Proceedings of the 36th Chinese Control Conference
July 26-28, 2017, Dalian, China

An echo state network based on Levenberg-Marquardt algorithm

Lei Wang[1,2], Cuili Yang[1,2], Junfei Qiao[1,2], Gongming Wang[1,2]

1. Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China
E-mail: jade_wanglei@163.com

2. Beijing Key Laboratory of Computational Intelligence and Intelligent System, Beijing 100124, China
E-mail: jade_wanglei@emails.bjut.edu.cn

Abstract: An abnormal solution might occur during the learning process of echo state network if the least singular value of reservoir state matrix is very close zero. To solve this problem, an echo state network based on Levenberg-Marquardt (LM-ESN) algorithm replacing linear regression for output weights is proposed and a new damping term is given. In the proposed method, it is demonstrated that the output weights sequence has quadratical convergence if the norm of error vector provides a local error bound. Simulations show that the new method could deal with abnormal solution problems effectively, also it has better performance and robustness for time series prediction than some existing methods.

Key Words: echo state network; Levenberg-Marquardt algorithm; time series prediction

1 Introduction

Recurrent neural network (RNN) is a significant nonlinear approach for modelling dynamical systems, and can approximate any dynamical systems with an arbitrary accuracy theoretically [1]. The typical recurrent neural network includes Hopfield neural network [2], Boltzmann neural network [3], Elman recurrent neural network [4], and echo state network (ESN) [5] et al. In these models, ESN is a simple and powerful approach and has drawn much attention. On the problem of Mackey-Glass chaotic time series prediction, the accuracy based on ESN is improved over 2400 times than previous techniques [5]. As a kind of novel recurrent neural network, ESN has numerous successful applications such as time-series prediction [6], nonlinear control [7], and nonlinear signal processing [8].

The performance of ESN is mainly determined by the reservoir, which has large numbers of neurons and is connected randomly and sparsely. In the ESN, only output weights are calculated by simple linear regression, other weights such as input weights, reservoir weights, output feedback weights remain unchanged once they are generated. Generally speaking, some methods have been proposed to compute the output weights, such as singular value decomposition (SVD) [9], pseudoinverse solution [5], Wiener-Hopf solution [10], ridge regression [11]. However, for the first three methods, an abnormal solution might occur during the training process, which may deviate from the real system [6].For the ridge regression method, it is difficult to determine ridge parameter. From [12], it is known that the generalization ability degrades due to large output weights, in other words, the input which slightly deviates from training data can result in relatively poor results. Furthermore, ESN model with large output weights might be unstable when it has output feedback. Based on these observations, in this paper, the Levenberg-Marquardt algorithm [13] is used to replace linear regression, and a new damping term is given. This model is called LM-ESN. It can effectively control the amplitude of output weights and improve the performance of ESN. LM-ESN includes two parts: Firstly, Levenberg-Marquardt algorithm is used to replace linear regression for output weights and the new damping term is given. Secondly, it is demonstrated that output weights sequence has quadratical convergence if the norm of error vector provides a local error bound.

The remainder of this article is organized as followed: In Section 2, a brief review of the classical ESN modle is given and an improved ESN model based on Levenberg-Marquardt algorithm is proposed. In Section 3, convergence analysis of the proposed algorithm is shown. In Section 4, two experiments are conducted: MSO problem and $PM_{2.5}$ series prediction, the results show good robustness and performance of the proposed algorithm. Finally, some conclusions are given.

2 ESN based on Levenberg-Marquardt Algorithm

2.1 Classical echo state network

ESN has three layers: input layer, reservoir and output layer. The recursive formula of ESN (without output feedback connections) is described as follows:

$$\mathbf{x}(n) = \mathbf{tanh}(\mathbf{W}^{in}\mathbf{u}(n) + \mathbf{W}\mathbf{x}(n-1)) , \quad (1)$$

$$\mathbf{y}(n) = \mathbf{W}^{out}(\mathbf{u}(n), \mathbf{x}(n)) , \quad (2)$$

where $\mathbf{u}(n) \in \mathbb{R}^K$ is the current input vector, $\mathbf{x}(n) \in \mathbb{R}^N$ is the internal state of reservoir, $\mathbf{y}(n) \in \mathbb{R}^L$ is the output. $\mathbf{W}^{in}$, $\mathbf{W}$, $\mathbf{W}^{out}$ are the weight matrices for input layer, reservoir and output layer, respectively. The dimension of $\mathbf{W}^{in}$, $\mathbf{W}$, $\mathbf{W}^{out}$ is $N \times K$, $N \times N$, and $L \times (K+N)$, respectively. Only output weights $\mathbf{W}^{out}$ needs to be calculated.

2.2 LM-ESN

Levenberg-Marquardt algorithm is a combination of steepest descent method and Gauss-Newton method. It not only has the speed advantage of Gauss-Newton method, but also has the stability of steepest descent method. Levenberg-Marquardt algorithm converges faster than gradient method with the approximate second-order derivative.

The calculation of output weights $\mathbf{W}^{out}$ based on Levenberg-Marquardt algorithm is equivalent to minimize

* This work was supported by the National Natural Science Foundation of China under Grants 61533002 and 61603012, Beijing Municipal Education Commission Foundation under Grant KM201710005025.

the objective function $E(\mathbf{W}^{out})$, which can be defined as following:

$$E(\mathbf{W}^{out}) = \frac{1}{2}\sum_{p=1}^{P}\sum_{j=1}^{M}(y_j^p - d_j^p)^2 = \frac{1}{2}\sum_{q=1}^{Q} e_q^2 \ , \quad (3)$$

where $e_q = y_j^q - d_j^q$, y_j^q is desired output, d_j^q is network output.

