Stock Forecasting with Feedforward Neural Networks and Gradual Data Sub-Sampling

Shekhar Gupta and Lipo Wang

School of Electrical and Electronic Engineering
Nanyang Technological University
Block S1, 50 Nanyang Avenue, Singapore 639798
elpwang@ntu.edu.sg

Abstract. We use feed-forward neural networks to forecast and trade the future index prices of the Standard and Poor's 500 (S&P 500). The effect of training the network with the most recent data, together with gradually sub-sampled past index data, has been studied in this research. We also study the effect of past NASDAQ 100 data on the prediction of future S&P 500. A daily trading strategy has been used, to buy/sell, according to the predicted prices, and hence calculate the directional efficiency and the rate of returns for different periods. We are able to obtain significantly higher returns compared to earlier work. There are numerous exchange traded funds (ETFs), which attempt to replicate the performance of S&P 500 by holding the same stocks in same proportion as the index, and therefore, giving the same percentage returns as S&P 500. Therefore, this study can be used to invest in any of the various ETFs, which replicates the performance of S&P 500.

Keywords: S&P 500, NASDAQ, neural networks, sub-sampling, prediction, stock.

1 Introduction

Stock prices are highly dynamic and non-linear, which is very hard to model by even the best financial models. Stock prices are affected by various factors, such as crude oil prices, exchange rates, interest rates, as well as political and economic climate. The information about future stock prices can also be studied merely by its historical prices. With globalization and ease of investment in international and national markets, many people are looking towards stock markets for profit. There is a high degree of uncertainty in the stock prices, which makes it difficult for investors to predict price movements. Hence prediction of stock prices becomes very important for financial analysts as well as the general public.

The Efficient Market Hypothesis (EMH) [1] states that no information can be used to predict the stock market in such a way as to earn greater profits from the stock market in an efficient market. There have been studies to show the accountability of the EMH [2], but other studies have proven otherwise [3]. Despite of the EMH, there are various views that oppose the EMH and support predictability of the stock market.

Many stock prediction methods have been used by investors and traders, e.g., fundamental analysis [1], technical analysis [1], multivariate regression [4], and artificial neural networks (NNs) [4]-[12]. There are two categories of prediction systems, i.e., one using the past prices and the other using fundamental data such as exchange rates, gold prices, interest rates, etc.. In particular, Rodrigues [6] used a relatively simple NN to predict and trade in the Madrid Stock Market Index. It used nine lagged inputs to predict the prices and make a Buy/Sell Decision, although this model did not perform well in a bullish market. Chang et al [8] used a NN model based on past prices and were able to obtain around 16% of returns per annum by the weekly prediction model, but it failed for daily prediction model.

2 Our Approach and Experimental Results

The main focus of the research is to predict future stock index prices, calculate directional efficiency and rate of returns, in order to develop a trading system capable of delivering high profit over a long period of time.

We study one of the most popular indexes, i.e., the Standard and Poor's 500 (S&P 500). The reason behind choosing the S&P 500 is that many exchange traded funds (ETFs) follow the performance of the S&P 500 and an intelligent trading system can help investors.

We focus on daily trading using feedforward NNs. The output of the NN is the next day's index price. The input to the NN is the past S&P 500 index prices, optionally with past NASDAQ 100 index prices. The number of inputs varies from 10 day lagged values to 40 day lagged values of the S&P 500 index closing prices. The main source of historical index prices is Yahoo Finance. 2 sets of data have been downloaded, i.e., (1) the closing prices of S&P 500 index from 9 January 1950 to 15 January 2010 and (2) the closing prices of S&P 500 and NASDAQ 100 index from 7 January 1991 to 15 January 2010. Due to public holidays, there were data missing at various days in the raw time series, which had to be adjusted to account for the missing values. A 5-day lagged average was used to fill in the missing data.

We train the NN with Levenberg-Marquardt algorithm. The performance measure is the Mean Square Error (MSE) function, with the Tan Sigmoid (TANSIG) as the transfer function. The number of hidden layers varies from 1 to 2. The number of neurons in the input layer varies from 10 to 40, with search intervals of either 5 or 10 during parameter adjustments. The output layer consists of only 1 neuron, i.e., the predicted price on the next working day. The number of neurons in the hidden layers depends on the number of neurons in the input layer.

The raw time series was divided into 3 sets of data, i.e., training, validation and testing data sets. These data sets which were in the form of a p*1 matrix was then divided into n*m matrices, which were used for batch training of the NN (we use * to denote "multiply" in this paper).

