An Adaptive Learning Rate for Training Ring-Structured Recurrent Network

Daqing Chen and Laiwan Chan
Email:lwchan@cse.cuhk.edu.hk

Dept. of Computer Science and Engineering
The Chinese University of Hong Kong, Shatin, Hong Kong

ABSTRACT

A new adaptive learning rate is proposed based on the Lyapunov stability theory for training the Ring-Structured Recurrent Network (RSRN). The adaptive rate is a sufficient condition to guarantee the stability and the most rapid convergence of the RSRN dynamic backpropagation algorithm, and it is easily determined in a direct and non-trial manner. Examples of training the RSRN to predict time series are used to demonstrate the efficiency of the learning rate. It has been found that by usage of the adaptive learning rate, the RSRN needs much smaller amount of training time and the resulting network could perform satisfactorily the prediction task.
KEYWORDS: recurrent network, learning algorithm, time series prediction

1. Introduction

As one promising learning rule, the *Real Time Recurrent Learning* (RTRL) Algorithm [7] has been widely used for *Recurrent Neural Networks* (RNNs) training. The RTRL is a type of gradient-based procedure and is developed as natural extension of the backpropagation rule used for the static feedforward networks learning. However, the algorithm is computationally expensive and suffers from the drawback of slow convergence. Stability and convergence properties of such method have not yet been rigorously analyzed, and the learning rate is usually determined in a trial manner.

In this paper, we focus on speeding up the learning process of RNNs. Based on the Lyapunov stability theory, an adaptive learning rate is proposed for training the *Ring-Structured Recurrent Network* (RSRN) [1][2][3]. The RSRN is a locally connected RNN and has much lower computational and storage complexities. The learning rate presented here is a sufficient condition to guarantee the stability and the most rapid convergence of the RSRN dynamic backpropagation algorithm. To evaluate the efficiency of the proposed learning rate, several simulation experiments of training the RSRN to predict time series of sunspot activity data are conducted, and the learning process and prediction capability for the network trained with fixed and adaptive learning rates are compared.

2. RSRN and Its Learning Algorithm

The RSRN is a particular type of locally connected recurrent network as shown in Fig. 1. Each hidden unit is connected to only a number of its neighbors including itself to form a ring-like structure recurrent links. It has been demonstrated that the RSRN could give expected dynamic mapping characteristics with a much lower cost of computation, storage and implementation on parallel architecture [1][2][3].

Consider a M-input-single-output RSRN with N hidden nodes, each being connected to its P neighboring nodes including itself. The whole dynamic equations of the network are accordingly described as

$$\begin{aligned} y(k) &= \sum_{j=1}^{N} w_j^{Out} x_j(k) \\ x_j(k) &= f(h_j(k)) \\ h_j(k) &= \sum_{i=1}^{M} w_{ij}^{In} I_i(k) + \sum_{l \in \mathcal{A}_P^j} w_{lj}^{RS} x_l(k-1) \end{aligned} \tag{1}$$

where k denotes time; $I_i(k)$ is the external input to the ith input node, $x_j(k)$ is the output of the jth hidden node, and $y(k)$ is the network output; w_{ij}^{In}, w_{lj}^{RS} and w_j^{Out} are the weights linked from the ith input node to the jth hidden node, the lth hidden node to the jth hidden node, and the jth hidden node to the output node, respectively; $\mathcal{A}_P^j$ represents the set of indices l for which hidden node l is one of the P neighbors of hidden node j; $f(\cdot)$ is the nonlinear hidden node transfer function.

Let $\{y_m(k)\}$ denote a scalar time series. Given the finite past T samples of $\{y_m(k)\}$, the network (1) is used to model the series. Let $y_m(k)$ be the desired response of the network at time k, and $e(k)$ be the prediction error, $e(k) = y_m(k) - y(k)$. Define the instantaneous squared error at time k as $\varepsilon(k) = \frac{1}{2}e^2(k) = \frac{1}{2}(y_m(k) - y(k))^2$. Thus, the objective function is obtained by summing $\varepsilon(k)$ over all the time k, that is

$$E = \frac{1}{2}\sum_k \varepsilon(k) = \frac{1}{2}\sum_k (y_m(k) - y(k))^2 \tag{2}$$

and an optimal set of network parameters can be determined by adjusting the parameters such that E is

minimized.

