CHEMICAL ENGINEERING TRANSACTIONS

VOL. 46, 2015

A publication of AIDIC

The Italian Association of Chemical Engineering
Online at www.aidic.it/cet

Guest Editors: Peiyu Ren, Yancang Li, Huiping Song

ISBN 978-88-95608-37-2; **ISSN** 2283-9216

DOI: 10.3303/CET1546095

Forecast of Opening Stock Price Based on Elman Neural Network

Jun Zheng

Baotou Teachers' College Network Information Center, Inner Mongolia, Baotou, 014030.
zhengjun_025@163.com

Elman Neural Network is a typical regression nerve network which is highly applicable in processing sequential data. In order to predicate accuratelythe fluctuation of stock prices, Elman Neural Network is applied in the forecast of the opening price of SSE (Shanghai Stock Exchange). This paper establishes mathematical models for the input layer, the hidden layer, connection layer and the output layer based on Elman Neural Network. The opening prices of SSE of 337 trading days from December, 2012 to April, 2014 are selected as the raw data for stimulated forecast, and its result proves the validity of this forecast model.

1. Research questions

Stocks have existed for over 200 years since Britain firstly issued shares in 1773, and have had an inestimable influence on the development of market economy. China's security market came into being in the early 1990s; although it has only developed for a little more than 20 years, great achievements have been made (ChongwenZhong (2014)) reported: there are over 2000 listed companies in Shanghai Stock Exchange and Shenzhen Stock Exchange and more than 50 million shareholders. China's stock market has taken preliminary shape and is gradually becoming an essential part of the security business or even the whole financial sector; in the meantime, stock investing has become an important part of the daily life of a growing number of people. However, the stock market constantly changes, sometimes going up and sometimes going down, which makes stock investing highly risky. With an increase in stock investing activities, an effective stock forecasting method is in urgent need, with which the trend of stock market can be predicted so that investors can avoid risks and enjoy higher rate of return. The stock value is significantly influenced by policy changes, interest rate changes, economic cycles and human manipulation. The entire stock market can be seen as a complicated non-linear system. Since Elman Neural Network has a strong non-linear approximation ability with features like self-learning and self-adaptation (S.C. Kremer (1995), Chun-Fei Hsu (2014)) reported, it will be very effective in predicting the opening price of the stock market.

2. Overview of Elman Neural Network

Elman Neural Network is a typical locally recurrent globally feed-forward network (D. P. Mandic and J. A. Chambers (2001)) reported. It was put forward in 1990 by J. L. Elmanfor speech processing, and later, on the basis of it, Pham et al introduced a modified Elman network, which is considered by most scholars as the standard Elman network. The feedback connections between and within layers in the Elman network enable it to show the time delay between the input and output, so it has a local memory function. The Elman network is widely employed in many fields such as sequence analysis, system identification and control (Jujie Wang et (2014), Penghua Li et (2014) Anqi Cui, Hua Xu and Peifa Jia (2011) reported, AksoySaadettin and Muhurcu Aydin (2011), Ciarlini P and Maniscalco U (2008)); its mathematical model is normally written as:

$$y(k) = g\left(w_3\left(w_1 x_c(k) + w_2 u(k-1)\right)\right),\quad x_c(k) = \alpha . x_c(k-1) + x(k-1) \tag{1}$$

Please cite this article as: Zheng J., 2015, Forecast of opening stock price based on elman neural network, Chemical Engineering Transactions, 46, 565-570 DOI:10.3303/CET1546095

3. Establishing stock forecasting model based on Elman Neural Network

3.1 Overall structure of the model

The structure of the proposed Elman Neural Network model is demonstrated as follows:

