

Cascade-Correlation Algorithm with Trainable Activation Functions

Fanjun Li (Corresponding author)

School of Mathematics, University of Jinan
106 Jiwei Road, Jinan 250022, Shandong, China
E-mail: ss_lifj@ujn.edu.cn

Ying Li

School of Science, Shandong Polytechnic University
Daxue Road, Jinan 250353, Shandong, China
E-mail: lzhbaby@sina.com.cn

Received: September 16, 2011
doi:10.5539/cis.v4n6p28

Accepted: October 17, 2011

Published: November 1, 2011

URL: <http://dx.doi.org/10.5539/cis.v4n6p28>

The research is financed by Natural Science Foundation of University of Jinan (XKY1030)

Abstract

According to the characteristic that higher order derivatives of some base functions can be expressed by primitive functions and lower order derivatives, cascade-correlation algorithm with tunable activation functions is proposed in this paper. The base functions and its higher order derivatives are used to construct the tunable activation functions in cascade-correlation algorithm. The parallel and series constructing schemes of the activation functions are introduced. The model can simply the neural network architecture, speed up the convergence rate and improve its generalization. The efficiency is demonstrated with the two-spiral classification and Mackay-Glass time series prediction problem.

Keywords: Neural networks, Cascade-correlation, Tunable activation functions, Generalization

1. Introduction

Generalization is a critical capacity for feedforward neural network. It is influenced by many factors (Simon Haykin, 1999; WEI Hai-Kun, 2001; Leonardo Franco, 2005). In the last number of years, many researchers have been making efforts to promote the generalization ability of neural networks. Two popular techniques for avoiding the over fitting are the regularization (Burden F, & Winkler D, 2008; R. Felix, 2011) and early stopping methods (Xing-xing Wu, 2009). To get the smallest system that will fit the data, pruning algorithms (Reed R. 1993; Md.Shahjahan, 2003) are widespread used. Pravin Chandra, YogeshSingh have experimentally verified the conjectures that imply the dependence of learning rate of FFANNs on activation functions used at the hidden units (P. Chandra, 2004). Gao Daqi, Yang Genxinga found that the learning abilities of multilayer feed forward neural networks depend on the types of activation functions (Gao Daqi, & Yang Genxing, 2003). ShuXing, Xu, Ming Zhang revealed that feedforward neural networks with the proposed neuron-adaptive activation function present several advantages over traditional neuron-fixed feedforward networks (ShuXing Xu, 2000). So professors gave a lot of new activation functions in the last years (M. Solazzi, 2004; Chine-Cheng Yu, 2002). Wu and Zhao addressed a new kind of neural model, which has trainable activation function (TAF) (WU You-Shou, & ZHAO Ming-Sheng, 2001). They incorporated certain degrees of freedom in the activation function. And in that paper, a special feasible domain of TAF was given, whereas the TAF model needs an effective and fast learning algorithm. So Shen and Wang presented a new multi-output neural model with tunable activation function (YANJUN SHEN, & BINGWEN WANG, 2004). The new model simplifies the neural network architecture, improves its accuracy and speeds up the convergence rate. Simulation results show that it has better capability and performance than the traditional multilayer feedforward neural network and the feed forward neural network with tunable activation functions.

Slow convergence is another factor which induces poor generalization of ANN, Back-Propagation (BP) for example. MD. ASADUZZAMAN and MD. SHAHJAHAN proposed "Fusion of Activation Functions" (FAF) in

which different conventional activation functions (AFs) are combined to compute final activation to make the learning faster (MD. ASADUZZAMAN, & MD. SHAHJAHAN, 2009). In 1989, Scott E. Fahlman and Christian Lebiere interpreted the reasons why the Back-Propagation learns so slowly, and put forward a new architecture and supervised learning algorithm for neural networks called Cascade-Correlation architecture (CC) (S. E. Fahlman, & C. Lebiere, 1989). In the architecture, hidden units are added to the network one at a time and do not change after they have been added. It learns very quickly and determines its own size and topology. But the algorithm has poor generalization for functional approximation. So Qun XU combined Cascade-Correlation algorithm with regularization theory (Qun XU, & Kenji NAKAYAMA, 1997), and got better generalization, especially for functional approximation. These ideas bring us new inspiration. Basted on the characteristic about sigmoid activation function that higher order derivative can be deduced quickly and accurately by primary function without any analysis and operation, we constructed a new TAF model. Combining both traditional CC algorithm and new TAF model, we got a fresh multilayer feed forward neural network, called Cascade-Correlation algorithm with trainable activation function (CCTAF). In fact, the model simplifies the neural network architecture, speeds up the convergence rate and improves its generalization, particularly for functional approximation. The parallel and series constructing schemes of the activation functions are introduced in this paper.

