A Normalized Gradient Descent Algorithm for Nonlinear Adaptive Filters Using a Gradient Adaptive Step Size

Danilo P. Mandic, Andrew I. Hanna, and Moe Razaz

***Abstract*—A fully adaptive normalized nonlinear gradient descent (FANNGD) algorithm for online adaptation of nonlinear neural filters is proposed. An adaptive stepsize that minimizes the instantaneous output error of the filter is derived using a linearization performed by a Taylor series expansion of the output error. For rigor, the remainder of the truncated Taylor series expansion within the expression for the adaptive learning rate is made adaptive and is updated using gradient descent. The FANNGD algorithm is shown to converge faster than previously introduced algorithms of this kind.**

***Index Terms*—Adaptive step size, gradient descent algorithms, neural networks, nonlinear adaptive prediction.**

I. Introduction

REAL-TIME signals are often nonstationary and the mathematical models that try to describe them are often very complex, nonlinear, and difficult to derive [1]. With the aid of nonlinear adaptive filters, this need for mathematical complexity is relaxed and replaced by simple nonparametric models, for instance, neural networks employed as nonlinear adaptive filters [2], [3]. It is well known that these adaptive filters are slow in converging to the optimal state [4], [5]. To achieve optimal performance, it is crucial that the learning rate, used in gradient-based training, is able to adjust in accordance with the dynamics of the network input and the neuron nonlinearity. Previously, a normalized nonlinear gradient descent (NNGD) algorithm was derived [6] to improve the convergence of a simple nonlinear FIR filter realized as a feedforward dynamical neuron, shown in Fig. 1.

The equations that describe the adaptation of such a filter are given by [2]

$$e(k) = d(k) - \Phi\left(\mathbf{x}^T(k)\mathbf{w}(k)\right) \tag{1}$$

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \eta(k)\Phi'\left(\mathbf{x}^T(k)\mathbf{w}(k)\right)e(k)\mathbf{x}(k) \tag{2}$$

where $e(k)$ is the instantaneous output error, $d(k)$ the desired output of the network $\mathbf{x}(k) = [x_1(k), x_2(k), \ldots, x_N(k)]^T$ the vector of input signals $\mathbf{w}(k) = [w_1(k), w_2(k), \ldots, w_N(k)]^T$ the vector of weights, $\Phi(\cdot)$ the nonlinear activation function, and $\eta(k)$ the learning rate. The cost function of this nonlinear gradient descent (NGD) algorithm is given by

Manuscript received June 6, 2001. The associate editor coordinating the review of this paper and approving it for publication was Prof. D. A. Pados.

The authors are with the School of Information Systems, University of East Anglia, Norwich, NR4 7TJ, U.K. (e-mail: d.mandic@uea.ac.uk; aih@sys.uea.ac.uk; mr@sys.uea.ac.uk).

Publisher Item Identifier S 1070-9908(01)11128-4.

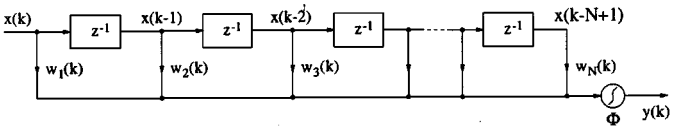

Fig. 1. Nonlinear neural adaptive filter.

$E_{\text{cost}}(k) = (1/2)e^2(k)$ [2]. The derivation of the NNGD algorithm starts from expressing the instantaneous output error (1) by a Taylor series expansion as [6]

$$e(k+1) = e(k) + \sum_{i=1}^{N}\frac{\partial e(k)}{\partial w_i(k)}\Delta w_i(k) + \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\frac{\partial^2 e(k)}{\partial w_i(k)\partial w_j(k)}\Delta w_i(k)\Delta w_j(k) + \text{h.o.t.} \tag{3}$$

where h.o.t. denotes the higher order terms of the remainder of the Taylor series expansion.

