

A Review on Web-Based Fuzzy Neural Networks for Stock Prediction

Abhishek Shukla

Dr. C. V. Raman University
abhishekshukla2009@gmail.com

Rohit Miri

Dr. C. V. Raman University
rohitmiri@gmail.com

S.R. Tondon

Dr. C. V. Raman University
s.r.tandon@gmail.com

Abstract- A web-based stock prediction system is developed based on a fuzzy neural network by using the past stock data to discover fuzzy rules and make future predictions. The learning algorithm is implemented. Input data to each network are the moving averages of the weekly stock data. The output simulation data are also the average values of the weekly stock data. After the input data are collected from the website for the specific term using web search techniques, the system is trained and then is able to make future predictions. To implement this stock prediction System, JSP (Java Server Page), JDK1.3, JSP server and IE server 5.0 are used.

Keywords: Prediction, Time series forecasting, Fuzzy Neural Network, Backpropagation Neural Network

I. INTRODUCTION

The financial market different from a lot of physical systems like we know the weather is that the financial market is a sort of complex feedback mechanism. What people expect prices to be affects the prices they observe and then the prices they observe then affects how they are going to form their expectations about what the prices will be in the next period. The market is basically an uncertain beast or an uncertain institution, it's an institution where people trade risk, swap risk, and that's why it's there. And so if it were possible to predict it there would be no risk. In individuals, I think there cannot be any publicly available system to predict a financial market. On the other hand, neural networks have been found useful in stock price prediction [1-2]. Both feedforward and recurrent neural networks have been investigated and good results have been obtained. That means the prediction software would be very useful to assist individuals in reaching a final decision. In this paper, assuming that it is possible to predict markets, a prediction system is developed using fuzzy neural networks with a learning algorithm to predict the future stock values. The system consists of several neural networks modules. These models are all used to learn the relationships between different technical and economical indices and the decision to buy or sell stocks. The inputs to the networks are technical and economic indices. The output of the system is the decision to buy and sell. There are several neural network methods for stock prediction, such as Time Series method, Recurrent neural network and Feed-forward neural network method, etc. When compared to these techniques, Fuzzy neural network is a very useful and effective method to process, which is explained in the later sections.

The learning algorithm is used to train the networks. Before learning starts, tolerances are defined for the output units. During learning, the weights are updated only when the output errors exceed the tolerances. The learning data for which the output errors do not exceed the tolerances are eliminated from the training data sets. The input data to each network are the moving averages of the weekly averaged data which are obtained directly by using a Java program from the website. The output simulation data is also the average values of the weekly stock data.

II. MODEL ANALYSIS

Time series forecasting analyzes past data and projects estimates of future data values. Basically, this method attempts to model a nonlinear function by a recurrence relation derived from past values. The recurrence relation can then be used to predict new values in the time series, which hopefully will be good approximations of the actual values. There are two basic types of time series forecasting: univariate and multivariate. Univariate models, like Box-Jenkins, contain only one variable in the recurrence equation. The equations used in the model contain past values of moving averages and prices. Box-Jenkins is good for short-term forecasting but requires a lot of data, and it is a complicated process to determine the appropriate model equations and parameters. Multivariate models are univariate models expanded to "discover casual factors that affect the behavior of the data." [3-4]. As the name suggests, these models contain more than one variable in their equations. Regression analysis is a multivariate model, which has been frequently compared with neural networks. Overall, time series forecasting provides reasonable accuracy over short periods of time, but the accuracy of time series forecasting diminishes sharply as the length of prediction increases. Many other computer-based techniques have been employed to forecast the stock market. They range from charting programs to sophisticated expert systems. Fuzzy logic has also been used. Expert systems process knowledge sequentially and formulate it into rules. They can be used to formulate trading rules based on technical indicators. In this capacity, expert systems can be used in conjunction with neural networks to predict the market. In such a combined system, the neural network can perform its prediction, while the expert system could validate the prediction based on its well-known trading rules. The advantage of expert systems is that they can explain how they

derive their results. With neural networks, it is difficult to analyze the importance of input data and how the network derived its results. However, neural networks are faster because they execute in parallel and are more fault tolerant.

