Financial Time Series Forecast Using Neural Network Ensembles

Anupam Tarsauliya, Rahul Kala, Ritu Tiwari, Anupam Shukla

Soft Computing & Expert System Laboratory, ABV - IIITM,
Gwalior-474010, India
{anupam8391, rkala001, drritutiwari, dranupamshukla }@gmail.com

Citation: A. Tarsauliya, R. Kala, R. Tiwari, A. Shukla (2011) Financial Time Series Forecast Using Neural Network Ensembles, *Proceedings of the International Conference on Swarm Intelligence, Springer Lecture Notes in Computer Science*, Chongqing, China, pp 480- 488.

Final Version Available At: http://link.springer.com/chapter/10.1007%2F978-3-642-21515-5_57

Abstract. Financial time series has been standard complex problem in the field of forecasting due to its non-linearity and high volatility. Though various neural networks such as backpropagation, radial basis, recurrent and evolutionary etc. can be used for time series forecasting, each of them suffer from some flaws. Performances are more varied for different time series with loss of generalization. Each of the method poses some pros and cons for it. In this paper, we use ensembles of neural networks to get better performance for the financial time series forecasting. For neural network ensemble four different modules has been used and results of them are finally integrated using integrator to get the final output. Gating has been used as integration techniques for the ensembles modules. Empirical results obtained from ensemble approach confirm the outperformance of forecast results than single module results.

Keywords: Neural Network, Ensemble, BPA, RBF, RNN, Time Series

1 Introduction

Financial forecasting is a standard benchmark example of time series analysis problem which is challenging due to high noise, volatility, and non-linearity [1]. These characteristics suggest that there is no complete information that could be obtained from the past behavior of financial markets to fully capture the dependency between the future price and that of the past. The domain contains some linear and nonlinear characteristics [2], and thus need to build a model is required which contains linear and nonlinear characteristics.

On the other hand, there is some risk to investment in the stock market due to its unpredictable behaviors. Thus, an intelligent prediction model for financial data would be of wider interest. ANNs are relatively recent method for business forecasting. The success of ANN applications can be qualified of their features and powerful pattern recognitions capability [3]. The use of ANN in this field has been growing due to their ability to model complex nonlinear systems on sample data. Back Propagation algorithm does the task of tuning of the ANN to enable it perform as desired [4]. Here the algorithm sets the various weights and biases of the ANN to their optimal values. The aim of training is to ensure that the network gives the closest desired outputs. The radial basis function network is another type of ANN that performs unsupervised or supervised learning. These networks, which are known for their ability to model complex scenarios in a very simple architecture, are used for classification, series prediction, control applications, and so forth. The radial basis function network uses radial basis functions as activation functions, which allows it to easily solve the problem or perform a mapping of the inputs to outputs [5].

Recurrent neural networks are a special type of ANN in which the output of one or more neurons is fed back into the network as inputs, forming the input for the next iteration. These networks offer a good means of machine learning in which past outputs may potentially affect the next outputs [6].The conventional training algorithms of the neural networks are prone to get struck at local minima. This is because of the gradient approaches, or similar approaches of tuning the parameters that they employ. Genetic Algorithms [7] use a variety of individuals, each of which presents an ANN. Each of the ANN or individual represents a point in the error space with some degree of performance or fitness value. These algorithms use the relative fitness or performance to generate or move the various individuals in this error space to finally enable convergence at the optimal value.

2 Algorithms and Methods

2.1 Backpropagation Algorithm

General BPA Neural Network architecture shown in Fig. 1 includes input layer, hidden layer and output layer. Each neuron in input layers are interconnected with neurons in hidden layers with appropriate weights assigned to them. Similarly each neuron of hidden layer in interconnected with output layer neuron with weights assigned to the connection. On providing learning data to the network, the learning values are passed through input to hidden and finally to output layer where response for input data is obtained. For optimizing the error obtained, the error values are back propagated to make changes in weights of input to hidden layer and hidden to output layer. With error back propagation input response are made converged to desired response [4] [8]. A general structure of BPA neural network has been shown below.

