FIT5186 Assignment Report

Combining Neural Networks and Financial Indices to Predict the Stock Trend in China

Yifei Hu, 2546****; Bimin Yin, 2546****

Southeast University & Monash University Joint Graduate School

22 May 2014

Content

Combine Neural Networks and Financial Indices to Predict the Stock Trend in China

Abstract

Since Shanghai Stock Exchange and Shenzhen Stock Exchange established in 1990 and 1991, people have shown great interest in investing into Chinese stock. Also, many researches have been conducted in order to develop a method which can predict the trend of the stock market [1]. In this paper, we try to build up several Neural Networks (NN) models incorporating some domain knowledge that admitted by many economists to predict the stock trend in Shanghai Stock Exchange. First we apply a single model to predict the whole stock market, then we clustered the stock into several categories and build up a model for each category, finally we try to apply a model to one specific stock in order to get a better accuracy. Multi-layer back-propagation neural networks, genetic adaptive probabilistic neural networks, general regression networks and unsupervised SOFM are trained with various parameters.

Key Words: Neural Networks, Financial Indices, Stock, Prediction

1. Introduction

The forecasting of the stock price is one of the most challenging tasks due to its uncertainty in the market, and so many potential factors which have hidden inner connections with each other will influence the price. What NN is good at happens to be digging out the inner relations among factors in a model, so in this stock forecasting situation, NN is a very powerful tool. A large number of studies have shown that time-series prediction methods based on multilayered Feed-forward Neural Networks has a relatively high capability to solve this kind of problems [2].

At this moment, most researches that have been done are more concentrated on the method, they want to use various prediction models and technologies to get a more precise result. The Research conduceted by Hadavandi, Shavandi and Ghanbari took fuzzy system and artificial neural networks as their prediction methods [3]; Another research used dynamic versions of a single-factor CAPM-based model and Fama and French's three-factor model [1]; Rough set theory which can overcome some inherent drawbacks in neural networks may be another useful tool to implement stock price prediction [4]. In this article, we try to examine the effect if we take some financial indices into consideration, and make them part of the parameters, except for purely change the model and techniques.

We want to build up various predictive models incorporating some specific financial indexes based on NN technology and use them to predict the trend of the stock market and the exact price level of the one certain stock share, and compare the accuracy of these two types of predictions. There are three different categories that we classify a certain stock share when predicting the trend of the stock: a. the price will increase more than 5% in three days; b. the price will decrease more than 5% in three days; c. the price will hold steady, neither increase more than 5% nor decrease 5% in three days.

2. Data Sets

2.1 Where is the data from

During the experiments, we fetch the data from a Chinese stock analysis software package called TongHuaShun, the data comes from Shanghai Stock Exchange in real time so the validity of the data is guaranteed. The samples we use to train and test the model consist of two parts, one includes opening price, closing price, lowest price, highest price, turnover, and total exchange of all the stock shares whose representative symbols ends up with a number 6 in Shanghai Stock Exchange in March 2014 (dataset1), the other part is the same features of one specific stock share whose symbol is 600757 from 11th May 2009 to 14th May 2014 (dataset2).

2.2 How do we preprocess the data

Let's define today as day0, yesterday as day-1, the day before yesterday as day-2, tomorrow as day1 and so on. According to the economists' experience, not only the opening and closing price but also the moving average and the exchange volume are considered very influential factors to the stock prices.

1. Add extra parameters. Just the price of day0 is not enough to make a good prediction, so we add a lot of features to the data, then different combination of the inputs will be experimented during the following training. The inputs added into the data set are: MA5 (moving average of the closing price of the last 5 days including day0), MA10, MA20, MA30, MA60, Close20 (closing price 20 days ago), Close10, Close5, Close3, Close2, Close1, Volume20 (total exchange volume 20 ago), Volume10, Volume5, Volume3, Volume2, Volume1. We pick more samples closer to day1, as you can see the data density is very high, for example, Close3, Close2, Close1, Close0, there is even no gap among them. The reason why we do this is because the closer to day1 the data is the more effect it will produce to the price of day1 which is what we want to forecast. Another issue we want mention is in dataset1, since the data is collected in one month, and the number of data entry of each stock share is only about 20, so we use MA2, MA3, MA5, MA7 instead of MA5, MA10, MA20, MA30, MA60, the same as closing prices and exchange volumes.

2. Remove unnecessary parameters and noisy data entries. After adding parameters to the original data set, the symbol, name and date is useless for the model training, delete them will reduce the size of the data and increase the efficiency. Also, those data entries with one or more empty column need to be removed or they will bring negative impact to the training result.