From (1) and (2), we have

$$\frac{\partial e(k)}{\partial w_i(k)} = -\Phi'\left(\mathbf{x}^T(k)\mathbf{w}(k)\right)x_i(k) \qquad i = 1, 2, \ldots, N \tag{4}$$

$$\Delta w_i(k) = \eta(k)\Phi'\left(\mathbf{x}^T(k)\mathbf{w}(k)\right)e(k)x_i(k) \qquad i = 1, 2, \ldots, N. \tag{5}$$

Truncating (3) and substituting (4) and (5) gives

$$e(k+1) = e(k)\left[1 - \eta(k)\left[\Phi'(\mathbf{x}^T(k)\mathbf{w}(k))\right]^2 \| \mathbf{x}(k) \|_2^2\right]. \tag{6}$$

From (6), we can solve for $\eta(k)$ to minimize the output error $e(k+1)$, which yields

$$\eta_{\text{opt}}(k) = \frac{1}{\left[\Phi'\left(\mathbf{x}^T(k)\mathbf{w}(k)\right)\right]^2 \| \mathbf{x}(k) \|_2^2 + C} \tag{7}$$

which is the learning rate of the normalized nonlinear gradient descent (NNGD) algorithm [3], [6]. The NNGD algorithm uses a first-order Taylor series expansion of the instantaneous output error, which forces the algorithm to include an unknown variable C into the calculation, namely, the second and higher order terms h.o.t. in (3). Although C is time varying, for simplicity, in the NNGD algorithm, this term was chosen to be a constant.

For nonlinear and nonstationary input signals, the statistics of the signal change in time and a static approximation of the truncated Taylor series expansion are often not good enough.

Therefore, there is a need for adjusting $C(k)$ using a gradient descent approach, i.e., according to the variation of the statistics of the input signal. Here, we derive a fully adaptive normalized nonlinear gradient descent (FANNGD) algorithm, by using a gradient adaptive $C(k)$ and show that the proposed algorithm converges faster than previously introduced algorithms of this kind.

II. The Fully Adaptive Nonlinear Normalised Gradient Descent Algorithm (FANNGD)

The adaptive learning rate in the NNGD algorithm comes from minimizing the error in (1) when expanded by the Taylor series expansion. The derivation of the adaptive learning rate is taken from the NNGD algorithm for a single layer network with one output node. Let us denote $\Phi'(\mathbf{x}^T(k)\mathbf{w}(k)) = \Phi'(k)$. Substituting the adaptive learning rate (7) into the weight adjustment (2) yields

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \frac{e(k)\Phi'(k)\mathbf{x}(k)}{[\Phi'(k)]^2 \parallel \mathbf{x}(k) \parallel_2^2 + C(k)} \tag{8}$$

To adaptively adjust the remainder of the Taylor series expansion on the run, we propose a gradient-based adaptation

$$C(k) = C(k-1) - \rho \nabla_{C(k-1)} \left[\frac{1}{2}e^2(k)\right] \tag{9}$$

where $\nabla_{C(k-1)}[(1/2)e^2(k)]$ is the gradient of the cost function $E_{\text{cost}}(k)$ with respect to $C(k-1)$, and ρ denotes the step size of the algorithm. The values of the adaptive remainder from the Taylor series expansion $C(k)$ when $k > 0$ are strongly coupled with the initial value $C(0)$. This kind of initial state dependence is typical of nonlinear systems.

From (9), we can now derive the gradient adaptive $C(k)$ equation as

$$C(k) = C(k-1) - \rho \frac{\Phi'(k)\Phi'(k-1)\mathbf{x}^T(k)\mathbf{x}(k-1)e(k)e(k-1)}{\left([\Phi'(k-1)]^2 \parallel \mathbf{x}(k-1) \parallel_2^2 + C(k-1)\right)^2}. \tag{10}$$

Combining (7) and (10), we obtain a fully adaptive nonlinear normalized gradient descent (FANNGD) learning algorithm for neural adaptive filters, for which the adaptation is described by (1) and (2) with the learning rate

$$\eta_{\text{opt}}(k) = \frac{1}{[\Phi'(k)]^2 \parallel \mathbf{x}(k) \parallel_2^2 + C(k)}. \tag{11}$$

