Hindawi Publishing Corporation
Discrete Dynamics in Nature and Society
Volume 2014, Article ID 193758, 6 pages
http://dx.doi.org/10.1155/2014/193758

Research Article

Levenberg-Marquardt Algorithm for Mackey-Glass Chaotic Time Series Prediction

Junsheng Zhao,[1] Yongmin Li,[2,3] Xingjiang Yu,[1] and Xingfang Zhang[1]

[1] *School of Mathematics, Liaocheng University, Liaocheng 252059, China*
[2] *School of Science, Huzhou University, Huzhou 313000, China*
[3] *School of Automation, Southeast University, Nanjing 210096, China*

Correspondence should be addressed to Junsheng Zhao; zhaojunshshao@163.com

Received 9 August 2014; Accepted 11 October 2014; Published 11 November 2014

Academic Editor: Rongni Yang

For decades, Mackey-Glass chaotic time series prediction has attracted more and more attention. When the multilayer perceptron is used to predict the Mackey-Glass chaotic time series, what we should do is to minimize the loss function. As is well known, the convergence speed of the loss function is rapid in the beginning of the learning process, while the convergence speed is very slow when the parameter is near to the minimum point. In order to overcome these problems, we introduce the Levenberg-Marquardt algorithm (LMA). Firstly, a rough introduction is given to the multilayer perceptron, including the structure and the model approximation method. Secondly, we introduce the LMA and discuss how to implement the LMA. Lastly, an illustrative example is carried out to show the prediction efficiency of the LMA. Simulations show that the LMA can give more accurate prediction than the gradient descent method.

1. Introduction

The Mackey-Glass chaotic time series is generated by the following nonlinear time delay differential equation:

$$\frac{dx(t)}{dt} = \frac{\beta x(t-\tau)}{1 + x^n(t-\tau)} + \gamma x(t), \tag{1}$$

where β, γ, τ, and n are real numbers. Depending on the values of the parameters, this equation displays a range of periodic and chaotic dynamics. Such a series has some short-range time coherence, but long-term prediction is very difficult.

Originally, Mackey and Glass proposed the following equation to illustrate the appearance of complex dynamics in physiological control systems by way of bifurcations in the dynamics:

$$\frac{dx}{dt} = \frac{\beta x_\tau}{1 + x_\tau^n} - \gamma x, \quad \gamma, \beta, n > 0. \tag{2}$$

They suggested that many physiological disorders, called dynamical diseases, were characterized by changes in qualitative features of dynamics. The qualitative changes of physiological dynamics corresponded mathematically to bifurcations in the dynamics of the system. The bifurcations in the equation dynamics could be induced by changes in the parameters of the system, as might arise from disease or environmental factors, such as drugs or changes in the structure of the system [1, 2].

The Mackey-Glass equation has also had an impact on more rigorous mathematical studies of delay-differential equations. Methods for analysis of some of the properties of delay differential equations, such as the existence of solutions and stability of equilibria and periodic solutions, had already been developed [3]. However, the existence of chaotic dynamics in delay-differential equations was unknown. Subsequent studies of delay differential equations with monotonic feedback have provided significant insight into the conditions needed for oscillation and properties of oscillations [4–6]. For delay differential equations with nonmonotonic feedback, mathematical analysis has proven

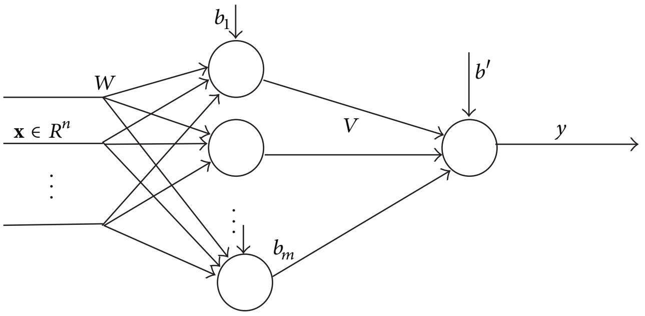


Figure 1: Multilayer Perceptrons.

