International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:1, No:5, 2007

Investigation of Some Technical Indexes in Stock Forecasting Using Neural Networks

Myungsook Klassen

***Abstract*—**Training neural networks to capture an intrinsic property of a large volume of high dimensional data is a difficult task, as the training process is computationally expensive. Input attributes should be carefully selected to keep the dimensionality of input vectors relatively small.

Technical indexes commonly used for stock market prediction using neural networks are investigated to determine its effectiveness as inputs. The feed forward neural network of Levenberg-Marquardt algorithm is applied to perform one step ahead forecasting of NASDAQ and Dow stock prices.

***Keywords*—**Stock Market Prediction, Neural Networks, Levenberg-Marquadt Algorithm, Technical Indexes

I. Introduction

STOCK markets are complex, nonlinear, and dynamic. The Random walk theory claims that stock price changes are serially independent, but traders and certain academics have observed that they are reasonably predictable. There are two major types of analysis for predicting stock prices-fundamental and technical. The fundamental analysis measures the intrinsic value of a particular stock by studying everything from the overall economy and industry conditions, to the financial condition and management of companies. It uses revenues, earnings, future growth, return on equity, profit margins, and other data to determine a company's underlying value and potential for future growth. Technical analysis is a method of evaluating stocks by analyzing statistics generated by market activity, past prices, and volume. It looks for peaks, bottoms, trends, patterns, and other factors affecting a stock's price movement. Future values of stock prices often depend on their past values and the past values of other correlated variables. Technical analysis looks for patterns and indicators on stock charts that will determine a stocks future performance.

Time-series problems are prediction applications with one or more time-dependent attributes. A time series is a set of numbers that measures the status of some activity over time. Time series analysis accounts for the fact that data points taken over time may have an internal structure (such as autocorrelation, trend or seasonal variation) that should be accounted for. It is an important research area in many applications such as economic forecasting, sales forecasting, stock market analysis, yield projections, inventory studies, workload projections, and census analysis. The stock market time-series analysis derives the stock's future movement from its historical movement, basing on the assumption that there exists strong enough correlation for prediction. The historical data can be used directly or they can be plugged into many technical indexes for further investigation.

There are many technical indexes used in stock market prediction. Moving average, exponential moving average, weighted moving average, moving average difference oscillator, relative strength index, volume, volume change, moving average convergence-divergence, momentum, rate of return, advance-decline, upside-downside volume ratio, high-low differential index, high-low ratio, volume, and historical volatility are some examples. There are many variations of these and new index terms may be derived from them. Each index has its own meaning, interpretation and many domain experts such as stock traders and consultants advise based on their ad hoc experience which ones to use. A comprehensive description of technical stock market indicators can be found in [4].

Traditionally forecasting research and practice had been dominated by statistical methods [4] [5], but results were insufficient in prediction accuracy. Recently, neural networks have been successfully applied in time-series problems to improve multivariate prediction ability. Neural networks have good generalization capabilities by mapping input values and output values of given patterns. Neural networks are usually robust against noisy or missing data, all of which are highly desirable properties in time series prediction problems. Yoon and Swales [8] compared neural networks to discriminant analysis with respect to prediction of stock market performance and found that the neural networks were superior to a statistical method of the discriminant analysis in its prediction.

Various neural network models have already been developed for the stock market analysis. Back propagation neural network is widely used for it. The network adopts the first order steepest descent technique as learning algorithm. Weights are modified in a direction that corresponds to the negative gradient of the error surface. To overcome the deficiencies of steepest descent learning such as a long

Manuscript received March 15, 2005.

Myungsook Klassen is with the computer science department , California Lutheran Oaks, CA, 91360, USA (phone: 805-493-3321; fax: 805-493-3479; e-mail: mklassen@clunet.edu).

training time and getting stuck in a local minima, some researchers [7] [9] [10] have investigated the use of genetic algorithm, simulated annealing, and Levenberg-Marquadt.