Gradient vector is defined as follows:

$$\begin{aligned}\mathbf{g}_k &= \left(\frac{\partial E(\mathbf{W}^{out})}{\partial w_1^{out}},\cdots,\frac{\partial E(\mathbf{W}^{out})}{\partial w_{K+N}^{out}}\right)^T \\ &= \left(\sum_{q=1}^{Q} e_q\frac{\partial e_q}{\partial w_1^{out}},\cdots,\sum_{q=1}^{Q} e_q\frac{\partial e_q}{\partial w_{K+N}^{out}}\right)^T = \mathbf{J}_k{}^T\mathbf{e}_k ,\end{aligned} \quad (4)$$

where $\mathbf{J}_k$ is a Jacobi matrix, $\mathbf{e}_k$ is an error vector.

$$\mathbf{J}_k \triangleq \mathbf{J}(\mathbf{W}^{out}(k)) = \begin{pmatrix} \frac{\partial e_1}{\partial w_1^{out}} & \frac{\partial e_1}{\partial w_2^{out}} & \cdots & \frac{\partial e_1}{\partial w_{K+N}^{out}} \\ \frac{\partial e_2}{\partial w_1^{out}} & \frac{\partial e_2}{\partial w_2^{out}} & \cdots & \frac{\partial e_2}{\partial w_{K+N}^{out}} \\ \vdots & \vdots & & \vdots \\ \frac{\partial e_Q}{\partial w_1^{out}} & \frac{\partial e_Q}{\partial w_2^{out}} & \cdots & \frac{\partial e_Q}{\partial w_{K+N}^{out}} \end{pmatrix}, \quad (5)$$

$$\mathbf{e}_k \triangleq \mathbf{e}(\mathbf{W}^{out}(k)) = (e_1, e_2, \cdots, e_Q)^T . \quad (6)$$

During each iteration of Levenberg-Marquardt algorithm, the output weights $\mathbf{W}^{out}$ will be replaced by the new value. The update rule based on Levenberg-Marquardt algorithm is

$$\mathbf{W}^{out}(k+1) = \mathbf{W}^{out}(k) - (\mathbf{J}_k{}^T\mathbf{J}_k + \mu_k\mathbf{I})^{-1}\mathbf{J}_k{}^T\mathbf{e}_k \ , \quad (7)$$

where μ_k is a damping term.

In Levenberg-Marquardt algorithm, if the damping term μ_k is very small (almost zero), (7) is approaching to Gauss-Newton algorithm, and if the damping term μ_k is very large, (7) approximates to the steepest descent method. Since Gauss-Newton method is the steepest convergence direction [14], the damping term μ_k should be as small as possible. There are many kinds of choices for damping term. It has been shown that Levenberg-Marquardt method has a quadratic convergence if the parameter is chosen as $\mu_k=\|\mathbf{e}_k\|^2$ under the local error bound condition [15]. However, the damping term $\mu_k=\|\mathbf{e}_k\|^2$ has some unsatisfactory properties. If the sequence is near solution set, $\mu_k=\|\mathbf{e}_k\|^2$ may be less than the machine precision, so it will has no effect. Meanwhile, if the sequence is far from solution set, the Levenberg-Marquardt step $\mathbf{d}_k$ will be very small. Consequently, iteration speed has no advantage. In [16], it has been shown that if $\mu_k= \theta\|\mathbf{e}_k\|+(1- \theta)\|\mathbf{J}_k^T\mathbf{e}_k\|$ ($\theta\in[0, 1]$) has a local error bound, the sequence generated by Levenberg- Marquardt method converges quadratically.

Based on these observations, a new damping term is given. Considering the Levenberg-Marquardt algorithm, each iteration can be described as following:

$$\begin{cases} \mathbf{W}^{out}(k+1) = \mathbf{W}^{out}(k) + \mathbf{d}(k), \\ \mathbf{d}_k = -(\mathbf{J}_k^T\mathbf{J}_k + \mu_k\mathbf{I})^{-1}\mathbf{J}_k^T\mathbf{e}_k, \\ \mu_k = \left\|\mathbf{J}_k^T\mathbf{e}_k\right\|^{\delta}, \delta\in[1,2]. \end{cases} \quad (8)$$

The main steps of LM-ESN can be summarized as followed:

Algorithm 1.

Step 1: Randomly generate a matrix $\mathbf{W}_0$ with the predefined sparsity and reservoir size according to uniform distribution in the interval [-1, 1]. The reservoir weight matrix is $\mathbf{W}= (\alpha_\mathbf{w}/\ \rho(\mathbf{W}_0))\ \mathbf{W}_0$, where $0<\alpha_\mathbf{w}<1$ and $\rho(\mathbf{W}_0)$ is the spectral radius of $\mathbf{W}_0$.

Step 2: Randomly generate an input weight matrix $\mathbf{W}^{in}$ according to uniform distribution in the interval [-1, 1], initialize output matrix $\mathbf{W}^{out}(0)$ and the reservoir states $\mathbf{x}(0)$.

Step 3: Drive the reservoir by the input signals as (1), collect the internal states at an initial transient $n_{\min}$.

Step 4: Compute the network output, error vector $\mathbf{e}_k$, objective function $E(\mathbf{W}^{out})$ and Jacobi matrix $\mathbf{J}_k$.

Step 5: Given $\varepsilon \geq 0$, if the norm of objective function's gradient $\|\mathbf{J}_k{}^T\mathbf{e}_k\| \leq \varepsilon$, stop; otherwise compute $\mathbf{d}_k= -(\mathbf{J}_k{}^T\mathbf{J}_k+\mu_k\mathbf{I})^{-1}\mathbf{J}_k{}^T\mathbf{e}_k$, where $\mu_k=\|\mathbf{J}_k{}^T\mathbf{e}_k\|^{\delta}$, $\delta\in[1, 2]$.

Step 6: Compute $\mathbf{W}^{out}(k+1) = \mathbf{W}^{out}(k) + \mathbf{d}_k$, go to *Step 4*.

Step 7: Test the trained LM-ESN.

