A NORMALIZED GRADIENT ALGORITHM FOR AN ADAPTIVE RECURRENT PERCEPTRON

Jonathon A. Chambers, Warren Sherliker

Department of Electrical and Electronic Engineering,
Imperial College,
London, UK.
{j.chambers,w.sherliker}@ic.ac.uk

Danilo P. Mandic

School of Information Systems,
University of East Anglia,
Norwich, UK.
d.mandic@uea.ac.uk

ABSTRACT

A normalized algorithm for on–line adaptation of a recurrent perceptron is derived. The algorithm builds upon the normalized backpropagation (NBP) algorithm for feedforward neural networks, and provides an adaptive learning rate and normalization for a recurrent perceptron learning algorithm. The algorithm is based upon local linearization about the current point in the state–space of the network. Such a learning rate is normalized by the squared norm of the gradient at the neuron, which extends the notion of normalized linear algorithms to the nonlinear case.

1. INTRODUCTION

A general class of Least Mean Square (LMS) based nonlinear algorithms can be expressed as

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \eta(k) F\left(\mathbf{x}(k)\right) g\left(\mathbf{w}(k), \mathbf{x}(k)\right) \quad (1)$$

where $\mathbf{w}(k) = [w_1(k), \ldots, w_N(k)]^T$ is the weight vector, $\eta(k)$ is the learning rate, $\mathbf{x}(k) = [x_1(k), \ldots, x_N(k)]^T$ is an input vector, k is the discrete time index, $F : \mathbb{R}^N \rightarrow \mathbb{R}^N$ usually consists of N copies of the scalar function f, and $g(\cdot)$ is typically a scalar function of the error e_k. Function F is related to data nonlinearities, which can impede the convergence of the algorithm. Function g is related to error nonlinearities, and it affects the cost function to be minimized. Error nonlinearities are mostly sign–preserving [1].
Numerous algorithms have been developed in order to improve the convergence of the LMS–based algorithms. The most popular variant of the LMS algorithm which uses an adaptive step size is the Normalized LMS (NLMS) algorithm. Its derivation involves optimization by the method of Lagrange multipliers.
For nonlinear systems, learning algorithms with an adaptive learning rate are most desirable [2]. Among them, annealing algorithms use a search–then–converge technique [3]. Backpropagation algorithms with an adaptive learning rate include algorithms with Lipschitz continuous constraints [4], and algorithms based upon the first and second order derivatives.
A class of linearized algorithms for nonlinear adaptive Volterra and Myriad filters has been proposed in [5]. For neural networks, a backpropagation algorithm based upon local linearization of the state space equations is the Normalized Backpropagation (NBP) algorithm [6]. In [7], normalization is provided using the Lagrange multiplier method applied to the state space of the network. Neural networks for on–line learning use direct gradient algorithms, such as the Real Time Recurrent Learning algorithm (RTRL).
Hence, there is a need for a real–time gradient based algorithm for a recurrent perceptron, with an adaptive learning rate which would impose stability and convergence similar to that produced by the NLMS for linear filters. We derive such an algorithm, which is locally optimal, in the sense that it minimizes the instantaneous squared error at the output of a recurrent perceptron, based upon the local properties of linearization provided via Taylor series expansion. Experimental results are presented to support the analysis.

2. ASPECTS OF LINEARIZATION

Linearization is used in order to examine the stability of the algorithm (1), where, in the case of the RTRL algorithm, F is an identity matrix, and g is some nonlinear, sign–preserving function of the output error. The linearization ought to be time–varying, due to the external input signal $\mathbf{x}$. An assumption that the learning rate η is sufficiently small to allow the algorithm to be linearized around its current point in the state space is also needed. The idea itself is not new. From Lyapunov stability theory, system

$$\mathbf{z}(k+1) = F(k, \mathbf{z}(k)) \quad (2)$$

can be analyzed via its linearized version

$$\mathbf{z}(k+1) = \mathbf{A}(k)\mathbf{z}(k) \quad (3)$$

0-7803-6293-4/00/$10.00©2000 IEEE

where $\mathbf{A}$ is the Jacobian of F and k denotes discrete time. This is the Lyapunov indirect method, and assumes that $\mathbf{A}(k)$ is bounded in the neighborhood of the current point in the state space, and that

$$\lim_{\|\mathbf{z}\|\to 0} \max_k \frac{\| F(k,\mathbf{z}) - A_k\mathbf{z} \|}{\| \mathbf{z} \|} = 0 \qquad (4)$$

which guarantees that time variation in the nonlinear terms of the Taylor series does not become arbitrarily large in time. Function g preserves the local nature of the results, and $\| \cdot \|$ denotes an arbitrary norm.

