Mathware & Soft Computing 12 (2005) 33-39

A Heuristic Forecasting Model for Stock Decision Making

D. Zhang, Q. Jiang and X. Li
School of Computer Science
Huazhong Univ. of Science & Technology, Wuhan, China
hustzdf@sina.com

Abstract

This paper describes a heuristic forecasting model based on neural networks for stock decision-making. Some heuristic strategies are presented for enhancing the learning capability of neural networks and obtaining better trading performance. The China Shanghai Composite Index is used as case study. The forecasting model can forecast the buying and selling signs according to the result of neural network prediction. Results are compared with a benchmark buy-and-hold strategy. The forecasting model was found capable of consistently outperforming this benchmark strategy.

1 Introduction

Neural networks are a class of generalized nonlinear nonparametric models inspired by studies of the human brain. They are robust and have good learning and generalization capabilities in data-rich environments [1], and are appropriate for clustering and prediction problems. Neural networks get their intelligence from learning process, and then this intelligence makes them to have the capability of auto-adaptability, association and memory to perform certain tasks. For a more detailed description of neural network, the interested reader is referred to the papers in [2,3,4]. Since neural network can learn valuable information from a mass of history information, so neural networks have obtained efficiently business applications [1, 3, 4, 5, 6, 7]. Especially, some researches [8, 9, 10] have shown that neural network performed better than conventional statistic approaches in financial forecasting.

The traditional view of neural networks is of a black-box program that emulates biological brain and learns to recognize patterns or categorize input data by being trained on a set of sample data from the domain. Learning through training and subsequently the ability to generalize broad categories from specific examples [3] is the unique perceived source of intelligence in neural networks. However, experienced neural networks application designers typically perform extensive knowledge engineering and incorporate a significant amount of domain knowledge into the design of neural networks even before the learning through training process has begun [11]. Design of optimal neural networks is problematic in that there exist a large number of alternative network physical architectures and learning methods, all of which may be applied to a given business problem.

Despite neural networks as forecasting tool have many advantages, however, due to the complexity and diversity of network architectures and learning methods, they still have some drawbacks, for example, overfitting and poor explanation capability and so on, which significantly affect the performance of neural network. In fact, neural network applications are frequently viewed as black boxes which mystically determine complex patterns in data. In order to enhance the learning capability of neural network, many researchers have improved neural networks by combining other techniques, for example, fuzzy logic [12], genetic algorithm [13]. In this paper, some heuristic strategies that use domain knowledge are presented to develop a forecasting model for enhancing learning capability of neural network and obtaining better trading performance. The actual results show that the heuristic forecasting model is efficient.

The rest of this paper is organized as follows. In section 2 we give a heuristic forecasting model. Computational results are described in section 3. Conclusions are summarized in section 4.

2 The Heuristic Forecasting Model

The feed-forward networks are the most widely used tools because they offer good generalization abilities and are readily to implement [14]. They can be used to identify patterns, which can then be used to predict the future. However, how to design neural network is still a difficult tasks [15]. The literature [11] presented some principles for the design of neural networks to improve the performance of neural network. In this paper, some heuristic designing strategies are presented as follows.

2.1 Heuristic Determination of Input Data

It is well known that stock market is a nonlinear dynamical system, and is affected mainly by factors like interest rates, inflation, economy and politics and so on. Although there is correlation, dependency and interaction between these factors is obvious, their relation is rather difficult to express in mathematical formulas. No formal methods currently exist for selecting input data for neural network solution to a domain. So forecasting financial markets is a really challenging task. It is well known that the closing price of stock market is one of the most important factors, and this price includes a lot of useful information, so the closing price time series is selected to predict the future trends of stock market. The aim in the paper is to mine future trends from a mass of the closing price data, and then provide decision-making for the different investors. In addition, the history closing price data about one month have more important affect on the next day price, so the size of input layer is 20 day.

In the process of modelling neural network, data is usually preprocessed before training in order to make network more effective [2]. Due to the activation function in this paper is a sigmoid function that squashes input data to [0,1], so the real stock data that varies from 250 to 2300 will not be useful. In order to make fitting range not saturate, all the patterns are scaled in [0.1,0.9].

