FAST NONLINEAR ADAPTIVE FILTERING USING A PARTIAL WINDOW CONJUGATE GRADIENT ALGORITHM

A. Neil Birkett and Rafik A. Goubran

Department of Systems and Computer Engineering
Carleton University, 1125 Colonel By Drive
Ottawa, Canada, K1S 5B6
Tel: (613) 788-5747, Fax: (613) 788-5727
birkett@sce.carleton.ca   goubran@sce.carleton.ca

ABSTRACT

In this paper a modified form of the partial conjugate gradient algorithm is presented for use in nonlinear filtering using neural networks.   The algorithm is based on using a gradient average window to provide a trade-off between convergence rate and complexity which, depending on the choice of averaging window, is (in both complexity and speed of convergence) intermediate between the conventional backpropagation (BP) algorithm and the Newton methods.   An additional simplification is introduced by replacing the calculated optimum step size α_k by a normalized step size $\bar{\alpha}$, in the same manner as the Normalized LMS algorithm.   This new algorithm is applied to a cascaded neural network/NLMS structure for the identification of a nonlinear system. This proposed algorithm demonstrates improved convergence rates with even small choices of window size.

FIGURE 1. Nonlinear system identification using a three layer feedforward neural network with an input delay line.

1.0 INTRODUCTION

The limitations of the conventional backpropagation algorithm include the uncertainty of finding the global minimum of the error function and excessively long training times required to obtain a small error output. The later shortcoming, i.e. the slow convergence to either a local or global minimum is the topic addressed in this study. Partial conjugate direction methods [1] can be regarded as being somewhat intermediate between the method of steepest descent (i.e. backpropagation) and Newton's method, in terms of complexity and convergence properties. Thus they give the designer the option of improving the convergence rate at the expense of increased complexity.

2.0 DESCRIPTION OF LEARNING ALGORITHM

Consider a multilayer feedforward network, such as the three layer network of Figure 1 . The basic mechanism behind most supervised learning rules is to update the network weights and bias terms until the mean-squared error between the network output and desired (i.e. target) signal is minimized to below a predetermined level. The error signal at the output of a neuron i at time n is defined by;
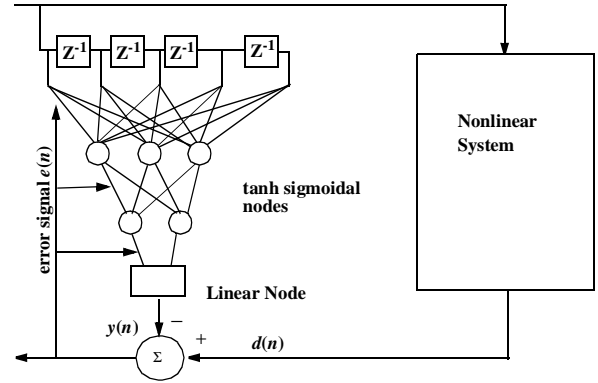
$$e_i(n) = d_i(n) - y_i(n) \qquad (1)$$

The instantaneous cost function E_{inst} at time n is defined as;

$$E_{inst}(n) = e^2(n) = \sum_{i=1}^{N_L} e_i^2(n) \qquad (2)$$

which is the instantaneous sum of squared errors of the network for N_L output nodes, in this case equal to one. We can define an alternate cost functions to be minimized, for example, a *partial* cost function $E_{partial}$ can be calculated by taking a window n_w of past cost functions calculated using the current weight vector $\mathbf{w}(n)$;

$$E_{partial}(n) = \sum_{i=0}^{n_w-1} E_{inst}(n-i)\Big|_{\mathbf{w}(n)} \qquad (3)$$

where $\mathbf{w}(n)$ is a weight vector consisting all the weights in the network, including bias weights. Specifically we may write the supervector $\mathbf{w}(n)$ as;

$$\begin{aligned} &[\mathbf{w}(n)]^T \\ &= [[\mathbf{w}^0(n)]^T, [\mathbf{w}^1(n)]^T, [\mathbf{w}^2(n)]^T, \dots [\mathbf{w}^L(n)]^T] \end{aligned} \qquad (4)$$

where $\mathbf{w}^l(n)$ is the weight vector connecting layer l to layer $l+1$ at time (n).