2.3 How to use the data

The data after preprocessing is still not usable for NeuroShell2. What we need to do is save the data as a well-formatted text, using comma or tab to separate the different columns of data for NeuroShell2 to recognize.

3. Training Issues

3.1 Architectures

NeuroShell2 provides us with multiple architectures of Neural Networks, see Figure 1. There are totally 5 different models which are Backpropagation(BP), Unsupervised (Kohonen), Probabilistic Neural Network (PNN), General Regression Neural Network (GRNN) and GMDH Network (Group Method of Data Handling or Polynomial Nets), and we use the first 4 models except the last one.

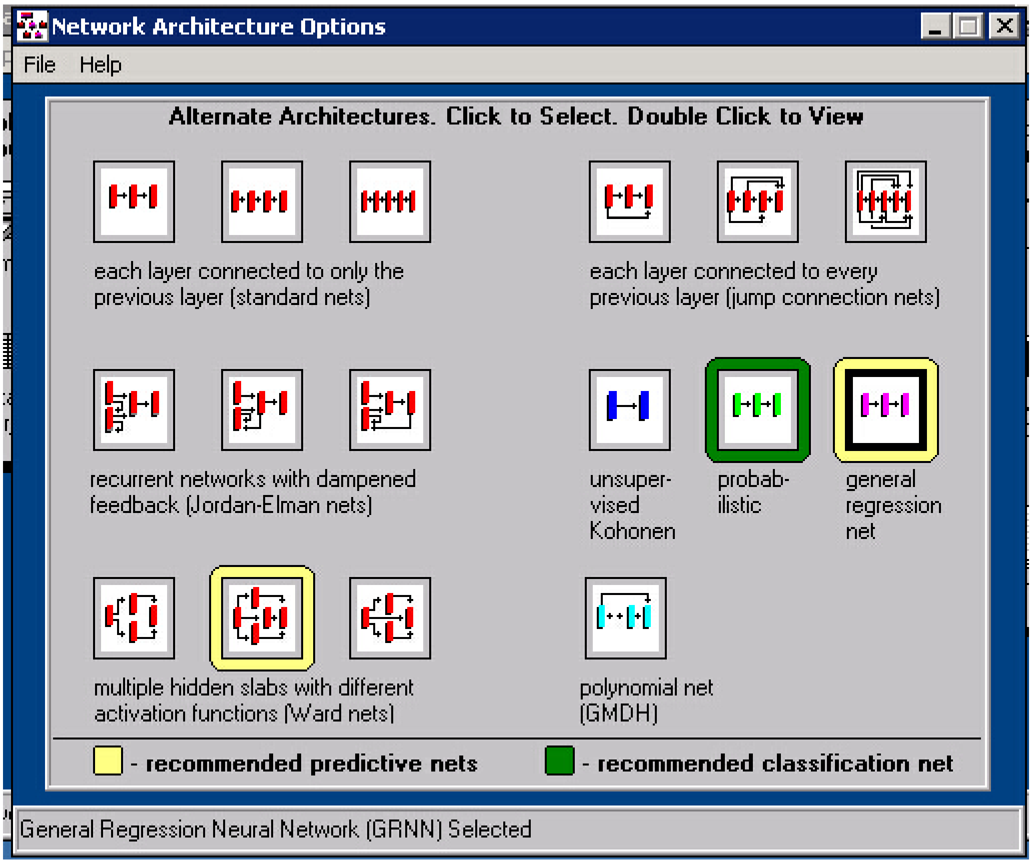

Figure 1: different models provided by NeuroShell2

Backpropagation networks are known for their ability to generalize well on a wide variety of problems and it is used for the vast majority of working neural network application because they tend to generalize well. Furthermore, Backpropagation networks have several variations.

1. The simplest one, each layer connects to the immediately previous layer, and it is easy to use and very practical.

2. Each layer connects to every previous layer.
3. Recurrent networks with dampened feedback from either the input, hidden and output layer. According to the document of NeuroShell2, this kind of sub-model in BP is very good at dealing with time series data.
4. Ward networks with multiple hidden slabs. These are some sort of advanced sub-model in BP, because once we have different hidden slabs we can assign different activation function to them so that the model can have a different view of the input data.

The Kohonen Self Organizing Map network is a type of unsupervised network, which has the ability to learn without being shown correct outputs in sample patterns. We use this model to cluster our sample data into several categories and then build up individual models for each of them, and that' one of the way we try to increase the accuracy of the prediction.