The major problem with applying expert systems to the stock market is the difficulty in formulating knowledge of the markets because we ourselves do not completely understand them. Neural fuzzy networks have an advantage over expert systems because they can extract rules without having them explicitly formalized. In a highly chaotic and only partially understood environment, such as the stock market, this is an important factor. It is hard to extract information from experts and formalize it in a way usable by expert systems. Expert systems are only good within their domain of knowledge and do not work well when there is missing or incomplete information. Neural networks handle dynamic data better and can generalize and make "educated guesses." Thus, neural networks are more suited to the stock market environment than expert systems. In the wide variety of different models presented so far, each model has its own benefits and shortcomings. The best way is that these methods work best when employed together. The major benefit of using a fuzzy neural network then is for the network to learn how to use these methods in combination effectively, and hopefully learn how the market behaves as a factor of our collective consciousness.

III. FUZZY NEURAL NETWORKS

A. Architecture

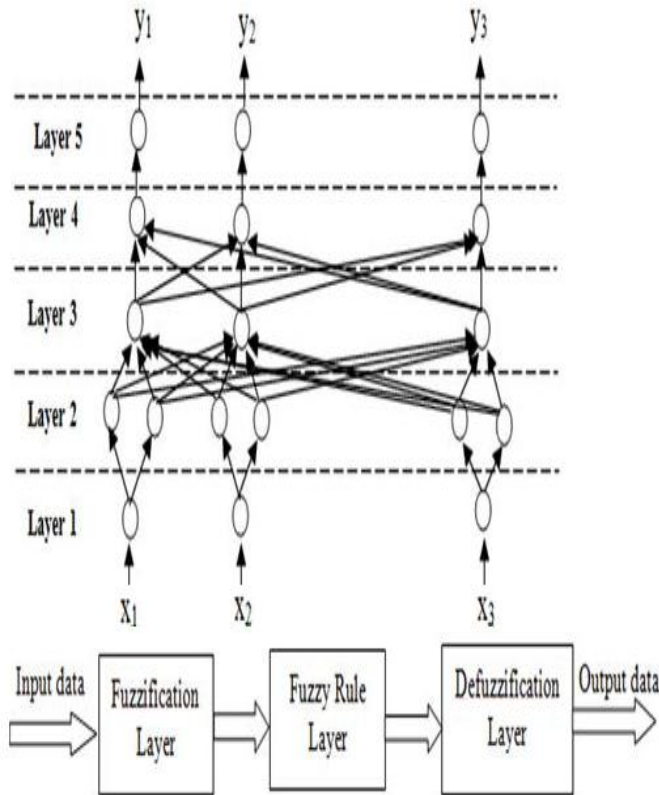


Fig. 1 : Architecture of the fuzzy neural network

According to the mechanism of fuzzy logic control system, the fuzzy neural network usually has 5 functional layers: (1) Layer 1 is the input layer. (2) Layer 2 is the fuzzification layer; (3) Layer 3 is the fuzzy reasoning layer which may consist of AND layer and OR layer; (4) Layer 4 is the defuzzification layer; (5) Layer 5 is the output layer. The architecture of a fuzzy neural network is described in Fig.1. Usually, the fuzzy neural network maps crisp inputs x_i ($i=1,2,\dots,n$) to crisp output y_j ($j=1,2,\dots,m$). A fuzzy neural network is constructed layer by layer according to linguistic variables, fuzzy IF-THEN rules, the fuzzy reasoning method and the defuzzification scheme of a fuzzy reasoning method and the defuzzification scheme of a fuzzy logic control system.

Each neuron in the fuzzification layer represents an input membership function of the antecedent of a fuzzy rule. One common method to implement this layer is to express membership functions as discrete points. Thus for a fuzzy rule "IF X_1 is A_1 and X_2 is A_2 ... THEN Y is B ", A 's characterize the possibility distribution of the antecedent clause " X is A ". Each of the hidden nodes is defined as a fuzzy reference point in the input space. The function of the defuzzification layer is for rule evaluation. Each neuron in this layer represents a consequent proposition "THEN Y is B " and its membership function can be implemented by combining one or two sigmoid functions and linear functions.