Many factors affect successful training of the back propagation neural networks: the number of hidden layers, the number of hidden units in each hidden layer, initial weights, activation function, learning rate and momentum rate. The learning rate is critical in the sense that too small value will have small convergence and too large value will make the search direction jump wildly and never converge. Input selection and its preprocessing are important. The dimension of the input vector (i.e. the number of input attributes) can affect the convergence speed of the network. The reduction in size of the input vector not only increases the convergence speed but also eliminates the possibility of duplicated information within the input variables that would affect network performance. Increasing the number of hidden layer neurons helps improving network performance, yet many problems could be solved with a very few neurons if only proper input attributes are selected

Understanding the significance of each technical index in predicting stock prices, especially in time-series fashion using neural network will help us reduce the number of inputs. Different technical indexes have been used as input attributes with different neural network models in hope of improving the prediction accuracy. Liu et al [6] used the back propagation neural networks using moving average, deviation from moving average, turnover moving average, and relative index to predict the buy/sell prediction. In Versace et al's work[7], values used are 'open', 'high', 'low', 'close' and 'volume" of a specific stock while Baba [9] used 'change of index', 'PBR', 'changes of the turnover by foreign traders', 'changes of current rates', and 'turnover in local stock market'.

In this research, an attempt was made to analyze effectiveness of some technical indexes derived from raw stock prices when they were used with time series stock price inputs in neural networks. "Volume", another widely used raw data, was not considered in the project. Levenberg-Marquardt back propagation neural network was used for this research. Two stock market indexes, NASDAQ Composite Index and Dow Jones Industrial Average Indexes, which have been the barometer for the US stock market for years, were used for the study.

II. THE LEVENBERG-MARQUARDT ALGORITHM

Gradient-based back propagation neural network has been most commonly used by researchers since it was introduced by Rumelhart in 1986. This simple gradient descent suffers from several convergence problems. One is vanishing gradient at the solution, meaning that the algorithm takes small steps toward a solution. The other is the algorithm takes large steps when the gradient is large which may cause overshooting the local minima. These convergence problems make the training with back propagation neural network slow and hard to learn. The added momentum term in the back propagation might help some of convergence problems. Alternatively, various second order learning methods had been proposed. Among these second order methods, Levenberg-Marquardt (LM) algorithm is one of the popular and effective methods. The algorithm was introduced to feed forward networks [11] for a number of years. It is a Hessian-based algorithm for nonlinear least square optimization. The algorithm take large steps down the gradient where the gradient is small such as near a local minima and takes small steps when the gradient is large.

In the gradient descent neural network learning algorithm, the object function $E(\mathbf{w})$ is the sum of all errors in every input attributes from all training patterns.

$$E(\mathbf{w}) = \frac{1}{2}\sum_{p=1}^{M}\sum_{i=1}^{K}(d_i^{(p)} - y_i^{(p)})^2$$

where $y_i^{(p)}$ and $d_i^{(p)}$ denote calculated output values and expected target values respectively for a training pattern p, and $\mathbf{w}$ is the weights column vector . The total number of training patterns is M and the number of output units is K. The local approximation of the objective function by a quadratic form is

$$E(\mathbf{w}_t + d\mathbf{w}_t) = E(\mathbf{w}_t) + \nabla E(\mathbf{w}_t)^T d\mathbf{w}_t + \frac{1}{2} d\mathbf{w}_t^T \nabla^2 E(\mathbf{w}_t) d\mathbf{w}_t$$

where $\nabla E(\mathbf{w}_t)$ is the gradient vector of the objective function and $\nabla^2 E(\mathbf{w}_t)$ is Hessian Matrix of the objective function. The weight update vector $d\mathbf{w}_t$ is calculated as

$$d\mathbf{w}_t = [\mathbf{J}_t^T \mathbf{J}_t + \mu_t \mathbf{I}]^{-1} \mathbf{JR}$$

where $\mathbf{I}$, $\mathbf{J}$ and μ_t are identity matrix, Jacobian matrix of the first derivatives of the residuals $(d_i^{(p)} - y_i^{(p)})$, and a scalar which controls the size of the trust region respectively. And R is a vector of size M*K calculated as follows $\left(d_1^1 - y_1^1, d_1^2 - y_1^2, \cdots, \cdots, d_2^1 - y_2^1, d_2^2 - y_2^2, \cdots, \cdots, d_M^K - y_M^K\right)$

LM method works extremely well in practice. But its need to invert matrices with dimensions equal to the total number of weight connections of the neural networks and to calculate Jacobian matrix of error functions require extra memory and execution time. Often times the inverse is usually implemented using clever pseudo-inverse methods such as singular value decomposition to reduce the cost of the update. For moderately sized problems however, this model is much faster than simple gradient descent.