This paper is organized as follows. In Section 2 we discuss the CC algorithm briefly. In Section 3 we give readers a brief introduction about TAF model. We elaborate the theory of CCTAF and realization algorithm in Section 4. Section 5 present two simulations of CCTAF. Section 6 concludes this paper.

2. Theory of CC algorithm

The Cascade Correlation algorithm was proposed by Fahlman & Lebiere. The architecture begins with all inputs and one or more output units without hidden ones. Every input is connected to every output unit by a connection with an adjustable weight. Then connections are trained by the Widrow-Hoff or “delta” rule etc for smaller training error (2.1 for example). E is defined as

$$E = \frac{1}{2} \sum_{o,p} (y_{op} - t_{op})^2 \quad (2.1)$$

Where y_{op} is the network output, while t_{op} is the expected output for pattern p . When the value of the error stops to change and does not meet the above requirements, a new unit called a candidate node is added to the network which receives a connection from each of the network’s original inputs and also from every pre-existing hidden unit. The aim of updating the weights, connected to the adding hidden unit by gradient ascent, is to maximize the correlation magnitude (2.2) between the output of the candidate node and network output error. S is defined as

$$S = \sum_o \left| \sum_p (V_p - \bar{V})(E_{p,o} - \bar{E}_o) \right| \quad (2.2)$$

Where o is the network output at which the error is measured and p is the training pattern. The quantities \bar{V} and \bar{E}_o are the values of V and E_o averaged over all patterns. The hidden unit’s input weights are frozen at the time the increment of the correlation is smaller than expected value. Then the output of the added node together with earlier inputs is connected to the output units, only the output connections are trained repeatedly. The cascade architecture with sigmoid functions is illustrated in Figure 1.

Since only one layer is trained at each training stage, CC requires no back-propagation of error signals, therefore learns very quickly, meanwhile a network is generated automatically without defining beforehand. In fact, CC has a tendency to construct a complexity network with some invalid nodes and has poor generalization for functional approximation.

3. TAF model

The TAF model was put forward in 2001 (WU You-Shou, & ZHAO Ming-Sheng, 2001). It is different from other neural networks for the tunable activation functions. Its general form is given by (3.1)

$$O = F(X, W, a) \quad (3.1)$$

Where O is the output of TAF neuron, X is the inputs signal vector, W is a weight vector, and a is a tunable parameter vector used to control the functions of the TAF model. In fact, the $F(\cdot)$ can be defined two functions

as shown in Figure 2. The function of $g(X, W)$ (called synapse function) is in charge of collecting input signal, and generating inner stimulation. The soma function ($f(S, a)$) with tunable parameters transform the inner stimulation to the output of the TAF nonlinearly. It is the parameters that enable the soma function to be adaptive to various questions.

In that paper, the function of $g(X, W)$ has three forms. And three rules were given for selecting feasible $f(\cdot)$. Meanwhile, an approach of selecting TAF came up with. It is shown as follows:

$$f(X, W, a) = \sum_{m=1}^M a_m \varphi_m(X, W) \quad (3.2)$$

Where $a_m (m = 1, 2, \dots, M)$ are all constants and tunable, $\varphi_m(X, W)$ is the m th basis function. Seven forms of $\varphi(\cdot)$ were formulated.

Compared to the traditional multilayer feed forward neural network, the TAF model can deal with many difficult problems easily. And it can simplify network's architecture. But lacking effective and fast learning algorithms influences its generalization.

4. Cascade-Correlation algorithm with trainable activation functions

A characteristic of some functions was found that their higher derivatives can be expressed by themselves, such as sigmoid function, Gauss function, and exponential function. We called it hereditability. The following is illustrated by the case of sigmoid. The definition of the i th derivative of $f(\cdot)$ is $f_i(\cdot)$

$$f_0(u) = \frac{1}{1 + e^{-\lambda u}} \quad (4.1)$$

$$f_1(u) = \frac{d}{du} f_0(u) = f_0(u)(1 - f_0(u)) \quad (4.2)$$

$$f_2(u) = \frac{d}{du} f_1(u) = f_1(u)(1 - 2f_0(u)) \quad (4.3)$$

$$f_3(u) = \frac{d}{du} f_2(u) = f_1(u)(1 - 6f_1(u)) \quad (4.4)$$

They can be normalized to $[0, 1]$ for easily being observed, as shown in Figure 3 (b). Two models will be formulated: parallel connection and series connection.