In addition, in order to eliminate noise of the closing price time series, the real stock data are smoothed by kernel smoothing method [16]. In order to filter noise, a kernel smoothing technique [6] is used. In detail, for a time series $Y : y_1, y_2, \cdots, y_n$, we can calculated a smoothed time series $\hat{Y} : \hat{y}_1, \hat{y}_2, \cdots, \hat{y}_n$ as following:

$$\hat{y}_i = \frac{\sum_{t=1}^{n} K_h(i-t) y_t}{\sum_{t=1}^{n} K_h(i-t)}, \tag{1}$$

where

$$K_h(x) = \frac{1}{h\sqrt{2\pi}} e^{-\frac{x^2}{2h^2}}.$$

2.2 Heuristic Modification of Error Function

Due to the nonlinearity inherent in stock market, the error function is unlikely to be globally convex and can have many local minima. In addition, the trend of up and down in the stock market is very important for making profits. Inspired by these ideas, a penalizing coefficient α is integrated into the error function similar to [18] in order to make prediction more effective, that is

$$MSE = \frac{1}{2} \sum_{i=1}^{N} \alpha (\tilde{x}_{t+i} - x_{t+i})^2, \tag{2}$$

where x_{t+i} is the real closing price in $t+i$ time, $\tilde{x}_{t+i}$ is a predict value with

$$\alpha = \begin{cases} \beta & \text{if } (\tilde{\mathrm{x}}_{\mathrm{t+i}} - \mathrm{x}_{\mathrm{t+i-1}})(\mathrm{x}_{\mathrm{t+i}} - \mathrm{x}_{\mathrm{t+i-1}}) > 0 \\ 2-\beta & \text{otherwise} \end{cases}, \beta \in (0,1).$$

It denotes that, if the predicted direction is the same as the real direction, α is given by a smaller constant, otherwise α is given by a bigger constant, namely, the incorrect predicted directions are penalized more heavily than the correct predicted directions.

It is noted that the modified error function is different from [18], the error function in this paper is more simple and efficient. In addition, the backpropagation algorithm, which is the most popular learning algorithm, is adopted to perform steepest descent on the error function (2).

2.3 Heuristic Design of Trading Strategies

Although neural networks have powerful learning and nonlinear mapping capabilities, they were earlier thought to be unsuitable for data mining because of the inherent black-box nature and poor explanation capability [1]. The investors can only trust output of neural network blindly. Recently, there has been many researches aimed to avoid those problems. In this paper, a trading strategy is considered to avoid the poor explanation capability of neural network. In fact, whether the forecasting model is effective or not in real world, it must be tested by the trading strategies. Thus the forecasting model can give us valuable suggestions for our investment decision-making. The presented trading strategy is as follows:

$$Sign = \begin{cases} Buy & \text{if } \mathrm{R_{BP}} > \mathrm{R_{BT}} \\ Sell & \text{if } \mathrm{R_{SP}} > \mathrm{R_{ST}} \text{ and Down} - \text{trend} \end{cases},$$

where R_{BT} and R_{ST} is a constant R_{BP} is the return of predict buying and can be calculated as $R_{BP} = \frac{max_{1\le i\le 30}(\tilde{x}_{t+i})-x_t}{x_t}$, $R_{SP}$is the return of predict selling and can be calculated as $R_{SP} = \frac{max_{1\le i\le 30}(\tilde{x}_{t+i})-x_t}{x_t}$. The forecasting model can tell us when the stock is bought or sold according to the presented trading strategies. denotes the trend in the past two days is down and it can be computed by the kernel smoothing method.

Generally, the output data cannot be directly used to predict, it must be post-processed, namely using the inverse of the preprocessing transformation. In addition, the real stock data are smoothed by kernel smoothing method according to (1), thus the up and down trend can be determined. The trading strategy in this section makes use of this trend.

