http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 4 (3): 289-292, 2005
ISSN 1812-5638

FLANN Based Forecasting of S&P 500 Index

[1]Banshidhar Majhi, Hasan Shalabi and Mowafak Fathi
College of Computer Engineering and Information Technology,
Al Hussein Bin Talal University, Ma'an, Jordan
[1]On leave from National Institute of Technology, Rourkela, India

Abstract: Neural networks are used extensively in time series forecasting because it can learn the non-linear pattern which is often present in time series data. Functional neural network is a modified neural network and is rapidly gaining popularity in different fields. The method of presenting the input to the network makes functional neural network different from the simple neural network. This study describes how functional neural network is used to predict the values of S&P 500 index of New York stock exchange.

Key words: Forecasting, multilayer perceptron, functional link neural networks, S&P 500 index, preprocessing

INTRODUCTION

Artificial Neural Network (ANN) models have been studied for many years with the hope of achieving human-like performance in different fields ranging from engineering to sociology, finance, etc. They are endowed with some unique attributes: universal approximation (input-output) mapping, the ability to learn from and adapt to their environment and the ability to invoke weak assumptions about the underlying physical phenomenon responsible for generation of input data. This model attempts to achieve good performance via dense interconnection of simple computational elements. Neural net models have great potential in areas where there is a non-linear pattern in the data, which the network has to learn. Forecasting is such a field, where we try to establish a pattern in the past data of a variable and in most practical problems that pattern is inherently non-linear. Neural networks are used extensively in electrical load forecasting, weather forecasting, market trend prediction and forecasting stock market indices. Lowe and Webb[1] has considered this problem using a Radial Basis Function (RBF) network and applied it to some examples of time series including simple chaotic maps, non-linear differential equations and stock market prediction. Some of the other ANN structures including recurrent neural networks applied to time series[2-6].

Multiple Layer Perceptron (MLP): Typically, the MLP consists of a set of sensory units or source nodes that consists of the input layer, one or more hidden layers of computation nodes and an output layer of computation nodes. The feed forward structure of the MLP refers to a network in which all nodes of a layer are fully connected through the synaptic weights to all nodes of the layer just above it. The input signal propagates through forward direction, on a layer-by-layer basis. The learning of the network is carried out in two phases. In the forward phase, an input pattern is applied to the input layer of the network and its effect propagates through the network layer by layer. The set of outputs of output layer constitutes the actual response of the network. During the forward phase, the weights of the networks are all fixed. In the backward phase, on the other hand, the synaptic weights are adjusted in accordance with the error-correction rule, most popularly known as Back Propagation (BP) algorithm.

The Functional Link ANN (FLANN): It has been found that Multi Layer Perceptron (MLP) suffers from slow convergence rate and high computational complexity. Even though an MLP with a single hidden layer is capable of universal approximation, in many practical applications more than one hidden layer are employed for better generalization capability. Further, to overcome the local minima problem, sometimes more number of nodes are added to the hidden layer. The increase in number of layers or the number of nodes in the hidden layer gives rise to computational burden on the network. Especially, the computational requirement for propagating equivalent errors (square error derivatives) backward, i.e. toward the hidden layer is very high.

We describe an alternate ANN structure called functional link ANN (FLANN). This network provides large reduction in computational requirement and possesses high convergence speed. This single layer ANN, with capability of formation of complex decision boundaries, was originally proposed by Pao[7].

Corresponding Author: Dr. Banshidhar Majhi, Assistant Professor, College of Computer Engineering and Information Technology, Al Hussain Bin Talal University, Ma'an, Jordan

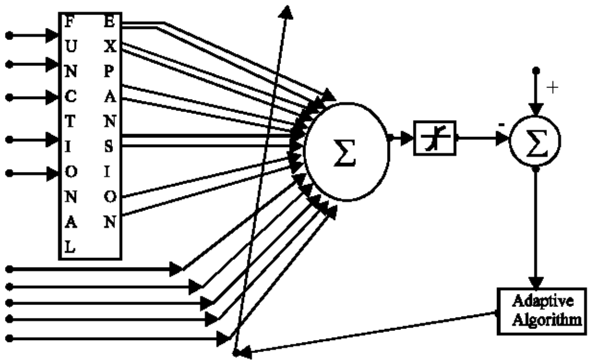

Fig. 1: Model of FLAAN structure

The processing action in MLP may be viewed in two ways. In one view, the nodes in the first layer construct hyper-planes, the nodes in the second layer constructs hyper-volumes and those in other layers specify the AND or OR decision rules[8]. From another viewpoint[9], the successive layers carryout a sequence of mappings until a final representation i.e. a mapping in a suitable space is found where the desired separation is possible.

