Available at WWW.MATHEMATICSWEB.ORG POWERED BY SCIENCE DIRECT®

PERGAMON Journal of the Franklin Institute 340 (2003) 363–370

Journal of The Franklin Institute

www.elsevier.com/locate/jfranklin

On "an improved approach for nonlinear system identification using neural networks"

Andrew I. Hanna[a,*], Danilo P. Mandic[b]

[a] *Signal and Image Informatics Group, Royal Society Wolfson Bioinformatics Laboratory, University of East Anglia, Norwich NR4 7TJ, UK*

[b] *Communications and Signal Processing Group, Department of Electrical and Electronic Engineering, Imperial College of Science, Technology and Medicine, London, UK*

Received 13 November 2001; received in revised form 1 July 2003; accepted 6 July 2003

Abstract

Nonlinear system identification and prediction is a complex task, and often non-parametric models such as neural networks are used in place of intricate mathematics. To that cause, recently an improved approach to nonlinear system identification using neural networks was presented in Gupta and Sinha (J. Franklin Inst. 336 (1999) 721). Therein a learning algorithm was proposed in which both the slope of the activation function at a neuron, β, and the learning rate, η, were made adaptive. The proposed algorithm assumes that η and β are independent variables. Here, we show that the slope and the learning rate are not independent in a general dynamical neural nétwork, and this should be taken into account when designing a learning algorithm. Further, relationships between η and β are developed which helps reduce the number of degrees of freedom and computational complexity in an optimisation task of training a fully adaptive neural network. Simulation results based on Gupta and Sinha (1999) and the proposed approach support the analysis.

Keywords: Neural networks; Nonlinear adaptive filtering; Adaptive learning rate; Delta-Bar-Delta; Isomorphic networks

1. Introduction

The backpropagation (BP) algorithm is the most widely used learning algorithm for multilayer feedforward neural networks. It is known that the convergence rate of

*Corresponding author. Tel.: +44-1603-592-442; fax: +44-1603-593-345.
E-mail addresses: aih@sys.uea.ac.uk (A.I. Hanna), d.mandic@ic.ac.uk (D.P. Mandic).

0016-0032/$30.00
doi:10.1016/j.jfranklin.2003.07.001

the BP algorithm is slow and the algorithm is susceptible to local minima. Furthermore, for convergence, the learning rate parameter η, in the gradient descent-based algorithm must be constrained according to the eigenanalysis of the correlation matrix of the input signal [1]. To this cause, optimization techniques have been developed that aim to increase the convergence rate and transform the learning algorithm so as to reduce the risk of local minima. Such adaptive algorithms include the momentum algorithm [2], normalised nonlinear gradient adaptive algorithms [3], adaptive slopes in the activation function [4] and various gradient adaptive learning rates [5–7]. Apart from the weights, the two parameters most often chosen to be adaptive are the learning rate, η and the slope of the nonlinear activation function, β.

This paper follows the approach given in [8], and provides an insight into the dependence relationship between the learning rate, η, and the slope of the activation function, β, within the framework of [9], thus reducing the number of degrees of freedom in the model. It is shown that it suffices to adapt only one of the two proposed parameters without any degradation in performance of the algorithm. Experimental results that employ an adaptive β confirm the analysis.

2. The delta-bar-delta with adaptive slope algorithm

Recently, a learning algorithm that uses the delta-bar-delta rule and an adaptive gain, β, in the activation function was proposed [9]. The adaptive learning rate, η, is calculated using Jacob's heuristics [10] to give

$$\Delta\eta(k) = \begin{cases} \kappa & \text{if } S(k-1)D(k) > 0, \\ -\gamma\eta(k-1) & \text{if } S(k-1)D(k) < 0, \\ 0 & \text{otherwise,} \end{cases} \tag{1}$$

where

$$D(k) = \frac{\partial J(k)}{\partial \mathbf{w}(k)}, \tag{2}$$

and

$$S(k) = (1-\xi)D(k) + \xi S(k-1), \tag{3}$$

where ξ and γ are small positive constants and $J(k) = 1/N \sum e^2(k)$, denotes the objective function calculated as the mean squared error. The learning rate update term is increased by a constant, κ or decreased exponentially by a term controlled by γ. This algorithm effectively belongs to the class of sign algorithms [1].