3 Convergence analysis

Let

$$e(\mathbf{W}^{out}) = 0 . \quad (9)$$

Suppose there are solutions for (9) and denote by Ω .

Define

$$\theta^k(\mathbf{d}) = \left\|\mathbf{J}_k\mathbf{d} + \mathbf{e}_k\right\|^2 + \mu_k\left\|\mathbf{d}\right\|^2 . \quad (10)$$

Considering the optimization problem:

$$\min_{\mathbf{d}} \theta^k(\mathbf{d}) \ . \quad (11)$$

Then (11) is equivalent to (7).

In order to study the convergence properties, the following assumption is given. If no other specified, all lemmas and theorems are under Assumption 1, and the operator $\|\cdot\|$ refers to standard L_2 norm.

Assumption 1.

(a) There exists a solution $\mathbf{W}_*^{out}$ for (9);

(b) $\mathbf{J}(\mathbf{W}^{out})$ is Lipschitz continuous on some neighborhood of $\mathbf{W}_*^{out}\in\Omega$,i.e. there are positive constants $L_1 > 0$ and $b_1 \in (0,1)$ satisfying

$$\left\|\mathbf{J}(\mathbf{W}_1^{out}) - \mathbf{J}(\mathbf{W}_2^{out})\right\| \leq L_1\left\|\mathbf{W}_1^{out} - \mathbf{W}_2^{out}\right\|, \quad (12)$$

$$\forall\mathbf{W}_1^{out}, \mathbf{W}_2^{out} \in N(\mathbf{W}_*^{out}, b_1) = \left\{\mathbf{W}^{out} \left| \left\|\mathbf{W}^{out} - \mathbf{W}_*^{out}\right\| \leq b_1\right.\right\}$$

(c) $\left\|e(\mathbf{W}^{out})\right\|$ has a local error bound on $N(\mathbf{W}_*^{out}, b_1)$ for (9), i.e. there is a constant $c_1 > 0$ satisfying

$$\left\|e(\mathbf{W}^{out})\right\| \geq c_1 dist(\mathbf{W}^{out}, \Omega), \forall\mathbf{W}^{out} \in \Omega, \quad (13)$$

where $dist(\mathbf{W}^{out}, \Omega) = \min_{\hat{\mathbf{W}}^{out}\in\Omega}\left\|\mathbf{W}^{out} - \hat{\mathbf{W}}^{out}\right\|$.

By Assumption 1, it can be obtained

$$\begin{aligned}&\left\|\mathbf{e}(\mathbf{W}_1^{out}) - \mathbf{e}(\mathbf{W}_2^{out}) - \mathbf{J}(\mathbf{W}_1^{out})(\mathbf{W}_1^{out} - \mathbf{W}_2^{out})\right\| \\ &\leq L_1\left\|\mathbf{W}_1^{out} - \mathbf{W}_2^{out}\right\|^2 ,\end{aligned} \quad (14)$$

$$\begin{aligned}&\left\|\mathbf{e}(\mathbf{W}_1^{out}) - \mathbf{e}(\mathbf{W}_2^{out})\right\| \leq L_2\left\|\mathbf{W}_1^{out} - \mathbf{W}_2^{out}\right\|, \\ &L_2 > 0, \forall\mathbf{W}_1^{out}, \mathbf{W}_2^{out} \in N(\mathbf{W}_*^{out}, b_1).\end{aligned} \quad (15)$$

Now the local convergence for Algorithm 1 is discussed, and the following three lemmas are given. And then the superlinear convergence theorem can be shown.

Suppose $\bar{\mathbf{W}}^{out}(k) \in \Omega$ satisfy the following:

$$\left\|\mathbf{W}^{out}(k) - \bar{\mathbf{W}}^{out}(k)\right\| = dist(\mathbf{W}^{out}(k), \Omega). \quad (16)$$

Lemma 1. If $\mathbf{W}^{out}(k) \in N(\mathbf{W}_*^{out}, b_1)$, then there is a constant $c_2 > 0$ satisfying

$$c_2^{\delta} dist(\mathbf{W}^{out}, \Omega)^{\delta} \le \mu_k = \left\|\mathbf{J}_k^T \mathbf{e}_k\right\|^{\delta} \le L_2^{\delta} \left\|\bar{\mathbf{W}}^{out}(k) - \mathbf{W}^{out}(k)\right\|^{\delta}. \quad (17)$$

Lemma 2. If $\mathbf{W}^{out}(k) \in N(\mathbf{W}_*^{out}, \frac{b_1}{2})$, then there are constants $c_3 > 0, c_4 > 0$ satisfying

(a) $\left\|\mathbf{d}_k\right\| \le c_3 dist(\mathbf{W}^{out}(k), \Omega)$;

(b) $\left\|\mathbf{J}_k \mathbf{d}_k + \mathbf{e}_k\right\| \le c_4 dist(\mathbf{W}^{out}(k), \Omega)^{\frac{2+\delta}{2}}$.

Lemma 3. If $\mathbf{W}^{out}(k+1), \mathbf{W}^{out}(k) \in N(\mathbf{W}_*^{out}, \frac{b_1}{2})$, there is a constant $c_5 > 0$ satisfying

$$dist(\mathbf{W}^{out}(k+1), \Omega) \le c_5 dist(\mathbf{W}^{out}(k), \Omega)^{\frac{2+\delta}{2}},$$

where $c_5 = \dfrac{c_4 + L_1 c_3^2}{c_1}$.

Theorem 1. If $\mathbf{W}^{out}(0)$ is close to Ω sufficiently, then $\left\{\mathbf{W}^{out}(k)\right\}$ converges to some solution $\bar{\mathbf{W}}^{out} \in \Omega$ superlinearly.