For the training purpose, if the inputs had p data items, where p = 15040, then this was divided into matrices of size n*m which is the input matrix, where n = 10, 15, 20, or 40, and m =15000. This was also divided into matrices of size 1*m, which would

be the target matrix for training purpose. Here n is the number of inputs in the NN, and m is the number of training sets in batch training.

After this, the training parameters were selected for training of the NN. Since the Levenberg-Marquardt training algorithm is fast, the number of epochs was limited to 100, so as to prevent over training of the NN. If the number of epochs were selected to be more, then this would affect the generalization property of the NN and would lead to memorization of the network, which can be disadvantageous in the prediction of future values.

The validation matrix was created from the raw data, i.e., by creating an n*t matrix, where n = 10, 15, 20 or 40 (same as the training data). t can vary according to the number of validations required, in our case, it varied from 200 to 400, depending on the size of the training data set. This validation was used for early stopping so as to prevent over-fitting of the NN. After the training of the NN, the testing for the trained network was done over different periods of time, ranging from 1 year to 2 years, i.e., from 250 to 420 prediction points, and with different market conditions, i.e., before, during and after recession conditions. The input test matrix was of the size n*u, where n = 10, 15, 20 or 40 (same as that of the training and validation matrix, i.e., the number of inputs) and u varied from 250 to 420, i.e., the number of testing sets in the batch.

The predicted outputs were then used to obtain the directional efficiency and rate of returns for the specified period of time. When the predicted price for the next day is less than today's price, a Sell decision is made. And when the predicted price for the next day is more than today's price, then a Buy decision is made. The Buy/Sell decision is made when the trading system shows a change in the trend. Based on these trading rules, the results are calculated over a period of time.

Further experiments were carried out to study effects of gradual data sub-sampling (GDSS) and effects of past NASDAQ 100 data on S&P 500 prediction, in comparison published results obtained with various other techniques.

In our GDSS, the older the data, the less relevant they will be for predicting the future, and therefore the lower the sampling frequency. This technique was applied on the 1st data set, i.e., from January 1950 till April 2008, for the purpose of training. Originally there were 15,200 sets of training data, which were reduced to 6,700 sets of training data in the following way:

a) 900 training data points were selected from the 1st 5400 data points.
b) 2000 training data points were selected from the next 6000 data points.
c) all remaining 3800 data points were selected.

This technique ensured that the historical trends are not ignored, and at the same time, the system shows more emphasis to the current market situations.

For this part, the 2 networks with best efficiency from the above research were used to train and test the network for the period of testing in various other research papers. The 2 networks which were used are as follows:

a) A network was used to train with the GDSS technique with 10 lagged values.
b) A network was used to train with the inclusion of the NASDAQ 100 index, with 10 lagged values each of NASDAQ 100 and S&P 500.

The period of comparison was from January 2004 till December 2004. Our results were compared with the following 3 sources:

a) The Trading Solutions [10], an online website for trading solutions software package.

b) The "integrating a piecewise linear representation" (IPLR) method [8], which determines trading points according to trends in the stock movements, by combining genetic algorithms and NNs.

c) Predicting the stock prices using a three artificial NN (3-ANN) system [9], one system each for bullish, bearish and choppy markets, respectively.

The results were as follows:

Table 1 Comparisons of rates of return by various techniques for January – December 2004. The last two results were obtained in this paper.

Technique	Trading Solutions [10]	IPLR [8]	3 – ANN System [9]	GDSS	Effect of NASDAQ
Rate of Return	11%	35.7%	18.36%	25.42%	16.34%

The rate of return was the maximum for the IPLR system, followed by our approach of sub-sampling. We will now show rates of returns for our GDSS technique for a number of periods:

Table 2 Rates of returns with our GDSS technique for different periods

Time of Testing	January 2004 – December 2004	April 2008 - January 2010	January 2009 – January 2010
Rate of Return	25.42%	47.19%	44.05%

Although the rates of return were not very high with our technique of data lessening for the period of January – December 2004, but for the period from April 2008 till January 2010, the rate of return was as high as 47.19%, which are way above the technique of IPLR for the period of January – December 2004.

Following if a table of Rate of Returns for the training NN, with the effect of NASDAQ 100.