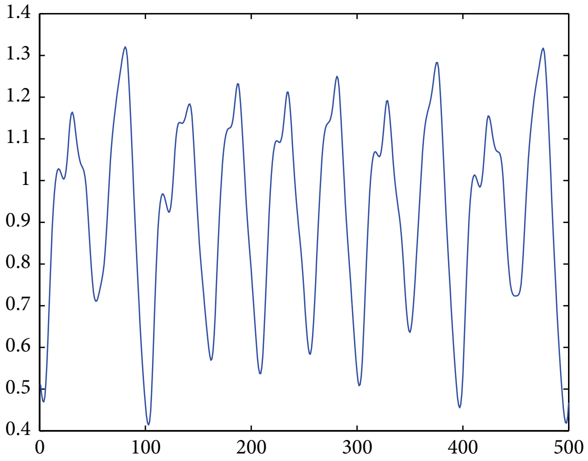


Figure 2: Mackey-Glass chaotic time series.

much more difficult. However, rigorous proofs for chaotic dynamics have been obtained for the differential delay equation $dx/dt = g(x(t-1))$ for special classes of the feedback function g [7]. Further, although a proof of chaotic dynamics in the Mackey-Glass equation has still not been found, advances in understanding the properties of delay differential equations is going on, such as (2), that contain both exponential decay and nonmonotonic delayed feedback [8]. The study of this equation remains a topic of vigorous research.

The Mackey-Glass chaotic time series prediction is a very difficult task. The aim is to predict the future state $x(t+\Delta T)$ using the current and the past time series $x(t)$, $x(t-1), \ldots, x(t-n)$ (Figure 2). Until now, there are many literatures about the Mackey-Glass chaotic time series prediction [9–14]. However, as far as the prediction accuracy is concerned, most of the results in the literature are not ideal.

In this paper, we will predict the Mackey-Glass chaotic time series by the MLP. While minimizing the loss function, we introduce the LMA, which can adjust the convergence speed and obtain good convergence efficiency.

The rest of the paper is organized as follows. In Section 2, we describe the multilayer perceptron. Section 3 introduces the LMA and discusses how to implement the LMA. In Section 4, we give a numerical example to demonstrate the prediction efficiency. Section 5 is the conclusions and discussions of the paper.

2. Preliminaries

2.1. Multilayer Perceptrons. A multilayer perceptron (MLP) is a feedforward artificial neural network model that maps sets of input data onto a set of appropriate outputs. A MLP consists of multiple layers of nodes in a directed graph, with each layer fully connected to the next one. Except for the input nodes, each node is a neuron (or processing element) with a nonlinear activation function. The multilayer perceptron with only one hidden layer is depicted as in Figure 1 [15].

In Figure 1, $\mathbf{x} = [x_1, x_2, \ldots, x_n]^T \in R^n$ is the model input, y is the model output, $W = \{w_{ij}\}$, $i = 1, 2, \ldots, n$, $j = 1, 2, \ldots, m$, is the connection weight from the x_i to the jth hidden unit, $V = \{v_j\}$ is the connection weight from the jth hidden unit to the output unit, b_j, $j = 1, 2, \ldots, m$, and b' are the bias.

The output of the multilayer perceptron described in Figure 1 is

$$y = \sum_{j=1}^{m} v_j h_j + b', \tag{3}$$

and the outputs of the hidden units are

$$h_j = \varphi\left(\sum_{i=1}^{n} w_{ij} x_i + b_j\right), \tag{4}$$

respectively, where $\varphi(\cdot)$ is the activation function. We will adopt the sigmoid function $\varphi(x)$ as the activation function; for example,

$$\varphi(x) = \frac{1}{1+e^{-x}}, \tag{5}$$

and the derivative of the activation function with respect to x is

$$\varphi'(x) = \frac{e^{-x}}{(1+e^{-x})^2}, \tag{6}$$

or

$$\varphi'(x) = \varphi(x)\left(1-\varphi(x)\right). \tag{7}$$

MLP provides a universal method for function approximation and classification [16, 17]. In the case of the function approximation, we have a number of observed data $(\mathbf{x}_1, y_1)$, $(\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_L, y_L)$, which are supposed to be generated by

$$y = f_0(\mathbf{x}) + \xi, \tag{8}$$

where ξ is noise, usually subject to Gaussian distribution with zero mean, and $f_0(\mathbf{x})$ is the unknown true generating function.

Given a set of observed data, sometimes called training examples, we search for the parameters

$$\theta = \left(b_1, \dots, b_m, w_{11}, \dots, w_{n1}, w_{12}, \dots, w_{n2}, \dots, w_{1m}, \dots, w_{nm}, b', v_1, \dots, v_m\right) \in R^{(n+1)m+m+1} \quad (9)$$

to approximate the teacher function $f_0(\mathbf{x})$ best, where T denotes the matrix transposition. A satisfactory model is often obtained by minimizing the mean square error.

