

行政院國家科學委員會專題研究計畫成果報告

混沌理論與傳遞神經網路系統

The Management of Chaotic Time-series Data

計畫編號：NSC 90-2416-H-004-031

執行期限：2001年8月1日至2002年7月31日

主持人：蔡瑞煌 政治大學資訊管理學系

計畫參與人員：曹美玲 政治大學資訊管理學系

一、中文摘要

我們在這個研究中，想了解能否設計出一個倒傳遞神經網路系統來模擬混沌系統？以及其能否對混沌系統作有效的預測？這些問題是本研究計畫想著力的議題。我們想從理論上探討：學習完混沌資料後的倒傳遞神經網路系統是否能重建其所學習的混沌模型。此外，本研究亦設計了一些實驗來檢測學習完混沌資料的倒傳遞神經網路系統是否為混沌系統，驗證的方法是檢驗其是否具有混沌資料的四個特性：有限性、非週期性、確定性、及對初始條件的敏感依賴。同時，更進一步地利用上述學習完的倒傳遞網路系統來預測所學習的混沌模型，這麼做的目的是想要了解：學習後的倒傳遞網路系統是一個混沌系統時，與學習後的倒傳遞網路系統不是一個混沌系統時，其預測能力的比較。

關鍵詞：倒傳遞神經網路系統、混沌系統

Abstract

There are no studies probed into a multilayered Perceptron with Back Propagation learning algorithm (BP system) after it has learned chaotic time-series data, nor any studies on whether BP system can effectively manage a chaotic time-series data. This study examines a BP system and explores if it can effectively manage a

chaotic time-series data. We find that the BP system may display various qualitative types of behavior for different values of weights: periodic cycles of different lengths and chaos. Also, chaotic time-series data with a large fluctuation will lead the BP system to yield a chaotic time-series data with a poor prediction effect. It seems that a BP system may manage badly a chaotic time-series data.

Keywords: The Back Propagation learning algorithm, Time-series Data, Chaos

二、緣由與目的

In modern finance, derivatives such as futures and options play increasingly prominent roles in risk management and price speculative activities. Owing to the high-leverage characteristic involved in derivative trading, investors can gain enormous profits with a small amount of capital if they can accurately predict the market's direction. Financial markets, however, can be influenced by many factors, such as, political events, general economic conditions, and traders' expectations. Predicting the financial market's movements is considered to be rather difficult in general. Movements in market prices are not random. Rather, they behave in a highly nonlinear, dynamic manner. The standard random walk assumption of futures prices may merely be a veil of randomness that shrouds a

messy nonlinear process (see, for example, Blank, 1991; DeCoster, Labys & Mitchell, 1992; Grudnitski & Osburn, 1993). To make the forecasting of futures prices more reliable, the application of Artificial Neural Networks (ANN), especially the multi-layered feed-forward network (Rumelhart, et. al., 1986), have received extensive attention (Grudnitski & Osburn, 1993; Hutchinson, Lo & Poggio, 1994; Tsaih, Hsu & Lai, 1998).

Since 1984, deterministic chaos has been hailed as a revolution in thought and attracted ever increasing attention of many scientists and technologists from diverse disciplines including biology, computation, engineering, economics, mathematics, meteorology, physics, statistics and many others. At the same time, since 1986, many researchers and practitioners have recognized that ANN is one of the ideal tools for managing the nonlinear environment. In the context of traditional statistical methods, the ANN can be considered as a multivariate nonlinear non-parametric inference technique that is data driven and model free. "Multivariate" implies that the ANN inputs comprise many different variables whose interdependencies and causative influences are exploited. "Nonparametric" and "model free" mean there are no presumptions regarding the relationship between the input and output variables. "Data driven" implies that the weights of the ANN are estimated from the (given) training data. One of the most popular ANN is the BP system, a layered feed-forward network with the back propagation learning algorithm.

The BP system adopts the layered feed-forward network structure, named a multi-layered Perceptron, with back propagation learning algorithm (Rumelhart,

et. al., 1986). The invention of the BP system results in the resurgence of ANNs because it can solve nonlinearly separable problems that remained unsolved by the previously invented Neural Networks, such as Perceptron (Rosenblatt, 1958).