The sigmoid function and its normalized derivatives are parallel operation. We install the candidate whose correlation score is the best. The function of hidden unit is defined as:

$$F(u) = \varphi_i \quad S_i = \max\{S_0, S_1, S_2, S_3\} \quad (4.5)$$

Where S_i is the correlation of E and φ_i . Parallel connection can reduce the chance that useless units are installed permanently, and accelerate the training.

Series connection need work out the weighted sum of sigmoid function and its normalized derivative, and the weight coefficients are tunable. Presented as follows:

$$F(u) = \sum_i a_i \varphi_i \quad (4.6)$$

Where a_i is the tunable coefficient. For maximizing the correlation, we need calculate $\frac{\partial S}{\partial w_i}$ and $\frac{\partial S}{\partial a_i}$ by

$$\frac{\partial S}{\partial w_i} = \sum_{p,o} \sigma_o(E_{p,o} - \bar{E}_o) \left(\sum_i a_i \varphi_{i+1} \right) I_{i,p} \quad (4.7)$$

$$\frac{\partial S}{\partial a_i} = \sum_{p,o} \sigma_o(E_{p,o} - \bar{E}_o) \varphi_i I_{i,p} \quad (4.8)$$

Where σ_o is the plus or minus of the magnitude in the S , I_{ip} is the i th input of the candidate unit when pattern p is imported. W and a are adjusted by Quick-Propagation. The processes of CCTAF are listed below:

Step 1. Train the initial net without hidden units until the mean square error reaches a minimum or invariability. If the performance is dissatisfied, turn to step 2;

- Step 2.** Install a hidden candidate node with TAF. Connect it to initial inputs and outputs of all the hidden units.
- Step 3.** Updating all weights connected to the candidate, and adjusting the tunable coefficient. When the correlation is maximized, we add the hidden candidate unit to the network and freeze its weights.
- Step 4.** Output of the added node together with earlier inputs is connected to the network output unit.
- Step 5.** Train the weights until E is smaller than expected value or invariability. Stop the whole process if the performance is satisfied, otherwise turn to step 2;
- So the hidden units are added one by one until the performance reaches the stopping criterion.

5. Simulations results

Two Simulations were conducted to evaluate the generalization of the CCTAF in the section. The Two-Spiral problem verifies the pattern classification of the CCTAF. Mackay-Glass time series prediction problem was used to illustrate the property of function approximation ability.

5.1 The Two-Spiral Problem

In this experiment, we adopt two-spiral problem because it is extremely challenging. It has become a common bench mark for neural network after proposed by Alexis Wieland firstly. The problem is to separate two classes of patterns in two intertwined spirals which cannot be linearly separated. Wu reported obtaining a solution with a 2-10-1 TAF model after 5794 epoch (WU You-Shou, & ZHAO Ming-Sheng, 2001). Fahlman solved the problem with the Cascade-Correlation algorithm using a sigmoidal activation function for both the output and hidden units and a pool of 8 candidate units (S. E.Fahlman, & C. Lebiere, 1989). The number of hidden units varied from 12 to 19, with an average of 15.2.

The training set consists of 194 X-Y values, half of which are to produce a +1 output and half a 0 output. And 19600 pots in $[-7,7] \times [-7,7]$ compose the test samples. We use CCTAF with sigmoidal activation function and normalized derivative (series connection) and a pool of 8 candidate units. We ran the problem 100 times successfully. The number of hidden units varied from 11 to 15, with an average of 12.9. And we got better generalization as Figure 4 (b). It follow from that CCTAF has the ideal ability of learning and generalization for pattern classification.