From Theorem 1, suppose $\left\{\mathbf{W}^{out}(k)\right\}$ converges to $\mathbf{W}_*^{out} \in \Omega$, and SVD of $\mathbf{J}(\mathbf{W}_*^{out})$ is

$$\mathbf{J}(\mathbf{W}_*^{out}) = \mathbf{U}^* \mathbf{\Sigma}^* \mathbf{V}^{*T} = (\mathbf{U}_1^*, \mathbf{U}_2^*) \begin{pmatrix} \mathbf{\Sigma}_1^* & \\ & \mathbf{O} \end{pmatrix} \begin{pmatrix} \mathbf{V}_1^{*T} \\ \mathbf{V}_2^{*T} \end{pmatrix} = \mathbf{U}_1^* \mathbf{\Sigma}_1^* \mathbf{V}_1^{*T},$$

where $\Sigma_1^* = diag(\sigma_1^*, \cdots, \sigma_2^*)$, $\sigma_1^* \ge \sigma_2^* \ge \cdots \ge \sigma_r^* > 0$, $rank(\mathbf{\Sigma}_1^*) = r$.

Suppose the SVD of $\mathbf{J}(\mathbf{W}^{out}(k)) \triangleq \mathbf{J}_k$ is as follows.

$$\mathbf{J}_k = \mathbf{U}_k \mathbf{\Sigma}_k \mathbf{V}_k^T = (\mathbf{U}_{k,1}, \mathbf{U}_{k,2}, \mathbf{U}_{k,3}) \begin{pmatrix} \mathbf{\Sigma}_{k,1} & & \\ & \mathbf{\Sigma}_{k,2} & \\ & & O \end{pmatrix} \begin{pmatrix} \mathbf{V}_{k,1}^T \\ \mathbf{V}_{k,2}^T \\ \mathbf{V}_{k,3}^T \end{pmatrix}$$

$$= \mathbf{U}_{k,1} \mathbf{\Sigma}_{k,1} \mathbf{V}_{k,1}^T + \mathbf{U}_{k,2} \mathbf{\Sigma}_{k,2} \mathbf{V}_{k,2}^T$$

where

$\Sigma_{k,1} = diag(\sigma_1^{(k)}, \cdots, \sigma_r^{(k)}), \Sigma_{k,2} = diag(\sigma_{r+1}^{(k)}, \cdots, \sigma_{r+q}^{(k)})$,

$\sigma_1^{(k)} \ge \cdots \ge \sigma_r^{(k)} \ge \sigma_{r+1}^{(k)} \ge \cdots \ge \sigma_{r+q}^{(k)} > 0$, $q \ge 0$.

$\mathbf{\Sigma}_{k,i}, \mathbf{U}_{k,i}, \mathbf{V}_{k,i}$ are denoted as $\mathbf{\Sigma}_i, \mathbf{U}_i, \mathbf{V}_i (i = 1,2,3)$ respectively. Therefore, the SVD of $\mathbf{J}_k$ can be written as

$$\mathbf{J}_k = \mathbf{U}_1 \mathbf{\Sigma}_1 \mathbf{V}_1^T + \mathbf{U}_2 \mathbf{\Sigma}_2 \mathbf{V}_2^T . \quad (18)$$

To prove $\left\{\mathbf{W}^{out}(k)\right\}$ converges quadratically, the following lemma similar to Lemma 4.3 of [17] is given.

Lemma 4. If $\mathbf{W}^{out}(k) \in N(\mathbf{W}_*^{out}, \frac{b_1}{2})$, then

(a) $\left\|\mathbf{U}_1 \mathbf{U}_1^T \mathbf{e}_k\right\| \le L_2 \left\|\mathbf{W}^{out}(k) - \bar{\mathbf{W}}^{out}(k)\right\|$;

(b) $\left\|\mathbf{U}_2 \mathbf{U}_2^T \mathbf{e}_k\right\| \le 2L_1 \left\|\mathbf{W}^{out}(k) - \mathbf{W}_*^{out}(k)\right\|^2$;

(c) $\left\|\mathbf{U}_3 \mathbf{U}_3^T \mathbf{e}_k\right\| \le L_1 \left\|\mathbf{W}^{out}(k) - \bar{\mathbf{W}}^{out}(k)\right\|^2$.

Now the quadratic convergence of the sequence is given as following.

Theorem 2. If the sequence $\left\{\mathbf{W}^{out}(k)\right\}$ is generated by Algorithm 1 without line search satisfying $\mathbf{W}^{out}(0)$ sufficiently close to Ω , then $\left\{\mathbf{W}^{out}(k)\right\}$ converges quadratically.

Proof. By (18), it can be got

$$(\mathbf{J}_k^T \mathbf{J}_k + \mu_k \mathbf{I})^{-1} = \mathbf{V}_1 (\mathbf{\Sigma}_1^2 + \mu_k \mathbf{I})^{-1} \mathbf{V}_1^T + \mathbf{V}_2 (\mathbf{\Sigma}_2^2 + \mu_k \mathbf{I})^{-1} \mathbf{V}_2^T .$$

Hence,

$$\mathbf{d}_k = -(\mathbf{J}_k^T \mathbf{J}_k + \mu_k \mathbf{I})^{-1} \mathbf{J}_k^T \mathbf{e}_k$$
$$= -\mathbf{V}_1 (\mathbf{\Sigma}_1^2 + \mu_k \mathbf{I})^{-1} \mathbf{\Sigma}_1 \mathbf{U}_1^T \mathbf{e}_k - \mathbf{V}_2 (\mathbf{\Sigma}_2^2 + \mu_k \mathbf{I})^{-1} \mathbf{\Sigma}_2 \mathbf{U}_2^T \mathbf{e}_k . \quad (19)$$