Probabilistic Neural Network (PNN) is a powerful model to categorize the input data especially applying genetic adaptive algorithm to it. Another model that can increase the generalization ability by applying genetic adaptive algorithm is General Regression Neural Networks (GRNN) which known for its ability to predict a continuous output value quickly and precisely.

3.2 Parameters

3.2.1 Number of Hidden Neurons

To compare the effect that caused by different number of hidden neurons, we have tested several groups of hidden neurons. First we use the default number in NeuroShell2 which is $\frac{1}{2}(N+K)+\sqrt{P}$

N is the number of input dimensions, K is the number of output neurons and P stands for the number of patterns in the training set. Then we increase the number gradually to see the difference.

3.2.2 Choice of Initial Weights

Initial weights strongly affect the final result of a NN model, so we try 3 different values: 0.1, 0.3 and 0.5 to see the consequence of changing the starting point.

3.2.3 Choice of Learning Rate

Guided by the documents of NeuroShell2, we alter the learning rate to adjust to different activation functions, for example, when the activation function of the hidden layer is linear, the learning rate should be relatively small, say 0.1, while in a classification scenario, the learning rate should get larger, say 0.9. Also two different values of momentum have been tested.

3.2.4 Choice of Activation Function

NeuroShell2 offer us various kinds of activation functions, we mainly use Gaussian function in the hidden neurons and linear function in the output layer which is suggested by the official document of NeuroShell2, and it turns out to have a good result in the prediction situation. However, logistic activation function is considered the best choice for classification. Besides, to make a comparison we also tried some other combination of activation functions including tanh and sine.

3.2.5 Choice of Input

We define a various group of input parameters in order to test which parameters will affect the prediction result more, and our target is to find out a best combination of parameters which can produce a prediction with relatively high accuracy.

Group1: only closing prices

Group2: closing prices + exchange volumes

Group3: closing prices + moving averages

Group4: closing prices + exchange volumes + moving averages

Group5: opening prices + closing prices + exchange volumes + moving averages

4. Results and Analysis

4.1 General Results

Generally speaking, clustered stock data and all stock data have a higher classification accuracy (more than 80 percent) compared with that o one specific stock data (which is around 77 percent), while the prediction R squares of clustered stock data and all stock are closer to 1 (which are around 0.99) compared with that of one specific stock data (which is only 0.90). Values of the average accuracy are shown in Table 5.1 and values of the average R square of different types of data are shown in Table 5.2.

Table 5.1. The average accuracy of different data sets

Data Sets	Average accuracy
All Stocks	0.8026
One Specific Stock	0.7742
Clustered Stocks – Class 1 (710)	0.8566
Clustered Stocks – Class 2 (241)	0.8203

Table 5.2. The average R square of different data sets

Data Sets	Average R square
All Stocks	0.9969
One Specific Stock	0.9091
Clustered Stocks – Class 1 (710)	0.9880
Clustered Stocks – Class 2 (241)	0.9899

At the first sight, the average R square of the prediction process is much higher numerically than the average accuracy of the classification process. However, this is not the case. The criterion of the classification accuracy indicates the accuracy directly as its name implies, while the criterion of the R square does not. The R square criterion is the coefficient of determination and indicates how well data points fit a statistical model, which can be simply a line or a curve. It provides a measure of how well observed outcomes are replicated by the model and ranges from 0 to 1 numerically. Data sets with the R square value closer to 1 fit better to a statistical model. A higher R square value does not necessarily promise a higher accuracy of the prediction nor a lower error rate of the prediction.

4.2 Comparisons and Analysis

4.2.1 Classification scenario

BP Architecture

In general, **data of all stocks and clustered data share more accurate results compared with data of a specific stock using the BP Architecture**, with the average accuracy of more than 80 percent on all and clustered data compared with around 70 percent on data of a specific piece of stock. Relative contribution factors graph is shown in Figure 5.1. The amount of volume, denoted as Factor 2, is one of contribution factors in stock classification. Factor 1 and Factor 6, 7, 8, 9, 10 and 11 are close prices different days ago, and they are also major contribution factors.