Using a gradient descent procedure, the dynamic backpropagation learning algorithm for updating the weights of RSRN is given by

$$W(k+1) = W(k) + \eta(-\frac{\partial \varepsilon(k)}{\partial W}) = W(k) + \eta e(k)\frac{\partial y(k)}{\partial W} \tag{3}$$

where η is a learning rate, $0 < \eta$, and W represents W^{Out}, W^{RS} or W^{In}, respectively. Furthermore, it is easy to be shown that

$$\frac{\partial y(k)}{\partial w_j^{Out}} = x_j(k), j = 1, \cdots, N \tag{4}$$

$$\frac{\partial y(k)}{\partial w_{mn}^{RS}} = \sum_j^N w_j^{Out}\frac{\partial x_j(k)}{\partial w_{mn}^{RS}}, j, m, n = 1, \cdots, N \tag{5}$$

$$\frac{\partial y(k)}{\partial w_{mn}^{In}} = \sum_j^N w_j^{Out}\frac{\partial x_j(k)}{\partial w_{mn}^{In}}, j, n = 1, \cdots, N; m = 1, \cdots, M \tag{6}$$

where

$$\frac{\partial x_j(k)}{\partial w_{mn}^{RS}} = f^{'}(h_j(k))[\delta_{jn}x_m(k-1) + \sum_{l \in \mathcal{A}_P^j} w_{lj}^{RS}\frac{\partial x_l(k-1)}{\partial w_{mn}^{RS}}] \tag{7}$$

$$\frac{\partial x_j(k)}{\partial w_{mn}^{In}} = f^{'}(h_j(k))[\delta_{jn}I_m(k) + \sum_{l \in \mathcal{A}_P^j} w_{lj}^{RS}\frac{\partial x_l(k-1)}{\partial w_{mn}^{In}}] \tag{8}$$

with initial conditions

$$\frac{\partial x_j(0)}{\partial w_{mn}^{RS}} = 0, \frac{\partial x_j(0)}{\partial w_{mn}^{In}} = 0$$

and δ_{jn} is a Kronecker delta equal to 1 when $j = n$ and zero otherwise.

3. Convergence and Stability Analysis of the Algorithm

The algorithm (3)-(8) represents a nonlinear dynamic system. Based on the Lyapunov stability theory, a sufficient condition to guarantee convergence of the algorithm can be given, which leads to an adaptive learning rate for stable and rapid convergence of the learning process.

Let a discrete-type quadratic error form $V(k) = \frac{1}{2}e^2(k)$ be used as the Lyapunov function [6]. The change of Lyapunov function due to the learning is accordingly given by $\Delta V(k) = V(k+1) - V(k) = \frac{1}{2}[e^2(k+1) - e^2(k)]$ and the error difference in the learning process can be obtained by $e(k+1) = e(k) + \Delta e(k) = e(k) + (\frac{\partial e(k)}{\partial W})^T \Delta W$, where ΔW denotes a change of an arbitrary weight W^{Out}, W^{RS} or W^{In} in the network, respectively. With the definition above, a general convergence theorem can be stated as

Theorem 1 Let η be the learning rate for the network weights of RSRN and $Y_{\max}$ be defined as $Y_{\max} = \max_k \|\partial y(k)/\partial W\|$, where $\|\cdot\|$ is the usual Euclidean norm. Then the convergence is guaranteed if η is chosen as $0 < \eta < \frac{2}{Y_{\max}^2}$, and the maximum learning rate $\eta_{\max}$ to guarantee the most rapid convergence is given by $\eta_{\max} = \frac{1}{Y_{\max}^2}$.

Proof: see [5][6].