Therefore,

$$\mathbf{e}_k + \mathbf{J}_k \mathbf{d}_k = (\mathbf{U}_1 \mathbf{U}_1^T + \mathbf{U}_2 \mathbf{U}_2^T + \mathbf{U}_3 \mathbf{U}_3^T) \mathbf{e}_k + \mathbf{J}_k \mathbf{d}_k$$
$$= (\mathbf{U}_1 \mathbf{U}_1^T + \mathbf{U}_2 \mathbf{U}_2^T + \mathbf{U}_3 \mathbf{U}_3^T) \mathbf{e}_k - \mathbf{U}_1 \mathbf{\Sigma}_1 (\mathbf{\Sigma}_1^2 + \mu_k \mathbf{I})^{-1} \mathbf{\Sigma}_1 \mathbf{U}_1^T \mathbf{e}_k$$
$$- \mathbf{U}_2 \mathbf{\Sigma}_2 (\mathbf{\Sigma}_2^2 + \mu_k \mathbf{I})^{-1} \mathbf{\Sigma}_2 \mathbf{U}_2^T \mathbf{e}_k$$
$$= (\mathbf{U}_1 \mathbf{U}_1^T - \mathbf{U}_1 \mathbf{\Sigma}_1 (\mathbf{\Sigma}_1^2 + \mu_k \mathbf{I})^{-1} \mathbf{\Sigma}_1 \mathbf{U}_1^T) \mathbf{e}_k$$
$$+ (\mathbf{U}_2 \mathbf{U}_2^T - \mathbf{U}_2 \mathbf{\Sigma}_2 (\mathbf{\Sigma}_2^2 + \mu_k \mathbf{I})^{-1} \mathbf{\Sigma}_2 \mathbf{U}_2^T) \mathbf{e}_k + \mathbf{U}_3 \mathbf{U}_3^T \mathbf{e}_k$$
$$= \mathbf{U}_1 (\mathbf{I} - \mathbf{\Sigma}_1 (\mathbf{\Sigma}_1^2 + \mu_k \mathbf{I})^{-1} \mathbf{\Sigma}_1) \mathbf{U}_1^T \mathbf{e}_k$$
$$+ \mathbf{U}_2 (\mathbf{I} - \mathbf{\Sigma}_2 (\mathbf{\Sigma}_2^2 + \mu_k \mathbf{I})^{-1} \mathbf{\Sigma}_2) \mathbf{U}_2^T \mathbf{e}_k + \mathbf{U}_3 \mathbf{U}_3^T \mathbf{e}_k$$
$$= \mu_k \mathbf{U}_1 (\mathbf{\Sigma}_1^2 + \mu_k \mathbf{I})^{-1} \mathbf{U}_1^T \mathbf{e}_k + \mu_k \mathbf{U}_2 (\mathbf{\Sigma}_2^2 + \mu_k \mathbf{I})^{-1} \mathbf{U}_2^T \mathbf{e}_k$$
$$+ \mathbf{U}_3 \mathbf{U}_3^T \mathbf{e}_k . \quad (20)$$

Since $\left\{\mathbf{W}^{out}(k)\right\}$ converges to $\mathbf{W}_*^{out}$, suppose $L_1 \left\|\mathbf{W}^{out}(k) - \mathbf{W}_*^{out}\right\| \le \dfrac{\sigma_r^*}{2}$ holds for all sufficient large k.

Hence

$$\left\|(\mathbf{\Sigma}_1^2 + \mu_k \mathbf{I})^{-1}\right\| \le \left\|\mathbf{\Sigma}_1^{-2}\right\| \le \frac{1}{(\sigma_r^* - L_1 \left\|\mathbf{W}^{out}(k) - \mathbf{W}_*^{out}\right\|)^2} < \frac{4}{\sigma_r^{*2}}, \quad (21)$$

$$\left\|(\mathbf{\Sigma}_2^2 + \mu_k \mathbf{I})^{-1}\right\| \le \mu_k^{-1}. \quad (22)$$

By (20), (21), (22), Lemma 1 and Lemma 4,

$$\left\|\mathbf{e}_k + \mathbf{J}_k \mathbf{d}_k\right\|$$
$$\le \mu_k \frac{4}{\sigma_r^{*2}} L_2 \left\|\mathbf{W}^{out}(k) - \bar{\mathbf{W}}^{out}(k)\right\|^2 + 3L_1 \left\|\mathbf{W}^{out}(k) - \mathbf{W}_*^{out}\right\|^2$$
$$\le \frac{4L_2^{1+2\delta}}{\sigma_r^{*2}} \left\|\mathbf{W}^{out}(k) - \bar{\mathbf{W}}^{out}(k)\right\|^{1+\delta} + 3L_1 \left\|\mathbf{W}^{out}(k) - \mathbf{W}_*^{out}\right\|^2$$
$$\le (\frac{4L_2^{1+2\delta}}{\sigma_r^{*2}} + 3L_1) \left\|\mathbf{W}^{out}(k) - \mathbf{W}_*^{out}\right\|^2$$
$$= c_6 \left\|\mathbf{W}^{out}(k) - \mathbf{W}_*^{out}\right\|^2 . \quad (23)$$

where $c_6 = \dfrac{4L_2^{1+2\delta}}{\sigma_r^{*2}} + 3L_1$.

It follows from (13), (14), (22) and Lemma 2 that

$$
\begin{aligned}
&c_1 dist(\mathbf{W}^{out}(k+1),\Omega)\\
&\le \left\|\mathbf{e}_{k+1}\right\| = \left\|\mathbf{e}_{k+1}+\mathbf{e}_k+\mathbf{J}_k\mathbf{d}_k-\mathbf{e}_k-\mathbf{J}_k\mathbf{d}_k\right\|\\
&\le \left\|\mathbf{J}_k\mathbf{d}_k+\mathbf{e}_k\right\|+\left\|\mathbf{e}_{k+1}-\mathbf{e}_k-\mathbf{J}_k\mathbf{d}_k\right\|\\
&\le c_6\left\|\mathbf{W}^{out}(k)-\mathbf{W}_*^{out}\right\|^2+L_1\left\|\mathbf{d}_k\right\|^2\\
&\le c_6\left\|\mathbf{W}^{out}(k)-\mathbf{W}_*^{out}\right\|^2+L_1c_3^2\left\|\mathbf{W}^{out}(k)-\bar{\mathbf{W}}^{out}(k)\right\|^2\\
&\le (c_6+L_1c_3^2)\left\|\mathbf{W}^{out}(k)-\mathbf{W}_*^{out}\right\|^2.
\end{aligned}
$$