3. RECURRENT PERCEPTRON

A general structure of a recurrent perceptron is shown in Figure 1. The equations which describe the recurrent perceptron are

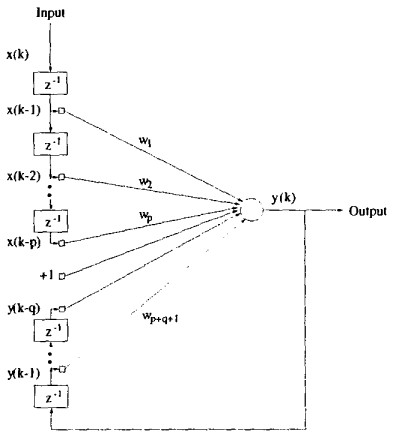


Figure 1: NARMA(p,q) recurrent perceptron

$$\begin{aligned} y(k) &= \Phi\left(v(k)\right) \\ v(k) &= \mathbf{w}(k)^T \cdot \mathbf{u}(k), \end{aligned} \qquad (5)$$

where $\mathbf{u}(k) = [y(k-1), \ldots, y(k-q), 1, x(k-1), \ldots, x(k-p)]^T$. The Bounded Input Bounded Output (BIBO) stability of (5) is preserved due to the saturation type nonlinearity Φ in (5), which is typically the logistic function $\Phi(v) = \frac{1}{1+e^{-\beta v}}$, with slope β.
The problem of gradient based training can be set as [8]

$$\begin{aligned} &\text{minimize} && \| \mathbf{w}(k+1) - \mathbf{w}(k) \|_p && (6) \\ &\text{subject to} && s(k) - \Phi\left(\mathbf{w}^T(k+1)\mathbf{u}(k)\right) = 0 && (7) \end{aligned}$$

where $\| \cdot \|_p$ denotes the $\mathcal{L}_p$ norm, and $s(k)$ is some teaching (desired) signal. The equations that define the adaptation of a recurrent perceptron are

$$\begin{aligned} e(k) &= s(k) - \Phi(v(k)) && (8) \\ \mathbf{w}(k+1) &= \mathbf{w}(k) - \eta \nabla_{\mathbf{w}(k)} e^2(k) && (9) \end{aligned}$$

where $e(k)$ is the instantaneous error at the output neuron, and $e^2(k)$ is the cost function to be minimized. The correction to the weight vector of the neuron, at the time instant k becomes [9]

$$\Delta\mathbf{w}(k) = 2\eta(k)e(k)\mathbf{\Pi}(k) \qquad (10)$$

where $\mathbf{\Pi}(k) = \left[\frac{\partial y(k)}{\partial w_1(k)}, \ldots, \frac{\partial y(k)}{\partial w_N(k)}\right]^T$ represents the gradient vector at the output of the neuron.

4. DERIVATION OF THE ALGORITHM

Notice that the weight vectors consist of two subvectors, namely $\mathbf{w} = [\mathbf{w}_a, \mathbf{w}_b]^T$, where $\mathbf{w}_a$ represents the weights associated with the feedback inputs, whereas the weights with the index b correspond to the weights associated with the external and bias inputs. Hence, the weight vector $\mathbf{w}$ can be split into two subvectors, $\mathbf{w}_a$ and $\mathbf{w}_b$.
In order to minimize the a posteriori error (7) via an adaptive learning rate, consider Taylor expansion of the instantaneous error

$$\begin{aligned} e(k+1) &= e(k) + \sum_{i=1}^{N} \frac{\partial e(k)}{\partial w_i(k)} \Delta w_i(k) \\ &+ \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \frac{\partial^2 e(k)}{\partial w_i(k) \partial w_j(k)} \Delta w_i(k) \Delta w_j(k) + \cdots \end{aligned} \qquad (11)$$