III. Experiment

For all the algorithms in the experiment, the order of the nonlinear adaptive filter was $N = 10$, $\beta = 1$, $\text{net}(k) = \mathbf{x}^T(k)\mathbf{w}(k)$, $\Phi(\text{net}(k)) = 1/(1+e^{-\beta * \text{net}})$, and $\eta_{\text{opt}}(0) = 0.3$. A Monte Carlo simulation with 300 runs was performed, where white noise $x(k)$ with zero mean and unit variance was passed through a nonlinear filter described by [7]

$$y(k+1) = \frac{y(k)}{1+y^2(k)} + x^2(k). \tag{12}$$

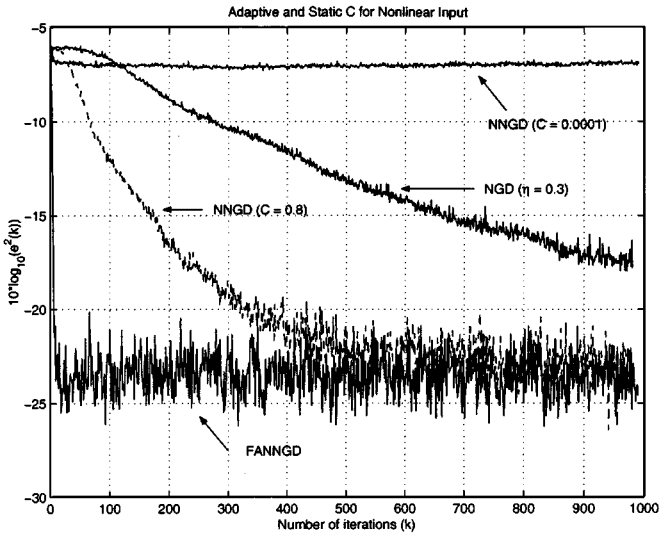

Fig. 2. Performance comparison between FANNGD, NNGD, and NGD for the nonlinear filter.

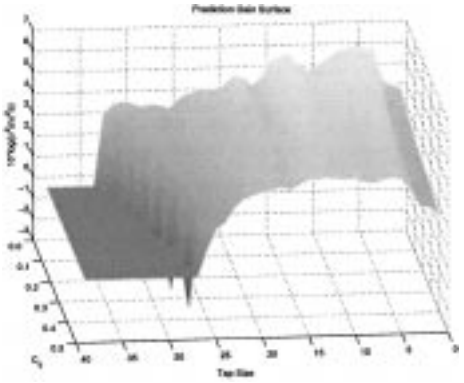

Fig. 3. Prediction gain variance, tap size, and C_0.

Fig. 2 shows that the FANNGD algorithm performs much better than some static values of $C(k)$. However, for the NNGD, we do not know the optimal value of $C(k)$ beforehand. The performance of the FANNGD algorithm is at least as good as the best choice of $C(k)$ in the NNGD algorithm.

As stated earlier, the initial value of $C(k)$ is crucial to the performance of the algorithm. A further experiment to find the optimal initial $C(k)$ at time instant $k = 0$ was carried out. The experiment measured the performance of the adaptive filter using the prediction gain $R_p = 10\log_{10}(\hat{\sigma}_s^2/\hat{\sigma}_e^2)$, where $\hat{\sigma}_s^2$ is the estimated signal variance, and $\hat{\sigma}_e^2$ is the estimated error variance. A predefined number of iterations was set to normalize the results. The performance of the nonlinear adaptive filter also relies on the input tap size. In all the previous experiments, the tap size was chosen $N = 10$. Fig. 3 shows the prediction gain against varying tap size and C_0. The input used was a white noise, with zero mean and unit variance which was then passed through a linear AR filter given by

$$y(k) = 1.79y(k-1) - 1.85y(k-2) + 1.27y(k-3) - 0.41y(k-4) + x(k). \tag{13}$$

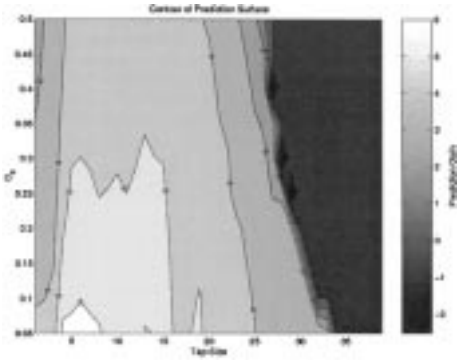

Fig. 4. Prediction gain variance contour plot.

