ijecs open access www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 3 Issue 9 September, 2014 Page No. 8347-8351

Performance analysis of Neural Network Algorithms on Stock Market Forecasting

Manasi Shah, Nandana Prabhu, Jyothi Rao

Student, ME Computers,
K.J.Somaiya College of Engineering,
Vidyavihar, Mumbai

Associate Professor,
Department of Information Technology,
K.J.Somaiya College of Engineering,
Vidyavihar, Mumbai

Associate Professor,
Department of Computer Engineering,
K.J.Somaiya College of Engineering,
Vidyavihar, Mumbai

ABSTRACT: Artificial Neural Networks (ANN) have been used in stock prediction extensively as it provides better results than other techniques. In this paper, different architectures of ANN, namely, simple feed forward back propagation neural network (FFBPNN), Elman Recurrent Network, Radial Basis Function network (RBFN) are implemented and tested to predict the stock price. Levenberg-Marquardt Back-propagation algorithm is used to train the data for both FFNN and Elman Recurrent Network. These techniques were tested with published stock market data of Bombay Stock Exchange of India Ltd., and from the results it is observed that FFBPNN gives better results than Elman Recurrent Network.

General Terms: Price Prediction, Stock Market, Artificial Neural Network (ANN).

Keywords: Feed Forward Back Propagation Network, Elman Recurrent Network, Radial Basis Function Network.

I. INTRODUCTION:

A stock market or exchange is the centre of a network of transactions where securities buyers meet sellers at a certain price. It is essentially dynamic, non-linear, complicated, nonparametric, and chaotic in nature. Stock market price depends of various factors which can be divided broadly as quantitative and qualitative factors.

Quantitative factors include daily open, close, high, low price of individual equities and even daily traded volume, stock market index, currency exchange rate, etc. **Qualitative factors** include socio-political factors, news, general economic conditions, commodity price index, political events, firms' policies, bank rate, exchange rate, investors' expectations, institutional investors' choices, movements of other stock market, psychology of investors, etc.

The **Efficient Market Hypothesis (EMH)** theory states that any form of information cannot be used for generating extraordinary profits from the stock market, as the stock prices always fully reflect all available information. The **Random Walk Hypothesis** states that stock price movement does not depend on past stock. Contradicting these theories, many studies show that it is possible to predict stock market satisfactorily using various techniques.

Technical analysis includes concepts such as the trending nature of prices, confirmation and divergence, and the effect of traded volume. **Fundamental analysis** is based on economic data of companies such as annual and quarterly reports, balance sheet, income statements, earnings forecast, past performance of the company, etc. In **Traditional Time Series Prediction**, the model analyzes historic data and attempts to approximate future values of a time series as a linear combination of these historic data. **Machine Learning** uses a set of samples to generate an approximation of the underling function that generated the data. The Nearest Neighbor, Support Vector Machine and the Neural Networks Techniques are methods that have been applied to market prediction. The efficiency of neural network in predicting stock market cannot be overlooked since accuracy of prediction using self-learning neural network has even been reached to 96%.[6]

Samek and Varacha [2] studies time series prediction using artificial neural networks. The special attention is paid to the influence of size of the input vector length. The system in [11] uses Adaptive Neuro-Fuzzy Inference System (ANFIS) for taking decisions based on the values of technical indicators.

The rest of the paper is organized as follows. Section II presents the methodology; technical indicators used and describe various architectures. In section III, implementation of ANN architectures is shown. Section IV shows the result obtained by implementation shown in section III and taking hidden neurons as 16. Conclusion is shown in Section V.

II. METHODOLOGY:

Ten technical indicators mentioned in Table 1 were selected as inputs of the proposed models [4]. Experimental results showed that average performance of ANN model (75.74%) was found significantly better than that of SVM model (71.52%).

To balance between generalization and over-fitting of ANN, we use only one hidden layer; as a three layer FFNN can model any input-ouput relationship.[6]

Preprocessing:

Normalization has been applied because of the high range of our dataset. Function *zscore* in MATLAB has been used to normalize data between 0 to 1, and -1 to +1.

Proposed Prediction Technique:

In this report, following structures of artificial neural networks are chosen to be tested: Multilayer feed-forward neural network, because of its wide usage, Elman neural network as the representative of the recurrent neural networks, radial basis function neural network, because it provides simple training with good prediction performance and adaptive neural network due to its simplicity.[1]

1 Multilayer feed-forward neural networks

Multilayer feed-forward neural networks have neurons structured in layers and the information flows only in one direction (from input to output). A feed-forward network with one hidden layer and enough neurons in the hidden layer can fit any finite input-output mapping problem.