The windowed conjugate gradient algorithm uses $E_{partial}$ in the calculation of the gradient for updating the weight vector each iteration. It should be noted that it is necessary that the previous values of the hidden layer outputs must be retained as well as the output layers in order to compute the windowed gradient.

The weight update formula minimizes $E_{partial}$ by the delta rule [2];

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \eta \frac{\partial E_{partial}}{\partial \mathbf{w}} \qquad (5)$$

We compute the gradient based on the average squared error of a *window* of training input/output pairs, rather than the complete set of input/output pairs (as is done in the batch training mode) or a single input/output pair (as is done in the individual update backpropagation mode). The errors however are backpropagated to previous layers in the same way as the conventional *backpropagation* [2] BP algorithm. The important point is that the window is moved for each new sample of the input that comes in i.e. it is a *sliding* window of *past* input/output pairs. The proposed algorithm based loosely on a linear version given in [3] is termed the *Windowed Fast Conjugate Gradient Algorithm* (WFCGA) and is summarized below;

Windowed Fast Conjugate Gradient Algorithm:

Initialization: Set weights and biases to random values.
For each iteration n, do *Steps 1 2 and 3*.

Step 1. *a)* Starting with an initial weight vector $\mathbf{w}_0$, compute the following;

$$\mathbf{g}_0 = [\nabla f(\mathbf{w}_0)]^T$$
$$= \left(\frac{2}{n_w}\right)\left[\sum_{i=0}^{n_w-1} \mathbf{g}_{inst}(n-i)\Big|_{\mathbf{W}_0(n),\, \mathbf{X}^0(n-i),\, d(n-i)}\right] \qquad (6)$$

where: $\mathbf{g}_{inst}(n-i)$ is the instantaneous gradient calculated with the current network weight vector $\mathbf{w}_0(n)$ and past inputs $\mathbf{x}^0(n-i)$ and $d(n-i)$. Both $\mathbf{g}_{inst}(n-i)$ and $\mathbf{w}_0(n)$ are vectors of length M, where M is the total number of weights in the network.

b) set $\mathbf{d}_0 = -\mathbf{g}_0$
c) compute the normalized step size parameter α according to;

$$\bar{\alpha} = \frac{\gamma}{\varepsilon + \|\mathbf{x}(n)\|^2} = \frac{\gamma}{\varepsilon + \mathbf{x}^T(n)\mathbf{x}(n)} \qquad (7)$$

Note that α could be replaced by a fixed step size here if desired;

Step 2. Repeat for $k=0,1, . n_w-1$ where $n_w \le m$
a) set $\mathbf{w}_{k+1} = \mathbf{w}_k + \alpha\mathbf{d}_k$
b) Compute an estimate of the gradient at $\mathbf{w}_{k+1}$;

$$\mathbf{g}_{k+1} = [\nabla f(\mathbf{w}_{k+1})]^T$$
$$= \left(\frac{2}{n_w}\right)\left[\sum_{i=0}^{n_w-1} \mathbf{g}_{inst}(n-i)\Big|_{\mathbf{W}_{k+1}(n),\, \mathbf{X}^0(n-i),\, d(n-i)}\right] \qquad (8)$$

c) Unless $k=n_w-1$, set $\mathbf{d}_{k+1} = -\mathbf{g}_{k+1} + \beta_k\mathbf{d}_k$, where;

$$\beta_k = \frac{\mathbf{g}_{k+1}^T\mathbf{g}_{k+1}}{\mathbf{g}_k^T\mathbf{g}_k} \qquad (9)$$

Note that if $\beta_k > 1$, go directly to Step three.
 Repeat *Step 2 a)*.

Step 3. Replace $\mathbf{w}_0$ by $\mathbf{w}_k$ and go back to *Step 1*.

