Journal of Computer Science and Cybernetics, V.27, N.1 (2011), 51–60

AN EXPDRIMENT RESULT BASED ON ADAPTIVE NEURO-FUZZY INFERENCE SYSTEM FOR STOCK PRICE PREDICT ON

BUI CONG CUONG[1], PHAM VAN CHIEN[2]

[1]*Institute of Mathematics*

[2]*Hanoi University of Science and Technology*

Tóm tắt. Trong những năm cuối thị trường tài chính thế giới đã thay đổi nhờ sự phát triển nhiều hệ thống tiên tiến. Nhằm khai thác dữ liệu thời gian thực đã phát triển những lĩnh vực mới như các hệ mờ nơron dành cho bài toán dự báo và như vậy làm sống lại quan tâm tới dự báo các chỉ số tài chính và chứng khoán. Bài báo này giới thiệu một thử nghiệm dùng hệ suy diễn mờ - nơron với một quy trình tính toán mới để dự báo giá chứng khoán.

Abstract. In the last years, the financial markets around the world have been modified by the rapid development of advance systems. The acquisition of high-frequency data in real times has developed new fields like neuro-fuzzy systems for forcasting problems, renewing also the interest in the forcasting of financial and stock market indexes. In this paper, we present an experiment result based on Adaptive Neuro-Fuzzy Inference System with a new computing procedure for stock price prediction.

1. INTRODUCTION

Artifical neural networks (ANN) have been successfully applied to a number of scientific and engineering fields in recent years, e.g. function approximation, system identification and control, image processing, time series prediction and so on [1-3, 6].

Time-series forcasting is an impotant research and application area. Much effort has been devoted over the past several decades to develop and improve the time-series forecasting models. Well established time series models include : (1) linear models, e.g., moving average, exponential smoothing and the autoregressive intergrated moving average (ARIMA); (2) nonlinear models, e.g., neural network models and fuzzy system models [4-8].

Neuro-fuzzy systems methods and statistical tools are different methods that can be used to predict financial indexes. Neural networks incorporate a large number of parameters which allows to learn the intrinsic non-linear relationship presented in time-series, enhancing their forcasting possibilities. ANN have been successfully applied to predict important financial and market indexes, like for example, Standart and Pool 500 (SP&500). Nikei 225 Index, the New York stock exchange composite index (NYSE index) and other.

Stock price prediction has always been a subject of investors and professional analysts. Nevertheless, finding out the best time to buy or to sell has remained a very difficult task because there are too many factors that influence stock. During the last decade, stocks and future traders have come to rely upon various types of intelligent systems. Lately, ANN and adaptive neuro-fuzzy inference system (ANFIS) have been applied to this area.

Other soft computing methods are also applied in the prediction of stock and these soft computing approaches are to use quantitative inputs, like technical indexes, qualitative factors, political effects, automate stock market forcasting and trend analysis.

In this paper, we will use an ANFIS with a new computing procedure for stock index forcasting. The remainder of the paper is organized as follows: Section 2 describes the architecture of the ANFIS, Section 3 presents some learning algorithms and Section 4 is devoted to an experiment result for VN Index stock index prediction. Finally, conclusions are drawn in Section 5.

2. ADAPTIVE NEURO-FUZZY INFERENCE SYSTEM

Adaptive neuro-fuzzy inference system (see [3, 6 - 8]) is the most popular neuro-fuzzy connectionist system that similar to a Sugeno type fuzzy inference systems (FIS). FIS can be efficiently used a bridge between the domain expert and a financial system. FIS works on knowledge bases that are in easily comprehensible óÀÌIF ... THENóÀÌ format. Neuro-fuzzy algorithms are assimilarly of neural networks and FIS. These algorithms are essentially adaptive, lucid and highly flexible. As they are essentiall fuzzy inference systems embedded into a neural network, they are also robust.

ANFIS architecture

Figure 1 shows a sample ANFIS structure using three inputs and two labels for each input. Generally, an ANFIS structure with n inputs and m labels for each input has 5 layers. The node functions in each layer are of the same function family as described on figure 1.