This approach to normalization of gradient adaptive algorithms for linear systems is provided in [10]. In linear systems, the terms after the second term vanish due to linearity of the system. In nonlinear systems, generally, many terms of the Taylor series are not negligible. However, as our intention is to provide a local linearization around the current point in the state–space of the network, and knowing that the logistic function behaves approximately linearly in the areas that are not saturated, we will also take into account only the first two terms of the expansion (11). It also helps to reduce the computational load associated with calculating higher order derivatives of (11). After neglecting the higher order terms in the Taylor series expansion (11), we have

$$\begin{aligned} e(k+1) &= e(k) - 2\eta(k)e(k) \sum_{i=1}^{N} \left[\frac{\partial y(k)}{\partial w_i(k)}\right]^2 \\ &= e(k) - 2\eta(k)e(k) \| \mathbf{\Pi}(k) \|_2^2 \end{aligned} \qquad (12)$$

The instantaneous squared error to be minimized is hence given by

$$e^2(k+1) = e^2(k) \left[1 - 2\eta(k) \| \mathbf{\Pi}(k) \|_2^2\right]^2 \qquad (13)$$

Differentiating with respect to η, the optimal value of learning rate $\eta_{OPT}(k)$ for a direct gradient trained recurrent perceptron becomes

$$\eta_{OPT}(k) = \frac{1}{2 \parallel \mathbf{\Pi}(k) \parallel_2^2} \tag{14}$$

This is closely related to the learning rate in the NLMS algorithm for linear adaptive filters. However, in order to converge, this algorithm has to be bounded by the corresponding normalized algorithm for Infinite Impulse Response (IIR) linear adaptive filters. Hence, the slope of the activation function β and the learning rate $\eta(k)$ are dependent [11]. For the normalized algorithm (14), to converge, function Φ ought to be a contraction, i.e. for any $a, b \in \mathbb{R}$, $|\Phi(b) - \Phi(a)| \leq |b - a|$.

5. CONVERGENCE OF THE PROPOSED ALGORITHM

From (12), we have

$$\begin{aligned} e(k+1) &= e(k) - 2\eta(k)e(k) \parallel \mathbf{\Pi}(k) \parallel_2^2 \\ &= e(k)\left[1 - 2\eta(k) \parallel \mathbf{\Pi}(k) \parallel_2^2\right] \end{aligned} \tag{15}$$

It is desirable that $|e(k)| \to 0$ as $k \to \infty$, which gives

$$\begin{aligned} |e(k+1)| &= \left|e(k)\left[1 - 2\eta(k) \parallel \mathbf{\Pi}(k) \parallel_2^2\right]\right| \\ &\leq |e(k)| \cdot \left|1 - 2\eta(k) \parallel \mathbf{\Pi}(k) \parallel_2^2\right| \end{aligned} \tag{16}$$

This will converge *uniformly* if and only if

$$\left|1 - 2\eta(k) \parallel \mathbf{\Pi}(k) \parallel_2^2\right| < 1 \tag{17}$$

which is a contractive behaviour.

6. MORE ON CONVERGENCE

Consider the error equation $e(k) = s(k) - \Phi\left(\mathbf{w}(k)^T\mathbf{u}(k)\right)$, but assume

$$s(k) = q(k) + \Phi\left(\tilde{\mathbf{w}}^T\mathbf{u}(k)\right) \tag{18}$$

where $\tilde{\mathbf{w}}(k)$ are optimal filter weights (not time varying). It follows then that

$$e(k) = q(k) + \Phi\left(\tilde{\mathbf{w}}^T\mathbf{u}(k)\right) - \Phi\left(\mathbf{w}^T(k)\mathbf{u}(k)\right) \tag{19}$$

Consider again the weight equation update

$$\mathbf{w}(k+1) = \mathbf{w}(k) + 2\eta(k)e(k)\mathbf{\Pi}(k) \tag{20}$$

From (19) and (20), we have

$$\begin{aligned} \mathbf{w}(k+1) &= \mathbf{w}(k) + 2\eta(k)q(k)\mathbf{\Pi}(k) \\ &+ 2\eta(k)\Phi\left(\tilde{\mathbf{w}}^T(k)\mathbf{u}(k)\right)\mathbf{\Pi}(k) \\ &- 2\eta(k)\Phi\left(\mathbf{w}^T(k)\mathbf{u}(k)\right)\mathbf{\Pi}(k) \end{aligned} \tag{21}$$