One of the serious problems in minimizing the mean square error is that the convergence speed of the loss function is rapid in the beginning of the learning process, while the convergence speed is very slow in the region of the minimum [18]. In order to overcome these problems, we will introduce the Levenberg-Marquardt algorithm (LMA) in the next section.

2.2. The Levenberg-Marquardt Algorithm. In mathematics and computing, the Levenberg-Marquardt algorithm (LMA) [18–20], also known as the damped least-squares (DLS) method, is used to solve nonlinear least squares problems. These minimization problems arise especially in least squares curve fitting.

The LMA is interpolates between the Gauss-Newton algorithm (GNA) and the gradient descent algorithm (GDA). As far as the robustness is concerned, the LMA performs better than the GNA, which means that in many cases it finds a solution even if it starts very far away from the minimum. However, for well-behaved functions and reasonable starting parameters, the LMA tends to be a bit slower than the GNA.

In many real applications for solving model fitting problems, we often adopt the LMA. However, like many other fitting algorithms, the LMA finds only a local minimum, which is always not the global minimum.

The least squares curve fitting problem is described as follows. Instead of the unknown true model, a set of N pairs of independent variables $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_N, y_N)$ are given. Suppose that $f(x, \theta)$ is the approximation model and $L(\theta)$ is a loss function, which is the sum of the squares of the deviations:

$$L(\theta) = \sum_{i=1}^{N} \left[y_i - f(\mathbf{x}_i, \theta)\right]^2. \quad (10)$$

The task of curve fitting problem is minimizing the above loss function $L(\theta)$ [21].

The LMA is an iterative algorithm and the parameter θ is adjusted in each iteration step. Generally speaking, we choose an initial parameter randomly, for example, $\theta_i \sim U(-1, 1)$, $i = 1, 2, \dots, n$, where n is the dimension of parameter θ.

The Taylor expansion of the function $f(\mathbf{x}_i, \theta + \Delta\theta)$ is

$$f(\mathbf{x}_i, \theta + \Delta\theta) \approx f(\mathbf{x}_i, \theta) + \frac{\partial f(\mathbf{x}_i, \theta)}{\partial\theta}\Delta\theta. \quad (11)$$

As we know, at the minimum θ^* of loss function $L(\theta)$, the gradient of $L(\theta)$ with respect to θ will be zero. Substituting (11) into (10), we can obtain

$$L(\theta + \Delta\theta) \approx \sum_{i=1}^{N} \left(y_i - f(\mathbf{x}_i, \theta) - \mathbf{J}_i\Delta\theta\right)^2, \quad (12)$$

where $J_i = \partial f(\mathbf{x}_i, \theta)/\partial\theta$, or

$$L(\theta + \Delta\theta) \approx \left\|\mathbf{y} - \mathbf{f}(\theta) - \mathbf{J}\Delta\theta\right\|^2. \quad (13)$$

Taking the derivative with respect to $\Delta\theta$ and setting the result to zero give

$$\left(\mathbf{J}^T\mathbf{J}\right)\Delta\theta = \mathbf{J}^T\left(\mathbf{y} - \mathbf{f}(\theta)\right), \quad (14)$$

where $\mathbf{J}$ is the Jacobian matrix whose ith row equals J_i and also $\mathbf{y}$ and $\mathbf{f}$ are vectors with ith component $f(x_i, \theta)$ and y_i, respectively. This is a set of linear equations which can be solved for $\Delta\theta$.

Levenberg's contribution is to replace this equation by a "damped version,"

$$\left(\mathbf{J}^T\mathbf{J} + \lambda I\right)\Delta\theta = \mathbf{J}^T\left[\mathbf{y} - \mathbf{f}(\theta)\right], \quad (15)$$

where I is the identity matrix, giving the increment $\Delta\theta$ to the estimated parameter vector θ.

The damping factor λ is adjusted at each iteration step. If the loss function $L(\theta)$ reduces rapidly, λ will adopt a small value, and then the LMA is similar to the Gauss-Newton algorithm. While the loss function $L(\theta)$ reduces very slowly, λ can be increased, giving a step closer to the gradient descent direction, and

$$\frac{\partial L(\theta)}{\partial\Delta\theta} = -2\mathbf{J}^T\left[\mathbf{y} - \mathbf{f}(\theta)\right]^T. \quad (16)$$

Therefore, for large values of λ, the step will be taken approximately in the direction of the gradient.

In the process of iteration, if either the length of the calculated step $\Delta\theta$ or the reduction of $L(\theta)$ from the latest parameter vector $\theta + \Delta\theta$ falls below the predefined limits, iteration process stops and then we take the last parameter vector θ as the final solution.