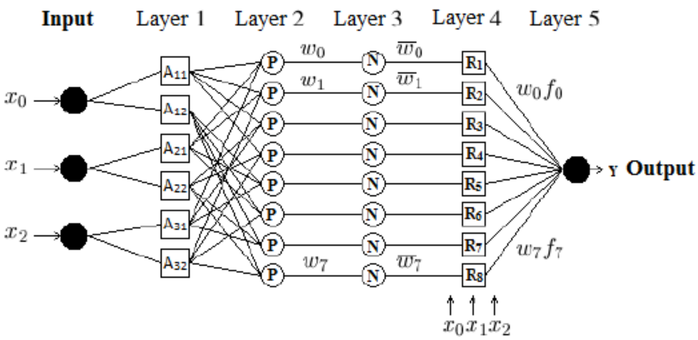

Figure 2.1. Sample ANFIS structure.

Layer 1: The first layer contains $n.m$ adaptive nodes (square nodes) with a node function:

$$O^1_{i,j} = \mu_{A_{i,j}}(X_i), \tag{2.1}$$

where X_i $(0 \le i \le n-1)$ is the i^{th} input, $A_{i,j}$ $(0 \le i \le n-1, 0 \le j \le m-1)$ is the j^{th} linguistic label of the i^{th} input, such as small, normal, large, etc. is the membership function of $A_{i,j}$ and it specifies the degree to which the give X_i satisfies the quantifier $A_{i,j}$. Usually we choose to be Generalized Bell or Gaussian membership function with minimum equal to 0 and maximum equal to 1:

$$\mu_{gbell}(x) = \frac{1}{1 + \left|\dfrac{x - c_k}{a_k}\right|^{2b_k}}, \qquad \mu_{gaussian}(x) = \exp\left[-\left(\frac{x - c_k}{s_k}\right)^2\right]. \tag{2.2}$$

Therefore, (c_k, a_k, bk) or (c_k, s_k) $(0 \leq k \leq n.m - 1)$ is non-linear parameter set of k^{th} node. When the values of these parameter change, the shape of membership function on linguistic label $A_{i,j}$ vary accordingly. In terms of calculation, we consider that $i * m + j = k$.

Layer 2: The second layer contains m^n fixed nodes (circle nodes) label P. The $k^{th}(0 \leq k \leq m^n - 1)$ node collects the incoming signals to do the T-norm and sends the result out:

$$O_k^2 = w_k = \prod_{i=0}^{n-1} \mu_{A_{i,j}}(X_i), \tag{2.3}$$

where w_k represents the firing strength rule. The computation of w_k requires clearly defined n nodes in the first layer that connected to it.

Layer 3: This layer contains m^n fixed nodes label N. The k^{th} node $(0 \leq k \leq m^n - 1)$ calculates the ratio of the k^{th} ruleóÀỄs firing strength to the sum of all rules firing strengths that called the weight or normalized firing strength of the k^{th} rule:

$$O_k^3 = \overline{w}_k = \frac{w_k}{\sum\limits_{i=0}^{m^n-1} w_i}. \tag{2.4}$$

Layer 4: The fourth layer contains m^n adaptive nodes in which the node fuction of the k^{th} node is:

$$O_k^4 = \overline{w}_k f_k = \overline{w}_k \left(\sum_{i=0}^{n-1} p_i^k x_i, + r_k\right) \tag{2.5}$$

where $(p_0^k, p_1^k, ..., p_{n-1}^k, r_k)$ is parameter set of the k^{th} node. Parameters in this layer are linear parameter and will referred to as consequent parameters of Takagi-Sugeno type fuzzy inference system.

Layer 5: There is only one node in last layer. It is a fixed node that computes the overall output as the summation of all incoming signals:

$$O_1^5 = output = y = \sum_{k=0}^{m^n-1} \overline{w}_k f_k = \frac{\sum\limits_{k=0}^{m^n-1} w_k f_k}{\sum\limits_{k=0}^{m^n-1} w_k}. \tag{2.6}$$

3. SOME ANFIS LEARNING ALGORITHMS

We consider a ANFIS with n inputs and m labels for each input. Assume that the node function of the first layer is Gaussian membership function.