The gain in the activation function in [9] is also being adapted according to a gradient descent-based approach as

$$\begin{aligned} \Delta\beta(k+1) &= -\eta_\beta \frac{\partial J(k)}{\partial \beta(k)} \\ &= \frac{\eta_\beta \delta(k) v(k)}{\beta(k)}, \end{aligned} \tag{4}$$

where η_β denotes the step size of the slope adaptation algorithm, $\delta(k)$ the local gradient and $v(k)$ the input to the nonlinear activation function. Here, notice that both η and β are being adapted. The algorithm proposed in [9] provides important results, as the learning rate for each synaptic weight is adapted according to a variant of the sign algorithm, and the slope of the activation function is adapted using a gradient descent-based approach. This does not provide consistency in the model but helps with the sensitivity of the algorithm compared to the original delta-bar-delta which employs a gradient adaptive-based approach [1]. The proposed algorithm indicates that the two adaptive parameters η and β are independent, which we address here.

The purpose of this paper is to show that the two parameters in [9] that have been made adaptive and assumed to be independent are connected by (for a comprehensive proof, consult Appendix A) [8]

$$\eta^{\mathrm{R}} = \eta\beta^2, \tag{5}$$

where η^{R} is the learning rate in a referent network for $\beta^{\mathrm{R}} = 1$. The fully adaptive algorithm proposed in [9] is a rigorous algorithm, as the experimental results in the paper clearly show. However, the system is dependent on two initial conditions, η_0 and β_0. Reducing the number of degrees of freedom from the model will result in some drift in performance due to the truncation of learning rates at each neuron. It is obvious that the relationship between η and β is strong for models with a single learning rate and gain at every neuron. However in the delta-bar-delta algorithm, each synaptic weight has its own learning rate increasing computation complexity, and sensitivity [1].

3. Sensitivity analysis and discussion

The proposed algorithm in [2] improves on the previously derived gradient adaptive delta-bar-delta algorithm, which is sensitive in the transient state, by adopting a sign variant alternative for the adaptation of the learning rate η. However, a learning rate at each weight within the neural network still results in a sensitive model that is computationally expensive. This paper shows that the proposed algorithm in [9] can be reduced in complexity by adopting the adaptive slope in the activation function as defined in Eq. (4) without a noticeable decrease in performance. However, we have highlighted the relationship between the learning rate, η, and the slope of the activation function, β. It can also be shown that adopting an adaptive learning rate results in a less sensitive neural network. To demonstrate this, consider a simple dynamical neuron where $\Phi(\cdot)$ denotes the nonlinearity at the neuron and $\mathbf{x}(k) = [x_1(k), x_2(k), \ldots, x_N(k)]^{\mathrm{T}}$ the input to the neuron for which the weight update algorithm is given by

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \eta e(k)\beta\Phi'(\beta\mathbf{x}^{\mathrm{T}}(k)\mathbf{w}(k))\mathbf{x}(k). \tag{6}$$

Therefore, for small η, the sensitivity of the weight update $\Delta\mathbf{w}(k)$ to η can be expressed as

$$\frac{\partial\Delta\mathbf{w}(k)}{\partial\eta} = \beta e(k)\Phi'(\beta\mathbf{x}^{\mathrm{T}}(k)\mathbf{w}(k))\mathbf{x}(k), \tag{7}$$

whereas the sensitivity of $\Delta\mathbf{w}(k)$ to β is given by

$$\frac{\partial\Delta\mathbf{w}(k)}{\partial\beta} = \eta e(k)\mathbf{x}(k)[\Phi'(\beta\mathbf{x}^{\mathrm{T}}(k)\mathbf{w}(k)) + \beta^2\Phi''(\beta\mathbf{x}^{\mathrm{T}}(k)\mathbf{w}(k))\mathbf{x}^{\mathrm{T}}(k)\mathbf{w}(k)]. \tag{8}$$

From Eqs. (7) and (8) it is clear that changing either the learning rate or the slope of the activation function has a direct effect on the other. Generally, for reasonable ranges of β and η the weight update algorithm is less sensitive if we choose to update the learning rate.