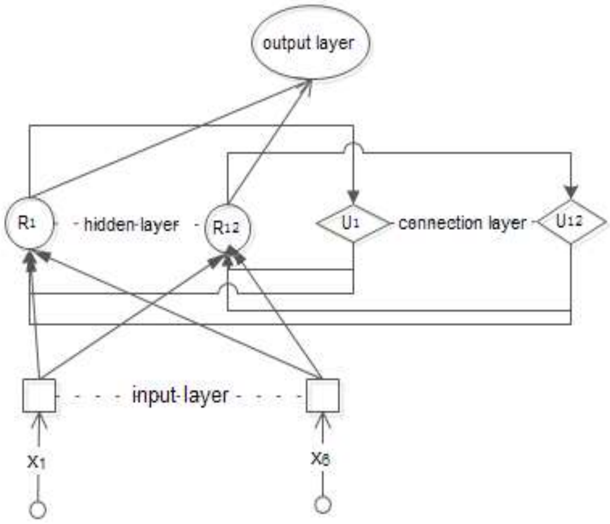

Figure 1: Overall structure of Elman Neural Network

3.2 The input layermodel

In this study, the closing prices of the first six trading days are utilized to predict the closing price of the seventh trading day. In accordance with the actual demand mentioned above, the number of nodes in the input layer is set as 6. Therefore, the mathematical model of input in the input layer is defined as:

$$\mathrm{Input}_{ik}^{(1)}(t) = [R_k(t),\quad U_k(t)]^T \tag{2}$$

In Equation 2, i stands for the number of nodes in the input layer, the value interval of which is 1≤i≤6 in this model, i∈ N; k refers to the number of input samples; t refers to the number of iterations; $R_k(t)$ stands for the input of sample space; $U_k(t)$ stands for the input in the connection layer. The mathematical model of output in the input layer is defined as:

$$\mathrm{Output}_{ik}^{(1)}(t) = \mathrm{Input}_{ik}^{(1)}(t) \tag{3}$$

3.3 The hidden layer model

The hidden layer is a critical model that links the input layer and the output layer. The hidden layer determines the performance of the network model. The number of nodes in the hidden layer has a great impact on the model: excessive nodes will lead to not only excessive training time of the network, but also over-training of it, which causes bigger error and poorer generalization ability; on the other hand, inadequate number of nodes will lead to insufficient training of the network, causing poorer fault tolerance. Therefore, the key issue in designing the hidden layer model is to properly determine the number of nodes on the layer. According to the empirical formula $M = \sqrt{m+n} + a$ (where m and n refer to the number of neurons in the output layer and the input layer respectively, and 1≤a≤10, a∈ C), and the experimental study, it can be seen that when the number of nodes in the hidden layer is 12, the network has the best performance, so the number of nodes in the hidden layer is determined as 12. Thus, the input model of the hidden layer is defined as:

$$\mathrm{Input}_{ik}^{(2)}(t) = \sum_{i=1}^{6} \sum_{l=1}^{p} \omega_{il}^{1}(t) R_{ik}(t) + \sum_{l=1}^{p} \omega_{lk}^{2}(t) U_{lk}(t) + \theta_l(t) \tag{4}$$

In Equation 4, p refers to the number of nodes in the hidden layer, which is set as 12 in this case; i stands for the number of nodes in the input layer; k refers to the input sample; t stands for the number of iterations; $\omega_{il}^{1}(t)$ refers to the connection weight between the input layer and the hidden layer, while$\omega_{lk}^{2}(t)$ refers to the connection weight between the connection layer and the input layer; $U_{lk}(t)$ stands for the output of the connection layer; $\theta_l(t)$ refers to the threshold value of the hidden layer. The output model of the hidden layer is defined as follows:

$$\mathrm{Output}_{lk}^{(2)}(t) = F(\mathrm{Input}_{lk}^{(2)}(t)) \tag{5}$$

p refers to the number of nodes in the hidden layer, and p=12; F refers to the activation function, which is defined as follows:

$$F(x) = \frac{2}{1+e^{-2x}} - 1 \quad (6)$$

3.4 The connection layer model

The connection layer enables the forecast model to have dynamic feedback and storage memory. The input model of the connection layer is defined as:

$$\mathrm{Input}_{lk}^{(c)}(t) = \mathrm{Output}_{lk}^{(2)}(t-1), \text{ when t=1 }, \mathrm{Input}_{lk}^{(c)}(1) = 0 \quad (7)$$