To further exhibit the extremes of the prediction error surface, a contour plot Fig. 4 shows the peak values for this data. Analysis of the contour plot highlights the best performance of the adaptive filter, which in this case was achieved with $0.05 < C_0 < 0.1$ and a tap input of size $4 < N < 14$. For nonlinear inputs, these values are slightly different.

IV. Convergence

Although the optimal learning rate for the FANNGD algorithm is given by (11), the algorithm should converge for a range of values of $\eta(k)$ and $C(k)$. It is our goal that $|e(k)| \to 0$ as $k \to \infty$. Also, for uniform convergence

$$|e(k+1)| \leq |1 - \eta(k) \left[\Phi'(\mathbf{x}^T(k)\mathbf{w}(k))\right]^2 \| \mathbf{x}(k) \|_2^2 \, |e(k)|. \tag{14}$$

Hence

$$|1 - \eta(k) \left[\Phi'(\mathbf{x}^T(k)\mathbf{w}(k))\right]^2 \| \mathbf{x}(k) \|_2^2 \, | < 1 \tag{15}$$

and

$$0 < \eta(k) < \frac{2}{\left[\Phi'(\mathbf{x}^T(k)\mathbf{w}(k))\right]^2 \| \mathbf{x}(k) \|_2^2}. \tag{16}$$

This gives the convergence boundary for $\eta(k)$. We can now use (16) to determine the convergence boundaries of $C(k)$, using (11)

$$\begin{aligned} 0 &< \frac{1}{\left[\Phi'(\mathbf{x}^T(k)\mathbf{w}(k))\right]^2 \| \mathbf{x}(k) \|_2^2 + C(k)} \\ &< \frac{2}{\left[\Phi'(\mathbf{x}^T(k)\mathbf{w}(k))\right]^2 \| \mathbf{x}(k) \|_2^2.} \end{aligned} \tag{17}$$

Solving the inequality with respect to $C(k)$ gives

$$\frac{-\left[\Phi'\left(\mathbf{x}^T(k)\mathbf{w}(k)\right)\right]^2 \| \mathbf{x}(k) \|_2^2}{2} < C(k) \tag{18}$$

which is the lower bound for $C(k)$ so that (14) holds.

The classical convergence analysis in terms of convergence in the mean, mean square, and steady state follows the known analysis from the literature [3].

V. Conclusions

A FANNGD algorithm for nonlinear neural adaptive filters has been derived. This algorithm has evolved from the previously derived NNGD algorithm, and it has been shown that the constant in the adaptive learning rate for the NNGD can be made adaptive according to a gradient descent-based method. Simulations on nonlinear input signals have shown that the FANNGD outperforms the NNGD and NGD algorithms.

References

[1] V. J. Mathews and Z. Xie, "Stochastic gradient adaptive filter with gradient adaptive step size," *IEEE Trans. Signal Processing*, vol. 41, pp. 2075–2087, June 1993.

[2] S. Haykin, *Neural Networks: A Comprehensive Foundation*. Englewood Cliffs, NJ: Prentice-Hall, 1999.

[3] D. Mandic and J. Chambers, *Recurrent Neural Networks for Prediction*. New York: Wiley, 2001.

[4] S. Kalluri and G. R. Arce, "A general class of nonlinear normalized adaptive filtering algorithms," *IEEE Trans. Signal Processing*, vol. 47, pp. 2262–2272, Sept. 1999.

[5] M. Moreira and E. Fiesler, "Neural networks with adaptive learning rate and momentum terms," Tech. Rep., IDIAP, vol. 4, 1995.

[6] D. P. Mandic, "NNGD algorithm for neural adaptive filters," *Electron. Lett.*, vol. 39, no. 6, pp. 845–846, 2000.

[7] K. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," *IEEE Trans. Neural Networks*, vol. 1, pp. 4–27, Jan. 1990.