4. Experimental results

To show the relationship between the learning rate, η and gain β the same experiment was performed on two different feedforward neural networks. The task involves prediction of a speech signal, shown in Fig. 1. The first neural network was initialised with $\beta = 1$, $\mathbf{w}_0$, and $\eta = 0.3$ and the second neural network was initialised with $\beta = 2$, $\mathbf{w}_0/\beta$, and $\eta = 0.075$, preserving dynamical relationship in Eq. (5). Fig. 2(a) shows the prediction error for the first neural network, and Fig. 2(b) shows the prediction error for the second neural network. Careful examination shows that both the prediction errors are identical at the same time instant, thus confirming the relationship between η and β.

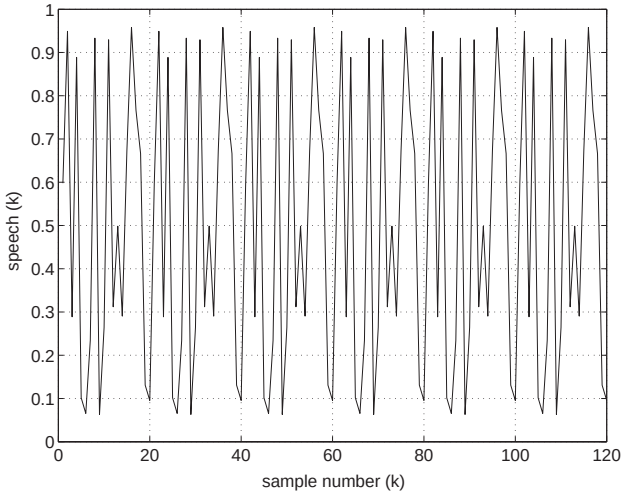

Fig. 1. Speech signal.

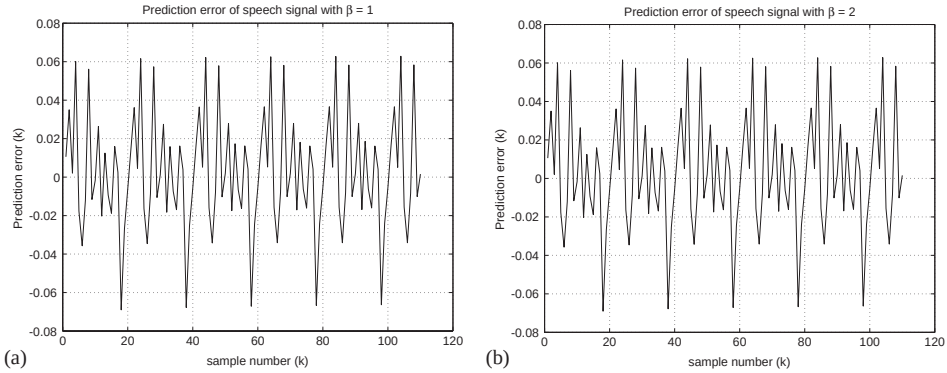

Fig. 2. Prediction errors for (a) $\beta = 1$ and (b) $\beta = 2$. (a) Prediction error with $\beta = 1$. (b) Prediction error with $\beta = 2$.