The output model of the connection layer is defined as:

$$U_{jk}(t) = \mathrm{Input}_{lk}^{(c)}(t) \quad (8)$$

3.5 The output layer model

In this study, the opening prices of the first six trading days are utilized to predict the opening price of the seventh trading day, so there is only one output in the output layer, and the number of nodes in the output layer is 1. The mathematical model of input in the output layer is defined as:

$$\mathrm{Input}_{ak}^{(3)}(t) = \sum_{l=1}^{p} \omega_{la}^{3}(t)\mathrm{Output}_{lk}^{(2)}(t) + \theta_a(t) \quad (9)$$

In this equation, a refers to the number of nodes in the output layer, and since the number of nodes in the output layer is 1, so a=1. k refers to the input sample; t stands for the number of iterations.
The output model of the output layer is defined as:

$$\mathrm{Output}_{ak}^{(3)}(t) = g(\mathrm{Input}_{ak}^{(3)}(t)) \quad (10)$$

In Equation 10, g is a linear function, which is defined as $g(x) = x$.

3.6 Network learning model

After the mathematical model of every layer in the Elman network has been determined, the stock price forecasting model still requires the weight coefficient and the threshold between layers, both of which would be determined through learning. In this study, the gradient descent algorithm is utilized as the learning algorithm (Wei Wu,Dong-po Xu and Zheng-xue Li (2008)) reported. The principle of gradient descent method is that for function $F(x)$, if there is definition at $x_0$and it is differentiable, then $F(x)$ decreases the fastest in the negative direction -$\nabla F(x_0)$ of gradient. In the training of the network, data transmit backward from the input layer, via the hidden layer and the connection layer, and are finally output via the output layer. Then, according to the gradient descent algorithm, weight coefficients and thresholds of the network are corrected layer by layer starting from the output layer in the direction of reducing errors (Qing Song (2010)) reported, and the following learning models can be derived from the gradient descent algorithm:
The adjustment model of the connection weight $\omega_{il}^{1}(t)$ between the input layer and the hidden layer:

$$\omega_{il}^{1}(t+1) = \omega_{il}^{1}(t) + \Delta\omega_{il}^{1}(t+1) \quad (11)$$

The adjustment model of the hidden layer threshold:

$$\theta_l(t+1) = \theta_l(t) + \Delta\theta_l(t+1) \quad (12)$$

The adjustment model of the connection weight between the hidden layer and the output layer:

$$\omega_{la}^{3}(t+1) = \omega_{la}^{3}(t) + \Delta\omega_{la}^{3}(t+1) \quad (13)$$

The adjustment model of the hidden layer threshold:

$$\theta_a(t+1) = \theta_a(t) + \Delta\theta_a(t+1) \quad (14)$$

The adjustment model of the connection layer weight:

$$\omega_{lk}^{2}(t+1) = \omega_{lk}^{2}(t) + \Delta\omega_{lk}^{2}(t+1) \quad (15)$$

4. Training and Testing of the Forecast Model

4.1 Data source

In this study, the opening prices of Shanghai Stock Exchange of 337 trading days from December, 2012 to April, 2014 are used as the source data. The data of the first 300 trading days are utilized as training samples for the Elman network model, to tune network model parameters; the data of the latter 37 trading days are utilized as testing samples, to test the network model that has been established.

4.2 Normalization of data

Since the ranges of the data collected are not unitary but variable, on the one hand, the establishment of the network model and the prediction cannot be processed in a unified framework; on the other hand, while the data input into the network model are restricted in certain intervals, the data collected do not necessarily meet the requirement, so in order to unify the range of input data, the source data need normalization.
The normalization method utilized in this study is to divide a vector by its length, so the result will stay between -1 and 1. The normalization model is described as follows:

$$p' = 2 \times \frac{p - \min(p)}{\max(p) - \min(p)} - 1 \quad (16)$$

4.3 Training of the forecast model

The training of the forecast model refers to a procedure in which the model connection weight is adjusted iteratively according to training samples (Alex Aussem(1999)) reported. The purpose of the training is to find out the essential connection between the opening prices of the stock market (Pham D T and Liu X (1996)) reported.
In this study, the opening prices of 337 trading days x_1, $x_2 \ldots x_{300}$ are rearranged to form the following matrix:

$$\begin{bmatrix} x_1 & x_2 & \cdots & x_{294} \\ x_2 & x_3 & \cdots & x_{295} \\ \vdots & \vdots & \ldots & \vdots \\ x_7 & x_8 & \cdots & x_{300} \end{bmatrix}$$

In this matrix, the vector of each column is an input sample. The first six elements in the column are input variables in the input layer and the last element (the seventh element) needs to be predicted.