5.2 Mackey–Glass Chaotic Time Series Prediction

Mackey–Glass Chaotic Time Series is a complex nonlinear dynamical time series, firstly investigated by Mackey and Glass, and difficult to approximate for many algorithms. It is recognized as a benchmark problem that has been used and reported by a number of researchers for comparing the learning and generalization ability of different models (Daijin Kim, & Chulhyun Kim, 1997; B. Samanta, 2011; Yusuf Oysal, 2005). The Mackey–Glass chaotic time series generated by the following chaotic differential delay equation:

$$\frac{dx(t)}{dt} = \frac{ax(t-\tau)}{1+x^n(t-\tau)} + bx(t) \quad (5.1)$$

Where a, b, τ, n are real numbers, and $x(t)$ is the value of time-series at time t . Depending on the values of the parameters, this equation displays a range of periodic and chaotic dynamics. In this simulation we define the parameters of the equation as $n=10, a=0.2, b=-0.1$ and $\tau=17$ according to earlier work fair. Ten thousands data points are generated with an initial condition $x(0)=1.2$ based on the fourth-order Runge–Kutta method with time step 0.1. 10000 data points were shown as Figure 5. By the following formats $[x(t-24), x(t-18), x(t-12), x(t-6), x(t)]$, we generated 500 data pairs of $(201 < t \leq 700)$ for training, and 500 data pairs $(1001 < t \leq 1500)$ for testing. Because the Testing sample is farther away from training, the prediction became more difficult. Neural network with CCTAF, including parallel connection and series connection, was used in the simulations. In parallel connection model, we used a pool of candidate consist of sigmoid function, normalized first and second derivative. We used sigmoid function, first and third normalized derivative in series connection model. The generalization of CCTAF is better than some other models as shown in table 1. According to the table, we found that CCTAF model has better generalization for function approximation.

6. Conclusion

In this paper, CCTAF was addressed based on traditional CC and TAF model including parallel connection and series connection. Sigmoid function has a characteristic that it can express its higher order derivative. According to that, we constructed TAF by sigmoid and its higher order derivatives for faster learning. Then the TAF was applied within the Cascade-Correlation architecture as activation function. It was evidently shown that CCTAF

model can solve problem with fewer hidden units and faster learning. The generalization of CCTAF when applied to the two-spiral and Mackey–Glass Chaotic Time Series Prediction is better compared with other some algorithms.