3.1. Back propagation learning algorithm

Back propagation algorithm (BP) was first introduced in the 1970s by Werbos [1]. The parameters set are updated through training data by gradient descent method (see [3,6]). We can see that most of the existing neural-network-based fuzzy systems are trained by the BP algorithm. It is well known that the algorithm is generally slow and likely to become trapped in local minimum. Hence, a fast learning algorithm for real-time applications is highly desirable.

3.2. Hybrid learning algorithm

The gradient algorithm is generally slow and likely to become trapped in local minima. Here a hybrid learning rule is proposed, which combines the gradient algorithm and the least squares estimate (LSE) to update parameters. From (2.6) we have:

$$output = F\left(\overrightarrow{I}, S\right),$$

where F is networkóÀȄs function, $\overrightarrow{I}$ is input vector and S is networkóÀȄs parameters set. We have:

$$y = \sum_{k=0}^{m^n-1} \overline{w}_k f_k = \sum_{k=0}^{m^n-1} \left[\overline{w}_k \left(\sum_{i=0}^{n-1} p_i^k x_i + r_k\right)\right]. \tag{3.1}$$

Therefore, y is a linear function of the parameters $\left(p_i^k, r^k\right)$.

We denote: S_1 is the parameters set in the first layer, S_2 is the parameters set in the 4^{th} layer.

- $S_1 = [c_{i,j}, s_{i,j}]$, where $i (0 \le i \le n-1)$ (inputóÀȄs index) and j $(0 \le j \le m-1)$ (index of corresponding linguistic label).
- $S_2 = \left[p_0^k, p_1^k, ..., p_{n-1}^k, r^k\right]$ where $(0 \le k \le m^n - 1)$.

Therefore, S can be decomposed into two sets:

$$S = S_1 \cup S_2. \tag{3.2}$$

Sine y is linear in S_2, for each given values of elements of S_1, we can use N training data into (3.1) to obtain a matrix equation:

$$AX = B, \tag{3.3}$$

where X is unknown vector whose elements are parameters in S_2. Assume that $|S_2| = M$ and the dimensions of A, X, B are $N \times M, M \times 1, N \times 1$. Because N (number of training data) is usually greater than M (number of parameter in the 4^{th} layer), this is an over determined problem generally there is no exact solution to equation (3.3). Instead, least squares estimate (LSE) of X, X^*, is sought to minimum least squared error $||AX - B||^2$, where $|| \bullet ||$ is Euclide norm. The most well-known formula for X^* uses the pseudo-inverse of X:

$$X^* = \left(A^T A\right)^{-1} A^T B, \tag{3.4}$$

where A^T is the transpose of A and $(A^T A)^{-1} A^T$ is the pseudo-inverse of A if $A^T A$ is non-singular. While equation (3.4) is concise in notation, but its time consuming when dealing with the matrix inverse and moreover, it becomes ill-defined if $A^T A$ is singular. Therefore, sequential formulas are used to compute X^*. This sequential method of LSE is more efficient especially when M is small. Let the i^{th} row vector of matrix A in equation (3.3) be a_i^T and the i^{th} elements of B be b_j^T, then X can be adjusted using the following sequential formulas:

$$X_{i+1} = Xi + S_{i+1} a_{i+1} \left(b_{i+1}^T - a_{i+1}^T X_i\right) \quad Si + 1 = Si - \frac{S_i a_{i+1} a_{i+1}^T S_i}{1 + a_{i+1}^T S_i a_{i+1}} \quad (i = \overline{0, N-1}), \tag{3.5}$$

where S_i is the covariance matrix and the least squares estimate X^* is equal to X_N. The initial conditions to sequential formulas (3.5) are $X_0 = 0$ and $S_0 = \xi I$, where ξ is a positive large number and I is the identity matrix of dimension $M \times M$.