Levenberg's algorithm has the disadvantage that if the value of damping factor λ is large, the inverse of $\mathbf{J}^T\mathbf{J} + \lambda I$ does not work at all. Marquardt provided the insight that we can scale each component of the gradient according to the curvature so that there is larger movement along the directions where the gradient is smaller. This avoids slow convergence in the direction of small gradient. Therefore, Marquardt replaced the identity matrix I with the diagonal matrix consisting of the diagonal elements of $\mathbf{J}^T\mathbf{J}$, resulting in the Levenberg-Marquardt algorithm [22]:

$$\left[\mathbf{J}^T\mathbf{J} + \lambda\,\mathrm{diag}\left(\mathbf{J}^T\mathbf{J}\right)\right]\Delta\theta = \mathbf{J}^T\left[\mathbf{y} - \mathbf{f}(\theta)\right], \quad (17)$$

and the LMA is as follows:

$$\theta_{t+1} = \theta_t + \Delta\theta, \quad (18)$$

where

$$\Delta\theta = \left[\mathbf{J}^T\mathbf{J} + \lambda\,\mathrm{diag}\left(\mathbf{J}^T\mathbf{J}\right)\right]^{-1}\mathbf{J}^T\left[\mathbf{y} - \mathbf{f}(\theta)\right]. \quad (19)$$

3. Application of LMA for Mackey-Glass Chaotic Time Series

In this section, we will derive the LMA when the MLP is used for the Mackey-Glass chaotic time series prediction. Suppose that we use the $x(t)-\Delta T_0, x(t-\Delta T_1),\ldots,x(t-\Delta T_{n-1})$ to predict the future variable $x(t+\Delta T)$, where $\Delta T_0=0$.

To implement the LMA, what we should do is calculate the Jacobian matrix $\mathbf{J}$, whose ith row equals J_i. According to (3) and (4), function $f(\mathbf{x},\theta)$ can be expressed as

$$f(\mathbf{x},\theta)=\sum_{j=1}^{m} v_j\varphi\left(\sum_{i=1}^{n} w_{ij}x\left(t-\Delta T_{i-1}\right)+b_j\right)+b'. \tag{20}$$

What we should do is calculate the $J_i=\partial f(\mathbf{x}_i,\theta)/\partial\theta$.

The derivatives of $f(\mathbf{x},\theta)$ with respect to θ are

$$\begin{aligned}
\frac{\partial f(\mathbf{x},\theta)}{\partial b_j} &= v_j\varphi\left(\sum_{i=1}^{n} w_{ij}x\left(t-\Delta T_{i-1}\right)+b_j\right)\\
&\quad\times\left[1-\varphi\left(\sum_{i=1}^{n} w_{ij}x\left(t-\Delta T_{i-1}\right)+b_j\right)\right],\\
\frac{\partial f(\mathbf{x},\theta)}{\partial w_{ij}} &= v_j\varphi\left(\sum_{i=1}^{n} w_{ij}x\left(t-\Delta T_{i-1}\right)+b_j\right)\\
&\quad\times\left[1-\varphi\left(\sum_{i=1}^{n} w_{ij}x\left(t-\Delta T_{i-1}\right)+b_j\right)\right]\\
&\quad\times x\left(t-\Delta T_{i-1}\right),\\
\frac{\partial f(\mathbf{x},\theta)}{\partial b'} &= 1,\\
\frac{\partial f(\mathbf{x},\theta)}{\partial v_j} &= \varphi\left(\sum_{i=1}^{n} w_{ij}x\left(t-\Delta T_{i-1}\right)+b_j\right),\\
i&=1,2,\ldots,n,\quad j=1,2,\ldots,m.
\end{aligned}\tag{21}$$

As we know, $\theta=(b_1,\ldots,b_m,w_{11},\ldots,w_{n1},w_{12},\ldots,w_{n2},\ldots,w_{1m},\ldots,w_{nm},b',v_1,\ldots,v_m)\in R^{(n+1)m+m+1}$, so J_i, the ith row of J, can be easily obtained according to (21). $\mathbf{J}$ is calculated and when MLP is used for Mackey-Glass chaotic time series prediction, the LMA can also be obtained.

4. Numerical Simulations

Example 1. We will conduct an experiment to show the efficiency of the Levenberg-Marquardt algorithm. We choose a chaotic time series created by the Mackey-Glass delay-difference equation:

$$\frac{dx(t)}{dt}=\frac{0.2x(t-\tau)}{1+x^{10}(t-\tau)}-0.1x(t), \tag{22}$$

for $\tau=17$.