Table 3 Rates of returns with effect of NASDAQ for different periods

Time of Testing	January 2004 – December 2004	April 2008 - January 2010	January 2009 – January 2010
Rate of Return	16.34%	34.77%	72.40%

Table 4 Training, Validation and Testing of Network with Further Lessening of Data from January 1991 till January 2010

		Period 1			Period 2		
		Return Value	Rate of Return	Directional Efficiency	Return Value	Rate of Return	Directional Efficiency
1	**10-5-1:**	536.65	38.66	56.43	399.06	45.68	55.16
2	**10-7-1:**	-121.22	-8.73	51.19	97.60	11.17	49.21
3	**10-9-1:**	655.17	47.20	57.62	384.83	44.05	54.76
4	**10-15-1:**	341.45	24.60	53.33	353.41	40.45	51.19
5	**10-21-1:**	556.71	40.46	58.57	310.84	34.37	54.76

Table 5 Training, Validation and Testing of Network with the effect of NASDAQ 100 for Continuous Data from January 1991 till January 2010

		Period 1			Period 2		
		Return Value	Rate of Return	Directional Efficiency	Return Value	Rate of Return	Directional Efficiency
1	**20-7-1:**	482.80	34.78	57.86	597.48	72.41	59.55
2	**20-11-1:**	399.25	28.59	54.29	436.05	57.95	55.06
3	**20-15-1:**	280.95	20.24	53.81	446.03	59.28	54.68
4	**20-22-1:**	89.67	6.52	53.10	283.53	37.68	54.68
5	**20-30-1:**	30.10	2.17	54.05	343.28	45.62	55.06

3 Conclusions and Discussions

The period from January 2004 till December 2004 was of relatively stable markets, where the annual increase in the index price was not very high. But the period from April 2008 till January 2010 included a period of recession where there was huge dip in the index prices and a market which was recovering from recession. Therefore, from the analysis done, it can be seen that our system was relatively stable and

efficient in giving good rate of return in all market situations, and high rate of return in special market situations, where there is more movement in the index values.

Also, the effect of NASDAQ was suitable to give high rate of returns, i.e., around 72.4% returns in good market conditions.

References

1. Malkiel, B. G., *A Random Walk Down Wall Street*. London, New York: W. W. Norton & Company, 1999.
2. Eugene F. Fama, Lawrence Fisher, Michael Jensen and Richard Roll. The Adjustment of Stock Market Prices to New Information. International Economic Review, Vol. X, Feb 1969, pp. 1-21.
3. Thaler, Werner F.M. De Bondt and Richard. Does the Stock Market Overreact? Journal of Finance, Vol. 40 Issue 3, July 1985, pp. 793-805.
4. P. C. Chang, Y.W.Wang, and W. N. Yang. An investigation of the hybrid forecasting models for stock price variation in Taiwan. Vol. vol. 21, no. 4, 2004, pp. 358–368.
5. S. C. Chi, H. P. Chen, and C. H. Cheng. A forecasting approach for stock index future using grey theory and neural networks. Proc. IEEE Int. Joint. Conf. Neural Netw. Vol. vol. 6, pp. 3850–3855, Jul. 1999.
6. Femdndez-Rodriguez, F., Gonzdlez-Martel, C, Sosviall-Rivero, S. On the Profitability of Technical Trading Rules based on Artificial Neural Networks: Evidence from the Madrid Stock Market. Economics Letter. 2000.
7. P.D. Yoo, M.H. Kim, and T. Jan. Karach, Financial Forecasting: Advanced Machine Learning Techniques in Stock Analysis, December, Proceedings of the 9th IEEE International Conference on Multitopic, pp. 1-7, 2005.
8. Pei-Chann Chang, Chin-Yuan Fan, and Chen-Hao Liu. Integrating a Piecewise Linear Representation Method and a Neural Network Model for Stock Trading Points Prediction. IEEE Transactions on Systems, Man, and Cybernetics—Part C: applications and reviews, Vol. 39, No. 1, January 2009.
9. Fu, C. Xiang, Predicting the Stock Market using Multiple Models. ICARCV 2006.
10. NeuroDimension, Inc. (2008). Trading solution company [Online]. Trading Solutions. [Online] Available: http://www.tradingsolutions.com/.
11. Lipo Wang and Xiuju Fu, *Data Mining with Computational Intelligence*, Springer, Berlin, 2005.
12. Lipo Wang (ed.), *Support Vector Machines: Theory and Applications*, Springer, Berlin, 2005.