By Theorem 1, it can be got $\left\|\mathbf{d}_{k+1}\right\| = O(\left\|\mathbf{d}_k\right\|^2)$, which implies that $\left\{\mathbf{W}^{out}(k)\right\}$ converges quadratically to $\mathbf{W}_*^{out}$, namely $\left\|\mathbf{W}^{out}(k+1)-\mathbf{W}_*^{out}\right\|^2 = O\left(\left\|\mathbf{W}^{out}(k)-\mathbf{W}_*^{out}\right\|^2\right)$. This completes the proof.

4 Simulations and results

4.1 Multiple Superimposed Oscillator Problem

The MSO time series [18-19] is derived by the following equation:

$$y(t)=\sum_{i=1}^{m}\sin(\alpha_i t), \tag{24}$$

where m denotes the number of sine waves. MSO is a benchmark problem for ESN and has drawn much attention. In this paper, m=2 (MSO2) is chosen to test the performance. The frequencies of the sine waves in MSO2 are taken from α_1=0.2, and α_2=0.311. The dataset contains 1100 values, the first 800 values are used for training, and the next values are used for testing. The discarding points are 100.

The normalized root mean square error (NRMSE) [18] is used as the evaluation criteria of model performance. NRMSE is given as followed:

$$\mathrm{NRMSE}=\sqrt{\sum_{t=1}^{T}\frac{(d_i(t)-y_i(t))^2}{T\sigma^2}}, \tag{25}$$

where $d_i(t)$ denotes the desired output, $y_i(t)$ is the prediction output, σ^2 is the variance of desired outputs, T is the numbers of $d_i(t)$.

The training NRMSE curve for MSO2 is presented in Fig.1. Simulation results based on LM-ESN with different damping terms for MSO2 are listed in Table 1. From Table 1, it can be seen that it just needs 52 iterations to finish training process with the new damping term, this means the average speed of the proposed algorithm is very fast and testing NRMSE is smaller than other choice of damping term.

The testing outputs and error for MSO2 are given in Fig.2 and Fig.3, respectively. It can be seen LM-ESN has smaller testing errors. The performance of LM-ESN is also compared with some existing methods, such as the origin ESN (O-ESN) [5], SCR [20]. Based on 100 independent simulations, the comparison of testing results with different approaches for MSO2 are described in Table 2. As seen from Table 2, LM-ESN needs slightly more training time than other methods, but has better prediction accuracy than O-ESN and SCR according to testing NRMSE.

In order to validate the robustness of LM-ESN, the successful design ratio is introduced by

$$R(\theta)=\frac{\sum_{i=1}^{I}h(e_i-\theta)}{I}, \tag{26}$$

where I is the number of experiments, e_i denotes the prediction error (NRMSE) of the ith experiment, $h(x)$=1 if $x\le 0$, otherwise $h(x)$=0 [19]. $R(\theta)$ is used to estimate the probability of obtaining a network whose prediction error is no more than the threshold θ. For $R(\theta)$, the higher, the better.

The successful design ratios with different thresholds are presented in Fig.4. It can be seen that LM-ESN offers higher successful design ratios than O-ESN for most thresholds.

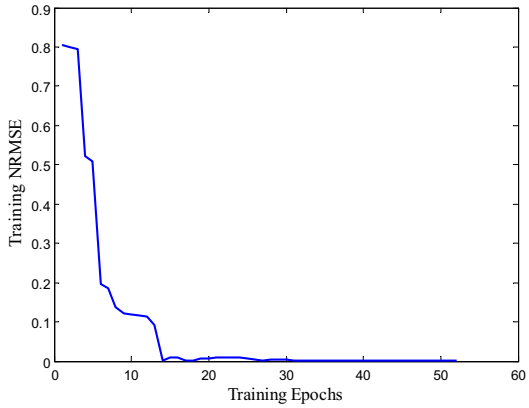

Fig.1. Training NRMSE curve for MSO2

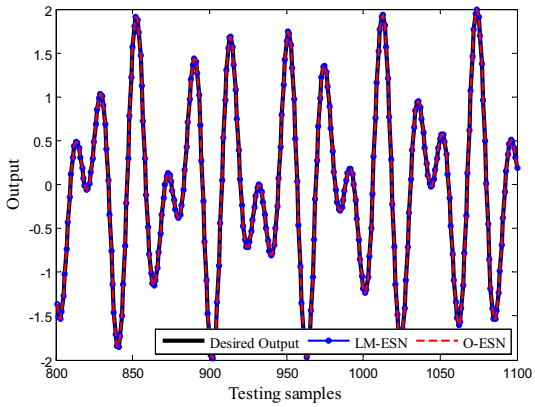

Fig.2. Testing outputs based on LM-ESN and O-ESN for MSO2

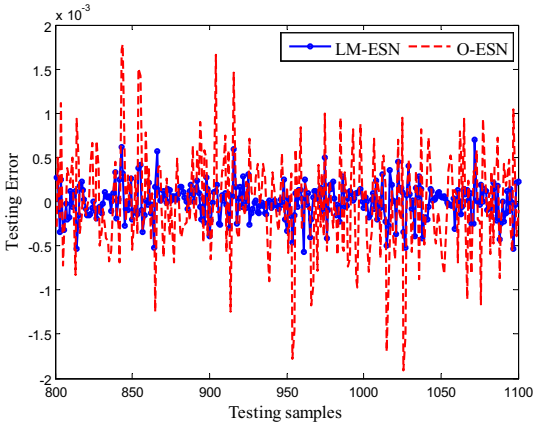

Fig.3. Testing error based on LM-ESN and O-ESN for MSO2

Table 1. Simulation results based on LM-ESN with different damping term for MSO2

Damping term	Iterations	Testing NRMSE	Reservoir size	Spectral radius	Sparsity
$\left\|\mathbf{J}_k^T\mathbf{e}_k\right\|^{\delta}$	**52**	**3.43e-5**	75	0.7500	0.1000
$\left\|\mathbf{J}_k^T\mathbf{e}_k\right\|$	314	4.72e-5	75	0.7500	0.1000
$\left\|\mathbf{e}_k\right\|^2$	500	2.95e-4	75	0.7500	0.1000