In the FLANN, which is based on the second viewpoint, the initial representation of a pattern is described in a space of increased dimensions. The concept of functional link is described in Fig. 1. In this model, each component of input vector is subjected to a functional expansion to yield the enhanced representation of the original pattern. The functions used in this may be a subset of orthonormal basis functions spanning over an n-dimensional representation space, such as cos(PI*x), sin(PI*x), cos(2*PI*x), sin(2*PI*x), ... and so on. Besides these trigonometric functions, other orthogonal functions such as Legendre, Chebyshev can be used. But the motivation behind using trigonometric function lies in Fourier series expansion of the time series. The trigonometric bsis functions which are given by {1, cos(PI*x), sin(PI*x), cos(2*PI*x), sin(2*PI*x), ..., cos(N*PI*x), sin(N*PI*x)} provide a compact representation of the function in the mean square sense. The degree of freedom, i.e., the number of basis functions needed in the case of polynomial basis function is more than that of the trigonometric basis function within a specified accuracy. When suitable trigonometric polynomials are used, after training, the FLANN weights represent a multidimensional Fourier series decomposition of a periodic version of the desired response function. Different applications of trigonometric expansion can be found by Giles and Maxwell[10,11]. For a thorough theoretical discussion of FLANN, the reader may refer[12,13].

Forecasting the S&P 500: There has been a great amount of interest on Wall Street for neural networks. Some application of Neural Network in stocks and finance can be found by different researchers[14-17]. The S&P 500 index is a widely followed stock index, like the Dow Jones Industrial Average (DIJA). It has a broader representation of stock market since this is average based on 500 stocks. Whereas, the DIJA is based on only 30. The problem, which is approached in this paper is to predict the S&P 500 index, given data of prior weeks.

Output of the network is the value of S&P 500 index ten weeks from now. Instead of predicting the absolute values of S&P 500, we train the network to predict the change in level of the index ten weeks ahead with respect to current value.

Deciding the inputs to the network is a very complex procedure and needs expertise of a stock market analyst. Only those variables, which have a relationship with the predicted index, are to be chosen as inputs. But one inherent strength of neural network is that if a relationship is weak, the network will ignore it automatically. But proper selection of input variables reduces the size of solution space and ensure quick convergence. So the five inputs chosen for our problem are:

- Previous closing values of S&P 500 index
- Breadth indicators for stock market like ratio of number of advancing issues to declining issues
- Other technical indicators, ratio of number of new highs to new lows achieved in the week for NYSE market. This gives some indication about the strength of an uptrend or downtrend.
- Interest rates, like short-term interest rates in the three-month treasury bill yield and long-term rates in the 30 year treasury bond yield.

Raw data for the period from January 4,1980 to May 27,1983 is taken as the training period, for a total of 178 weeks of data. These 178 facts were randomly divided into training set consisting of 155 weeks data, which is used to train the network and test consisting of 23 weeks data. Test set is used to test the generalisation capability of the network after training. Because over training a network results in memorisation of the input and poor forecasting outside the training set. So the training and testing should be done in tandem until performance of the network over the test set deteriorates. In our experiment, we tested the performance of the network over the test set after every 50 iteration of training.

The next step is to highlight the pattern in the data. For each of the five inputs, we use a function to highlight rate of change (ROC) of features. We use the following function proposed by Jurik.

ROC(n) = (input(t) - BA(t-n))/(input(t) + BA(t-n))

where, input (t) is the input's current value and BA (t-n) is a five unit block average of adjacent values centred around the value n periods ago. Because we are predicting the stock index value 10 weeks ahead, ROC (10) is used. So the inputs to the network are current values, ROC (10) values for each of the five input variables making the number of inputs 10.