Such a series has some short-range time coherence, but long-term prediction is very difficult. The need to predict such a time series arises in detecting arrhythmias in heartbeats.

The network is given no information about the generator of the time series and is asked to predict the future of the time series from a few samples of the history of the time series. In our example, we trained the network to predict the value at time $T+\Delta T$, from inputs at time T, $T-6$, $T-12$, and $T-18$, and we will adopt $\Delta T=50$ here.

In the simulation, 3000 training examples and 500 test examples are generated by (22). We use the following multilayer perceptron for fitting the generated training examples:

$$x(t+50)=\sum_{j=1}^{20} V_j\varphi\left(\sum_{i=1}^{4} x\left(t-6(i-1)\right)w_{ij}+b_j\right)+b'; \tag{23}$$

for example, the number of the hidden units is $n=20$ and the dimension of the input is $m=4$.

Let $f(\mathbf{x},\theta)=\sum_{j=1}^{20} V_j\varphi(\sum_{i=1}^{4} x(t-6(i-1))w_{ij}+b_j)+b'$; then, we can obtain the following equation according to (21):

$$\begin{aligned}
\frac{\partial f(\mathbf{x},\theta)}{\partial b_j} &= v_j\varphi\left(\sum_{i=1}^{20} w_{ij}x\left(t-6\times(i-1)\right)w_{ij}+b_j\right)\\
&\quad\times\left[1-\varphi\left(\sum_{i=1}^{4} w_{ij}x\left(t-6\times(i-1)\right)w_{ij}+b_j\right)\right],\\
\frac{\partial f(\mathbf{x},\theta)}{\partial w_{ij}} &= v_j\varphi\left(\sum_{i=1}^{20} w_{ij}x\left(t-6\times(i-1)\right)+b_j\right)\\
&\quad\times\left[1-\varphi\left(\sum_{i=1}^{4} w_{ij}x\left(t-6\times(i-1)\right)w_{ij}+b_j\right)\right]\\
&\quad\times x\left(t-6\times(i-1)w_{ij}+b_j\right),\\
\frac{\partial f(\mathbf{x},\theta)}{\partial b'} &= 1,\\
\frac{\partial f(\mathbf{x},\theta)}{\partial v_j} &= \varphi\left(\sum_{i=1}^{4} w_{ij}x\left(t-6\times(i-1)\right)w_{ij}+b_j\right),\\
i&=1,2,\ldots,20,\quad j=1,2,3,4.
\end{aligned}\tag{24}$$

The initial values of the parameters are selected randomly:

$$\begin{gathered}
W_{ij}\sim U(2,4),\qquad V_j\sim U(1,2),\\
i=1,2,\ldots,21,\quad j=1,2,3,4,5.
\end{gathered}\tag{25}$$

The learning curves of the error function and the fitting result of LMA and GDA are shown in Figures 3, 4, 5, and 6, respectively.

The learning curves of LMA and GNA are shown in Figures 3 and 4, respectively. The training error $\sum_{i=1}^{3000}(y_i-f(\mathbf{x}_i,\theta))^2$ of LMA can reach 0.1, while the final training error of GDA is more than 90. Furthermore, the final mean test error $(1/500)\sum_{i=5001}^{5500}(y_i-f(\mathbf{x}_i,\theta))^2=0.0118$ of LMA is much smaller than 0.2296, which is the final test error of GDA.

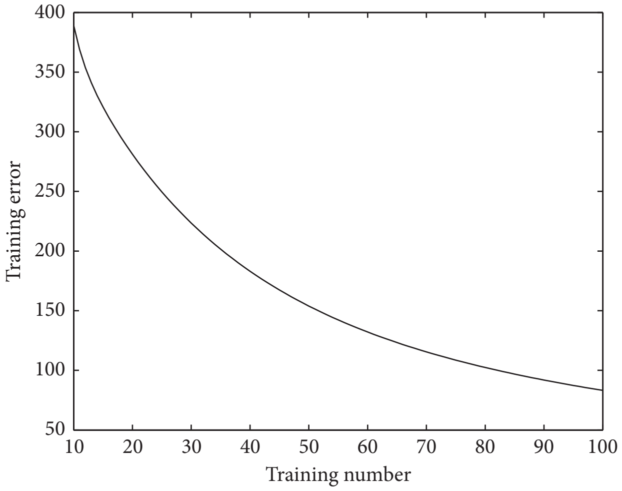


Figure 3: The GDA learning curve of the error function.