References

- B. Samanta. (2011). Prediction of chaotic time series using computational intelligence. *Expert Systems with Applications*, 38, 11406–11411. <http://dx.doi.org/10.1016/j.eswa.2011.03.013>
- Burden F, & Winkler D. (2008). Bayesian regularization of neural networks. *Methods Mol Biol*, 458, 23-42. http://dx.dox.org/10.1007/978-1-60327-101-1_3
- Chine-Cheng Yu, Yun-Ching Tang, & Bin-DaLin. (2002). An Adaptive Activation Function for Multilayer Feedforward Neural Networks. Proceeding of IEEE TENCON'02, 645-650. <http://dx.doi.org/10.1109/TENCON.2002.1181357>
- Daijin Kim & Chulhyun Kim. (1997). Forecasting Time Series with Genetic Fuzzy Predictor Ensemble. *IEEE TRANSACTIONS ON FUZZY SYSTEMS*, 5(4).
- Gao Daqi, & Yang Genxing. (2003). Influence of Variable Scales and Activation Function on the Performance of multiplayer Feed forward Neural Network. *Pattern Recognition*, 36, 869-878. [http://dx.doi.org/10.1016/S0031-3203\(02\)00120-6](http://dx.doi.org/10.1016/S0031-3203(02)00120-6)
- Leonardo Franco, Jos'e M. Jerez-Aragon'es, & Jos'eM. Bravo Montoya. (2005). Role of Function Complexity and Network Size in the Generalization Ability of Feedforward Networks. Proceedings: IWANN. LNCS 3512. 1–8. http://dx.dox.org/10.1007/11494669_1
- M. Solazzi, & A. Uncini. (2004). Regularizing neural networks using flexible multivariate activation function. *Neural Networks*, 17, 247-260. [http://dx.doi.org/10.1016/S0893-6080\(03\)00189-8](http://dx.doi.org/10.1016/S0893-6080(03)00189-8)
- MD. ASADUZZAMAN, & MD. SHAHJAHAN. (2009). Faster Training Using Fusion of Activation Functions for Feed Forward Neural Networks. *International Journal of Neural Systems*, 19(6), 437–448. <http://dx.doi.org/10.1142/S0129065709002130>
- Md. Shahjahan, M.A.H.Akhand, & K.Murase. (2003). A Pruning Algorithm for Training Neural Network Ensembles. SICE Annual Conference in Fukui August 4-6, 628-633.
- P. Chandra, & Y. Singh. (2004). A Case for the Self-adaptation of activation function in FFANNS. *Neural Computing*, 56, 447-454. <http://dx.dox.org/10.1016/j.neucom.2003.08.005>
- Qun XU, & Kenji NAKAYAMA. (1997). Avoiding Weight-illgrowth: Cascade Correlation Algorithm with Local Regularization. *Neural networks international conference on*
- R. Felix Reinhart, & Jochen J. Steil. (2011). A constrained regularization approach for input-driven recurrent neural networks. *Differential Equations and Dynamical Systems*, 19(1), 27-46. <http://dx.dox.org/10.1007/s12591-010-0067-x>.
- Reed R. (1993). Pruning algorithms: A survey. *IEEE Trans. Neural Networks*, 4, 740-747. <http://dx.dox.org/10.1109/72.248452>
- S. E.Fahlman, & C. Lebiere. (1989). The Cascade-Correlation Learning Architecture. *Neural Information Processing Systems*, 524-532.
- ShuXing Xu, & Ming Zhang. (2000). Justification of a Neuron-Adaptive Activation Function. *IJCNN Proceedings*, 3, 465-470. <http://dx.doi.org/10.1109/IJCNN.2000.861351>
- Simon Haykin. (1999). *Neural Networks: A Comprehensive Foundation*. (2nd ed.). Tom Robbins – Prentice Hall, Inc. 205-208.
- WEI Hai-Kun, XU Si-Xi, & SONG Wen-Zhong. (2001). Generalization Theory and Generalization methods for Neural Networks. *ACTA AUTOMATICA SINICA*, 27(6), 806-815.
- WU You-Shou, & ZHAO Ming-Sheng. (2001). A neural model with trainable activation function and its MFNN supervised learning. *SINCE IN CHINA (series F)*, 44 (5), 366-375. <http://dx.doi.org/10.1007/BF02714739>
- Xing-xing Wu, & Jin-guo Liu. (2009). A New Early Stopping Algorithm for Improving Neural Network Generalization. *Intelligent Computation Technology and Automation*, 1, 15-18. <http://doi.ieeecomputersociety.org/10.1109/ICICTA.2009.11>
- YANJUN SHEN, & BINGWEN WANG. (2004). A New Multi-output Neural Model with Tunable Activation

Function and its Applications. *Neural Processing Letters* 20, 85–104. <http://dx.doi.org/10.1007/s11063-004-0637-4>

Yusuf Oysal. (2005). Time Delay Dynamic Fuzzy Networks for Time Series Prediction. *Lecture Notes in Computer Science*. LNCS 3514, 775-782. http://dx.doi.org/10.1007/11428831_96

Table 1. Comparison Results

Method	Traing	Prediction Error (RMSE)
CCTAF(parallel connection)	500	0.0106
CCTAF(series connection)	500	0.0080
GFPE (coarse Partition)	500	0.038011 (9 partition)
GFPE (Fine-Tuning)	500	0.037873 (9 partition)
GFPE (Ensemble)	500	0.026431
Lee&Kim	500	0.0816
Wang (product operator)	500	0.0907
Min operator	500	0.0904
ANFIS(adaptive-network-based fuzzy inference system)	500	0.007
Auto Regressive Model	500	0.19
Cascade-Correlation Neural Network	500	0.06
Back-Prop Neural Network	500	0.02
6th-order Polynominal	500	0.04
Linear Predictive Method	500	0.55