Table 2. Simulation results based on LM-ESN, O-ESN and SCR for MSO2

Method	Training time (s)	Testing NRMSE	Reservoir size	Spectral radius	Sparsity
LM-ESN	**61.13**	**3.43e-05**	75	0.7500	0.1000
O-ESN[5]	55.17	6.61e-05	75	0.7500	0.1000
SCR[20]	40.77	9.41e-04	75	0.7500	0.0133

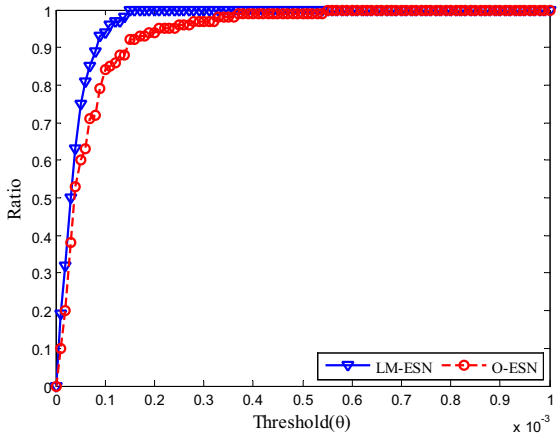

Fig.4. Successful design ratio based on LM-ESN and O-ESN for MSO2

4.2 $PM_{2.5}$ Series Prediction

Fine particulate matter ($PM_{2.5}$), which may cause several kinds of diseases, has already been a major pollutant in many cities in China. It has great influence on human health, air quality and visibility. Therefore, it is very important to study models and systems so that the concentration of $PM_{2.5}$ can be real-time monitored and predicted [21-22]. The target is to predict hourly average concentrations of $PM_{2.5}$ in later one hour in one city of China. The input data, such as SO_2, NO_2, CO, O_3, air temperature, relative humidity, wind speed, PM_{10}, $PM_{2.5}$, are obtained from China's air quality online analysis platform. The 1144 sets of data are collected from 1 December 2015 to 18 January 2016 [23]. The first 700 values are used for training, and the number of testing set is 444. The discarding points are 100.

Before the simulation, input samples of the model are normalized into [-1, 1] by

$$\mathbf{u}(i)' = \frac{\mathbf{u}(i) - \min(\mathbf{u}(i))}{\max(\mathbf{u}(i)) - \min(\mathbf{u}(i))} \times 2 - 1, \qquad (27)$$

The training desired outputs are also normalized into [-1, 1] by similar formula as (27). After simulation, the outputs of the trained model are converted.

The testing outputs and error for $PM_{2.5}$ are given in Fig.5 and Fig.6, respectively. It can be seen that LM-ESN has better prediction performance and the error of LM-ESN is smaller than O-ESN. The detailed results are summarized in Table 3 based on 100 independent simulations. From the comparison of testing NRMSE and their relative parameters in Table 3, it can be seen that LM-ESN has better prediction performance than O-ESN and SCR, but needs more training time.

The successful design ratios of different methods are presented in Fig.7, it can be seen that LM-ESN has higher successful design ratios than O-ESN for $PM_{2.5}$.

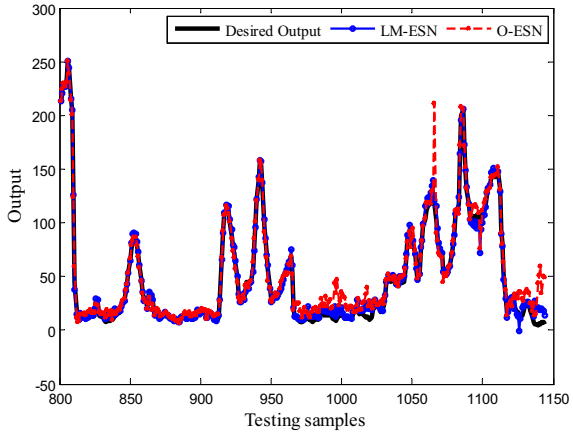

Fig.5. Testing outputs based on LM-ESN and O-ESN for $PM_{2.5}$

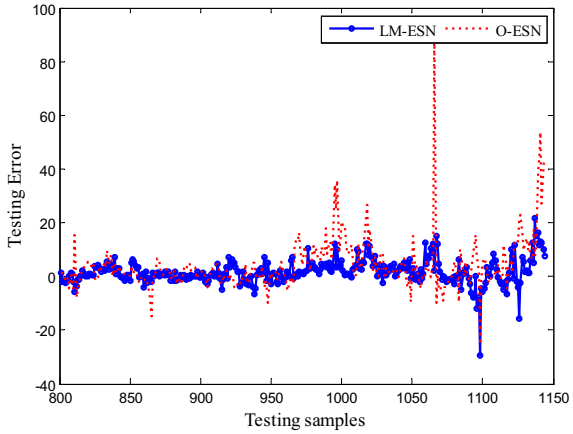

Fig.6. Testing error based on LM-ESN and O-ESN for $PM_{2.}$

Table3.Simulation results based on LM-ESN, O-ESN and SCR for $PM_{2.5}$ series

Method	Training time (s)	Testing NRMSE	Reservoir size	Spectral radius	Sparsity
LM-ESN	**156.36**	**0.5260**	500	0.7500	0.1000
O-ESN[5]	71.18	0.7329	500	0.7500	0.1000
SCR[20]	117.23	1.1374	500	0.7500	0.0020