Based on the Theorem 1, the optimal learning rates for the weights W^{Out}, W^{RS}, and W^{In} of RSRN can be determined as

Theorem 2 Let η^{Out}, η^{RS}, and η^{In} be the learning rates for the weights W^{Out}, W^{RS}, and W^{In} of RSRN, respectively, and let each nonlinear hidden node transfer function in the network is specified by $f(\cdot) = \frac{1-e^{-(\cdot)}}{1+e^{-(\cdot)}}$. Then the algorithm (3)-(8) converges if $0 < \left|w_{lj}^{RS}\right| < \frac{1}{N\sqrt{P+\varepsilon}}$, $j = 1, \cdots N$, $l \in \mathcal{A}_P^j$, and the learning rates are chosen as

$$0 < \eta^{Out} < \frac{2}{N} \tag{9}$$

$$0 < \eta^{RS} < \frac{2(1-\sqrt{\delta})^2}{(W_{\max}^{Out})^2 N^2 (P+1)} \tag{10}$$

$$0 < \eta^{In} < \frac{2(1-\sqrt{\delta})^2}{(W_{\max}^{Out})^2 I_{\max}^2 N^2 M} \tag{11}$$

where $\varepsilon > 1$, $W_{\max}^{Out} = \max_k \|W^{Out}(k)\|$, $I_{\max} = \max_k \|I(k)\|$, and $\delta = \sqrt{\frac{P+1}{P+\varepsilon}}$.

Proof: see [4].

Remark The learning rates presented by Theorem 2 are very sufficient conditions to guarantee the stability and convergence of the RSRN learning process, and the values of which are usually very small. ¿From the practical application point of view, those conditions may need to be relaxed further [4].

4. Simulation Experiments

To illustrate the efficiency of the proposed learning rate, several experiments are presented in this section.

In the following simulation study, the RSRN is trained to perform single-step prediction for the time series produced by the yearly averaged sunspot activity from years 1700 to 1979. The network architecture is chosen as $M = 13$ and $N = 5$. These input nodes correspond to 12 time delays of the series and a constant input ($\equiv 1$) used for the bias of hidden node. Initial network parameters are set randomly to values in the intervals [-0.1,0.1]. The first 100 samples and the last 168 samples of the series are used for training the network and testing the prediction capability of the resulting network, respectively. For the sake of comparison, let $P = N$ and $\varepsilon = N$.

Example 1 The network is trained by using the algorithm (3) - (8) with fixed learning rates and the adaptive learning rates (9) - (11), respectively. Fig. (2) gives the evolution of the sum of squared error in the network learning process for different learning rates, where $\eta^{Out} \equiv \eta^{RS} \equiv \eta^{In} \equiv \eta$ for each set of fixed learning rates. It is obvious that the learning processes controlled by the dynamic backpropagation rule with the adaptive rates are stable and rapid convergence. However, the evolution determined by fixed learning rates are closely related to a proper choice of the learning rate η. For a small value of η the convergence is guaranteed but the speed is slow; on the other hand if η is too big, the algorithm becomes unstable.

Example 2 The network is trained by using the algorithm (3) - (8) with learning rates chosen carefully based on a lot of trial simulations. In such a way, it is possible to determine a set of good learning rates for network training. Compared to this, however, the adaptive learning rate is determined in a direct way. Especially, by properly relaxing the sufficient conditions, the network learning process can be speeded up considerably, and the resulting networks are more powerful in time series prediction. The related experiment results are given in Table 1 and Fig. 3 - Fig. 5, where the adaptive learning rates are determined by the relaxed sufficient conditions as [4]

$$\eta^{Out} = \frac{2}{N} \times 0.01 \tag{12}$$

$$\eta^{RS} = \frac{2}{(W_{\max}^{Out})^2 N^2 (P+1)} \tag{13}$$

$$\eta^{In} = \frac{2}{(W_{\max}^{Out})^2 I_{\max}^2 N^2 M} \tag{14}$$

and the fixed learning rates correspond to the parameter set $\eta^{Out} \equiv 0.1$, $\eta^{RS} \equiv 0.001$, and $\eta^{In} \equiv 0.01$.

Summing up the experiments and discussion made above, we can conclude that the adaptive learning rate presented here has apparent advantage. By usage, a network can learn dynamic mapping more effectively.