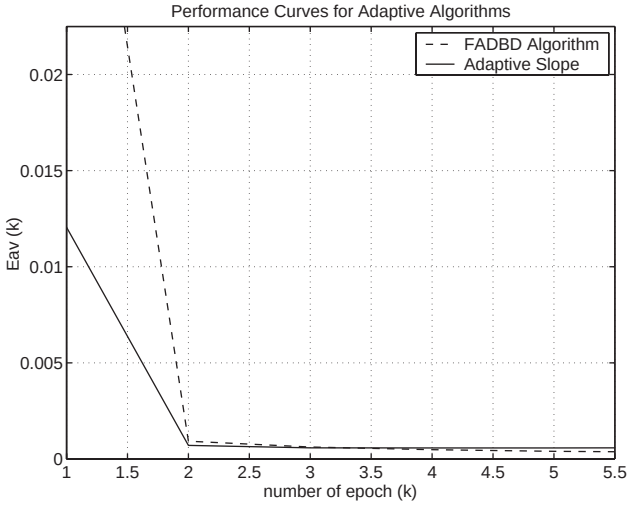

Fig. 3. Comparison of adaptive algorithms.

The second experiment was prediction of a known input signal, $u(k)$, defined by $u(k) = \sin(2\pi k/250)$, $1 \leqslant k \leqslant 500$ with the fully adaptive delta-bar-delta (FADBD) algorithm given in [9] and the adaptive slope part of the FADBD algorithm as defined by Eq. (4) with $\eta_\beta = 0.5$, $\xi = 0.7$, $\eta_0 = 0.3$, $\beta_0 = 1$ and $\kappa = 0.01$. It is clear from Fig. 3 that the performance of the backpropagation with adaptive slope (BPAS) algorithm closely matches that of the FADBD algorithm in [9]. The performance curves show some drift that is accounted for by the truncation of learning rates in the BPAS algorithm, however the computation complexity is greatly reduced.

5. Conclusions

The relationship between the learning rate, η and the slope of the activation function β has been highlighted in the framework of the results from [9], which uses an adaptive learning rate together with an adaptive slope in order to achieve better identification of nonlinear systems. It has been shown that only either the learning rate or the slope need be made adaptive thus reducing the number of degrees of freedom and computational complexity. This way the sensitivity and stability of the delta-bar-delta algorithm have been improved. Experimental results have shown the performance of the proposed algorithm consistent with that in [9].

Appendix A

To show the relationship between η and β, two isomorphic feedforward neural networks are used with outputs $y(k)$ and $y^{\mathrm{R}}(k)$.[1] In the analysis it must be shown that

$$y(k) = y^{\mathrm{R}}(k) \Leftrightarrow \Phi(net(k)) = \Phi^{\mathrm{R}}(net^{\mathrm{R}}(k)), \tag{A.1}$$

where $\mathbf{x}(k) = [x_1(k), x_2(k), \ldots, x_N(k)]^{\mathrm{T}}$ denotes the tap input, $\mathbf{w}(k) = [w_1(k), w_2(k), \ldots, w_N(k)]^{\mathrm{T}}$ the weight vector, $\Phi(\cdot)$ the nonlinearity at the activation function and $net(k) = \mathbf{x}^{\mathrm{T}}(k)\mathbf{w}(k)$. Following the approach in [8] we can show that,

$$\begin{aligned}
&\Phi(net(k)) = \Phi^{\mathrm{R}}(net^{\mathrm{R}}(k)) \\
&\quad \Leftrightarrow \Phi^{\mathrm{R}}(\beta net(k)) = \Phi^{\mathrm{R}}(net^{\mathrm{R}}(k)) \\
&\quad \Leftrightarrow \beta net(k) = net^{\mathrm{R}}(k) \\
&\quad \Leftrightarrow \beta(\mathbf{x}^{\mathrm{T}}(k)\mathbf{w}(k)) = \mathbf{x}^{\mathrm{T}}(k)\mathbf{w}^{\mathrm{R}}(k) \\
&\quad \Leftrightarrow \beta\mathbf{w}(k) = \mathbf{w}^{\mathrm{R}}(k),
\end{aligned} \tag{A.2}$$

leaving the term $\beta\mathbf{w}(k) = \mathbf{w}^{\mathrm{R}}(k)$. Since $\mathbf{w}(k) = \mathbf{w}(k-1) + \Delta\mathbf{w}(k-1)$ it follows that,