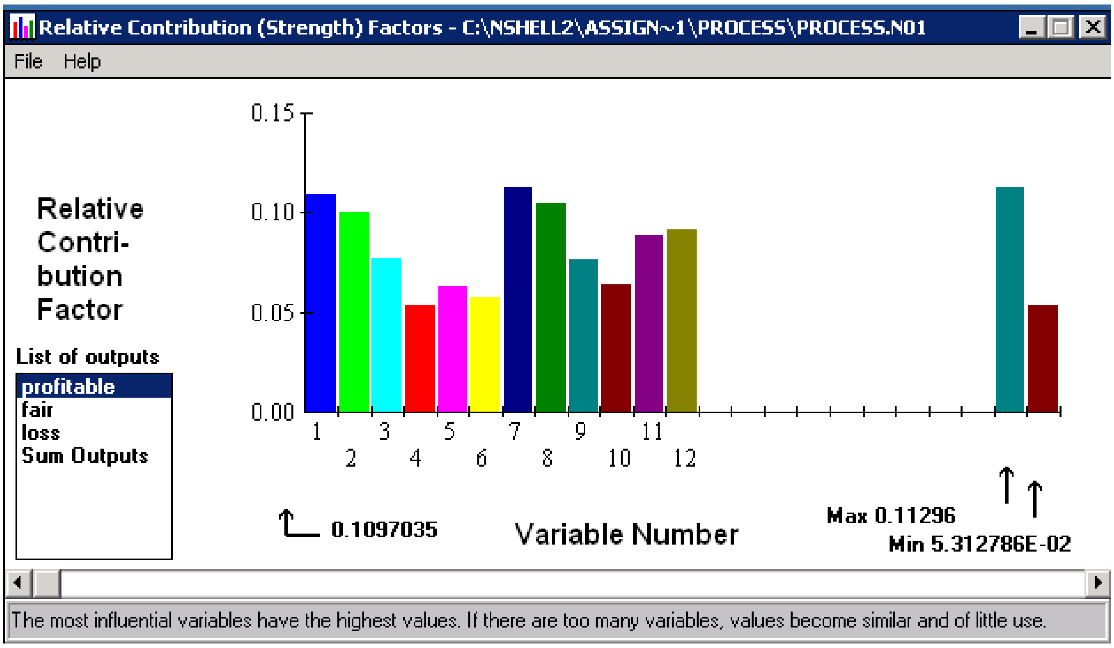

Figure 5.1. Relative contribution factors graph in BP Architecture

We select first 200 patterns and the output stock market curve is well fitting the actual one. The actual stock market curve and the output one are shown together in Figure 5.2.

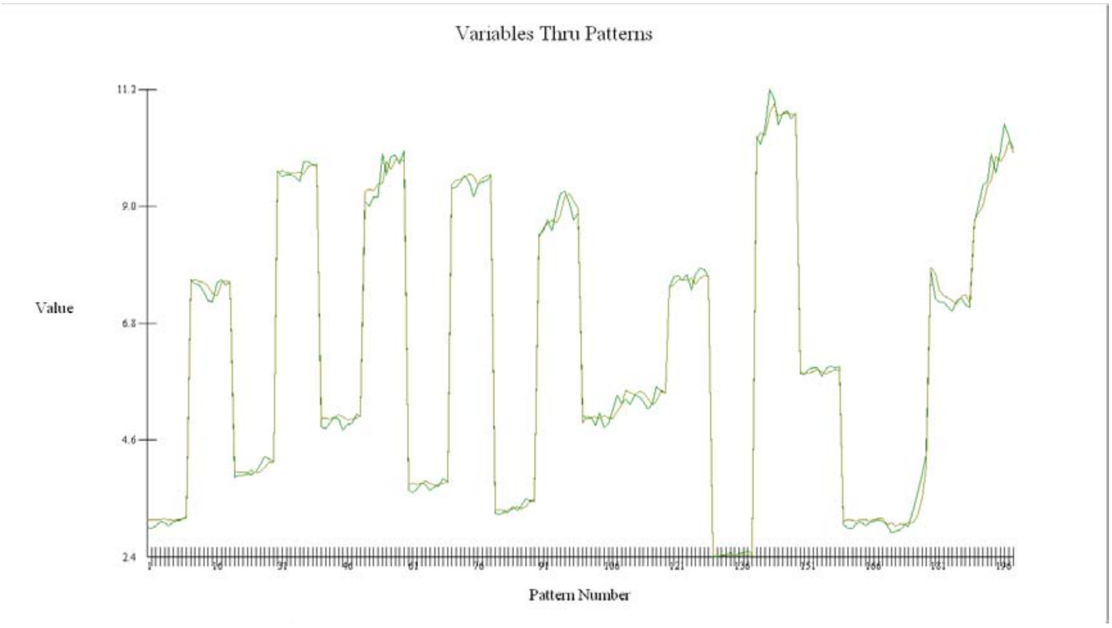

Figure 5.2. The actual stock market curve and the output one

1. **Standard connections**

   We first use Three Layer BP neural network (3L-BP).