There have been research and applications of the BP system to chaotic time-series data (Wong, 1991) (Matsuba, Masui & Hebishima, 1992) (Adachi et. al., 1992), time-series forecasting of financial markets (Azoff, 1994), and uncovering nonlinear structure in a stock market (Abhyankar et. al., 1997). Nevertheless, they have not examined the BP system after being trained on (chaotic) time-series data, and made a further study of whether it is proper to apply a BP system to such a time-series data.

Suppose we train the BP system with chaotic time-series data. Its purpose is to predict the sequence $x_{m+1} \sim x_{m+q}$ based on the current input sequence $x_1 \sim x_m$. The BP system may have $m+q-1$ input nodes, p hidden nodes and q output nodes, and all hidden and output nodes use the following hyperbolic tangent (tanh) activation function (cf. (Rumelhart, et. al., 1986)):

$$\tanh(x) \equiv \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (1)$$

That is, the following output value of the l^{th} output node tries to estimate x_{m+l} :

$$O_l \equiv \tanh\left({}_3w_{l0} + \sum_{i=1}^p {}_3w_{li} \tanh\left({}_2w_{i0} + \sum_{j=1}^{m+q-1} {}_2w_{ij} x_j\right)\right) \quad (2)$$

where ${}_2w_{i0}$ is the bias of the i^{th} hidden node, ${}_2w_{ij}$ is the weight between the variable x_j and the i^{th} hidden node, ${}_3w_{l0}$ is the bias of the l^{th} output node, and ${}_3w_{li}$ is the weight between i^{th} hidden node and the l^{th} output node. The BP system tries to catch the time structure embedded in time-series data. A question for such application is that, suppose the training time-series data is chaotic, can the

BP system after learning generate exactly the same chaotic time-series data?

The learning algorithm of the BP system adopts the generalized delta rule. That is, the learning algorithm is defined to minimize an objective function to find the optimal arrangement of weights via using the gradient descent method. The sum of error square is usually adopted as the objective function. The highly nonlinear property of the objective function leads to the notorious predicament of the relatively optimal learning result.

Hornik et. al., (1989) mention that the BP system acts just as an approximator. This conclusion comes from the fact that the value of the error term at the conclusion of learning is usually not zero. Even if the error term value at the end of learning is zero, it merely means that the BP system has perfectly learned the training data, not that the BP system has any perfect prediction ability. (Barron, 1991; Barron, 1992) mention that “when the network is exposed to test data that has not been seen before, the network function acts as an estimator of new points of the target function.” Due to the facts that the BP system is an estimator, and that the chaos has the property of sensitive dependence on initial condition, it seems the BP system will not generate the same chaotic time-series data.

Yet further questions remain: Will the time-series data yielded from the BP system be a chaotic one? If the time-series data yielded from the BP system is a chaotic one, how well does this time-series data mimic the time-series data the BP system just learns? If the BP system yields a non-chaotic time-series data, how large will the error be? These questions are unsolved, and motivate

this study.

For this study, we set up an experiment with the following finite-difference equation (cf. (Kaplan & Glass, 1995b)) adopted to derive time-series data that are used as the training data for the BP system:

$$x_{t+1} = f(x_t) \equiv R x_t (1 - x_t) \quad (3)$$

To answer the questions mentioned above, we examine the time-series data yielded from the BP system after learning, and see how well it predicts the behavior of the trained time-series data. To examine if the BP system yields a chaotic time-series data, we verify the value of Lyapunov exponent of the time-series data generated from the BP system.

三、結果與討論

Equation (3) with variant values of R is used here, and the experimental design is arranged as five blocks of variant values of R: (1) less than 3.5, (2) 3.56 to 3.65, (3) 3.72 to 3.76, (4) 3.81 to 3.86, and (5) greater than 3.9. For each value of R, there are 2000 training data¹ generated from equation (3). That is, the pairs of input and its associated desired output of training data are $\{(x_t, x_{t+1}), t = 0, 1, \dots, 1999\}$, where $x_{t+1} = R x_t (1 - x_t)$.