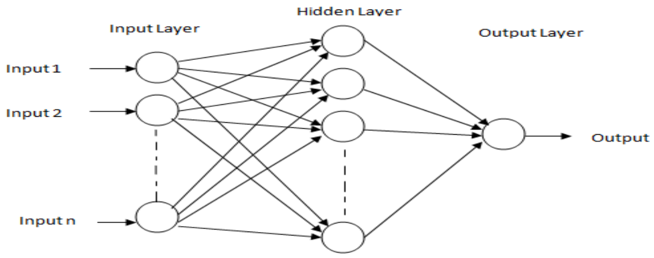

Fig. 1. General architecture of a Backpropagation Neural Network

2.2 Radial Basis Neural Network

Radial Basis Function Networks [5] has a simple 3 layer ANN architecture. The first layer is the input layer where the inputs are applied. The second layer is the hidden layer. The last layer is the output layer where the system delivers its output. These networks however differ in the manner in which they process information for the generation of the outputs from the inputs.

Consider the input space of the problem being considered. Each neuron of the hidden layer in the RBN corresponds to a location in the input space. The various neurons are spread all round the input space. The input itself is a point in this input space consisting of some value of different attributes. At the time of processing of the inputs, each of these neurons calculates its distance from the input. The outputs of the various neurons of the hidden layer serve as the inputs of the output layer. This layer is a linear layer and simply computes the weighted sum of the various hidden layer neurons. Each connection of this layer corresponds to the weight that is multiplied by the associated hidden neuron. In this way the system generates the final output.

2.3 Recurrent Neural Network

Recurrent Neural Networks architecture shown in Fig. 2 [6] are cyclic in nature unlike the other models that were arranged in a manner that cycles can never be formed. The conventional ANNs have a big limitation of their static nature. The information flow is only forward where the predeceasing layers process data and forward it to the next layers. These networks allow backward connections where every neuron gets the feedback from the forward layers as well as itself [11]. This allows the ANN to again process data and again transmit the output for further processing by the other layers in backward and forward direction to which it is connected. In this manner there is a lot of dynamism which drives these networks. Further the algorithm operates in timestamps or iterations where a unit processing is performed by each of the neurons in a single iteration. The output of the system continuously changes with time as the layers undergo changes driven by the feedback connections.

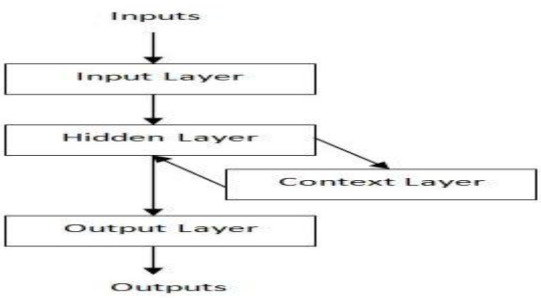

Fig. 2. General architecture of a Recurrent Neural Network

2.4 Evolutionary BPA Neural Network

A genetic algorithm performs a parallel stochastic search. It is parallel in the sense that many solutions in the population are considered simultaneously and the fittest solutions are chosen for reproduction. It is stochastic in the sense that the solutions are randomly selected for refinement and the likelihood of a solution being selected is enhanced by the quality of the solution or its fitness, and the search direction is also chosen randomly. Genetic Algorithm evolves ANNs by fixing the values and the weights and biases of the various nodes i.e. the GA optimizes the network parameters for better performance [13].

Steps followed for evolution of ANN [12][13] are problem encoding, creation of random initial state, fitness evaluation, and genetic operator including selection, crossover, mutation and elite, generate next generation, testing and verification.

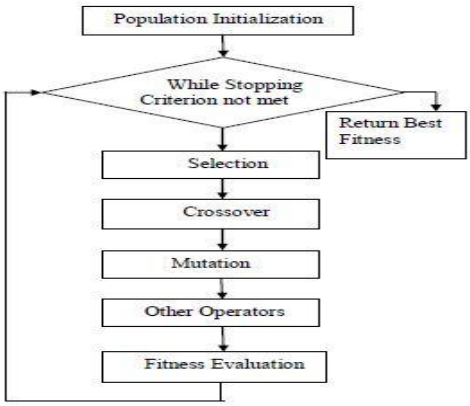

Fig. 3. Flow Diagram for working of Genetic Algorithm

2.5 Ensembles

Ensembles make use of multiple ANNs to solve the same problem. Each of the ANN is given the complete part of the problem input. Each ANN solves the problem using its own approach. All these ANNs or modules return a solution to the integrator which then computes the final output [14]. The input is given by the system to each of the modules. Each module is similar in regard to the inputs and the outputs. Each module represents an independent ANN of its own which is trained separately using the same training data. The training may hence be carried out in parallel among the various ANNs or modules. In this manner all the modules or ANN solve the same problem and make their distinct contributions. All the outputs are communicated to a central integrator that carries the next task of computing the final output of the system based on the system module outputs. Justification to ensembles revolve around the fact that they are more resistant to accidental under-training of a single ANN, or it being accidentally struck at a local minima, since the effect may be averaged by others.