   3L-BP performs well in classification.

   By altering different parameters in 3L-BP, namely, the scale function, the learning rate, the momentum, the initial weight, the number of neurons in hidden layer and different input parameters, we notice that **the increase of neurons in hidden layer significantly** leads to an increase in the accuracy of classification by improving accuracy from around 70 percent to almost 80 percent on data of a specific stock. The improvement is shown in Figure 5.3.

mod	module	scale	activation functi	learning rat	hidde	cali	input	accuracy/R square
CATE	3L-BP	〈0,1〉	gaussian/logistic	0.1/0.1/0.3	21	100	MA5,10,20,30,60&Close20,10,5,3,2,1,0	0.6954
CATE	3L-BP	〈0,1〉	gaussian/logistic	0.1/0.1/0.3	30	100	MA5,10,20,30,60&Close20,10,5,3,2,1,0	0.7900
CATE	3L-BP	〈0,1〉	gaussian/logistic	0.1/0.1/0.3	40	100	MA5,10,20,30,60&Close20,10,5,3,2,1,0	0.8041

Figure 5.3. Significant improvement by increasing number of hidden neurons

The increase of relevant input also brings about an increase, not significant, although, in the accuracy by improving from 80 percent to 81 percent. Changes in the learning rate, the momentum and the initial weight accelerate the velocity of learning process significantly, but improve the accuracy slightly (increase of less than 1 percent). Changes in other parameters, such as the choice of activation functions and the number of calibration interval, can hardly improve the classification results.

2. **Recurrent Networks and Ward Networks**

The Recurrent Network architecture has a poorer performance compared with the 3L-BP architecture.

The increase of relevant input brings about an increase, not significant, although, in the accuracy by improving from 69 percent to 73 percent, shown in Figure 5.4.

mode	module	scale	activation functi	learning ra	hidde	cali	input	accuracy/R square
CATE	JE-net(3)	<0,1>	gaussian/logistic	0.9/0.5/0.3	38,3	100	MA5,10,20,30,60&Close20,10,5,3,2,1,0	0.6888
CATE	JE-net(2)	<-1,1>	gaussian/logistic	0.9/0.5/0.3	38,38	100	MA5,10,20,30,60&Close20,10,5,3,2,1,0	0.7286

Figure 5.4. Improvement by increasing number of hidden neurons

By altering other parameters, such as scale function or activation function in Recurrent Network as well as Ward Networks, we find out that no significant improvement can be seen compared with BP with standard connections.

PNN Architecture

Generally, **clustered data and data of a specific stock share more accurate classification results compared with data of all stocks using PNN Architecture**, with average accuracy of more than 90 percent in clustered data and data of a specific stock compared with around 80 percent in stocks data. PNN Architecture also presents better results, namely, higher accuracy, than BP Architecture, by having accuracy of almost 90 percent compared with accuracy of around 80 percent. The PNN Architecture, in general, is superior to the BP Architecture in classification. This is shown in Figure 5.5.

mode	module	scale	activation functi	learning ra	hidde	cali	input	accuracy/R square
CATE	3L-BP	<0,1>	gaussian/logistic	0.1/0.1/0.3	21	100	MA5,10,20,30,60&Close20,10,5,3,2,1,0	0.6954
CATE	GA-PNN	<0,1>	/	/		/	MA5,10,20,30,60&Close20,10,5,3,2,1,0	0.9286

Figure 5.5. BP and GA-PNN classification accuracy

1. **PNN with Genetic Adaptive learning (GA-PNN)**

GA-PNN performs much better than the BP architecture in general.

The number of relevant inputs has a significant influence on the accuracy of classification on the data of a specific stock. This is shown in Figure 5.6. (Volumes are less important compared with close price and average close prices.)

mod	module	e fu	tivation funct	l	ate/momentum/init	io	input	accuracy/R square
CATE	GA-PNN	<-1, 1>	/		Smoothing Factor 0.5	/	close, M2_close-M7_close, close-7-close-1	0.9461
CATE	GA-PNN	<-1, 1>	/		Smoothing Factor 0.5	/	Volume, Volume-7-Volume-1, close-7-close-1	0.6639

Figure 5.6 Improvement by changing number of relevant inputs

By altering other parameters in GA-PNN, namely, the scale function and different input parameters, we find out that the scale function brings about the increase in the accuracy accordingly, but not significantly.