III. STOCK MARKET PREDICTION SYSTEM

A. *Technical Indicators used for the study*

Moving Average(MA) This is perhaps the oldest and the most widely used technical indicator. It shows the average value of stock price over time. A simple moving average with the time period n is calculated by $MA(t) = 1/n\sum_{i=0}^{i=n} C_{t-i}$ where C_t is a stock price at time t. The shorter the time period, the more reactionary a moving average becomes. A typical short term moving average ranges from 5 to 25 days, an intermediate-term from 5 to 100, and long-term 100 to 250 days. In our experiment, the window of the time interval n is 5.

Exponential Moving Average (EMA) An exponential moving average gives more weight to recent prices, and is calculated by applying a percentage of today's closing price to yesterday's moving average. The equation for n time period is $EMA(t) = \frac{2}{n+1}C_t + (1 - \frac{2}{n+1}) * MA(t)$. The longer the period of the exponential moving average, the less total weight is applied to the most recent price. The advantage to an exponential average is its ability to pick up on price changes more. In our experiment, the window of the time interval n is 5.

Change in Exponential Moving Average Difference of two EMAs. This can be considered as momentum.
. $EMA\_diff(t) = EMA(t) - EMA(t-1)$.

Change in Moving Average Difference of two MAs. This can be considered as momentum.
$MA\_diff(t) = MA(t) - MA(t-1)$.

Moving Average Convergence/Divergence (MACD) Difference between two exponential moving averages, normally one short moving average and one long moving average. In our experiment, 8 day EMA and 17 day EMA are used. $MACD(t) = Short\_EMA(t) - Long\_EMA(t)$

Rate of returns of Stocks (RRS)

$$x_i = \frac{c_i - c_{i-j}}{c_i \lambda_j} \quad j = 1, 2, \ldots N$$

$$\lambda_j = \frac{2}{p-N} \sum_{i=N+1}^{p} \frac{|c_i - c_{i-j}|}{c_i}$$

where p is the number of stock data in the training set and N is the number of daily lags in the input. In our experiment, the time period N is set to 5.

Relative Strength Index (RSI) An oscillator, introduced by J. Welles Wilder, Jr., is based upon the difference between the average gain vs. the average loss over a given period. The RSI compares the magnitude of a stock's recent gains to the magnitude of its recent losses.

The general formula for the RSI is

$$RSI = 100 - [100/(1 + RS)]$$

where $RS =$ (Average Gain of n-day up)/(Average Loss of n-day down). A high RSI occurs when the market has been rallying sharply and a low RSI occurs when the market has been selling off sharply. It is the most popular over bought/oversold indicator. In our experiment, the time period n is set to 10 (10 weeks).

IV. EXPERIMENTAL RESULTS

A. *Neural network Architecture*

The network, Levenberg-Marquardt back propagation, with one output unit for the next week's predicted average closing price was used for experiments. One hidden layer with 5 to 8 units was used. The number of input units were four time-series values C_{t-1}, C_{t-2}, C_{t-3} and C_{t-4} plus each of derived technical index terms. Trainings were done between 50 and 100 epochs with a learning rate of 0.001, a learning rate decrease rate of 0.1 and its increase rate of 10. A logsig activation function was used. The training termination condition was set as the total epoch of 100 or the goal of 1e-5. For trainings, mean square errors(MSE) fell in the range of 0.0002 to 0.00001.

B. *Data and Preprocessing*

Composite daily closing, high, and low prices for NASDAQ and Dow from June 10, 2002 to Jan 7, 2005 and from March 4, 2002 to Jan 7, 2005 were obtained from http://finance.yahoo.com. Average closing prices for 5 days from Monday through Friday during the time period specified were computed to create 135 weekly averages. A time series of inputs was created with past four weekly averages. The first sample input was weekly closing prices from June 10th , June 17th, June 24th and June 30th in 2002 and its output was the weekly closing price from July 8th, 2002 .