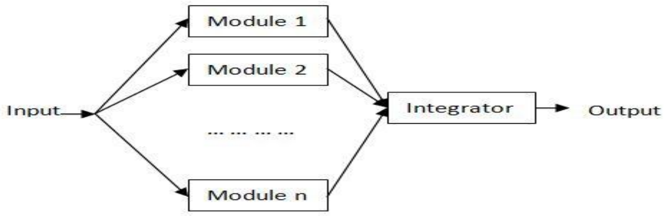

Fig. 4. General Architecture of a Neural Network Ensemble

In gating as integration technique [15], we try to combine the various outputs of the modules and generate the final output of the system by a simple function. The easiest implementation of this is to make the integrating function take a weighted or simple mean of the various outputs of the modules. This gives the final system output of the system.

3 Experiment and Results

3.1 Research Data

We have used two different data sets for our research. The data (un-normalized) have been collected from Prof. Rob J Hyndman's website http://robjhyndman.com/TSDL/ . Data sets analyzed are as: Daily closing price of IBM stock, Jan. 01 1980 - Oct. 08 1992 [10], Daily S & P 500 index of stocks, Jan. 01 1980 - Oct. 08 1992 [10]. The first few data indexes of series are used for the research. For training, 80% of the data of the series has been used and remaining 20% is used for testing.

Table 1. Time Series Data Sets Description

Time Series	Standard Deviation	Mean	Count
Daily IBM	5.736916	60.89908	500
Daily S&P	10.1308	123.3728	500

3.2 Methodology

Time series dataset is divided into training and testing dataset. Dataset division carried out is based on the random probability followed which bifurcates the dataset into training and testing dataset. Training dataset which is 80% of the original dataset is used for defining and training of the neural network. Testing dataset which is remaining 20% of the original dataset is used for performance measure. Performance measurement carried out in the experiment is root mean square error.

After dataset division, normalization followed by logarithmic scale conversion is carried out to draw better timely relation between index values. Thus processed training dataset is used for defining and supervised learning of the neural network. For whole experiment, total number of input neurons and total number of output neuron are kept 10 and 01 respectively. Neural Network training followed by testing is carried out one by one on all four modules to record the individual performance for the individuals. Corresponding graphs between actual and predicted values are plotted to analyze the performance graphically.

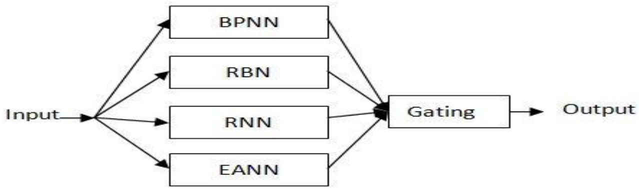

Fig. 5. Used Architecture of Neural Network Ensemble

For neural network ensemble used architecture as shown in figure 5, backpropagation, radial basis, recurrent and evolutionary neural networks are considered as individual modules of ensemble. Each modules gives output for given input to the ensemble system. Output of these modules as discussed is integrated by gating technique. In gating, weighted mean is followed for outcomes as final output result. Thus found outcome of the neural network ensemble is compared with the individual performance of standalone modules. Graph for actual and predicted values is also plotted for neural network ensemble for carrying out the graphical analysis of individual and ensemble performances.

3.3 Empirical Results

Table 2 shown below represents the mean root mean square error for both of the time series Daily IBM and Daily S&P.

Table 2. Results Obtained for Time Series

Methodology	Daily IBM (Mean RMSE)	Daily S&P (Mean RMSE)
BPA	1.9823	2.8913
RBN	1.2261	1.5217
RNN	0.9918	1.1371
EANN-BPA	1.4725	2.1237
Ensemble	0.9743	1.1261

3.4 Graphical Analysis

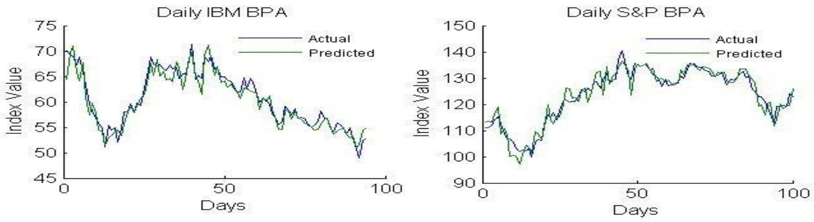

Fig. 6. Graphs for Actual and Predicted Values for Daily IBM and Daily S&P using traditional Backpropagation Algorithm with mean RMSE = 1.9823 and 2.8913 respectively.