Now the gradient algorithm and the least squares estimate can be combined to update the parameters in an ANFIS. Each epoch of this hybrid learning procedure is composed of a forward pass and a backward pass. In the forward pass, S_1 is fixed, we use input date to compute each nodeóÀỄs output until the matrices A and B in equation (3.3) are obtained and the parameters set S_2 are identified by LSE method. After that, the function signals keep going forward until the output error is computed. In the backward pass, S_2 is fixed, we use gradient descent method to update S_1.

4. AN ADAPTIVE NEURO-FUZZY INFERENCE SYSTEM FOR STOCK INDEX PREDICTION

4.1. Input parameters selection and data preprocessing

According to financial research, we find that there are factors affecting the objectivity of the stock market of Vietnam in general and the VN Index in particular. So, we decided to add three new parameter along with three conventional parameters to the forecast. The new system has six inputs and one output. The proposed ANFIS is a new model belonging to the new class of knowledge-based ANFIS models. The data preprocessing was treated as presented in [4].

4.2. Computer simulation program

We have built a computer simulation program to test the proposed model. The Interface of program is illustrated in figure 2. Our computer program includes the following modules:

- *Module 1:* Automatically update data from the Internet: Update the new transaction data including gold price (from www.sjc.vn), USD Exchange rates (from www.vietcombank.com.vn /exchangerates) A92 petrol retail price and VN Index price (from www.cophien68.com).

- *Module 2:* Data pre-processing and training data creation: Get data from module 1 then make data classification and data pre-processing, in oder to creat training data.
- *Module 3:* Training and prediction: Do the network training using back propagation learning rule and hybrid learning rule, defuzzification and forecasting.

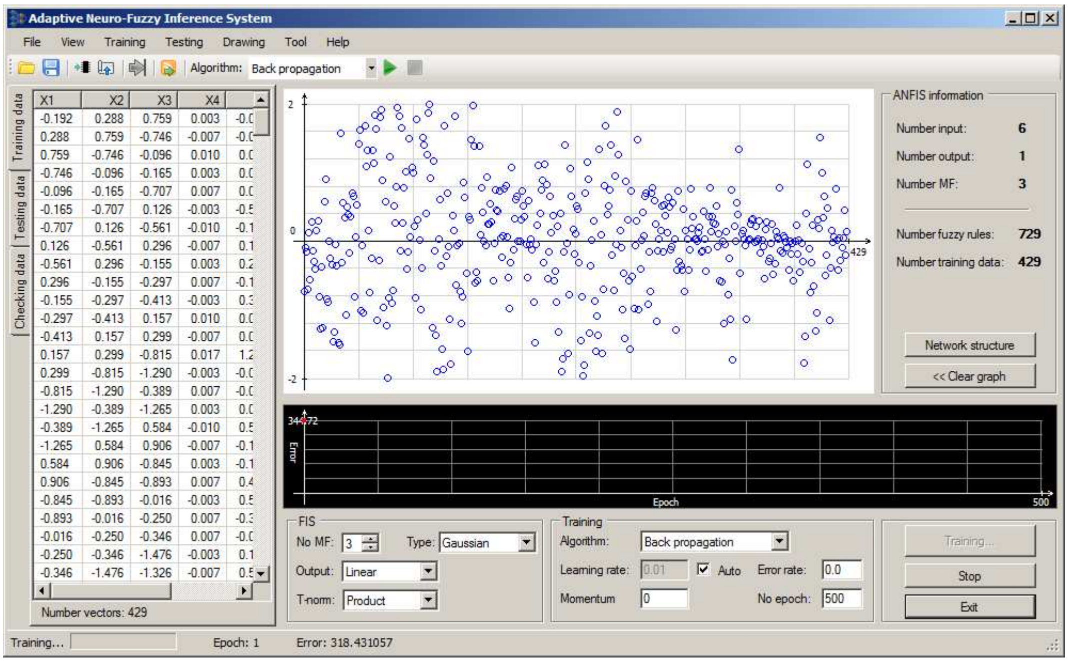

Figure 4.2. Computer simulation program

4.3. Implement

Based on built computer program and obtained data, we have the experiments consist of the following steps:

Step 1: Data for training . The interval to train the ANFIS (called traning-set) was taken as the daily closing values of VN Index from Juanary 2009 to September 2010. The results were tesed by predicting values of the same index since October to December 6-2010 (test-set). At all, 433 points were used during the training process and 46 to test the trained ANN. Taking the data preprocessing base on [4], we obtain following result (table 1).