4.4 Testing of the forecast model

After the training of the network model, the network needs to be tested with testing samples. The opening prices ofShanghai Stock Exchange of the latter 37 out of the 337 trading days are utilized to test the forecast model. Before the testing, the testing samples need to be normalized according to Equation 16, and the output of the forecast model also needs to be reversely normalized in order to obtain normal data. The reversed normalization model is described as follows:

$$p' = 0.5(p + 1) \times (\max(p) - \min(p)) + \min(p) \quad (17)$$

5. Analysis and evaluation of experiment results

After training the Elman Neural Network established with the data of 300 trading days, and the training results and the actual results are shown in the following Fig 2:
It can be seen from Fig 2 that the training results and the actual results match very well.
In order to better evaluate the performance of the network, on the condition that parameters of the trained network model remain unchanged, the testing samples are utilized to test the established model. The test results are shown in Table 1 and Fig 3:

Table 1: Test results

Period	forecasted value	actual value	absolute error	relative error
307	2015.313425	2026.224	-10.9106	0.005384684
308	2030.474114	2019.98	10.49411	0.005195157
309	2024.298481	2017.22	7.078481	0.003509028
310	2020.816727	1987.679	33.13773	0.016671569
311	1995.641654	2050.827	-55.1853	0.026908826
312	2057.628295	2063.323	-5.6947	0.002759968
313	2067.806992	2070.574	-2.76701	0.001336348
314	2066.338839	2060.812	5.526839	0.002681874
315	2060.675963	2046.851	13.82496	0.00675426
316	2048.897426	2043.045	5.852426	0.00286456
317	2045.907503	2031.005	14.9025	0.007337502
318	2033.873784	2049.423	-15.5492	0.007587119
319	2052.027645	2063.497	-11.4694	0.005558213
320	2066.324399	2037.552	28.7724	0.014121062
321	2037.91241	2054.53	-16.6176	0.008088268
322	2056.821305	2100.651	-43.8297	0.020864815
323	2105.30551	2105.876	-0.57049	0.000270904
324	2104.624669	2130.368	-25.7433	0.012083983
325	2129.713469	2127.409	2.304469	0.001083228
326	2126.652801	2125.902	0.750801	0.000353168
327	2124.915249	2097.213	27.70225	0.013209077
328	2096.415877	2108.947	-12.5311	0.005941886
329	2108.457653	2091.48	16.97765	0.008117531
330	2089.618038	2085.98	3.638038	0.001744043
331	2083.092073	2062.787	20.30507	0.009843514
332	2062.427368	2068.084	-5.65663	0.002735204
333	2067.599232	2064.156	3.443232	0.001668106
334	2064.649926	2060.538	4.111926	0.001995559
335	2060.125047	2033.337	26.78805	0.013174426
336	2034.900326	2001.896	33.00433	0.016486534
337	2005.643197	2020.438	-14.7948	0.007322572