2. **PNN with Iterative, non-adaptive learning**

PNN with non-adaptive learning learns much faster compared with GA-PNN, while presents a poorer classification accuracy on all three data sets. PNN with non-adaptive learning may not be suitable for classification.

4.2.2 Prediction scenario

BP Architecture

Generally, **data of all stocks and clustered data share higher R square values compared with data of a specific stock using BP Architecture,** with average R square of more than 0.98 in all and clustered data compared with around 0.9 on data of a specific piece of stock. Factor 1 is the close price of yesterday, which is definitely the most significant contribution factor in the prediction of stock price. The relative contribution factors graph is shown in Figure 5.7.

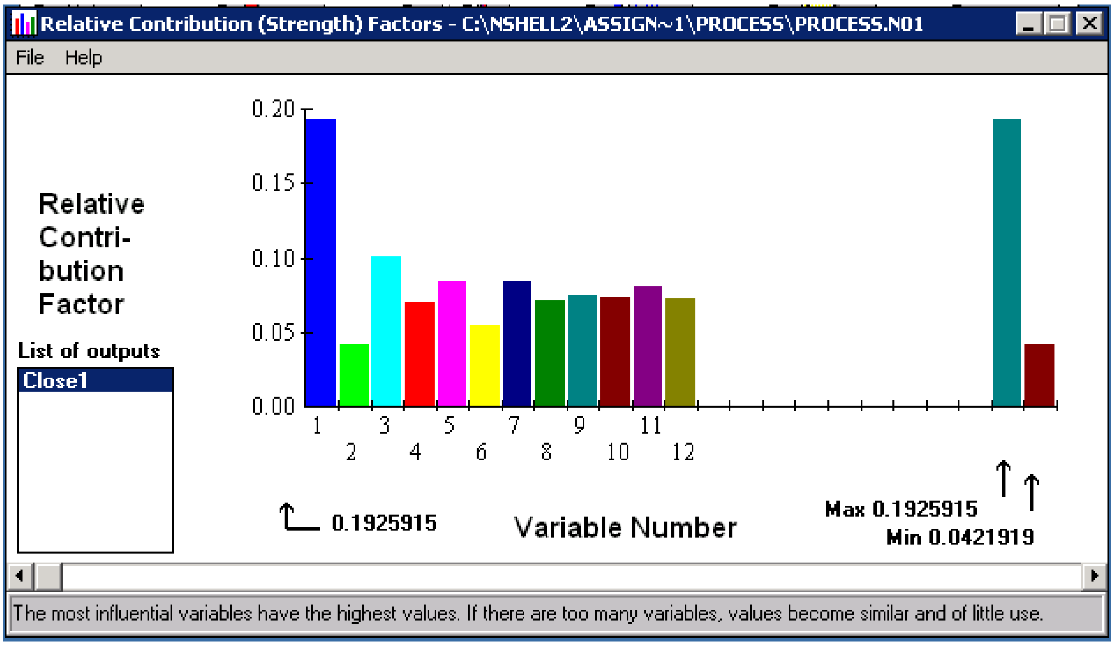

Figure 5.7 Relative contribution factors graph in BP Architecture

The actual stock market curve and the output one are shown in Figure 5.8. We select first 200 patterns and the output stock market curve is well fitting the actual one.