The BP system tries to catch the time structure embedded in the training time-series data. The output value O_t of BP system is arranged as follows:

$$O_t \equiv g(x_t) = \tanh\left(3w_0 + \sum_{i=1}^3 3w_i \tanh(2w_{i0} + 2w_{i1}x_t)\right) \quad (4)$$

After learning, O_t is a function of x_t , and tries to estimate x_{t+1} of equation (3).

For each case, each network system has 10 repetitions with different initial weights.

¹ We have tried 4000 training data instead of 2000 training data and the experimental results are similar. It seems that 2000 training data are sufficient to describe the time structure of the time-series data generated from equation (3).

The stopping rule for learning is the satisfaction of either the value of $\sum_{t=0}^{1999} (x_{t+1} - O_t)^2$ less than 10^{-25} or learning iterations greater than 250000.

After learning, we assess the following MRE (mean ratio of error) with respect to the learning and the prediction. To test the learning effect, the pairs of input and its associated desired output are $\{(x_t, x_{t+1}), t = 0, 1, \dots, 1999\}$ generated from equation (3) with x_0 being designed as in Table 1; to test the prediction effect, the pairs of input and its associated desired output are $\{(x_t, x_{t+1}), t = 0, 1, \dots, 1999\}$ generated from equation (3) with $x_0 = 0.49999$. In principle, the smaller MRE, the better the result.

$$MRE = \left(\frac{\sum_{t=0}^{1999} \left(\frac{X_{t+1} - O_t}{X_{t+1}} \right)^2}{2000} \right)^{1/2}$$

To examine if the BP system g yields a chaotic time-series data, we generate a time-series data from the BP system, i.e., $\{O_t, t = 0, 1, \dots, 2000\}$, with O_0 equals x_0 designed as in Table 1 and

$$O_{t+1} \equiv g(O_t) = \tanh(3w_0 + \sum_{i=1}^3 3w_i \tanh(2w_{i0} + 2w_{i1} O_t)) \quad (6)$$

The Lyapunov exponent (i.e., λ_1) regarding the time-series data generated from the BP system is calculated with the following equation:

$$\lambda_1 = \log \| J_1 \cdot J_2 \cdot \dots \cdot J_{2000} \| \quad (7)$$

where J_t is the derivative of O_t , i.e.,

$$J_t \equiv (1 - O_t^2) \sum_{i=1}^3 (1 - (\tanh(2w_{i0} + 2w_{i1} O_{t-1}))^2) 3w_i \quad (8)$$

If the value of λ_1 is positive, we say the time-series data generated from the BP system is a chaotic one.

Table 2 shows the test summary if the time-series data generated from the BP system is chaotic, and the average MRE values regarding the learning and the prediction. The following facts are obtained from the experimental results:

1. All experimental results show that, regardless what the trained time-series data are, the BP system acts as an approximator.
2. From Table 2, when the trained time-series data is non-chaotic, the BP system after learning may yield a non-chaotic or chaotic time-series data; as for the cases of chaotic training time-series data, the BP system after learning may also yield a non-chaotic or chaotic time-series data.
3. Most of the non-chaotic training time-series data lead to yielding a non-chaotic time-series data. However, in the cases of $R = 3.74, 3.83$ or 3.84 , where the characteristic of the time-series data is a stable cycle of odd period, most of λ_1 of the BP system are positive. As for the cases of chaotic training time-series data, there is no pattern.
4. The average MRE values of all cases are rather small, except the case of $R = 4.0$. It seems that the chaotic time-series data with a large fluctuation leads the BP system, after learning, to generate a chaotic time-series data with a poor prediction effect.
5. With respect to non-chaotic time-series data, the average MRE values of learning and prediction were almost the same.

In summary, the multi-dimensional finite-difference equation (6) (i.e., the BP system) may display various qualitative types of behavior for different values of $2w_{ij}$ and $3w_i$: periodic cycles of different lengths and chaos. It seems that the BP system may yield a non-chaotic or chaotic time-series data is relevant to the complexity of the family of equation (6).