3 Computational Results

The heuristic forecasting model (HFM) and forecasting model without heuristic (FM) were implemented in C++ for Dos on a PC. The performance of both forecasting models were evaluated by trading the SEE stock market index from May 1996 to May 2003. The training patterns in this paper are the closing price of 500 trading days. Other parameters are designed as follows: The initial weights and thresholds are in $[-0.5, 0.5], n = 20, m = 5, R_{BT} = 0,05, R_{ST} = 0.04, \beta = 0.5$. Here, n is the number of input neuron, m is the number of hidden neuron. A backpropagation neural network with one input layer and one output layer and one hidden layer is selected to forecast. All the training patterns are selected randomly to train network about 15000 times, the process is repeated until the stop criteria is met, where the stop criteria is $thecurrenttrainingerror > 1.3 * Min(allthetrainingerror)$. The accuracy of trend in 30 days between HFM and FM can be seen in Fig. 1. The comparisons of returns between HFM, FM and the buy and hold strategy can be seen in Fig. 2. All returns were calculated after taking the actual transaction costs for each transaction into consideration, namely the transaction costs for buying and selling is 1% respectively. The number of the total trading is 41, and the number of selling is 20, the percentage of profitable trading among the total number of trading is 82.9% for the heuristic forecasting model, however, the percentage of profitable trading among the total number of trading is only 73% for forecasting model without heuristic.

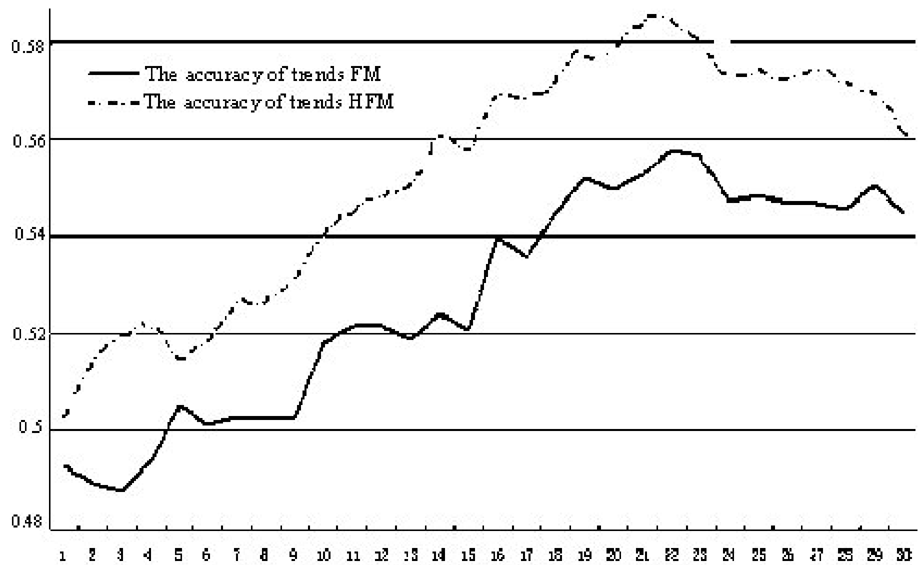

Fig. 1. The accuracy of trends in 30 days between HFM and FM.