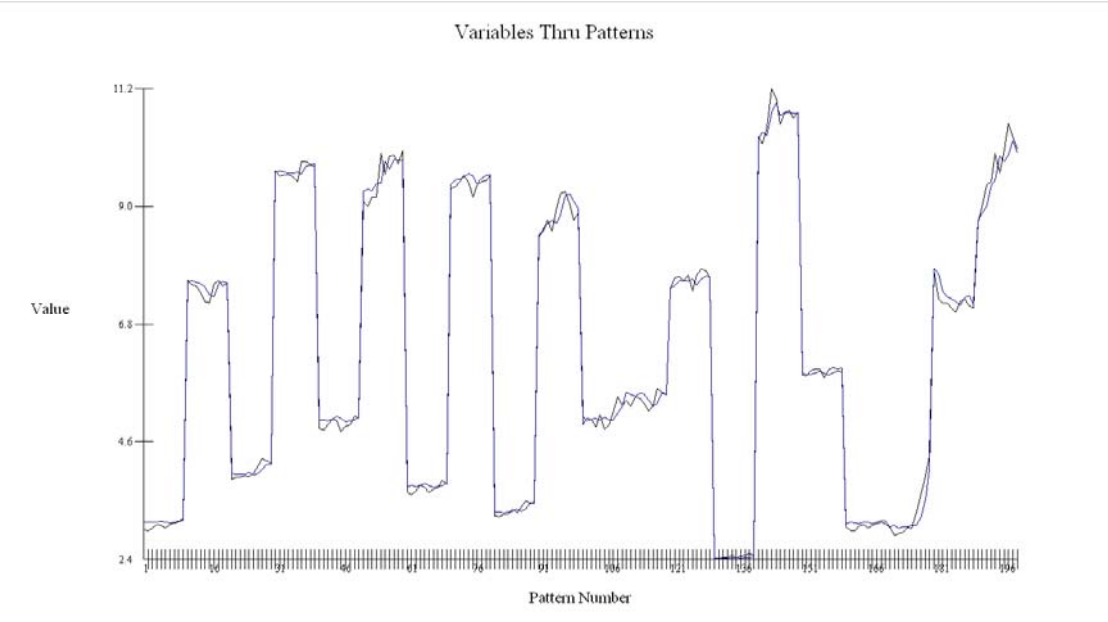

Figure 5.8. The actual stock market curve and the output one

1. **Standard connections**

   We use Three Layer BP neural network (3L-BP).

   The 3L-BP architecture performs well in the prediction of close price of stock generally.

By altering different parameters in 3L-BP, namely, the scale function, the learning rate, the momentum, the initial weight, the number of neurons in hidden layer and different input parameters, we notice that the number of neurons in hidden layer has an influence on the R square of prediction. The hidden layer which has the recommended number of neurons results in the highest R square value. This is shown in Figure 5.9.

mod	module	scale	activation functi	learning rat	hidde	cali	input	accuracy/R square
PRED	3L-BP	<0,1>	gaussian/linear	0.05/0.1/0.1	30	100	MA5,10,20,30,60&Close20,10,5,3,2,1,0	0.9738
PRED	3L-BP	<0,1>	gaussian/linear	0.1/0.5/0.3	40	100	MA5,10,20,30,60&Close20,10,5,3,2,1,0	0.9731
PRED	3L-BP	<0,1>	gaussian/linear	0.1/0.5/0.3	50	100	MA5,10,20,30,60&Close20,10,5,3,2,1,0	0.9461

Figure 5.9. Recommended neuron number results in the highest R square

The increase of relevant input can hardly improve the prediction results in the R square compared with the relatively high R square value of 0.98 on data of all stocks.

2. **Recurrent Networks and Ward Networks**

BP Architecture with Recurrent Networks presents a relatively lower R square on data of a specific stock compared with BP Architecture with standard connections. This is shown in Figure 5.11.

mod	module	scale	activation functi	learning rat	hidde	cali	input	accuracy/R square
PRED	3L-BP	<0,1>	gaussian/linear	0.1/0.5/0.3	40	100	MA5,10,20,30,60&Close20,10,5,3,2,1,0	0.9731
PRED	JE-net(2)	<-1,1>	gaussian/logistic	0.1/0.5/0.3	38,38	100	MA5,10,20,30,60&Close20,10,5,3,2,1,0	0.9072

Figure 5.11. BP Architecture with Recurrent Networks presents a lower R square

Changes the scale function and the number of neurons in hidden layer do have influence on the results. This is shown in Figure 5.12.

mod	module	scale	activation functi	learning rat	hidde	cali	input	accuracy/R square
PRED	JE-net(2)	<-1,1>	gaussian/logistic	0.1/0.5/0.3	38,38	100	MA5,10,20,30,60&Close20,10,5,3,2,1,0	0.9072
PRED	JE-net(3)	<0,1>	gaussian/linear	0.1/0.5/0.3	38,1	100	MA5,10,20,30,60&Close20,10,5,3,2,1,0	0.6593
PRED	JE-net(3)	<-1,1>	gaussian/linear	0.1/0.5/0.3	38,1	100	MA5,10,20,30,60&Close20,10,5,3,2,1,0	0.3862

Figure 5.12. Influence of the scale function and the number of hidden neurons

By altering different parameters in Ward Networks, we find out that no significant improvement can be observed compared with BP with standard connections.

