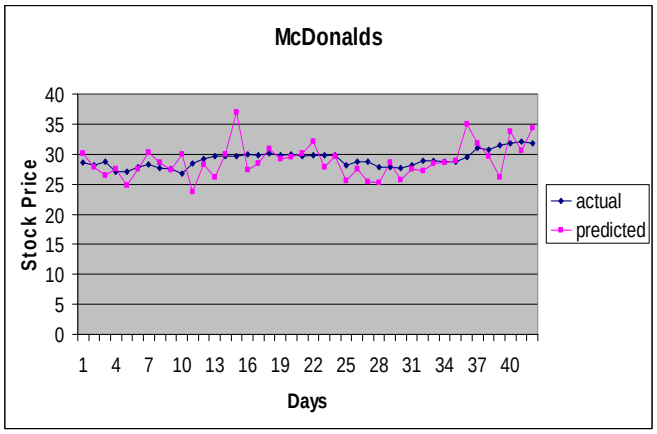
for each view-update during training
If the validation error is higher than the lowest value seen
Increment count
If count equals change-frequency
If the learning rate minus decrement is greater than zero
Lower the learning rate by decrement
Reset count
Continue
Else
Stop training.

The purpose of the heuristic is to start with an aggressive learning rate, which will quickly find a coarse solution, and then to gradually decrease the learning rate to find a finer solution. In the evaluation, the heuristic algorithm is compared to the "simple" method of training where training continues until either the number of epochs grows to the epochs limit or the total squared error drops to the error limit.

Techniques such as boosting were applied to give better performance from the start. Parameter weights were initialised with values after careful assessment of each stock and varied considerably. The network was trained over a series data of 40 days using batch training over both fixed and variable epochs for different stocks. The learning rate was gradually decreased over the time period.

Some results are shown below after batch training of 10,000 iterations.





7. Forecasting

The coefficient of determination from Drossu and Obradovic (1996) is used to compare the networks' forecasting performance and is given in Equation

$$r^2 = 1 - \frac{\sum_{i=1}^{n}(x_i - \hat{x}_i)^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

The number of data points forecasted is n, Xi is the actual value, $\hat{x}_i$ is the forecast value, and $\bar{x}$ is the mean of the actual data. The coefficient of determination can have several possible values:

$$r^2 = \begin{cases} 1 & \text{if } \forall i\ \hat{x}_i = x_i \\ 0 > k > 1 & \text{if } \hat{x}_i \text{ is a better forecast than } \bar{x} \\ 0 & \text{if generally } \hat{x}_i = \bar{x} \\ k < 0 & \text{if } \hat{x}_i \text{ is a worse forecast than } \bar{x} \end{cases}$$

A coefficient of determination close to one is preferable. To compute the coefficient of determination in the evaluation, the forecasts for the networks trained on the original, are compared to the original data series.

8. Conclusion

The framework deployed was a robust and computationally inexpensive network. The main advantage of this approach is the possibility of easy modification in the functions depending on particular stocks and indices. This makes the network responsive to fluctuations and trends while at the same time achieving low error rates. Its efficiency lends itself to a future implementation of a real time stock predictor which will predict on a minute by minute basis.

We also drew some important empirical conclusions. First, in some cases the heuristic worked better than the simple method and quickly trained a neural network to generalize the underlying form of the data series, without over fitting. In other cases, such as with the less hidden layer network trained on the original data series, the simple training method produced better results. Unfortunately, one training method does not appear to be clearly better than the other. Second, it appears from the training time and/or error that it is difficult or impossible to determine the network's forecasting ability. The only way to determine forecasting ability is to make a forecast, compare it to known values, and compute the coefficient of determination. Third, the committee method of forecasting proved better in only a couple of cases.

One goal of the research was to determine the performance of various neural network architectures in forecasting various data series. The empirical evaluation showed that increasingly noisy data series increasingly degrade the ability of the neural network to learn the underlying form of the data series and produce good forecasts. Depending on the nature of the data series, taking the moving average may or may not be appropriate to smooth the fluctuations. Non stationery seen in the mean also degraded the network's forecasting performance. We also concluded that lesser number of hidden units produced better results for simpler data but was not very suitable for complex data series.

An important limitation is the inability of the network to assimilate qualitative information. This is compensated to some extent by the inclusion of outlook parameters. Thus while this framework alone is insufficient for investment purposes, this approach can be extremely helpful to an analyst or an informed investor who can use this network in supplement to their predictions.

While it may never be possible to provide definite predictions, research must continue in pursuit of better profit opportunities and more complete understanding of stock market mechanisms.

9. Acknowledgements

Since this research was undertaken during the 3rd year of our undergraduate study, we would primarily like to acknowledge our professors and colleagues.

Neural Networks – Prof. Ashish Pandey, PhD IIT Delhi

Digital Signal Processing – Prof. U.S. Tiwary, PhD IIIT Allahbad

Dr. G.C. Nandi, Dean, IIIT Allahbad

References:

[1] H.White, Economic Prediction Using Neural Networks :The Case Of IBM Daily Stock Returns ,*IEEE International conference on Neural Networks vo.l 2,* 1988

[2] H.C.Romesburg, *Cluster Analysis for Researchers* (Belmont, Lifetime Learning Publications, 1984).

[3] E..Schoenberg, Stock Prediction Using Neural Networks ,A project Report ,*Neurocomputing vol2* June 1990 17-54

[4] S.Dutta and S.Shekar, A non liner approach of neural computation, *Neural Computation* 19998 453-450

[5] M.J. Pring, *Technical Analysis Explained* (New York, McGraw Hill, 1985)

[6] Drossu, R., & Obradovic, Z. (1996). Rapid Design of Neural Networks for Time Series Prediction. IEEE Computational Science & Engineering, Summer 1996, 78-89.

[7] Hebb, D. O. (1949). The Organization of Behavior: A Neuropsychological Theory. New York: Wiley & Sons.