$$\beta\Delta\mathbf{w}(k) = \Delta\mathbf{w}^{\mathrm{R}}(k). \tag{A.3}$$

Knowing that $\Delta\mathbf{w}(k) = \eta\delta(k)\mathbf{x}(k)$, where $\delta(k)$ denotes the local gradient, it can be shown that

$$\begin{aligned}
&\beta\eta\delta(k)\mathbf{x}(k) = \eta^{\mathrm{R}}\delta^{\mathrm{R}}(k)\mathbf{x}(k) \\
&\quad \Leftrightarrow \beta\eta\delta(k) = \eta^{\mathrm{R}}\delta^{\mathrm{R}}(k) \\
&\quad \Leftrightarrow \beta\eta\delta(k) = \beta^2\eta\delta^{\mathrm{R}}(k) \\
&\quad \Leftrightarrow \delta(k) = \beta\delta^{\mathrm{R}}(k).
\end{aligned} \tag{A.4}$$

There are now two cases for $\delta(k) = \beta\delta^{\mathrm{R}}(k)$, the first is in the output layer and the second is in the hidden layer. Letting the subscript $(\cdot)_{\mathrm{L}}(k)$ denote parameters in the

[1] The superscript $(\cdot)^{\mathrm{R}}$ denotes the terms in the referent network where $\beta^{\mathrm{R}} = 1$.

output layer and $(\cdot)_l(k)$ denote the parameters in the hidden layer we state,

$$\delta_{\mathrm{L}}(k) = \Phi'(net_{\mathrm{L}}(k))e(k), \tag{A.5}$$

$$\delta_{l,i}(k) = \Phi'(net_{l,i}(k)) \sum_{j=1}^{N_{l+1}} \delta_{l+1,j} w_{l+1,i,j}, \tag{A.6}$$

where N_l denotes the number of neurons in layer l and $w_{l,i,j}$ the weight connecting neuron i in layer $(l-1)$ to neuron j in layer l. For a neuron in the output layer, it stands that,

$$\beta_{\mathrm{L}} \Phi'^{\mathrm{R}}(net_{\mathrm{L}}^{\mathrm{R}}(k))e^{\mathrm{R}}(k) = \Phi'(net_{\mathrm{L}}(k))e(k). \tag{A.7}$$

From Eq. (A.1) it follows that $e^{\mathrm{R}}(k) = e(k)$ to give,

$$\begin{aligned} &\beta_{\mathrm{L}} \Phi'^{\mathrm{R}}(net_{\mathrm{L}}^{\mathrm{R}}(k)) = \Phi'(net_{\mathrm{L}}(k)) \\ &\quad \Leftrightarrow \beta_{\mathrm{L}} \Phi'^{\mathrm{R}}(\beta_{\mathrm{L}} net(k)) = \Phi'(net_{\mathrm{L}}(k)) \\ &\quad \Leftrightarrow \Phi^{\mathrm{R}}(\beta_{\mathrm{L}} net(k)) = \Phi(net_{\mathrm{L}}(k)). \end{aligned} \tag{A.8}$$

For a neuron not in the output layer we obtain,

$$\begin{aligned} &\beta_l \delta_l^{\mathrm{R}}(k) = \delta_l(k) \\ &\quad \Leftrightarrow \beta_l \Phi'^{\mathrm{R}}(net_l^{\mathrm{R}}(k)) \sum_{i=1}^{N_{l+1}} \delta_{l+1}^{\mathrm{R}}(k) \mathbf{w}_{l+1}^{\mathrm{R}}(k) \\ &\quad = \Phi'(net_l(k)) \sum_{i=1}^{N_{l+1}} \delta_{l+1}(k) \mathbf{w}_{l+1}(k). \end{aligned} \tag{A.9}$$