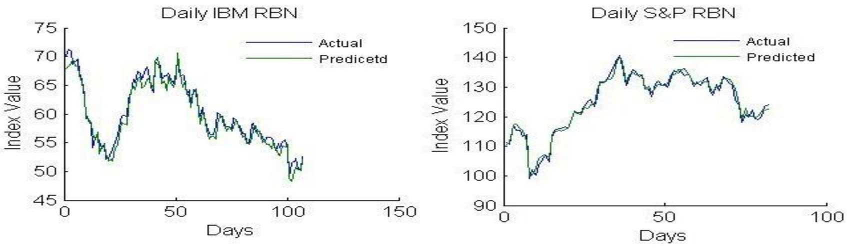

Fig. 7. Graphs for Actual and Predicted Values for Daily IBM and Daily S&P using Radial Basis Network with mean RMSE = 1.2261 and 1.5217 respectively.

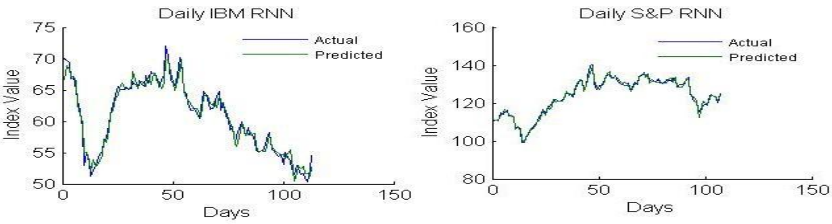

Fig. 8. Graphs for Actual and Predicted Values for Daily IBM and Daily S&P using Recurrent Neural Network with mean RMSE = 0.9918 and 1.1371 respectively.

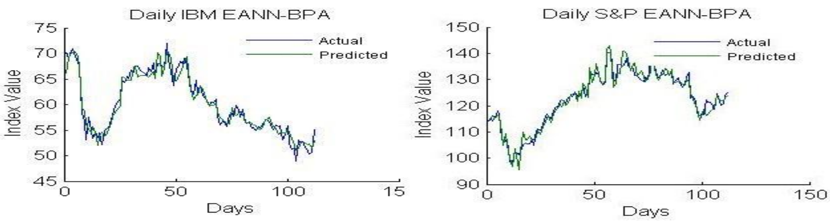

Fig. 9. Graphs for Actual and Predicted Values for Daily IBM and Daily S&P using Evolutionary BPA Neural Network with mean RMSE = 1.4725 and 2.1237 respectively.

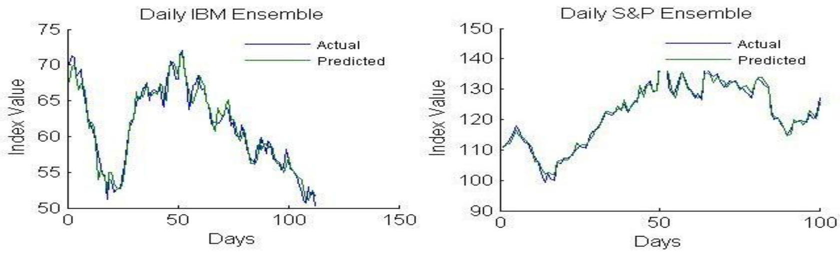

Fig. 10. Graphs for Actual and Predicted Values for Daily IBM and Daily S&P using Neural Network Ensemble with mean RMSE = 0.9743 and 1.1261 respectively.

4 Conclusions

Neural network ensemble has been used in order to improve the financial forecasting performance. It is based on the taking different learning models as individual modules, where each module accounts for the performance of the final result of the system. Output values from the individual modules are integrated using gating as integration technique in order to have the final output results. Justification to ensembles revolve around the fact that they are more resistant to accidental under-training of a single ANN, or it being accidentally struck at a local minima, since the effect may be averaged by others. As a performance measurement root mean square error is used. Neural network ensembles are computationally efficient and it shows a very good behavior to estimation values. Accuracy results and comparison graph of actual and predicted index values shows the better performance of neural network ensemble over performances of individual module used. Order of performance on the basis of results can be adjudged as Ensembles > RNN > RBN > EANN-BPA > BPA.