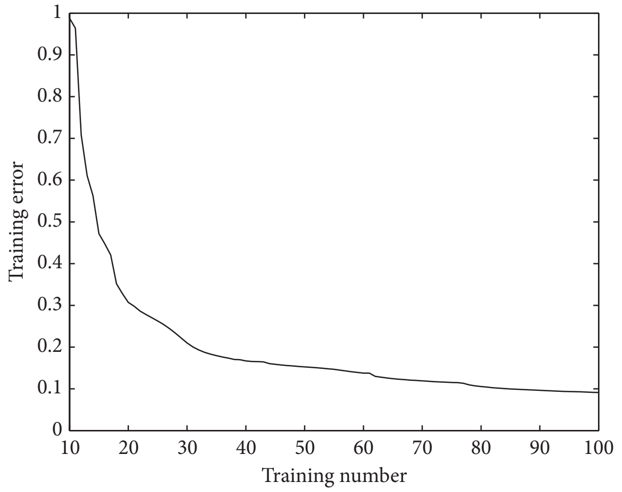


Figure 4: The LMA learning curve of the error function.

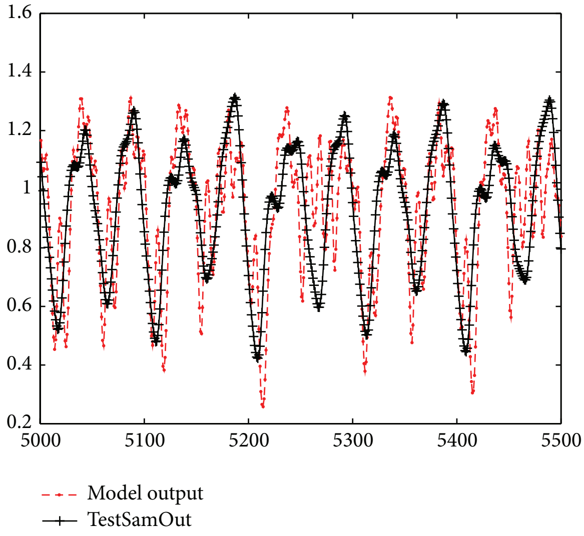


Figure 5: Fitting of Mackey-Glass chaotic time series, GDA.

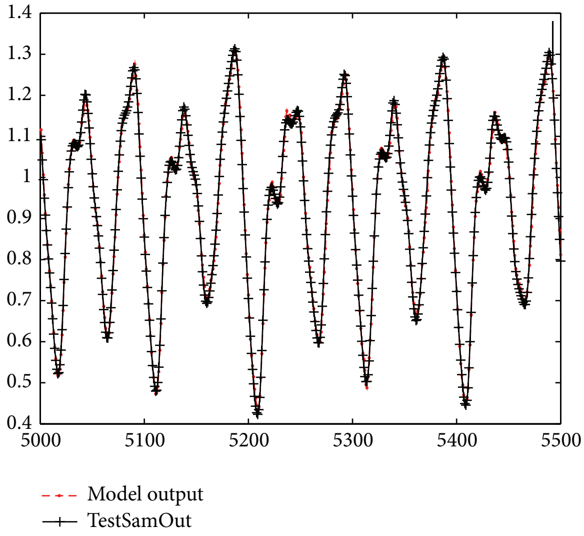


Figure 6: Fitting of Mackey-Glass chaotic time series, LMA.

As far as the fitting effect is concerned, the performance of LMA is much better than that of the GDA. This is very obvious from Figures 5 and 6.

All of these suggest that when we predict the Mackey-Glass chaotic time series, the performance of LMA is very good. It can effectively overcome the difficulties which may arise in the GDA.

5. Conclusions and Discussions

In this paper, we discussed the application of the Levenberg-Marquardt algorithm for the Mackey-Glass chaotic time series prediction. We used the multilayer perceptron with 20 hidden units to approximate and predict the Mackey-Glass chaotic time series. In the process of minimizing the error function, we adopted the Levenberg-Marquardt algorithm. If reduction of $L(\theta)$ is rapid, a smaller value damping factor λ can be used, bringing the algorithm closer to the Gauss-Newton algorithm, whereas if an iteration gives insufficient reduction in the residual, λ can be increased, giving a step closer to the gradient descent direction. In this paper, the learning mode is batch. At last, we demonstrate the performance of the LMA. Simulations show that the LMA can achieve much better prediction efficiency than the gradient descent method.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

This project is supported by the National Natural Science Foundation of China under Grants 61174076, 11471152, and 61403178.