5. Conclusions

In this paper, a new adaptive learning rate is proposed for using the dynamic back propagation algorithm effectively to train the ring-structured recurrent network. Such a learning rate is developed based on the Lyapunov stability theory, and is optimal in that it can guarantee both the stability and the rapid convergence of the network learning evolution. Especially, by properly relaxing the sufficient conditions, the resulting learning rates can speed up the learning process considerably. In addition, the rate is easily determined in a direct and non-trial manner. In the experiments of training the RSRN to perform time series prediction, the networks trained with

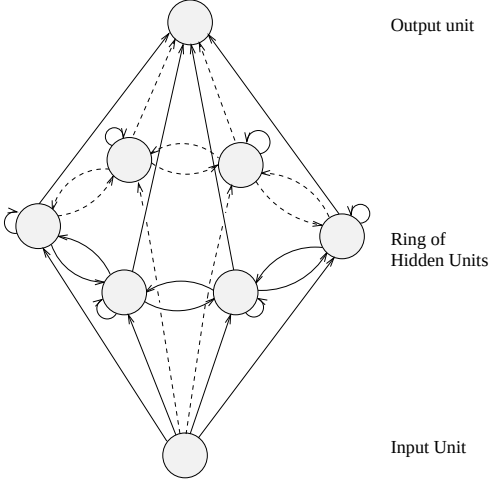

Figure 1: A ring-structured recurrent network with its simplest form.

the adaptive and some predetermined learning rates have been compared in terms of learning process and prediction capability. It has been demonstrated that by use of the adaptive learning rate, the RSRN needs only much smaller amount of training time and the resulting network could perform satisfactorily the prediction task.

References

[1] Chan, Lai-Wan and Young Fung-Yu (1993). Ring-structured recurrent neural network. Proc. 1993 World Congress on Neural Networks, Vol. 4, pp.328-331.

[2] Chan, Lai-Wan (1994). Is Fully-Connected Recurrent Network Necessary? Technical Report: CS-TR-94-11, The Department of Computer Science and Engineering, The Chinese University of Hong Kong.

[3] Chan, Lai-Wan and Young, Evan Fung-Yu (1995). Locally Connected Recurrent Networks. Technical Report: CS-TR-95-10, The Department of Computer Science and Engineering, The Chinese University of Hong Kong.

[4] Chen, Daqing and Chan, Laiwan (1998). A Stability Based Adaptive Learning Rate for Training Ring-Structured Recurrent Network, Technical Report: CS-TR-98-03, The Department of Computer Science and Engineering, The Chinese University of Hong Kong.

[5] Ku, Chao-Chee and Lee, Kwang Y. (1995). Diagonal Recurrent Neural Networks for Dynamic Systems Control. IEEE Trans. Neural Networks, Vol. 6, No. 1, pp.144-155.

[6] Tokai, I., (1991). Learning Control Using Neural Networks. Proc. of 1991 IEEE International Conference on Robotics and Automation, pp. 740-745.

[7] William, R. J. and Zipser, D. (1989). A learning algorithm for continually running fully recurrent neural network. Neural Computation, Vol. 1, pp.270-280.

Acknowledgement The authors would like to thank The Research Grants Council, HK for support.

Table 1: Sum of squared error of the networks trained with different learning rates.

Learning Rates	Stop Learning	Test on Learning Samples	Prediction
Adaptive (12) - (14)	0.356887	0.349604	1.142144
$\eta^{In} = .01, \eta^{RS} = .001, \eta^{Out} = .1$	0.546778	0.562580	1.425479
$\eta^{In} = .01, \eta^{RS} = .001, \eta^{Out} = .01$	0.607001	0.599954	1.334655
$\eta^{In} = .001, \eta^{RS} = .001, \eta^{Out} = .1$	0.620160	0.621735	1.356867
$\eta^{In} = .01, \eta^{RS} = .01, \eta^{Out} = .1$	0.527440	0.523087	1.367918
$\eta^{In} = .01, \eta^{RS} = .1, \eta^{Out} = 0.1$	1.600618	2.288551	5.12505