Step 2: Creat training set. Training vector is 7-dimensions vector:

$$M_i = \left(Z^i_{t-2}, Z^i_{t-1}, Z^i_t, U^i_t, G^i_t, P^i_t, Z^i_{t+1}\right), \quad (i = 1, ..., 430)$$

where $Z^i_{t-2}, Z^i_{t-1}, Z^i_t, Z^i_{t+1}$ are stock values in three days $t-2, t-1, t, t+1$; U^i_t, G^i_t, P^i_t areUSD/VND exchange rates, gold value and A92 petrol retail price on day t.
Therefore, we obtain 430 vectors for training and 46 vectors for testing.

Step 3: Build an ANFIS. We choose an ANFIS with three labels for each input and membership function is Gaussian membership function (see on Figure 3). Therefore our ANFIS structure has 6 input parameters and $3^6 = 729$ fuzzy rules.

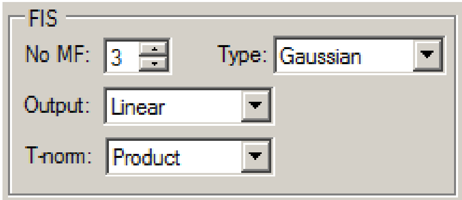

Figure 4.3. Fuzzy inference system

Step 4: Train the ANFIS using BP and hybrid learning algorithms in 500 epoches and learning rate was adjusted automatically (Figure 4).

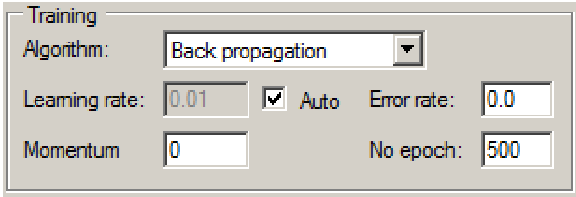

Figure 4.4. ANFIS training

Step 5: Test the performance of the model using 46 testing vectors. The result was illustrated on Figure 5.

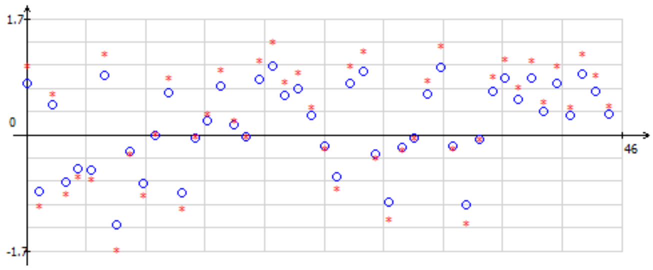

Figure 4.5. Test the performance of the model(circle node is actual ouput, * is output by model).

Table 1. Adjusted coefficient

Input type	Adjusted coefficient
VN Index price	100
Gold price	75
USD exchange rates	140
A92 petrol retail price	85

4.4. Comparison

By using BP and hybrid learning algorithms, we obtain the result as described on Table 2.

Table 2. Algorithms performance comparison

Algorithm	Learning time	Error rate
Back propagation learning rule	60 min 32 sec	57.28726
Hybrid learning learning rule	55 min 21 sec	42.42525

Table 3. Models performance comparison (for the first 22 vectors)