四、計畫成果自評

This is the first study probed into a multilayered Perceptron with Back Propagation learning algorithm (BP system)

after it has learned chaotic time-series data. The purpose of this study is on whether BP system can effectively manage a chaotic time-series data. It seems that the result is rather solid, and I will rewrite as a paper and try to make it published.

五、參考文獻

1. Abhyankar, A., Copeland, L. & Wong, W. (1997). Uncovering nonlinear structure in real-time stock-market indexes: The S&P 500, the DAX, the Nikkei 225, and the FTSE-100. *Journal of Business & Economic Statistics*, 15, 1-14.
2. Adachi, M., Aihara, K. & Kotani, M. (1992). Learning Strange Attractors by Back-Propagation Neural Networks. *Proceedings of IJCNN*, Beijing, II, 569-574.
3. Azoff, E. (1994). *Neural Network Time-series Forecasting of Financial Markets*. John Wiley & Sons, West Sussex, England.
4. Barron, A. R. (1991). Complexity regularization with application to artificial neural networks. In G. Roussas (Eds.), *Nonparametric Functional Estimation and Related Topics* (pp. 561-576).
5. Barron, A. R. (1992). Neural net approximation. *Proceedings of the Seventh Yale Workshop on Adaptive and Learning Systems*, 69-72.
6. Blank, S. (1991). Chaos in futures market? A nonlinear dynamical analysis. *J. Futures Markets*, 11, 711–728.
7. DeCoster, G., Labys, W., & Mitchell, D. (1992). Evidence of chaos in commodity futures prices. *J. Futures Markets*, 12, 291–305.
8. Devaney, R. (1989). *An Introduction to Chaotic Dynamical Systems*. 2nd edition, Addison-Wesley, Inc.
9. Grudnitski, G., & Osburn, L. (1993). Forecasting S&P and gold futures prices: an application of neural networks. *J. Futures Markets*, 13, 631–643.
10. Hornik, K., Stinchcombe, M. and White, H. (1989). Multi-layer feedforward networks are universal approximators. *Neural Networks*, 2, 359-366.
11. Hutchinson, J., Lo, A., & Poggio, T. (1994). A nonparametric approach to pricing and hedging derivative securities via learning networks. *J. Finance*, 49 (3), 851- 889.
12. Kaplan, D., and Glass, L. (1995a). *Understanding Nonlinear Dynamics*. (pp. 27-28) New York: Springer-Verlag.
13. Kaplan, D., and Glass, L. (1995b). *Understanding Nonlinear Dynamics*. (pp. 29-31) New York: Springer-Verlag.
14. Kaplan, D., and Glass, L. (1995c). *Understanding Nonlinear Dynamics*. (pp. 314-318) New York: Springer-Verlag.
15. Kaplan, D., and Glass, L. (1995d). *Understanding Nonlinear Dynamics*. (pp. 324-327) New York: Springer-Verlag.
16. Ott, E., Sauer, T. and Yorke, J. (1994). *Coping With Chaos – Analysis of Chaotic Data and the Exploitation of Chaotic Systems*. 1st edition, Wiley Inc.
17. Matsuba I., Masui H. & Hebishima S. (1992). Prediction of Chaotic Time-Series Data using Optimized Neural Networks. *Proceedings of IJCNN*, vol. I, pp. 340-345, Beijing.
18. Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological Review*, 65, 386-408.
19. Rumelhart, D.E., Hinton, G.E., and

- Williams, R. (1986). Learning internal representation by error propagation. *Parallel Distributed Processing*, Vol. 1 (pp. 318-362). Cambridge, MA: MIT Press.
20. Tsaih, R., Hsu, Y., & Lai, C. (1998). Forecasting S&P 500 Stock Index Futures with the Hybrid AI system. *Decision Support Systems*, Vol. 23, No. 2, pp. 161-174.
21. Wong, F.S. (1991). Time-series Forecasting Using Back-Propagation Neural Networks. *Neurocomputing*, 2, pp. 147-159