The misalignment vector can be expressed as

$$\mathbf{v}(k) = \mathbf{w}(k) - \tilde{\mathbf{w}} \tag{22}$$

Now, subtract $\tilde{\mathbf{w}}$ from either side of (21), so we have

$$\begin{aligned} \mathbf{v}(k+1) &= \mathbf{v}(k) + 2\eta(k)q(k)\mathbf{\Pi}(k) \\ &-2\eta(k)\left[\Phi\left(\mathbf{w}^T(k)\mathbf{u}(k)\right) - \Phi\left(\tilde{\mathbf{w}}^T\mathbf{u}(k)\right)\right]\mathbf{\Pi}(k) \end{aligned} \tag{23}$$

For Φ a contraction mapping the term in the square brackets from (23) is bounded from above by $\alpha\left|\mathbf{u}^T(k)\mathbf{v}(k)\right|$, $0 < \alpha \leq 1$. Further analysis towards the weight convergence becomes rather involved because of the nature of $\mathbf{\Pi}$. Let us denote $\mathbf{w}^T(k)\mathbf{u}(k) = net(k)$. Since

$$\mathbf{\Pi}(k) = \frac{\partial y(k)}{\partial \mathbf{w}(k)} = \Phi'\left(net(k)\right)\left[\mathbf{u}(k) + \mathbf{w}_a^T(k)\mathbf{\Pi}_a(k)\right]$$

let us restrict ourselves to an approximation

$$\mathbf{\Pi}(k) \to \Phi'\left(\mathbf{w}^T(k)\mathbf{u}(k)\right)\mathbf{u}(k)$$

This does not affect the generality of the result, since it is possible to return to the $\mathbf{\Pi}$ terms, after the convergence result is obtained. In fact, this approximation of $\mathbf{\Pi}$ resembles a single–layer, single–neuron feedforward normalized algorithm [6]. Therefore

$$\begin{aligned} \mathbf{v}(k+1) &\leq \mathbf{v}(k) + 2\eta(k)q(k)\Phi'\left(net(k)\right)\mathbf{u}(k) \\ &- 2\eta(k)\mathbf{u}^T(k)\mathbf{v}(k)\alpha\Phi'\left(net(k)\right)\mathbf{u}(k) \end{aligned} \tag{24}$$

For a contractive activation function, $\Phi'\left(net(k)\right)$ is also bounded as $0 < |\Phi'(net(k))| \leq 1$ [12], and can be replaced by $\Phi'(\cdot) < \gamma \leq 1$. We do not take into account the algebraic sign of Φ', because of the sign–preserving nature of the nonlinear function Φ. Now (24) becomes

$$\begin{aligned} \mathbf{v}(k+1) &\leq \mathbf{v}(k) + 2\eta(k)q(k)\mathbf{u}(k)\gamma \\ &- 2\eta(k)\mathbf{u}(k)\mathbf{u}^T(k)\mathbf{v}(k)\gamma\alpha \end{aligned} \tag{25}$$

If we now include the zero mean noise assumption, and the *independence* assumption between η, $\mathbf{u}$, and $\mathbf{v}$, we have

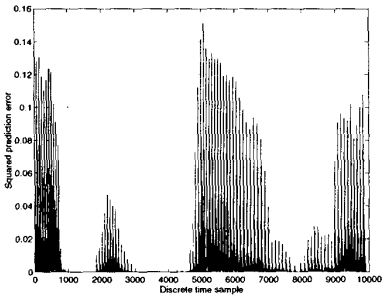
$$E\left[\mathbf{v}(k+1)\right] = E\left[\mathbf{v}(k)\right]E\left[\mathbf{I} - 2\gamma\eta\mathbf{u}(k)\mathbf{u}^T(k)\alpha\right] \tag{26}$$

where $E[\cdot]$ is the expectation operator. For convergence, $0 < E\left[\parallel \mathbf{I} - 2\gamma\eta(k)\mathbf{u}(k)\mathbf{u}^T(k)\alpha \parallel\right] < 1$, which for the upper limit of α and γ gives $0 < \eta(k) < E\left[\frac{1}{\mathbf{u}^T(k)\mathbf{u}(k)}\right]$[1]. This means that the NLMS algorithm is the upper bound for the simplified recurrent perceptron algorithm analyzed. Also, by continuity, an IIR version of the NLMS algorithm is the bound for the single–neuron RTRL algorithm. The mean square and steady state mean square convergence analysis follows the same form.