Real values	New model's ouput	Old model's ouput
448.9	447.2	446.1
457.3	456.6	452.8
457.1	456.0	454.9
449.4	448.1	450.3
451.3	450.2	448.0
446.7	445.8	442.4
441.6	441.6	440.6
433.5	435.1	437.5
426.9	427.5	430.1
425.5	426.0	427.8
430.7	431.2	426.4
426.5	425.1	424.8
430.8	432.9	429.1
434.5	433.1	428.9
439.9	438.6	432.0
439.9	439.2	436.0
446.4	443.9	440.7
451.6	448.3	449.5
449.9	448.0	451.6
457.4	454.8	454.2
464.4	462.7	459.1
465.6	465.2	464.9
RMSE	1.549046628	3.738679884

We also have tested with old traditional model (5 conventional input parameters) and obtain the result as illustrated on Table 3. In order to evaluate the forcasting accuracy, we obtain the root mean squared error (RMSE) as:

$$RMSE = \sqrt{\frac{1}{n}\sum_{k=1}^{n}(y_k - p_k)^2},$$

where y_k represents the forcast of the real value p_k, and n is the number of predicted events.

According to the result we can find out that the proposed model makes a higher performance than the traditional model.

5. CONCLUSION

This work performs a predictive study of the principal index of the Vietnam stock price market through new ANFIS model. According to the result we can find out that the proposed model makes a higher performance than the traditional model.

Applying ANFIS to predict stock index it was encountering following issues:

Training neural network issues: Although the hybrid training algorithm proved more effective than the BP algorithm, however, when the network structure is complicated and the training set is large, the network training time increases, the rate of convergence is very slow.

Data problems: Principles of ANFIS and fuzzy inference in general is to create systems of law from the available data, thus requiring very high accuracy of training data. Meanwhile, the collection being full of data accuracy is not an easy work.

Futher work should be done with new variables and some new computing procedures.

REFERENCES

[1] S. Haykin, *Neural Networks: A Comprehensive Foundation*, Second Edition, Prentice Hall, NewJersay, 1999.

[2] B.C. Cuong and N.D. Phuoc (Eds.), *Fuzzy Systems, Neural Networks And Applications*, Second Ed., Science and Technology Pub., Hanoi, 2006.

[3] B.C.Cuong, "Artifical neural networks, Lecture at the Center for Talent Engineers", Hanoi University of Technology, 2007.

[4] B.C. Cuong and T.D. Hoan, A neural fuzzy system and a soft computing procedure for predicting exchange rate, *The Proceedings of the* 20^{th} *Scientific Conference, Section: Applied Mathemtics and Informatics*, Hanoi University of Technology, 2006 (pp.9-13).

[5] B.C. Cuong, L.Q. Phuc and N.T.A. Binh, A Combination of context-fuzzy clustering method an learning with forgetting algorithm in a neural network model to generating fuzzy rules, *Jour. of Computer Science and Cybernetics* **24** (4) (2008) 295-306.

[6] C.T. Lin and C.S.G. Lee, *Neural Fuzzy Systems*, Prentice Hall, London, 1996.

[7] J.S.R.Jang,C.I. Sun and E.Mizutani, *Neuro-fuzzy and soft computing : a computational approach to learning and machine intelligence*, Prentice-Hall, NJ , 1997.

[8] L. Rutkowski, *Flexible neuro-fuzzy systems structure, learning and performance*, Kluwer Academic Pub., 2004.

[9] R.Zemouri, D.Racoceanu and N. Zerhouni, Recurrent radial basis function network for time-series prediction, *Engineeering Applications and Artifical Intelligence* **16** (2003) 453-463.

[10] E.H. Ruspini, P.P. Bosnissone, and W. Pedrycz (Eds.), *Handbook of fuzzy computation*, Institute of Physics Pub. , Bristone, 1998.

[11] S.H. Chun and S.H. Kim, Data mining for financial prediction and trading: application to single and multiple markets, *Expert Systems with Applications* **26** (2004) 131-139.

[12] W. Leigh, R.Hightower, and N.Modani, Forcasting the New York stock exchange composite index with past price and interest rate on condition of volume spike, *Expert Systems with Applications* **28** (2005) 1-8.

[13] P.C. Chang and C. H. Liu, A TSK type Fuzzy Rule Based System for stock price prediction, *Expert Systems with Applications* **34** (2008) 135-144.

Received on January 11 - 2011