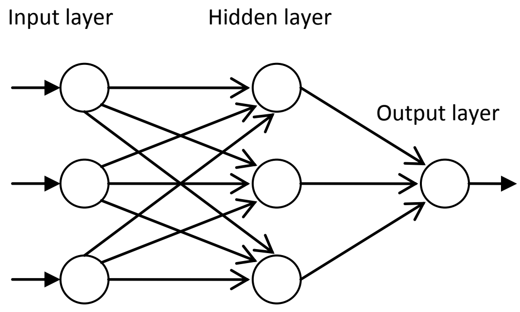

Figure 1: Feed Forward Network

2 Recurrent neural networks

Elman networks are feed-forward networks with the addition of layer recurrent connections with tap delays. Elman networks with one or more hidden layers can learn any dynamic input-output relationship arbitrarily well, given enough neurons in the hidden layers. Elman neural networks were selected as a representative of large group of recurrent neural networks.

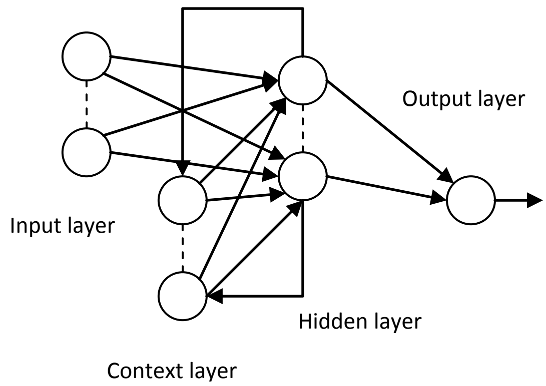

Figure 2: Elman Recurrent Network

3 Radial basis function neural networks

Typical RBFNN contains two layers, while the hidden layer utilizes radial basis transfer function and output layer employs linear transfer function.

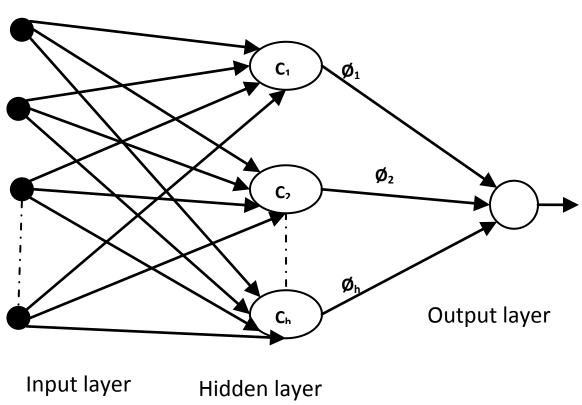

Figure 3: Radial Basis Network

III. EXPERIMENTATION AND RESULTS:

The data used for network training and validation comprises of daily figures for equities listed in IT sector in BSE, i.e., Financial Technologies, Geometric Ltd, Infosys Ltd and Wipro Ltd from January 2001 to January 2014.
Each data set is divided into two parts, one is used for training and the other is used for testing.

1 Multilayer feed-forward neural networks

The training function used are Levenberg-Marquart algorithm (*trainlm*), Gradient descent with adaptive learning rate back-propagation (*traingda*) and Gradient descent with momentum and adaptive learning rate back-propagation (*traingdx*) algorithm built in MATLAB Neural Network Toolbox.

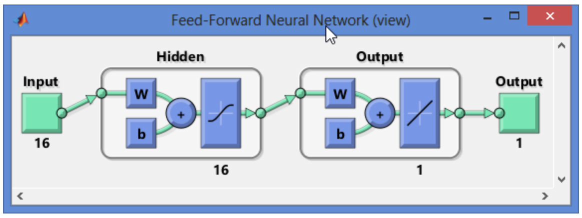

Figure 4: Feed Forward Neural Network on MATLAB

The tested structures had 2,5,10,16,20,50 neurons with Hyperbolic tangent sigmoid transfer function(*tansig*), Log-sigmoid transfer function(*logsig*), Radial basis transfer function(*radbas*) in the hidden layer and one neuron with linear transfer function(*purelin*) in the output layer.