Out of the total 130 samples(although 135 weeks, due to a time series of 5, the sample size is 5 less), first 110 patterns were used during training and the remaining 20 patterns were used to test the neural network's generalization capability. For the MACD technical index, daily high and low prices were used to compute 8 week and 17 week exponential moving averages. The number of training patterns were 109 from June 22, 2002 due to a longer moving average time period of 17. All data, original time series values and computed index values were normalized between 0.75 and 0.25. After each run, normalized prices were converted back to regular prices for evaluation.

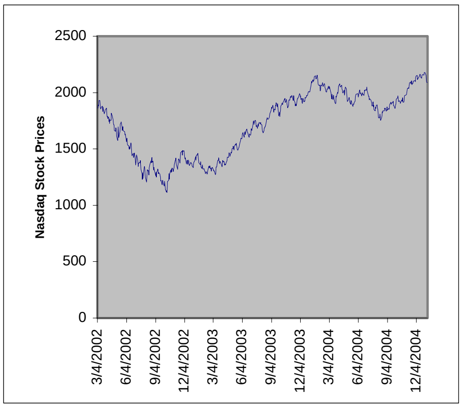

Fig. 1Nasdaq Stock Index Mar 02 – Jan 05

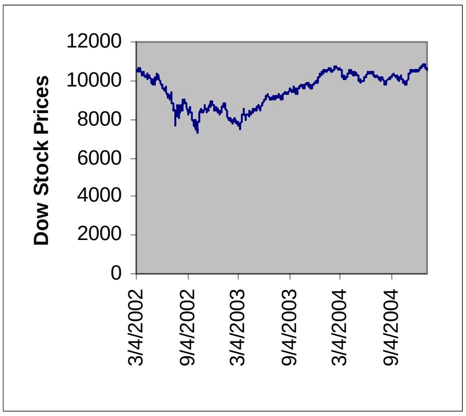

Fig. 2 Dow Stock Index Mar 02- Jan 05

C. Results

Once the network was trained, its performance was measured by a difference between an expected output (in this case, the actual NASDAQ closing price for that week) and the output computed by the network for all testing patterns. Technical indexes were added to four original time-series values to evaluate their goodness. Each time, only one index was added to the original four input attributes. For each set of inputs, the total nine experiments were executed. The percent of the average differences from all testing patterns and for all nine trials are shown in T.

TABLE I NASDAQ ALTERNATIVE TECHNICAL INDEXES PREDICTION RESULTS

Inputs	Difference in Percent
Time Series	1.70
Time Series plus Moving Average(MA)	1.64
Time Series plus Exponential Moving Average(EMA)	1.58
Time Series plus MA Difference	1.17
Time Series plus EMA Difference	1.51
Time Series plus Rate of RRS	1.74
Time Series plus RSI	2.53
Times Series plus MACD	1.45

TABLE II DOW ALTERNATIVE TECHNICAL INDEXES PREDICTION RESULTS

Inputs	Difference in Percent
Time Series	1.07
Time Series plus Moving Average(MA)	0.95
Time Series plus Exponential Moving Average(EMA)	0.90
Time Series plus MA Difference	0.93
Time Series plus EMA Difference	0.94
Time Series plus Rate of RRS	0.96
Time Series plus RSI	1.11
Times Series plus MACD	0.70

V. CONCLUSION AND DISCUSSION

When four time series values $C_{t-1}, C_{t-2}, C_{t-3}$ and C_{t-4} were used as inputs, the average differences between actual values and predicted values for all testing patterns were 0.70% and 1.07% in NASDAQ and Dow respectively. The results are in line with those from other researcher [1] [2].

Moving average related technical indexes, MA, EMA, MA difference, EMA difference showed improved prediction results when added to the time series values. In both cases, EMA showed a better prediction than MA which was anticipated. In MA, past observations are weighted equally while EMA assigns exponentially decreasing weights. Recent observations are given relatively more weight in forecasting than the older observations. As long as there is no abrupt change of a stock index trend, the EMA should predict the following time period price better.