References

1. Donglin Wang; Yajie Li; , "A novel nonlinear RBF neural network ensemble model for financial time series forecasting," Advanced Computational Intelligence (IWACI), 2010 Third International Workshop on , vol., no., pp.86-90, 25-27 Aug. 2010
2. Hui-ren Zhou; Ying-hui Wei; , "Stocks market modeling and forecasting based on HGA and wavelet neural networks," Natural Computation (ICNC), 2010 Sixth International Conference on , vol.2, no., pp.620-625, 10-12 Aug. 2010
3. Ning Yang-cui; Zheng Xiao-xian; Zhao Jing; Jiang Gui-juan; , "Forecasting the natural forest stand age based on artificial neural network model," Computer and Communication Technologies in Agriculture Engineering (CCTAE), 2010 International Conference On , vol.3, no., pp.536-539, 12-13 June 2010

4. Fangwen Zhai; Qinghua Wen; Zehong Yang; Yixu Song; , "Hybrid forecasting model research on stock data mining," New Trends in Information Science and Service Science (NISS), 2010 4th International Conference on , vol., no., pp.630-633, 11-13 May 2010
5. Karimi, B.; Menhaj, M.B.; Saboori, I.; , "Robust Adaptive Control of Nonaffine Nonlinear Systems Using Radial Basis Function Neural Networks," IEEE Industrial Electronics, IECON 2006 - 32nd Annual Conference on , vol., no., pp.495-500, 6-10 Nov. 2006
6. Lalit Gupta, Mark McAvoy, James Phegley, Classification of temporal sequences via prediction using the simple recurrent neural network, Pattern Recognition, Volume 33, Issue 10, October 2000, Pages 1759-1770, ISSN 0031-3203
7. Elisaveta G. Shopova, Natasha G. Vaklieva-Bancheva, BASIC--A genetic algorithm for engineering problems solution, Computers & Chemical Engineering, Volume 30, Issue 8, 15 June 2006, Pages 1293-1309, ISSN 0098-1354
8. Tsung-Lin Lee, Back-propagation neural network for the prediction of the short-term storm surge in Taichung harbor, Taiwan, Engineering Applications of Artificial Intelligence, Volume 21, Issue 1, February 2008, Pages 63-72, ISSN 0952-1976
9. David Casasent, Xue-wen Chen, Radial basis function neural networks for nonlinear Fisher discrimination and Neyman-Pearson classification, Neural Networks, Volume 16, Issues 5-6, Advances in Neural Networks Research: IJCNN '03, June-July 2003, Pages 529-535, ISSN 0893-6080
10. Hipel and McLeod Time Series Modelling of Water Resources and Environmental Systems, 1994, Elsevier.
11. Mohammad Assaad, Romuald Bone, Hubert Cardot, A new boosting algorithm for improved time-series forecasting with recurrent neural networks, Information Fusion, Volume 9, Issue 1, Special Issue on Applications of Ensemble Methods, January 2008, Pages 41-55, ISSN 1566-2535
12. Yan Guo; Lishan Kang; Fujiang Liu; Huashan Sun; Linlu Mei; , "Evolutionary Neural Networks Applied to Land-cover Classification in Zhaoyuan, China," Computational Intelligence and Data Mining, 2007, CIDM 2007. IEEE Symposium on, vol., no., pp.499-503, March 1 2007-April 5 2007
13. Randall S. Sexton, Jatinder N. D. Gupta, Comparative evaluation of genetic algorithm and backpropagation for training neural networks, Information Sciences, Volume 129, Issues 1-4, November 2000, Pages 45-59, ISSN 0020-0255
14. Chih-Fong Tsai, Jhen-Wei Wu, Using neural network ensembles for bankruptcy prediction and credit scoring, Expert Systems with Applications, Volume 34, Issue 4, May 2008, Pages 2639-2649, ISSN 0957-4174
15. Nikunj C. Oza, Kagan Tumer, Classifier ensembles: Select real-world applications, Information Fusion, Volume 9, Issue 1, Special Issue on Applications of Ensemble Methods, January 2008, Pages 4-20, ISSN 1566-2535