Table 1: List of Technical Indicators

Technical Indicator	Formula
Typical Price (M)	$\frac{H_t + L_t + C_t}{3}$
Moving Average (MA)	$C_t + C_{t-1} + C_{t-2} + C_{t-3} + \cdots . + C_{t-n+1}$
Stochastic %K	$\frac{C_t - L_n}{H_n - L_n} * 100$
Stochastic %D	$\sum_{i=0}^{n-1} \%K_{t-i}/n$
Momentum	$C_t - C_{t-4}$
Rate of Change	$\frac{C_t}{C_{t-n}} * 100$
Larry Williams %R	$\frac{H_n - C_t}{H_n - L_n} * 100$
AD Oscillator	$\frac{H_t - C_{t-1}}{H_t - L_t}$
Disparity 5 Days	$\frac{C_t}{MA_5} * 100$
Disparity 10 Days	$\frac{C_t}{MA_{10}} * 100$
Price Oscillator	$\frac{MA_5 - MA_{10}}{MA_5}$

Commodity Channel Index	$\frac{M_t - SM_t}{0.015 * D_t}$, $D_t = \frac{\sum_{i=1}^{n} \lvert M_{t-i+1} - SM_t \rvert}{n}$, $SM_t = (\sum_{i=1}^{n} M_{t-i+1})/n$
Price Volume Trend	$\frac{C_t - C_{t-1}}{C_{t-1}} * V$

C: Closing Price, L: Low Price, H: High Price

2 Recurrent neural networks

In this article the hidden layer contained ten neurons with *tansig*, *logsig*, *radbas* and the output layer of the Elman neural network used linear transfer function (*purelin*). The *trainlm*, *traingda*, *traingdx* algorithm was used for the training.

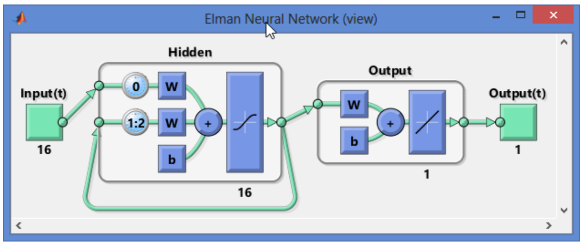

Figure 5: Elman Recurrent Network in MATLAB

3 Radial basis function neural networks

Function *newrb* adds neurons to the hidden layer of a radial basis network until it meets the specified mean squared error goal. (MATLAB Neural Network Toolbox function *newrbe* adds as many radbas neurons to the hidden layer as the size of input.

IV. RESULTS:

Figure 6 shows predicted v/s actual output by applying Feed-Forward Back-Propagation Neural Network on Hybrid Indicators on Infosys data. The output obtained by this network is best when compared to other two architectures.

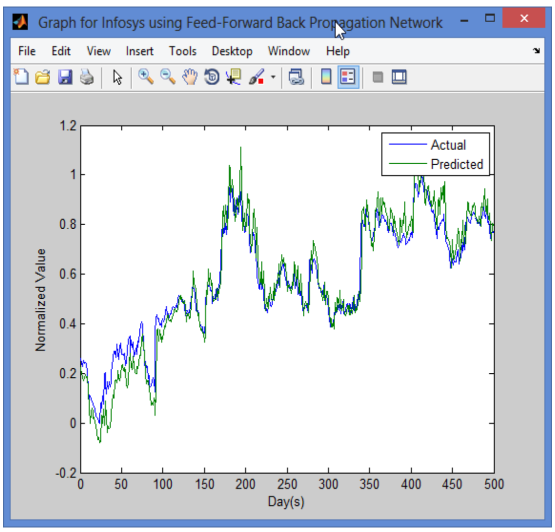

Figure 6: Output of FFBPNN for Hybrid Indicators

Figure 7 shows output shows predicted v/s actual output by applying Elman Recurrent Neural Network on Hybrid Indicators on Infosys data.

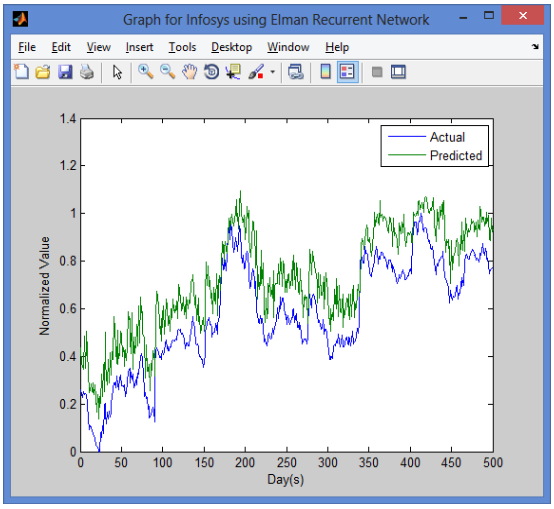

Figure 7: Output of Elman Recurrent for Hybrid Indicators

Figure 8 shows output shows predicted v/s actual output by applying Radial Basis Neural Network on Hybrid Indicators on Infosys data.