GRNN Architecture

Generally, **clustered data and data of all stocks share more, not significantly though, accurate results compared with data of a specific stock using GRNN**

Architecture, with average R square of more than 0.99 in clustered data and data of all stocks, compared with around 0.97 in data of a specific stock. GRNN Architecture also presents a better results, namely, higher R square, than BP Architecture, by having R square of almost 0.99 compared with R square of around 0.98. This is shown in Figure 5.13.

mod	module	scale	activation functi	learning ra	hidde	cali	input	accuracy/R square
PRED	3L-BP	<0,1>	gaussian/linear	0.05/0.1/0.1	30	100	MA5,10,20,30,60&Close20,10,5,3,2,1,0	0.9738
PRED	GA-GRNN	<0,1>	/	/		/	MA5,10,20,30,60&Close20,10,5,3,2,1,0	0.9935

Figure 5.13 BP and GA-GRNN prediction R square

1. **GRNN with Genetic Adaptive learning (GA-GRNN)**

   By altering different parameters in GA-GRNN, namely, the scale function and different input parameters, we find out that the scale function and the increase of relevant input do have influence on the R square of prediction, although not significantly because the R square is near to 1.

2. **GRNN with Iterative, non-adaptive learning**

   GRNN with non-adaptive learning learns much faster compared with GA-GRNN, while presents a poor prediction R square on all data sets. GRNN with non-adaptive learning may not be suitable for prediction.

5.Limitations

Sound results we have, our study is limited by several factors listed as follows:

1. **Representativeness**

   An indication of overfitting is the nearly perfect prediction results with the R square of 0.99. This may be caused by statistical factors we choose cannot fully describe factors of stock market. Some professional factors are too hard to process and comprehend.

2. **Limited scale**

   The data we collected are of a relatively small scale which may not be capable of representing data of the whole stock market nor a specific stock. This may also result in the overfitting phenomena in the classification and prediction, and the model fits too well to the data which may have a poorer ability of generalization.

3. **Irrelevant factors**

   The results presented show that classification and prediction on both data of all stocks and clustered data are more accurate than those on data of a specific stock. This may be caused by some statistical factors we have chosen are not that relevant and have less influence on results. In the classification and prediction on data of a specific stock, we choose close prices of 1, 2, 3, 5, 10 and 20 days ago and compute average stock prices within 5, 10, 20, 30 and 60 days. Close prices which are too far away have less contribution to the close price of the current day.

6. Conclusion

NN technology can be used to predict the trend and the exact price level of the stock market by combining with financial indices, and can provide more accurate result compared with traditional statistical methods despite some limitations. By applying various NN technology architectures, such as BP Architecture, GA-PNN Architecture and GA-GRNN Architecture, to data sets for classification and prediction, we can get sound results with high accuracy. The GA-PNN architecture can improve the accuracy more significantly compared with BP Architecture and the GA-GRNN architecture can predict more accurately than the BP Architecture and thus they are more suitable for the classification and prediction. Clustered data after clustering by the Unsupervised Model can have more accurate prediction by applying GA-PNN or GA-GRNN on them. NN technology can be applied to improve the accuracy of prediction of stock price and further investigation can be conducted to bring out more accurate prediction.

References

[1] Q. Cao, M. Parry, and K. Leggio, "The three-factor model and artificial neural networks: predicting stock price movement in China", *Annals of Operations Research,* vol. 185, 2011, pp. 25-44.

[2] M. P. Naeini, H. Taremian, and H. B. Hashemi, "Stock market value prediction using neural networks", *2010 International Conference on Computer Information Systems and Industrial Management Applications (CISIM)*, 2010, pp. 132-136.

[3] E. Hadavandi, H. Shavandi, and A. Ghanbari, "Integration of genetic fuzzy systems and artificial neural networks for stock price forecasting", *Knowledge-Based Systems,* vol. 23, 2010, pp. 800-808.

[4] C.-H. Cheng, T.-L. Chen, and L.-Y. Wei, "A hybrid model based on rough sets theory and genetic algorithms for stock price forecasting", *Information Sciences*, vol. 180, 2010, pp. 1610-1629.

Appendix

Meanings of Variables

1. For data of a specific stock

Date: the corresponding date of stock records

Amount: the total amount of money in one day

Low: the lowest price in one day

High: the highest price in one day

Open#: the opening price of a specific stock # days ago

Close#: the closing price # days ago

Close-1: tomorrow' closing price

Volume#: the Volume of a piece of stock # days ago

MA#: #-day moving arrange of stack price

2. For data of all stocks

Symbol: the symbol (ID) of a specific stock

Name: the name of a specific stock

Date: the corresponding date of stock records

Low: the lowest price in one day

High: the highest price in one day

Amount: the total amount of money in one day

Open#: the opening price all stocks # days ago

Close#: the closing price # days ago

Close-1: tomorrow' closing price

Volume#: the Volume of a piece of stock # days ago

MA#: #-day moving arrange of stack price