B. Learning Algorithm

An n -input- l -output fuzzy neural network has m fuzzy IF-THEN rules which are described by

$$\text{IF } x_1 \text{ is } A_1^k \text{ and } \dots \text{ and } x_n \text{ is } A_n^k \text{ THEN } y \text{ is } B^k,$$

where x_i and y are input and output fuzzy linguistic variables, respectively. Fuzzy linguistic values A_i^k and B^k are defined by fuzzy membership functions as follows,

$$\mu_{A_i^k}(x_i) = \exp\left[-\left(\frac{x_i - a_i^k}{\sigma_i^k}\right)^2\right] \quad (1)$$

$$\mu_{B^k}(y) = \exp\left[-\left(\frac{y - b^k}{\eta^k}\right)^2\right] \quad (2)$$

the n -input- l -output fuzzy neural network with simple fuzzy reasoning is defined below:

$$f(x_1, \dots, x_n) = \frac{\sum_{k=1}^m b^k [\prod_{i=1}^n \mu_{A_i^k}(x_i)]}{\sum_{k=1}^m [\prod_{i=1}^n \mu_{A_i^k}(x_i)]} \quad (3)$$

Given n -dimensional input data vectors x^p (i.e., $x^p = (x_1^p, x_2^p, \dots, x_n^p)$) and one-dimensional output data vector y^p for $p=1,2,\dots,N$, (i.e., N training data sets). The energy function for p is defined by

$$E^p = \frac{1}{2} [f(x_1^p, \dots, x_n^p) - y^p]^2 \quad (4)$$

For simplicity, let E and f^p denote E^p and $f(x_1^p, \dots, x_n^p)$, respectively.

After training the centers of output membership functions $(\frac{\partial E^p}{\partial b^k})$, the widths of output membership functions $(\frac{\partial E^p}{\partial \sigma^k})$, the centers of input membership functions $(\frac{\partial E^p}{\partial a^k})$ and the centers of input membership functions $(\frac{\partial E^p}{\partial \eta^k})$, then we obtain the training algorithm [5 -7]:

$$b^k(t+1) = b^k(t) - \theta \left. \frac{\partial E^p}{\partial b^k} \right|_t \quad (5)$$

$$\sigma^k(t+1) = \sigma^k(t) - \theta \left. \frac{\partial E^p}{\partial \sigma^k} \right|_t \quad (6)$$

$$a^k(t+1) = a^k(t) - \theta \left. \frac{\partial E^p}{\partial a^k} \right|_t \quad (7)$$

$$\eta^k(t+1) = \eta^k(t) - \theta \left. \frac{\partial E^p}{\partial \eta^k} \right|_t \quad (8)$$

Where, η is the learning rate and $t=0,1,2,\dots$

The main steps using the learning algorithm as follows:

- Step 1. Present an input data sample, compute the corresponding output;
- Step 2. Compute the error between the output(s) and the actual target(s);
- Step 3. The connection weights and membership functions are adjusted;
- Step 4. At a fixed number of epochs, delete useless rule and membership function nodes, and add in new ones;
- Step 5. IF Error > Tolerance THEN go to Step 1 ELSE stop.

When the error level drops to below the user-specified tolerance, the final interconnection weights reflect the changes in the initial fuzzy rules and membership functions. If the resulting weight of a rule is close to zero, the rule can be safely removed from the rule base, since it is insignificant compared to others. Also, the shape and position of the membership functions in the Fuzzification and Defuzzification Layers can be fine tuned by adjusting the parameters of the neurons in these layers, during the training process.