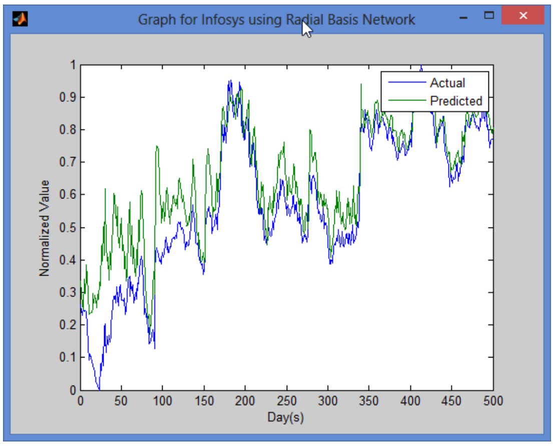

Figure 8: Output of RBF for Hybrid Indicators

V. CONCLUSION:

Using technical indicators along with historical time series data yields better results as compared to results obtained by using only technical indicators as shown in

Table 2. Feed forward back propagation network is better than Elman recurrent network. Elman is only used for historical data and research purposes nowadays. Levenberg-Marquardt back-propagation, when used as a training function, gives better accuracy than training using Gradient descent with adaptive learning rate back-propagation. Radial Basis Network also gives promising results but it takes lot of time to train the network, if error goal is high. From our experiment, we conclude that number of neurons in hidden layer should be between *n* to *2n* where *n* is the number of nodes in input layer and considering that output layer has only one node.

REFERENCES:

[1] Sugandha Saha, "Comparison of different Neural Network Models for Stock Market Prediction, Twenty-Fifth International Conference on Software Engineering and Knowledge Engineering, June 2013".

[2] David Samek and Pavel Varacha, "Time series prediction using artificial neural networks: single and multi-dimensional data", International Journal Of Mathematical Models And Methods In Applied Sciences, Issue 1, Volume 7, 2013.

[3] Ayodele et al., "Stock Price Prediction using Neural Network with Hybridized Market Indicators", Journal of Emerging Trends in Computing and Information Sciences, VOL. 3, NO. 1, January 2012 ISSN 2079-8407.

[4] Kara et al, "Predicting direction of stock price index movement using artificial neural networks and support vector machines: The sample of the Istanbul Stock Exchange", Expert Systems with Applications 38 (2011) 5311–5319.

[5] Samarth Agrawal, Manoj Jindal, G. N. Pillai , "Momentum Analysis based Stock Market Prediction using Adaptive Neuro-Fuzzy Inference System (ANFIS)", Proceedings of IMECS 2010.

[6] Abhishek Kar, "Stock Prediction using Artificial Neural Networks", Dept. of Computer Science and Engineering, IIT Kanpur.

Table 2: Performance Measure for Technical Indicators and Hybrid Indicators using Neural Network Architectures

Data Set	Architecture	Technical Indicators			Hybrid Indicators		
		RMSE	MAPE	ACCURACY	RMSE	MAPE	ACCURACY
INFOSYS	FFBPNN	0.0958	11.5602	92.2361	92.2361	7.2302	**94.9715**
	ELMAN	0.1052	14.8701	90.4099	0.1628	21.7713	84.7908
	RBF	0.1119	9.7581	91.6234	0.0771	8.3235	94.6030
WIPRO	FFBPNN	0.1927	24.7584	83.9847	0.1223	17.0915	**89.7812**
	ELMAN	0.2684	41.1723	76.5044	0.2530	38.9206	75.9443
	RBF	0.1335	14.5337	89.1947	0.2107	19.7995	83.4504
FINANCIAL	FFBPNN	0.1570	16.4628	86.3105	0.1192	12.7133	**90.2159**
	ELMAN	0.1686	24.8067	84.1721	0.2214	28.2560	79.4238
	RBF	0.1293	11.4043	90.0570	0.1514	22.2847	86.7702
GEOMETRIC	FFBPNN	0.0918	10.2483	92.5993	0.0745	7.9855	**94.3886**
	ELMAN	0.1501	14.6201	87.0882	0.0946	11.9969	92.1638
	RBF	0.2052	28.0125	82.0614	0.2363	16.3455	82.2150