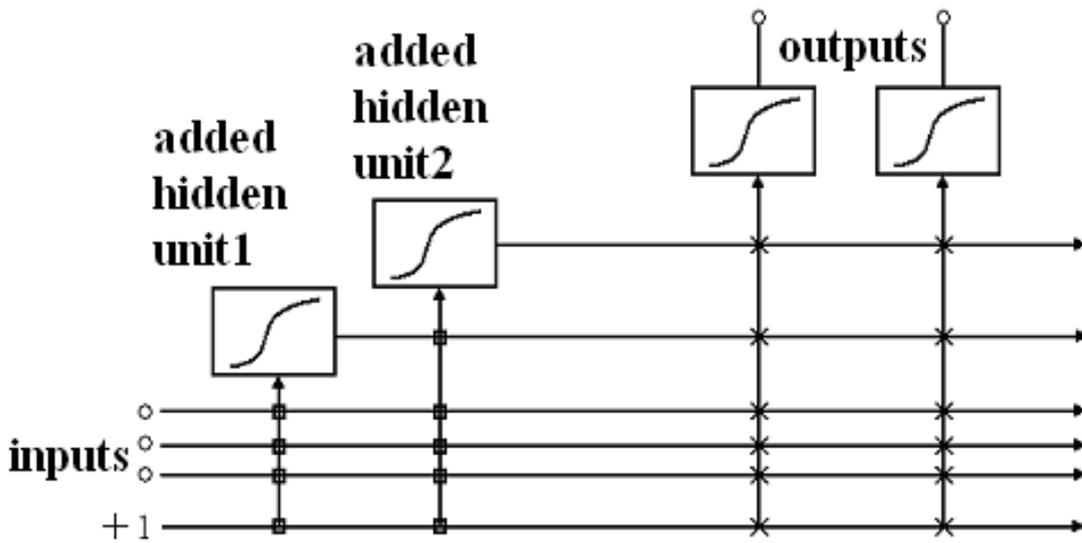


Figure 1. The Cascade architecture with two outputs and two added hidden units

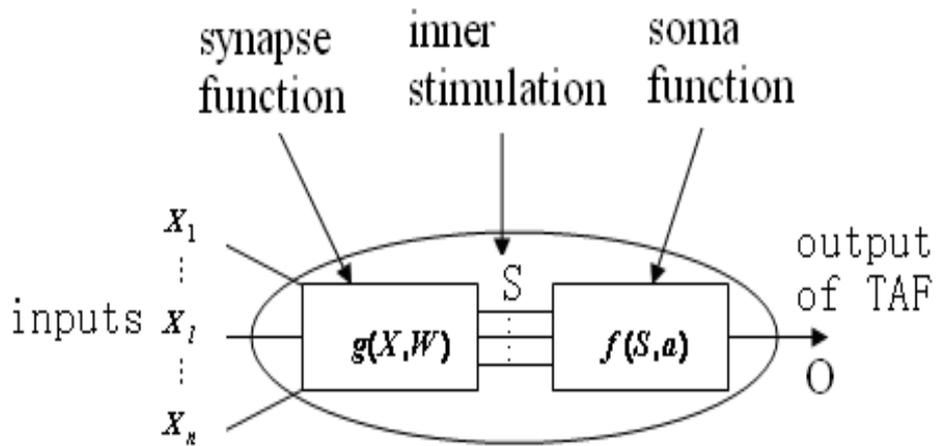


Figure 2. The architecture of general TAF neuron model