IV. SYSTEM IMPLEMENTATION

A. Input Data & Software Parameters

The system can predict future for any stock or market index. For example to predict Microsoft stock values for some

days ahead, historical data is required. Data for neural networks is probably the most important aspect for training. Without user intervention a particular stock data is obtained from the Internet directly as user's requirement, in this process, user can determine which stock and how long of price information he wants to get. Preprocessing operation is needed as preparatory step for next stage. As an example of preprocessing, the downloadable data is daily price, it is need to be grouped by week, and the average of each week are calculated for the next step. A HTML parser that coded in java is used to retrieve data. This program parses the whole file line by line to get the needed information and insert them into the database. All of training data are arranged for 2-input-1-output system as this format: (d1, d2 --- d3), (d2, d3 ---d4), where, for the first vector, d1 and d2 are inputs and d3 is an output, and in the second vector, d2 and d3 are inputs and d4 is an output.

There are some parameters, which we have to use during the training period. They are error, threshold, tolerance, etc. The accurate results depend on these parameters. To achieve our target rapidly, some parameters are preprocessed. For example, in the fuzzy member function 1 and 2, there are two important parameters: center and width of the fuzzy set. Their initial values have critical impact on the performance of system. The random value generated by machine is not optimal. In this system, we propose a simple way to optimize the initial value: sort all of data obtained from internet, and then divide data into five groups, get the average value of each group, these values are used in member function as the initial value of fuzzy set center.

The prediction algorithm takes all these parameters as input. This algorithm is called when we click predict future from the system after entering the stock symbol. This algorithm makes the neural network learn. The algorithm returns the future values as output witch are eventually stored in results table for each stock. We will keep track of these results until we click on the average error for all simulations for this particular stock. Java program displays all the results of predictions for any particular stock at a time on the web page and clears the contents of the table.

B. Overview of Implementation

A full run of the program implementation will be described, going through all the main features of the program:

- Download historical data from the Internet.
- A program is written through which user can get the data as his/her requirement. Arrange all of data into the format that used for the next step, and insert data into database.
- Algorithm, which trains the neural networks using the mean square error as stop criterion for learning, while never exceeding the maximum number of cycles which can take testing data from the initial date to the user entered date, and predicts the future stock closing values.
- A program is written, which compares the predicted values with real values.

One of the most important factors to construct a neural network depends on what the network will learn. A neural

network must be trained on some input data. The two major problems in implementing the training are: defining the set of input to be used (the learning environment) and deciding on an algorithm.

V. SIMULATION

Using the developed system to predict the future stock values with Fuzzy neural Networks we can do some analysis to know the performance of the Back-Propagation Algorithm.

By using the past historical data, if we predict stock values for future 5 weeks from Back-Propagation algorithm we are now able to compare the predicted values with the real values. Table 1 and Figure 2 show the prediction and real values of the weekly average of the stock of I2 Technologies, Inc. The input past historical data is from 2001.1 to 2001.5. Table 2 and Figure 3 show the prediction and real values of the weekly average of the stock of Cypress Semiconductor Corporation. The input past historical data is from 2000.6 to 2001.1. The simulation results are very close to the real values.

When we changed the number of the input past historical data, the more input data we have the better training and get more close results. This means that, more the available data for predicting financial markets, the greater the chances of an accurate forecast. For the same test case if we decrease the maximum training error parameter from 0.0001 to 0.000015, we are getting more close results. The resulting future stock values are closer than the future stock values with high training error.

Simulations are done on several other stocks, such as Microsoft, Oracle and IBM, etc.. From the simulation results it is conclusive that, the average error for simulations using lot of data is small compared to the average error using less data and the more data for training the neural network, the better prediction it gives.

Table 1. The weekly average values of the stock of I2 Technologies, Inc(from 2001.5)

	1 st week	2 nd week	3 rd week	4 th week	5 th week
Web-based	20.763	19.851	19.386	19.2	19.116
Real valued	21.874	20.77	17.568	17.24	17.712

Table 2. The weekly average values of the stock of Cypress Semiconductor Corporation(from 2001.1)

	1 st week	2 nd week	3 rd week	4 th week	5 th week
Web-based	25.014	24.227	23.658	23.326	23.139
Real valued	25.008	23.338	23.872	20.575	19.358