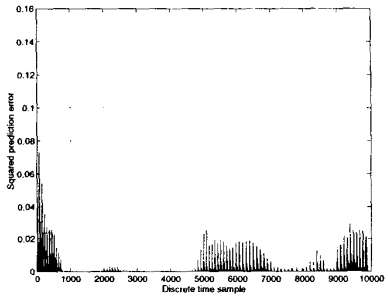
[1] Using the independence assumption, $\mathbf{u}(k)\mathbf{u}^T(k)$ is a diagonal matrix, and $E\left[\parallel \mathbf{u}(k)\mathbf{u}^T(k) \parallel\right]$ can be replaced by its trace $\mathbf{u}^T(k)\mathbf{u}(k)$.

7. EXPERIMENTAL RESULTS

Figure 2 shows the comparison of instantaneous squared prediction errors for the RTRL and NRTRL for a nonstationary (speech) signal. The NRTRL algorithm from Figure



(a) Standard RTRL algorithm



(b) Normalized RTRL algorithm

Figure 2: Squared instantaneous prediction errors for the RTRL and NRTRL algorithms

2(b), clearly achieves significantly better performance than the RTRL algorithm (Figure 2(a)). To quantify this, if the measure of performance is the standard prediction gain, the NRTRL achieved 7 dB better performance than the RTRL algorithm.

8. SUMMARY

A normalized version of the Real Time Recurrent Learning (RTRL) algorithm has been analyzed. This has been achieved via local linearization around the current point in the state space, as is common in stability theory in automatic control. Such an algorithm provides an adaptive learning rate normalized by the 2–norm of the gradient vector at the neuron. Experimental results on a nonstationary signal support the analysis.

9. REFERENCES

[1] W. A. Sethares, "Adaptive algorithms with nonlinear data and error functions," *IEEE Transactions on Signal Processing*, vol. 40, no. 9, pp. 2199–2206, 1992.

[2] S. C. Douglas and A. Cichocki, "On–line step–size selection for training of adaptive systems," *IEEE Signal Processing Magazine*, vol. 14, no. 6, pp. 45–46, 1997.

[3] H. Fang, G. Gong, and M. Qian, "Annealing of iterative stochastic schemes," *SIAM Journal of Control and Optimization*, vol. 35, pp. 1886–1907, 1997.

[4] G. D. Magoulas, M. N. Vrahatis, and G. S. Androulakis, "Improving the convergence of backpropagation algorithm using learning rate adaptation methods," *Neural Computation*, vol. 11, pp. 1769–1796, 1999.

[5] S. Kalluri and G. R. Arce, "A general class of nonlinear normalized adaptive filtering algorithms," *IEEE Transactions on Signal Processing*, vol. 47, no. 8, pp. 2262–2272, 1999.

[6] D. P. Mandic and J. A. Chambers, "Towards an optimal learning rate for backpropagation," *Neural Processing Letters*, vol. 11, no. 1, pp. 1–5, 1999.

[7] R. Hahnloser, "Learning algorithms based on linearization," *Network: Computation in Neural Systems*, vol. 9, pp. 363–380, 1998.

[8] S. C. Douglas, "A family of normalized LMS algorithms," *IEEE Signal Processing Letters*, vol. 1, no. 3, pp. 49–51, 1994.

[9] R. Williams and D. Zipser, "A learning algorithm for continually running fully recurrent neural networks," *Neural Computation*, vol. 1, pp. 270–280, 1989.

[10] E. Soria-Olivas, J. Calpe-Maravilla, J. F. Guerrero-Martinez, M. Martinez-Sober, and J. Espí-Lopez, "An easy demonstration of the optimum value of the adaptation constant in the LMS algorithm," *IEEE Transactions on Education*, vol. 41, no. 1, p. 81, 1998.

[11] D. P. Mandic and J. A. Chambers, "Relationship between the slope of the activation function and the learning rate for the RNN," *Neural Computation*, vol. 11, no. 5, pp. 1069–1077, 1999.

[12] D. P. Mandic and J. A. Chambers, "Global asymptotic stability of nonlinear relaxation equations realised through a recurrent perceptron," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP–99)*, vol. 2, pp. 1037–1040, 1999.