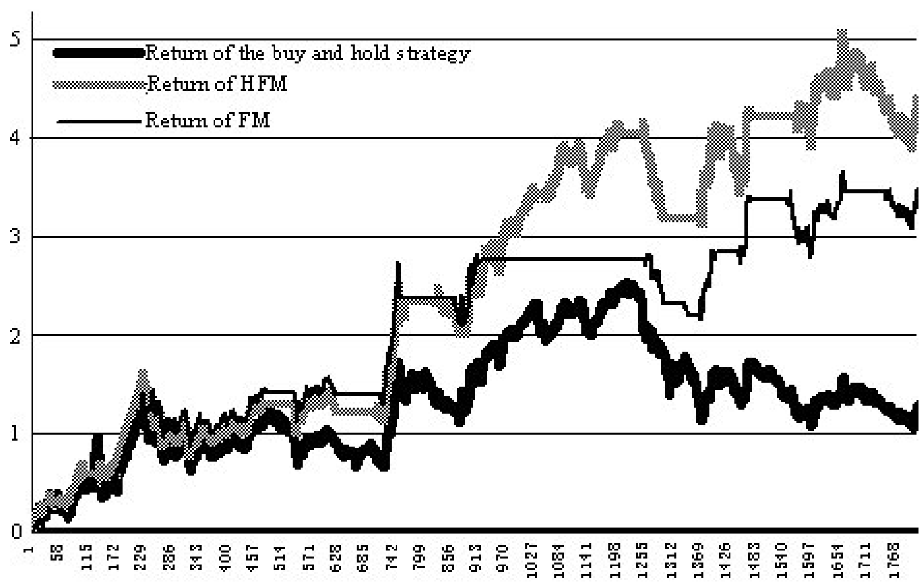

Fig. 2. The comparisons of returns between HFM, FM and the buy and hold strategy

4 Conclusions

For Fig. 1, the average trend accuracy is 55% over 30 days for HFM, for FM, only 52.6%. For Fig. 2, the return achieved by HFM and FM is 4.39 and 3.48, respectively, and the return achieved by the buy and hold strategy is $R_{bh} = 1.47$. The former is about three times as large as the latter. So the forecasting model is superior to the buy and hold strategy. In addition, from Fig. 2, it can know that the presented forecasting model performs better in bear market than in bull market, it denotes that the forecasting model have better capability of controlling risk.

The results of trading about seven years to Shanghai Composite Index show that the heuristic forecasting model was encouraging. This forecasting model can be applied to actual trading and is very efficient for combinational investment. The future work is to strive for making the model more adaptive to the application and applying the model to other market and individual stock.

References

[1] Sushmita Mitra, Sankar K. Pal, and Pabitra Mitra, Data Mining in Soft Computing Framework: A Survey. IEEE Transactions on Neural Networks 13(1) (2002), 3-14.

[2] Zurada J.M., An introduction to artificial neural networks systems. St. Paul: West Publishing (1992).

[3] Widrow B., Rumelhart D.E., and Lehr M.A., Neural networks: applications in industry, business and science. Communications of the ACM 37(3) (1994), 93-105.

[4] Ralf Herbrich, Max Keilbach, Thore Graepel, Peter Bollmann-Sdorra, and Klaus Obermayer, Neural Networks in Economics: Background, Applications, and New Developments. Advances in Computational Economics 11 (1999), 169-196.

[5] R.A. Ribeiro, H.-J. Zimmermann, R.R. Yager and J. Kacprzyk (Eds.), Soft Computing in Fnancial Engineering. Physica-Verlag (Springer-Verlag), Heidelberg and New York (1999).

[6] Kate A.Smith, Jatinder N.D. Gupta, Neural networks in business: techniques and applications for the operations researcher. Computers & Operations Research 27 (2000),1023-1044.

[7] Hoptroff R.,The principles and practice of time series forecasting and business modelling using neural nets. Neural Computing & Applications 1 (1993), 59-66.

[8] Refenes A.N., A.Zapranis, and G. Francis, Stock performance modeling using neural network: A comparative study with regression models. Neural Network 5 (1994), 961-970.

[9] Hill T.,O'Connor M.,and Remus W., Neural Network models for times series forecasting, Management Science 42 (7) (1996), 1082-1091.

[10] Y.S.Abu-Mostafa, A.F. Atiya, M. Magdon-Ismail, and H. White, Neural networks in financial engineering. IEEE Transactions on Neural Networks 12(4) (2001), 653-656.

[11] Steven Walczak and Narciso Cerpa, Heuristic principles for the design of artificial neural networks. Information and software technology 41(2) (1999), 109-119.

[12] Lotfi A. Zadeh, Fuzzy Logic, Neural Networks, and Soft Computing. Communications of the ACM 37(3) (1994), 77-84.

[13] Paul G. Harrald and Mark Kamstra, Evolving Artificial Neural Networks to Combine Financial Forecasts. IEEE Transactions on Evolutionary Computation 1(1) (1997), 40-52.

[14] H. White, Connectionist nonparametric regression: Multilayer feedforward networks can learn arbitrary mappings. Neural Networks 3(5) (1990), 535-549.

[15] Bartlett P., and Ben-David S., Hardness Results for Neural Networks Approximation Problems. Theoretical Computer Science 284(1) (2002), 53-66.

[16] Wand M.P. and Jones M.C, Kernel Smoothing. Chapman and Hall, London (1995).

[17] Caldwell R.B, Performances metrics for neural network-based trading system development. NeuroVet Journal 3(2) (1995), 22-26.

[18] Yao J.T.,H.L.Poh., Equity forecasting: a case study on the KLSE index, Neural Networks in Financial Engineering. Proceedings of 3rd International Conference On Neural Networks in the Capital Markets, Oct 1995, London, A. N. Refenes, Y. Abu-Mostafa, J. Moody and A. Weigend (Eds.),World Scientific (1996), 341-353.