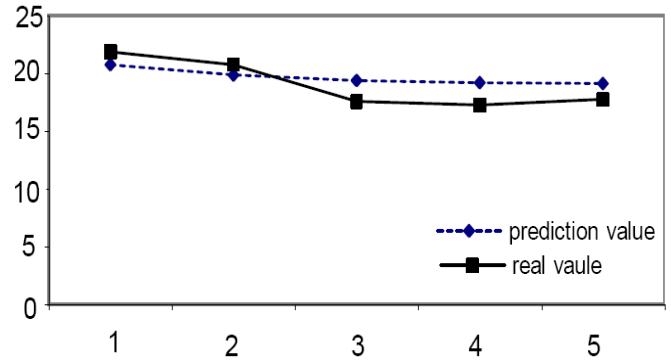


Fig. 2 : The simulation values and real values of the stock

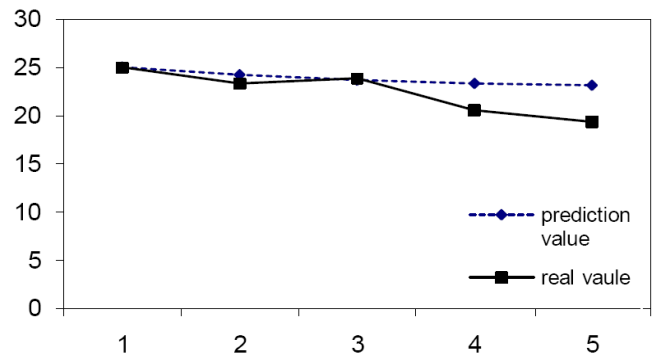


Fig.3 : the simulation values and real values of the stock of Cypress Semiconductor Corporation (from 2001.1)

VI. CONCLUSION

After completing several simulations for predicting several stocks based on the past historical data using fuzzy neural network with the Back-Propagation learning algorithm, it is conclusive that the average error for simulations using lots of data is smaller than that using less amount of data. That is, the more data for training the neural network, the better prediction it gives. If the training error is low, predicted stock values are close to the real stock values.

One possibility for future work is to update the system, so that it can read the past stock data automatically from the web. In this way the system will become Internet ready for predicting any stock market and is ready at any time. Another possibility for future work is to update the system, which can allow to trade the stock, which means users can manage to buy and sell the stock after seeing the prediction values from this system. In this way the system can eventually become Internet ready to be used anywhere in the world at any time. It is also possible to make the fuzzy neural Web-based stock prediction system as a commercial application by updating such that it gives more user-friendly functionality and by giving more valuable information to the users on palm machine.

REFERENCES

- [1] Lee,C.H and Park,K.C., “*Prediction of monthly transition of the composition stock price index using recurrent back-propagation*”, Int. Conf. On Artificial Neural Networks, Brighton,UK, pp.1629-1632, 1992.
- [2] D.T.Pham and X.Liu, “*Neural Networks for Identification, Prediction and Control*”, Springer, 1999.
- [3] J. Ellman, “*Finding structure in time*”, Cognitive Science, pages 179-211, 1990.
- [4] J. Robert and Van Eyden, “*The Application of Neural Networks in the Forecasting of Share Prices*,” Finance and Technology Publishing, 1996.
- [5] Witold Pedrycz, Abraham Kandel and Yan-Qing Zhang, “Neurofuzzy Systems,” In Fuzzy Systems: Modeling and Control, pages 312–363, Kluwer Academic Publishers, 1999.
- [6] Y.-Q. Zhang, M. D. Fraser, R. A. Gagliano and A. Kandel, “*Granular Neural Networks for Numerical-Linguistic Data Fusion and Knowledge Discovery*,” Special Issue on Neural Networks for Data Mining and Knowledge Discovery, *IEEE Transactions on Neural Networks*, Vol. 11, No. 3, pp.658-667, May, 2000.
- [7] Y.-Q. Zhang and A. Kandel, “Compensatory Genetic Fuzzy Neural Networks and Their Applications,” Series in Machine Perception Artificial Intelligence, Volume 30, World Scientific, 1998.