The calculation of the instantaneous gradient $\mathbf{g}_{inst}(n-i)$ in (6) and (8) is done by evaluating (10) through (12) as follows;

$$e(n-i) = d(n-i) - N[\mathbf{W}_{k+1}(n), \mathbf{x}^0(n-i)] \qquad (10)$$

is the network error output using weights $\mathbf{w}_{k+1}(n)$ with input vector $\mathbf{x}^0(n-i)$;

$$g_{ij}^l(n-i) = \delta_j^{l+1}(n-i) \cdot x_i^l(n-i) \qquad (11)$$

is the *local gradient* of a particular weight where;

$$\delta_j^l(n-i)=$$

$$\begin{cases} -2e(n-i)f'(s_j^L(n-i)) & \dots l = L \\ \\ f'(s_j^l(n-i)) \cdot \sum_{k=1}^{N_{L+1}} \delta_k^{l+1}(n-i) \cdot w_{jk}^l(n) & \dots 1 \le l \le L-1 \end{cases} \qquad (12)$$

Note that the vector $\mathbf{g}_{inst}(n-i)$ has the same size as the supervector $\mathbf{w}_k(n)$ and is formed by placing individual $g_{ij}^l(n-i)$ in much the same way that $\mathbf{w}_k(n)$ is formed in (4).

Complexity: The complexity of the WFCGA is $O(mn_w^2)$ since in *Step 2*, the weights are updated n_w times per iteration and the calculation of the averaged gradient is $O(mn_w)$. Thus for $n_w=1$, it is equal in complexity to the BP algorithm.

3.0 SIMULATION

In this section, we apply the WFCGA to the identification of a nonlinear system constructed by generating a signal x which is then hard limited and then convolved with an exponentially decaying 50 tap impulse. The input signal x is obtained by a first order autoregressive (AR) process according to the equation $x(n)=0.9x(n-1)+0.2v(n)$ where $v(n)$ is a unit variance white noise sequence. The hard limiter has a linear region up to 0.5, beyond which the output is clipped with a limiting function which has a slope of 0.2. The neural network consists of a 50 tap input delay line followed by one hidden layer. The system is illustrated in Figure 2.
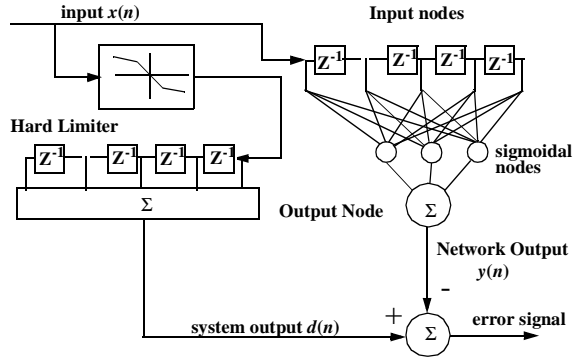
FIGURE 2. System identification model. The system to be identified is a fixed nonlinearity consisting of a linear portion up the value of 0.5 followed by a squashing function of slope =0.2. The output of this nonlinearity is then passed through a dispersive channel consisting of an exponentially decaying random noise impulse of length 50 taps.

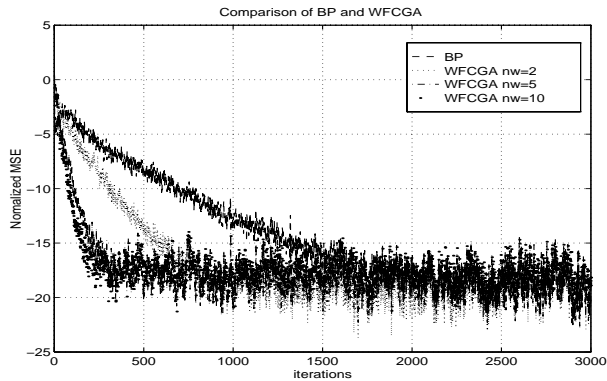The results illustrated in Figure 6 show that for the AR input, the

FIGURE 3. Comparison of the normalized MSE using the BP and WFCGA algorithms with n_w=2, 5 and 10 for the system identification model of Figure 2. A first order autoregressive signal is used to model the input signal x. Two hundred independent trials are used in the averaging process.

WFCGA converges at a rate much faster than the conventional BP algorithm, depending on the size of the gradient averaging window n_w. The larger the choice of n_w, the higher the convergence rate. The final misadjustment is approximately -18dB for all cases.