References

[1] M. C. Mackey and L. Glass, "Oscillation and chaos in physiological control systems," *Science*, vol. 197, no. 4300, pp. 287–289, 1977.

[2] L. Glass and M. C. Mackey, "Mackey-Glass equation," *Scholarpedia*, vol. 5, no. 3, p. 6908, 2010.

[3] J. Hale, *Theory of Functional Differential Equations*, Springer, New York, NY, USA, 1977.

[4] J. Mallet-Paret and R. D. Nussbaum, "A differential-delay equation arising in optics and physiology," *SIAM Journal on Mathematical Analysis*, vol. 20, no. 2, pp. 249–292, 1989.

[5] H. O. Walther, "The 2-dimensional attractor of $dx/dt = \mu x(t) + f(x(t-1))$," *Memoirs of the American Mathematical Society*, vol. 113, no. 544, p. 76, 1995.

[6] J. Mallet-Paret and G. R. Sell, "The Poincaré-Bendixson theorem for monotone cyclic feedback systems with delay," *Journal of Differential Equations*, vol. 125, no. 2, pp. 441–489, 1996.

[7] B. Lani-Wayda and H.-O. Walther, "Chaotic motion generated by delayed negative feedback, part II: construction of nonlinearities," *Mathematische Nachrichten*, vol. 180, pp. 181–211, 2000.

[8] G. Röst and J. H. Wu, "Domain-decomposition method for the global dynamics of delay differential equations with unimodal feedback," *Proceedings of The Royal Society of London. Series A. Mathematical, Physical and Engineering Sciences*, vol. 463, no. 2086, pp. 2655–2669, 2007.

[9] M. Awad, H. Pomares, I. Rojas, O. Salameh, and M. Hamdon, "Prediction of time series using RBF neural networks: a new approach of clustering," *International Arab Journal of Information Technology*, vol. 6, no. 2, pp. 138–143, 2009.

[10] M. Awad, "Chaotic time series prediction using wavelet neural network," *Journal of Artificial Intelligence: Theory and Application*, vol. 1, no. 3, pp. 73–80, 2010.

[11] I. López-Yáñez, L. Sheremetov, and C. Yáñez-Márquez, "A novel associative model for time series data mining," *Pattern Recognition Letters*, vol. 41, pp. 23–33, 2014.

[12] A. C. Fowler, "Respiratory control and the onset of periodic breathing," *Mathematical Modelling of Natural Phenomena*, vol. 9, no. 1, pp. 39–57, 2014.

[13] N. Wang, M. J. Er, and M. Han, "Generalized single-hidden layer feedforward networks for regression problems," *IEEE Transactions on Neural Networks and Learning Systems*, 2014.

[14] N. Wang, "A generalized ellipsoidal basis function based online self-constructing fuzzy neural network," *Neural Processing Letters*, vol. 34, no. 1, pp. 13–37, 2011.

[15] H. K. Wei, *Theory and Method of the Neural Networks Architecture Design*, National Defence Industry Press, Beijing, China, 2005.

[16] H. Wei and S.-I. Amari, "Dynamics of learning near singularities in radial basis function networks," *Neural Networks*, vol. 21, no. 7, pp. 989–1005, 2008.

[17] H. Wei, J. Zhang, F. Cousseau, T. Ozeki, and S.-I. Amari, "Dynamics of learning near singularities in layered networks," *Neural Computation*, vol. 20, no. 3, pp. 813–843, 2008.

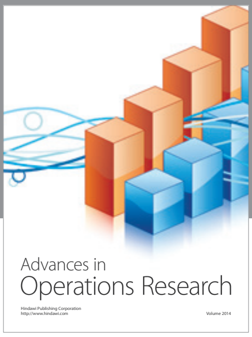
[18] K. Levenberg, "A method for the solution of certain non-linear problems in least squares," *Quarterly of Applied Mathematics*, vol. 2, pp. 164–168, 1944.

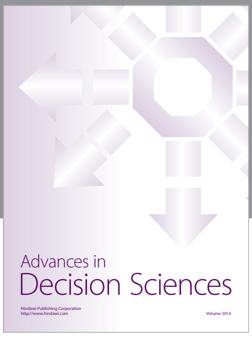
[19] M. T. Hagan and M. B. Menhaj, "Training feedforward networks with the Marquardt algorithm," *IEEE Transactions on Neural Networks*, vol. 5, no. 6, pp. 989–993, 1994.