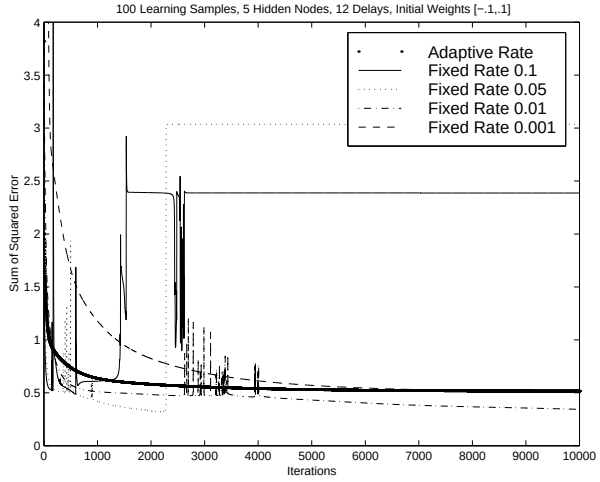

Figure 2: The learning process controlled by different learning rates.

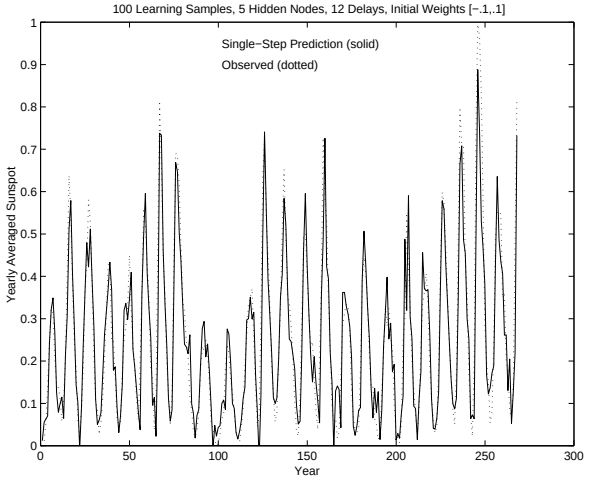

Figure 4: Prediction produced by the network trained with adaptive learning rates.

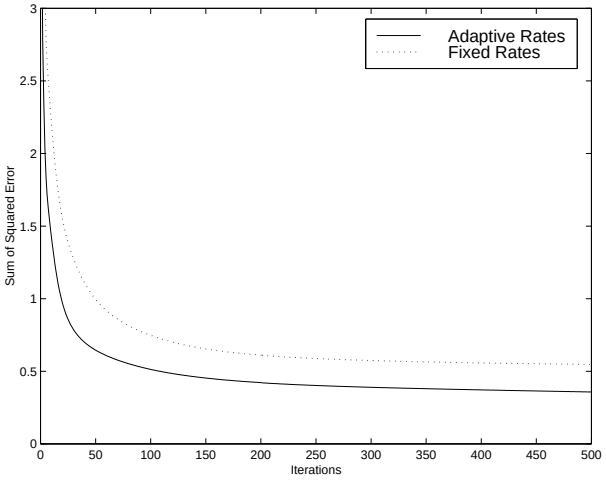

Figure 3: The learning process controlled by different learning rates.

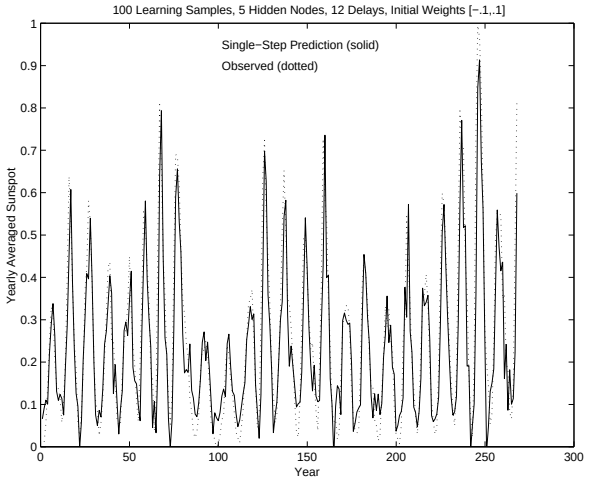

Figure 5: Prediction produced by the network trained with fixed learning rates.