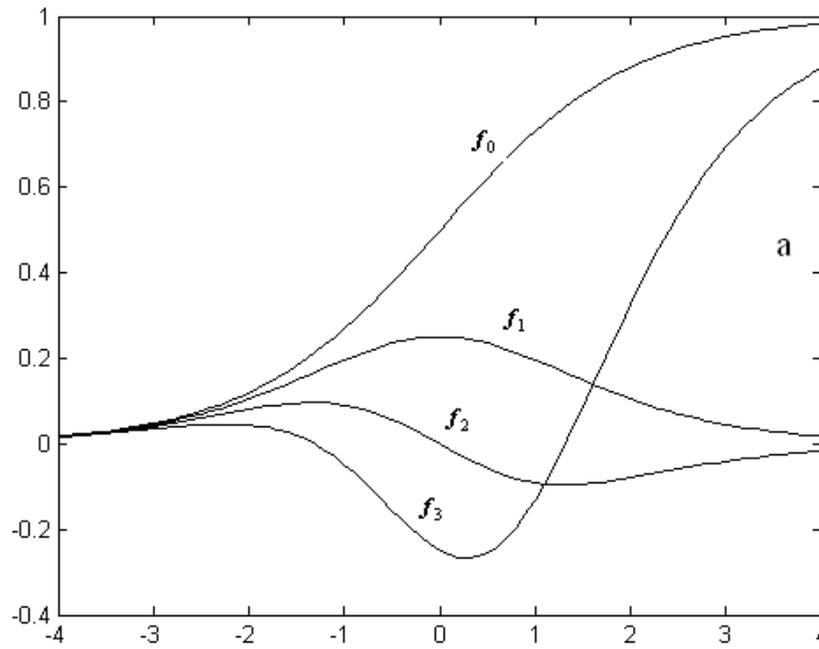


Figure 3. (a) Sigmoid function and its derivative

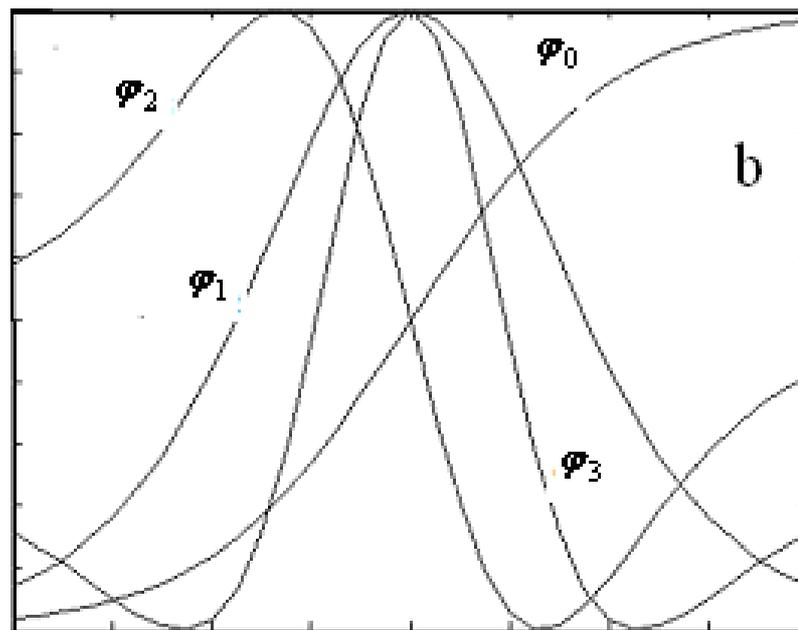


Figure 3. (b) Sigmoid function and its normalized derivative

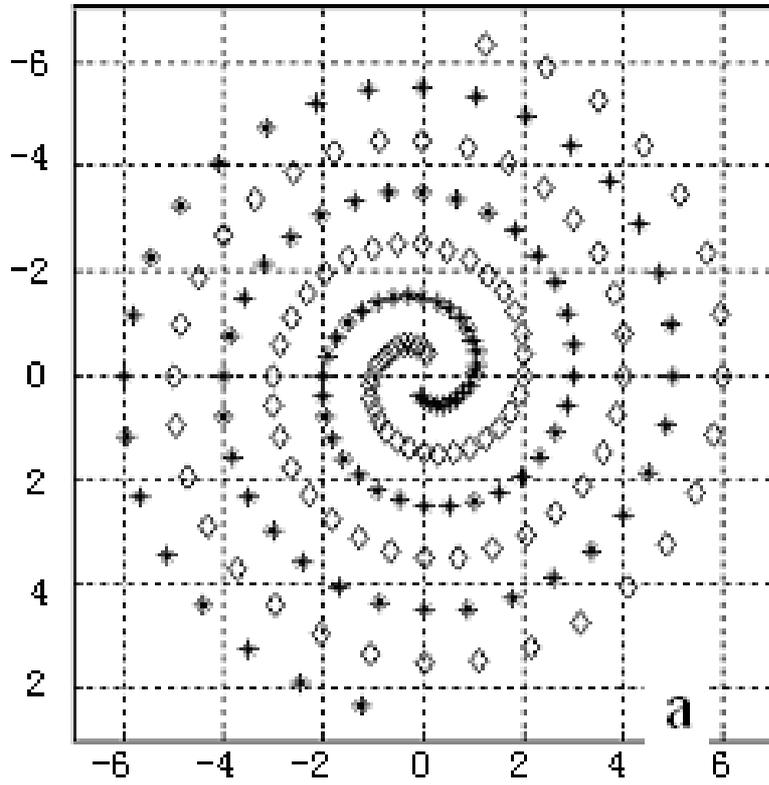