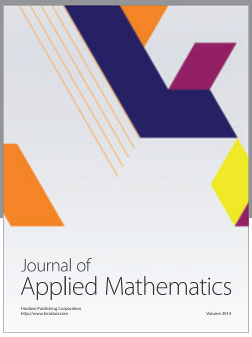
[20] W. Guo, H. Wei, J. Zhao, and K. Zhang, "Averaged learning equations of error-function-based multilayer perceptrons," *Neural Computing and Applications*, vol. 25, no. 3-4, pp. 825–832, 2014.

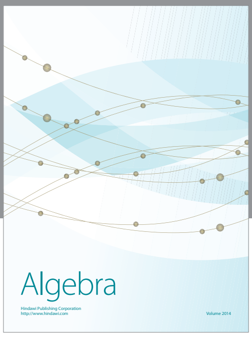
[21] P. E. Gill and W. Murray, "Algorithms for the solution of the nonlinear least-squares problem," *SIAM Journal on Numerical Analysis*, vol. 15, no. 5, pp. 977–992, 1978.

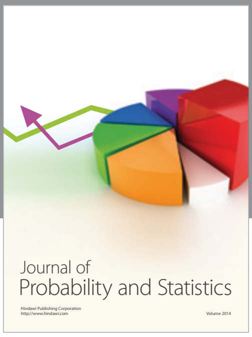
[22] D. W. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," *SIAM Journal on Applied Mathematics*, vol. 11, no. 2, pp. 431–441, 1963.


Advances in
Operations Research

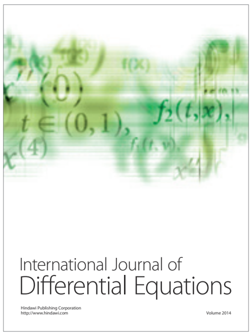

Advances in
Decision Sciences


Journal of
Applied Mathematics

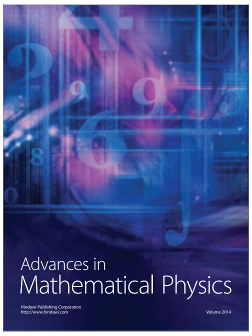

Algebra

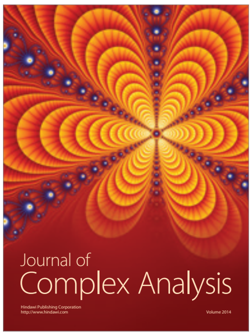

Journal of
Probability and Statistics

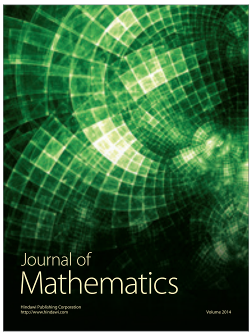
The Scientific
World Journal

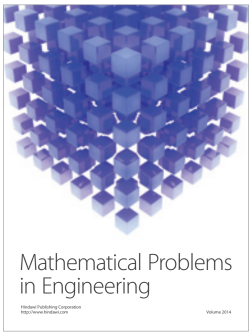

International Journal of
Differential Equations

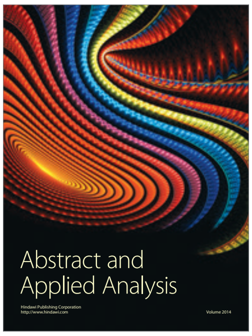
International Journal of
Combinatorics

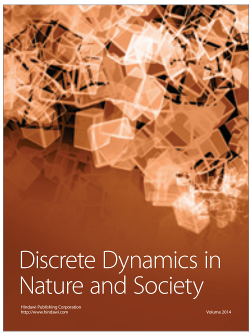

Advances in
Mathematical Physics

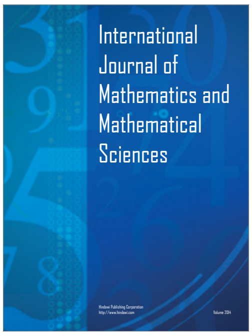

Journal of
Complex Analysis


Journal of
Mathematics


Mathematical Problems
in Engineering

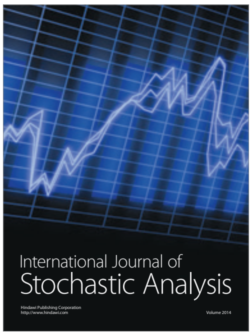

Abstract and
Applied Analysis


Discrete Dynamics in
Nature and Society


International
Journal of
Mathematics and
Mathematical
Sciences

Journal of
Discrete Mathematics

Journal of
Function Spaces


International Journal of
Stochastic Analysis

Journal of
Optimization