We have already shown in Eq. (A.8) that $\beta_{\mathrm{L}} \Phi'^{\mathrm{R}}(net_{\mathrm{L}}^{\mathrm{R}}(k)) = \Phi'(net_{\mathrm{L}}(k))$ so we are left with,

$$\begin{aligned} &\sum_{i=1}^{N_{l+1}} \delta_{l+1}^{\mathrm{R}}(k) \mathbf{w}_{l+1}^{\mathrm{R}}(k) = \sum_{i=1}^{N_{l+1}} \delta_{l+1}(k) \mathbf{w}_{l+1}(k) \\ &\quad \Leftrightarrow \sum_{i=1}^{N_{l+1}} \delta_{l+1}^{\mathrm{R}}(k) \beta_{l+1} \mathbf{w}_{l+1} = \sum_{i=1}^{N_{l+1}} \delta_{l+1}(k) \mathbf{w}_{l+1}(k), \end{aligned} \tag{A.10}$$

therefore $\delta_{l+1}^{\mathrm{R}}(k)\beta_{l+1} = \delta_{l+1}(k)$. Returning to the weight update algorithm we get

$$\begin{aligned} \mathbf{w}^{\mathrm{R}}(k) &= \mathbf{w}^{\mathrm{R}}(k) + \Delta \mathbf{w}^{\mathrm{R}}(k) \\ &= \mathbf{w}^{\mathrm{R}}(k) + \beta \Delta \mathbf{w}(k) \\ &= \mathbf{w}^{\mathrm{R}}(k) + \beta \eta \delta(k) \mathbf{x}(k) \\ &= \mathbf{w}^{\mathrm{R}}(k) + \beta \eta \beta \delta^{\mathrm{R}}(k) \mathbf{x}(k) \\ &= \mathbf{w}^{\mathrm{R}}(k) + \eta^{\mathrm{R}} \delta^{\mathrm{R}}(k) \mathbf{x}(k). \end{aligned} \tag{A.11}$$

From the last two lines of Eq. (A.11) it can be shown that two isomorphic feedforward neural networks behave the same in the dynamic sense if

$$\eta^{\mathrm{R}} = \eta \beta^2 \tag{A.12}$$

thus the relationship of learning rate and gain in a feedforward neural network has been established.

References

[1] S. Haykin, Adaptive Filter Theory, 3rd Edition, Prentice-Hall, Englewood Cliffs, NJ, 1996.
[2] M. Moreira, E. Fiesler, Neural networks with adaptive learning rate and momentum terms, IDIAP Technical Report 04 (95) (1995).
[3] A.I. Hanna, D.P. Mandic, M. Razaz, A normalised backpropagation learning algorithm for multilayer feed-forward neural adaptive filters, Proceedings of the XI IEEE Workshop on Neural Networks for Signal Processing, 2001, pp. 63–72, Falmouth, Massachusetts, USA.
[4] D.P. Mandic, I.R. Krcmar, On training with slope adaptation for feedforward neural networks, Proceedings of the Fifth IEEE Seminar on Neural Network Applications in Electrical Engineering (NEUREL-2000), 2000, pp. 42–45, Belgrade, Yugoslavia.
[5] D.P. Mandic, A.I. Hanna, M. Razaz, A normalised gradient descent algorithm for nonlinear adaptive filters using a gradient adaptive step size, IEEE Signal Process. Lett. 8 (11) (2001) 295–297.
[6] W. Ang, B. Farhang-Boroujeny, A new class of gradient adaptive step-size LMS algorithms, IEEE Trans. Signal Process. 49 (4) (2001) 805–810.
[7] V.J. Mathews, Z. Xie, Stochastic gradient adaptive filter with gradient adaptive step size, IEEE Trans. Signal Process. 41 (6) (1993) 2075–2087.
[8] G. Thimm, P. Moerland, E. Fiesler, The interchangeability of learning rate and gain in backpropagation neural networks, Neural Comput. 8 (2) (1996) 451–460.
[9] P. Gupta, N. Sinha, An improved approach for nonlinear system identification using neural networks, J. Franklin Inst. 336 (1999) 721–734.
[10] R.A. Jacobs, Increased rates of convergence through learning rate adaptation, Neural Networks 1 (1988) 295–307.