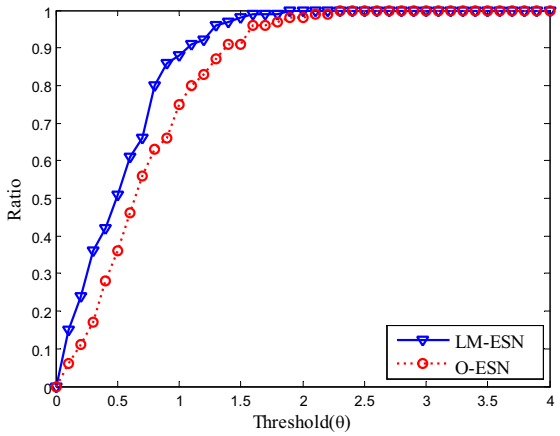

Fig.7. Successful design ratio based on LM-ESN and O-ESN for $PM_{2.5}$

5 Conclusion

In this paper, to solve the abnormal solution problem, an echo state network based on Levenberg-Marquardt algorithm is proposed. LM-ESN uses Levenberg-Marquardt algorithm to replace linear regression for output weights, and select damping term self-adaptively. It can effectively control the amplitude of output weights and improve the performance. Simulations results on two time-series prediction problems show that LM-ESN has better prediction performance than some existing ESN models.

References

[1] A. M. Schäfer, H. G. Zimmermann, Recurrent neural networks are universal approximators, *International Journal of Neural Systems*, 17 (04): 253-263, 2007.

[2] B. Song, Y. Zhang, Z. Shu, et al. Stability analysis of Hopfield neural networks perturbed by Poisson noises, *Neurocomputing*, 196: 53-58, 2016.

[3] J. Gao, J. Yang, G. Wang, et al. A novel feature extraction method for scene recognition based on centered convolutional restricted Boltzmann machines. *Neurocomputing*, 214:708-717, 2016.

[4] R. Chandra, Competition and collaboration in cooperative coevolution of Elman recurrent neural networks for time-series prediction, *IEEE Transactions on Neural Networks and Learning Systems*, 26(12): 3123-3136, 2015.

[5] H. Jaeger, H. Haas, Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication, *Science*, 304 (5667):78-80, 2004.

[6] D. Li, M. Han, J. Wang, Chaotic time series prediction based on a novel robust echo state network, *IEEE Transactions on Neural Networks and Learning Systems*, 23 (5): 787-799, 2012.

[7] Y. Pan, J. Wang, Model predictive control of unknown nonlinear dynamical systems based on recurrent neural networks, *IEEE Transactions on Industrial Electronics*, 59 (8):3089-3101, 2012.

[8] Y. Xia, B. Jelfs, M. M. Van Hulle , An augmented echo state network for nonlinear adaptive filtering of complex noncircular signals, *IEEE Transactions on Neural Networks*, 22 (1): 74-83, 2011.

[9] C. M. Alaíz, J. R. Dorronsoro, On the learning of ESN linear readouts, *Conference of the Spanish Association for Artificial Intelligence*, 124-133, 2011.

[10] Q. Song, Z. Feng, M. Lei, Stable training method for echo state networks with output feedbacks, *International Conference on Networking, Sensing and Control (ICNSC)*, 159-164,2010.

[11] A. Rodan, P. Tiňo, Simple deterministically constructed cycle reservoirs with regular jumps, *Neural Computation*, 24(7):1822-1852,2012.

[12] H. Jaeger, Reservoir riddles: suggestions for echo state network research, *IEEE International Joint Conference on Neural Networks*, pp. 1460-1462, 2005.

[13] J. M. Wu, Multilayer potts perceptrons with Levenberg–Marquardt learning, *IEEE Transactions on Neural Networks*, 19 (12):2032-2043, 2008.

[14] T. Xie, H. Yu, J. Hewlett, B. Wilamowski , Fast and efficient second-order method for training radial basis function networks, *IEEE Transactions on Neural Networks and Learning Systems*, 23 (4): 609-619, 2012.

[15] N. Yamashita, M. Fukushima, On the rate of convergence of the Levenberg-Marquardt method, *Computing* (*Suppl*), 15:239-249, 2001.

[16] C.F. Ma, L.H. Jiang, Some research on Levenberg–Marquardt method for the nonlinear equations, *Applied Mathematics and Computation*, 184(2): 1032-1040, 2007.

[17] J. Fan, J. Zeng, A Levenberg–Marquardt algorithm with correction for singular system of nonlinear equations, *Applied Mathematics and Computation*, 219 (17): 9438-9446, 2013.

[18] D. Koryakin, J. Lohmann, M. V. Butz , Balanced echo state networks, *Neural Networks*, 36: 35-45, 2012.

[19] Y. Xue, L. Yang, S. Haykin, Decoupled echo state networks with lateral inhibition, *Neural Networks*, 20 (3):365-376, 2007.

[20] A. Rodan, P. Tiňo, Minimum complexity echo state network, *IEEE Transactions on Neural Networks*, 22(1):131-144, 2011.

[21] X. Feng, Q. Li, Y. Zhu, Artificial neural networks forecasting of $PM_{2.5}$ pollution using air mass trajectory based geographic model and wavelet transformation, *Atmospheric Environment*, 107:118-128, 2015.

[22] B. T. Ong, K. Sugiura, K. Zettsu, Dynamically pre-trained deep recurrent neural networks using environmental monitoring data for predicting $PM_{2.5}$, *Neural Computing and Applications*, 1-14, 2015.

[23] China's air quality on-line monitoring analysis platform, *Air quality index*, [Online], Available: http://www.aqistudy.cn/,2014.