The input values are normalised to keep them within the range -1 to +1. Normalisation is done according to the following formula. The output of the network is the percentage of change in S&P 500 index after 10 weeks to the current value of the index.

Output = 100 X ((S & P 10 weeks ahead) - (S & P this week)) / (S & P this week)

The output value is also normalised between 0 and 1.

Multi Layer Perceptron (MLP) approach: We have used a three layer network with 10, 3 and 1 number of nodes in the input, hidden and output layer, respectively. The back propagation algorithm is used to train the network and the weights are updated on a batch basis. Hyperbolic tangent function is used as the threshold function at each node in the hidden and output layer.

Functional Link ANN (FLANN) approach: We functionally expanded the five ROC (10) values. Instead of feeding ROC (10) values directly, COS and SIN of each of the five were fed to the network. The rest five inputs i.e. the actual values of the five variables were fed as usually i.e. without any functional expansion. So the total number of inputs to the FLANN was 15. No hidden layer was used and the output layer contained only one neuron. So the total number of neurons in the network was 16 (15 in the input layer and 1 in the output layer) and the number weights was 15, where as number neurons and number weights in the MLP network was 14 and 33, respectively. The Fig. 1 shows the structure of FLANN used in our model.

Simulation results: Here, we give a comparison of performance of two approaches in terms of number of operations required In each of them, convergence characteristics and the forecast values of S&P index. No of operations required in the two approaches are shown in Table 1.

Table 1: Operation counts in MLP and FLANN

Operation	MLP	FLANN
Addition	70	45
Multiplication	143	62
Tanh( )	4	1
Sin( ) / Cos( )	-	10
Total	257	118

Table 2: Comparison of errors in FLANN and MLP

FLANN		MLP	
Training set error	Test set error	Training set error	Test set error
0.057	0.055	0.064	0.072
0.053	0.053	0.054	0.058
0.054	0.053	0.061	0.065
0.053	0.053	0.064	0.066
0.051	0.052	0.059	0.063
0.055	0.055	0.061	0.068
0.050	0.051	0.060	0.061
0.052	0.051	0.067	0.073
0.051	0.050	0.066	0.076
0.052	0.051	0.071	0.079

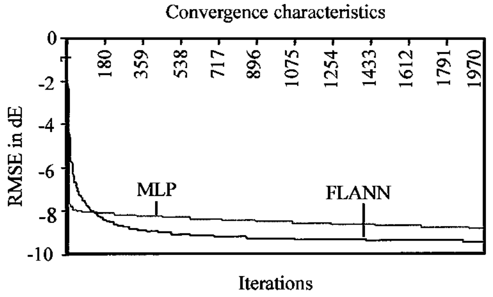

Fig. 2: Convergence charcateristics of FLAAN and MLP

From Table 1, it is clearly evident that the computational complexity is drastically reduced in FLANN as compared to the MLP structure. This is due to the fact that major computational burden on the MLP is due to error propagation for calculation of square error derivative of each node in all the hidden layers. On the other hand, since FLANN has no hidden layer the number of operations are very less. Both the programs were run 10 times and the Root Mean Square Error (RMSE) for training set and test set in both approaches are shown in the Table 2.

Table 2 shows the RMSE (Root Mean Square Error) for training and test set obtained from the different runs of the both method. One conclusion is that FLANN structure gives consistently better performance than MLP.

Figure 2 shows the convergence characteristic of the two approaches. From 10 runs of each method the best results found for each is plotted in the following graph (i.e. Run No. 7 of FLANN and Run No. 2 of MLP of the previous table). It can be seen that the convergence rate of FLANN is much better than that of MLP. In MLP the final error found after 2000 iterations was -8.88 dB, where as that in case of FLANN was -9.49 dB.