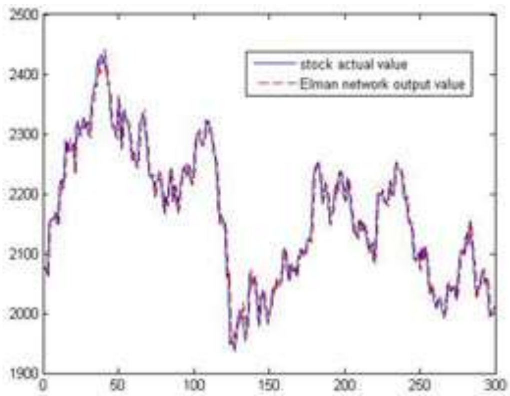

Figure 2: Test results of training data

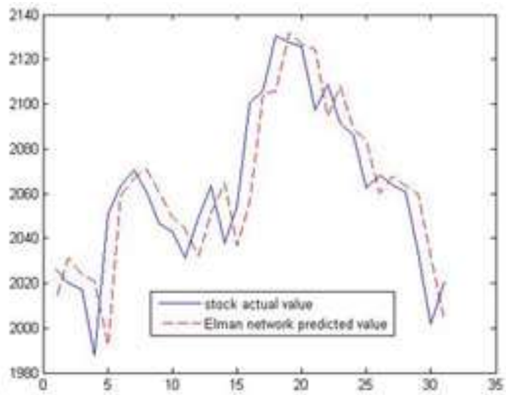

Figure 3: Test results

It can be seen from Table 1 and Fig 3 that the predicted values and the actual values are very close; the biggest relative error is only 0.026908826, and the tendency of the predicted values is consistent with that of the actual values. In summary, according to the experiment, the network model established on the basis of Elman Neural Network has good performance in predicting the opening price index of Shanghai Stock Exchange.

References

Aussem A., 1999, Dynamical recurrent neural networks towards prediction and modeling of dynamical systems, Neurocomputing, 28, 207-232, doi: 10.1016/S0925-2312(98)00125-8

Chandra R., 2014, Memetic cooperative coevolution of Elman recurrent neural networks, Soft Computing, 18(8), 1549-1559, DOI: 10.1007/s00500-013-1160-1

Ciarlini P., Maniscalco U., 2008, Wavelets and Elman Neural Networks for monitoring environmental variables, Journal of Computational and Applied Mathematics, 221(2), 302-309, DOI: 10.1016/j.cam.2007.10.040

Cui A.Q., Xu H., Jia P.F., 2011, An Elman neural network-based model for predicting anti-germ performances and ingredient levels with limited experimental data, Expert Systems with Applications, 38(7), 8186-8192, DOI: 10.1016/j.eswa.2010.12.164

Elman J.L., 1990, Finding structure in time, Cognitive Science, 14, 179-211, DOI: 10.1207/s15516709cog1402_1

Hsu C.F., 2014, Adaptive backstepping Elman-based neural control for unknown nonlinear systems, Neurocomputing, 136, 170-179, DOI: 10.1016/j.neucom.2014.01.015

Kremer S.C., 1995, On the computational power of Elman-style recurrent networks, Neural Networks, 6(4), 1000-1004, DOI: 10.1109/72.392262

Li P.H., Li Y.G., Xiong Q.Y., Chai Y., Zhang Y., 2014, Application of a hybrid quantized Elman neural network in short-term load forecasting, International Journal of Electrical Power & Energy Systems, 55, 749-759, DOI: 10.1016/j.ijepes.2013.10.020

Mandic D.P., and Chambers J.A., 2001, Recurrent Neural Networks for Prediction: Learning Algorithms, Architecture and Stability, 2001: Wiley, DOI: 10.1002/047084535X

Pham D.T., Liu X., 1996, Training of Elman networks and dynamic system modeling, International Journal of Systems Science, 27(2), 221–226, DOI: 10.1080/00207729608929207

Saadettin A., Aydin M., 2011, Elman neural network-based nonlinear state estimation for induction motors, Turkish Journal of Electrical Engineering and Computer Sciences, 19(6), 861-875, DOI: 10.3906/elk-1002-406

Song Q., 2010, On the Weight Convergence of Elman Networks, Neural Networks on IEEE Transactions, 21(3), 463-480, DOI: 10.1109/TNN.2009.2039226

Wang J.J., Zhang W.Y., Li Y.N., Wang J.Z., Dang Z.L., 2014, Forecasting wind speed using empirical mode decomposition and Elman neural network, Applied Soft Computing, 23, 452-459, DOI: 10.1016/j.asoc.2014.06.027

Wu W., Xu D.P., Li Z.X., 2008, Convergence of gradient method for Elman networks, Applied Mathematics and Mechanics, 29(9), 1231-1238, DOI: 10.1007/s10483-008-0912-z

Zhong C.W., 2014, Empirical research on the impact of China's stock market on the real economy, Exploration of economic problems, 00(04), 34-39, doi: 10.3969/j.issn.1006-2912.2014.04.006