Figure 4. (a) Training sample

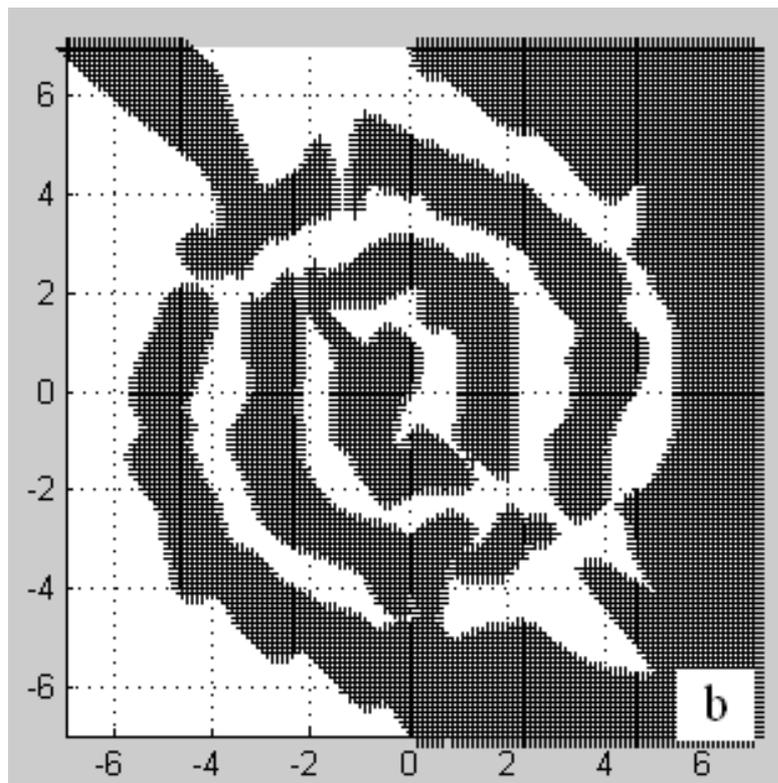


Figure 4. (b) Test result

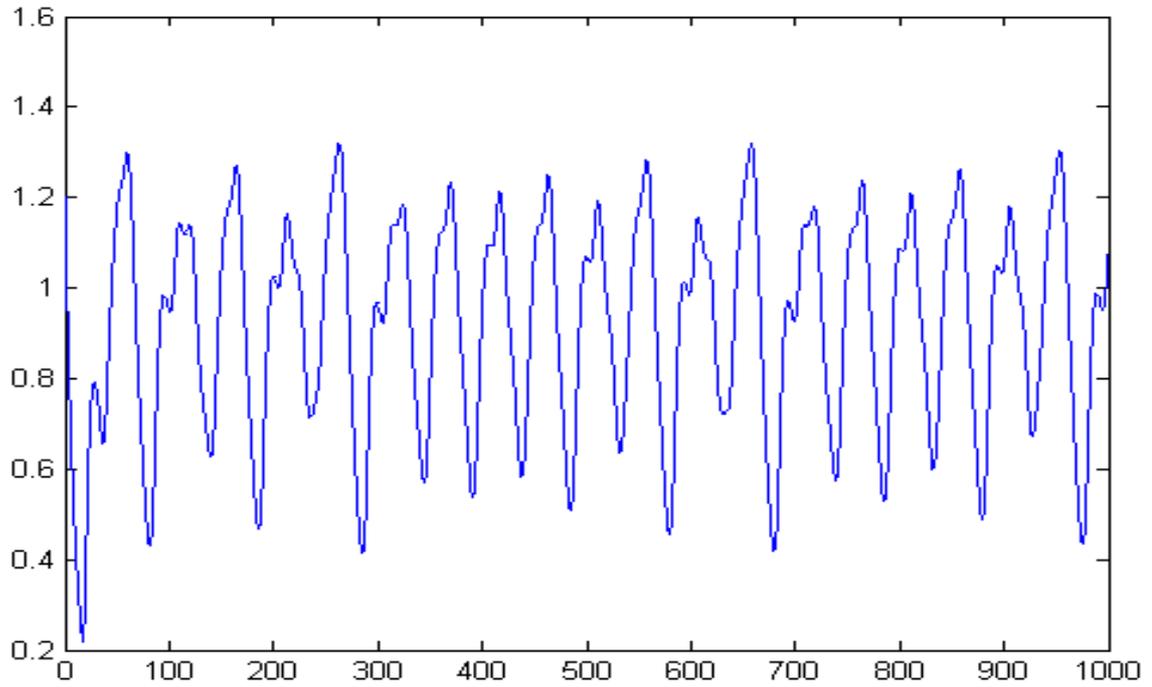


Figure 5. Ten thousand sample points of a chaotic Mackey–Glass time series