4.0 APPLICATION TO NONLINEAR ACOUSTIC ECHO CANCELLATION

The specific application addressed here is nonlinear adaptive filtering and system identification, where a short term nonlinearity is followed by a long tail impulse, for example, in nonlinear *acoustic echo cancellation* where a where a nonlinear loudspeaker and reverberant room must be identified. This is illustrated in Figure 4 and is based on the Partial Adaptive Acoustic

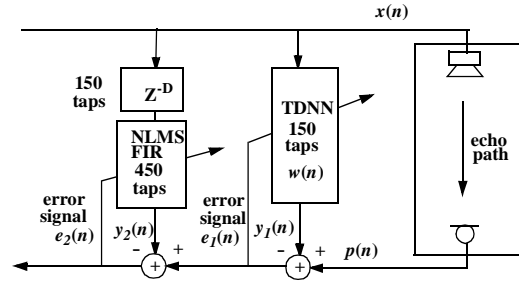Echo Canceller (AEC) structure in Reference [4]. Referring to

FIGURE 4. Adaptive nonlinear AEC using a neural network structure cascaded with a NLMS filter based on the partial adaptive structure. The TDNN cancels the first part of the AIR and an FIR trained with the NLMS cancels the tail portion. Signal $e_2(n)$ is the residual signal left after the echo has been cancelled.

Figure 4, the output $e_2(n)$ is given as;

$$e_2(n) = e_1(n) - y_2(n) = p(n) - y_1(n) - y_2(n) \qquad (13)$$

where $p(n)$ is the microphone (primary) signal, $y_1(n)$ is the output of the TDNN and $y_2(n)$ is the output of the delayed NLMS filter. Expanding, we obtain;

$$e_2(n) = p(n) - y_1(n) - \sum_{i = N_1 + 1}^{N_2} w(n)x(n - i) \qquad (14)$$

where $w(n)$ are the NLMS tap weights and $x(n)$ is the information vector, N_1 is the delay length of the TDNN section and N_2 is the total impulse length. The TDNN consists of 150 taps in the delay line, and 2 and 3 nodes respectively in the 1st and 2nd hidden layers. The NLMS section has 450 taps such that the total impulse response is 600 taps.

The NLMS algorithm [2] is the baseline by which performance of alternative models is measured but it is incapable of reducing nonlinear distortion. A measure of the AEC performance is the Echo Return Loss Enhancement (ERLE) which is defined as;

$$ERLE(dB) = \lim_{N \to \infty} \left[10 \log \frac{E[p^2(n)]}{E[e^2(n)]} \right] \cong 10 \log \left[\frac{\sigma^2_p}{\sigma^2_e} \right] \qquad (15)$$

where σ^2_p and σ^2_e refer to the variances of the primary and error signals respectively and E is the statistical expectation operator.

4.1 Simulation Results

In this section, a 50 tap neural network is employed in the identification of a room impulse constructed by passing nonlinearly distorted noise through a truncated room transfer function similar to the structure shown in Figure 2 . The application of the windowed conjugate gradient algorithm using different window sizes is shown in Figure 5 and is compared to the conventional backpropagation algorithm. The figure illustrates that even mod-

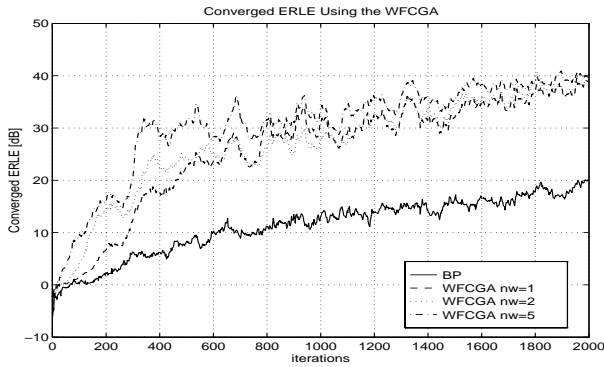est sizes of gradient window can substantially improve convergence speed.

FIGURE 5. Comparison of the BP and WFCGA with nw=1,2, and 5. Substantial improvement in convergence is obtained with even modest window sizes.