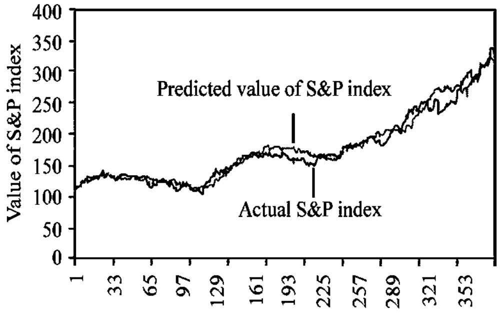

Fig. 3: Predicted Vs. actual values of S&P index by FLLAN structure

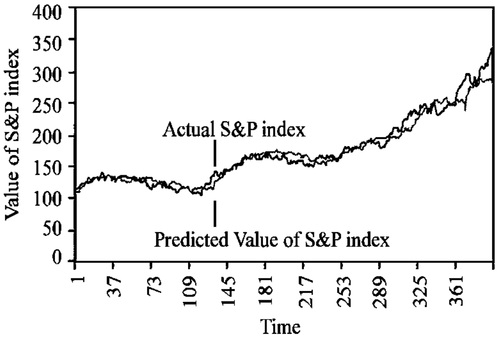

Fig. 4: Predicted Vs. actual values of S&P index by MLP Structure

Finally, Fig. 3 and 4 show the forecast S&P values by FLANN and MLP structures after training respectively, plotted against the actual values of the index during the same period. Table 2 depicts the better performance results over the training set and test set of FLANN as compared to those of MLP.

CONCLUSIONS

In this study we have discussed two different neural network architectures namely, FLANN and MLP to forecast the S&P index and a comprehensive comparative analysis has been made. The prediction results obtained from FLANN structure are better than that of MLP. In addition, the single layer structure of FLANN makes it computationally very efficient and provides faster convergence.

REFERENCES

1. Lowe, D. and A.R. Webb, 1991. Time series prediction by adaptive networks: A dynamical systems perspective. IEE Proceedings (London), Part F, 138: 17-24.
2. Anastasio, T.J., 1991. A recurrent neural network model of velocity storage in the vestibulo-ocular reflex. Advances in Neural Information Processing Systems, 3: 32-38.
3. Back, A.D. and A.C. Tsoi, 1998. A low-sensitive recurrent neural network. Neural Computations, 10: 165-188.
4. Cleeramans, A., D. Servan-Schreiber and J.L. McClelland, 1989. Finite state automata and simple recurrent networks. Neural Computation, 1: 372-381.
5. El Hihi, S. and Y. Bengio, 1996. Hierachical recurrent neural networks for long-term dependecies. Advances in Neural Information Processing Systems, 8: 493-499.
6. Giles, C.L., G.Z. Sun, H.H. Chen, H.H. Chen, Y.C. Lee and D. Chen, 1990. Higher order recurrent networks and grammatical inference. Advances in Neural Information Processing Systems, 2: 380-387.
7. Yoh-Han Pao, 1989. Adaptive Pattern Recognition and Neural Networks. Addison-Wesley Publishing Company, Inc.
8. Lippmann, R.P., 1987. An introduction to computing with neural nets. IEEE ASSP Magazine, 4: 4-22.
9. Nilsson, N.J., 1965. Learning Machines: Foundations of Trainable Pattern Classifying Systems. McGraw-Hill, New York.
10. Giles, C.L. and T. Maxwell, 1987. Learning invariance and generalisation in higher-order neural networks. Applied Optics, 26: 4972-4978.
11. Sejnowski, T.J., 1986. Highter-order Boltzman machines. American Institute of Physics Conference Proceedings, No. 151: Neural Networks for Computing, Snowbird, Utah, pp: 398-403.
12. Sobajic, D., 1988. Neyral nets for control of power systems. Ph.D Thesis, Case Western Reserve University, Cleveland, OH.
13. Patra, J.C., G. Panda, R.N. Pal and B.N. Chatterji, 1997. Neural networks for signal applicaions. J. Computer Sci. Informatics, 27: 34-58.
14. Rao, V. and Rao Hayagriva, 1996. C^{++} Neural Networks and Fuzzy Logic. BPB Publication, New Delhi, pp: 377-416.
15. Konstenius, Jeremy G., 1994. Trading the S&P with a Neural Network. Technical Analysis of Stocks and Commodities. Technical Analysis Inc., Seattle.
16. Trippi, R. and E. Turban, (Eds.), 1993. Neural Networks in Finance and Investing. Probus Publishing, Chicago.
17. Yoon, Y. and G. Swales, 1993. Predicting Stock Price Performance: A Neural Network Approach. Neural Networks in Finance and Investing, Probus Publishing, Chicago, pp: 329-339.