MA difference and EMA difference are the first derivatives of MA and EMA respectively and are supposedly to capture increase/decrease of MA values and EMA values. In NASDAQ case, the prediction rate was improved by 28.6% from the predicted value difference of 1.64% to 1.17% for MA and 4.4% for EMA. In Dow case, prediction capability was also increased, but only slight. MACD showed the most improvement of prediction capability. It can be interpreted that when a longer MA along with a short MA are used, the index is able to capture the stock price trend well.

The prediction rates with RSI became lower when used with time series values. RSI is overbought/oversold indicator, computed using values such as the number of closing price “up” days and “down” days. The characteristic of the index is different from the time series values which are closing prices. As a result, adding the index term didn’t contribute the neural network generalize the stock price trend any better. In fact, the result shows the opposite. This case demonstrates that simply adding more input attributes does not help neural network prediction strength.

To test our finding further, we did the testing with MA, EMA, MA difference, EMA difference, RSI and RRS plus four time series values. The predicted stock value had 5.28% difference from the expected prices. This is over 3 times higher error than when only 4 time series inputs used. During the training, the network output error was observed: initially it went down and stayed in the plateau for a long time and then start going down to a very small MSE near 1.40E-06 with the similar epoch of 100. This observation is explained that the network learned the trend of input data set in a meaningful manner initially when it reached the plateau, but with further iterations, it was over trained and generalized the input data in an unexpected way.

To sum up, the efficiency of back propagation neural network was most improved by addition of the technical index term MACD. Using a small number of relevant and good attributes make the neural network training time short and generalize better. The key is to use a small number of good attributes.

References

[1] Andrew S. Andreou et al. Testing the probability of the Cyprus Stock Exchange: The Case of an Emerging Market. Proceedings of the IEEE-ENNS IJCNN. 2000.

[2] S. I. Ao Hybrid Intelligent System for pricing the Indices of Dual-Listing Stock Markets. Proceedings of the IEEE International Conference on Intelligent Agent Technology. 2003

[3] Man-Chung Chan, Ch-Cheong Wong, and Ch-Chung Lam. Financial Time Series Forecasting by neural Network Using Conjugate Gradient Learning Algorithm and Multiple Linear Regression Weight Initialization. Computing in Economics and Finance 2000. July 2000.

[4] Robert W. Colby, and Thomas A. Meyers. The Encyclopedia of technical Market Indicators, Dow Jones-Irwin, Homewood, IL, 1988.

[5] Richard Johnson, and Dean Wichern. Applied Multivariate Statistical Analysis. 4th Ed. Prentice Hall, New Jersey, 1998.

[6] H. Tong, & K. S. Lim. Threshold Autoregressive, Limit Cycles and Cyclical Data. Journal of the Royal Statistical Society Series B, 42(3), 245-292. 1980.

[7] Qiong Liu, Xin Lu, Fuji Ren and Shingo Kuroiwa. "Automatic Estimation of Stock Market Forecasting and Generating the Corresponding Natural language Expression", IEEE Proceedings of the International Conference on Information Technology: Coding and Computing. 2004.

[8] M. Versace, R. Bhatt, O. Hinds, and M. Shiffer. "Predicting the exchange traded fund DIA with a combination of genetic algorithms and neural networks." Expert Systems with applications, 2004. Elsevier.

[9] Yoon, Y. and G. Swales, "Predicitng stock price performance," proceeding of the 24th Hawaii International Conference on System Sciences, 4, 156-162, 1997.

[10] N. Baba, I. Naoyuki, and A. Hiroyuki. "Utilization of Neural Networks & GAs for Constructing Reliable Decision Support Systems to Deal Stocks." Proceedings of IEEE-INNS-ENNS International Joint Conference on Neural Networks. 2000.

[11] N. Ampazis, and S. J. Peranonis. "Levenberg-Marquardt Algorithm with adaptive Momentum for the Efficient Training of Feedforward Networks." Proceedings of IEEE-INNS-ENNS International Joint Conference on Neural Networks. 2000.

[12] M. T. Hagan and M. B. Menhaj. "Training Feedforward networks with the Marquardt Algorithm." IEEE Transactions on Neural Networks, 5(6), 989-993, November 1994.