4.2 Experimental Results

In this section, a loudspeaker is excited with a filtered noise signal such that a volume of 100 dB sound pressure level (as measured at 0.5m) is obtained. At this level, distortion is produced in the loudspeaker. We apply the nonlinear algorithms in an effort to model the nonlinearity and thus obtain improved ERLE over the linear case. Figure 6 shows the convergence of the network of
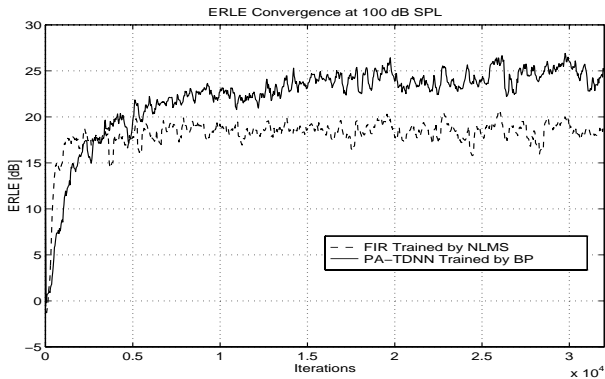

FIGURE 6. Experimental Results. ERLE convergence curves for the partial adaptive TDNN (PA-TDNN) structure and the linear structure at high volume. The nonlinear architecture using BP is capable of achieving improved ERLE performance *but has a slower convergence rate.*

Figure 4 compared to a purely linear filter (updated using the NLMS). The nonlinearities present in the loudspeaker limit the performance of the linear architecture, however, the nonlinear architecture is capable of more accurately modelling the system, however the *initial convergence is slower and this is due to the slow convergence of the conventional backpropagation algorithm*. Now by combining the neural architecture with the

WFCGA, *both* high steady state ERLE and good convergence can be obtained, as shown in Figure 7 .
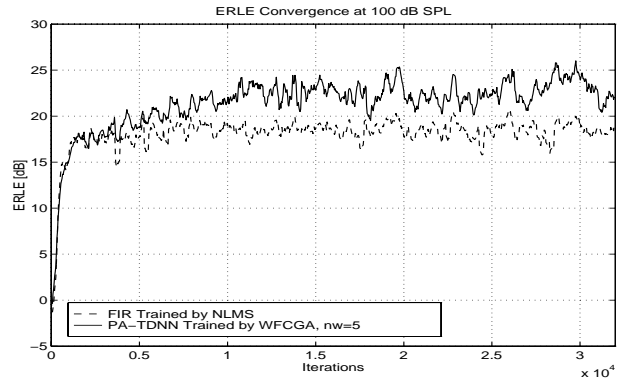

FIGURE 7. Experimental results using the WFCGA. The nonlinear architecture using BP is capable of achieving improved ERLE performance *with an improved convergence rate.*

5.0 SUMMARY

This paper introduces a variant of the partial adaptive gradient algorithm based on using a gradient averaging window and normalized step size to replace the optimum step size. The WFCGA has reduced complexity compared to the full conjugate gradient algorithm and simulations show it has much faster convergence even for low values of gradient averaging window when compared to the conventional BP algorithm. Experimental results using a combined neural network/NLMS structure to identify a loudspeaker/room at high volumes show that the new architecture is capable of improved system identification compared to a linear structure but has slower convergence due to the use of the BP algorithm. The combination of the above architecture with the WFCGA provides both a fast convergence rate and improved system identification in nonlinear environments.

6.0 REFERENCES

[1]   M. R. Hestenes, *Conjugate Direction Methods in Optimization*, Springer-Verlag, 1980.

[2]   S. Haykin, *Neural Networks: A comprehensive foundation*, Macmillan Publishing Co., Englewood Cliffs, NJ: , 1994

[3]   G. K. Boray, M. D. Srinath, "Conjugate Gradient Techniques for Adaptive Filtering", *IEEE Trans. on Circ. and Sys.* Vol. CAS-1, Jan. 1992, pp. 1-10.

[4]   A.N. Birkett, R. A. Goubran, "Acoustic Echo Cancellation Using NLMS-Neural Network Structures", Proceedings *ICASSP95*, pp 3035-3038.

[5]   C. Charlambous, "Conjugate Gradient Algorithms for efficient Training of Artificial Neural Networks", *Proc. IEEE*, Vol. 139, No. 3, pp. 301-310, 1992.

[6]   R. Battiti, "First and second order methods for learning: Between steepest descent and Newtons method", *Neural Computation*, Vol. 4, No. 2, pp 141-166, 1992.