A ROBUST GENERAL NORMALISED GRADIENT DESCENT ALGORITHM

Danilo P. Mandic[1]*, Dragan Obradovic*[2]*, and Anthony Kuh*[3]

[1]Department of Electrical and Electronic Engineering, Imperial College London, UK
[2]Corporate Technology, Siemens AG, Munich Germany
[3]Department of Electrical Engineering, University of Hawaii, USA
d.mandic@imperial.ac.uk, dragan.obradovic@siemens.com, kuh@spectra.eng.hawaii.edu

ABSTRACT

A modification of the generalised normalised gradient descent (GNGD) algorithm is introduced which caters for the steady state performance of the algorithm. This is achieved by combining the search–then–converge and gradient adaptive step size approaches. This way, the proposed algorithm exhibits very fast convergence and small steady state error. Simulations on linear and nonlinear signals in prediction setting support the analysis.

1. INTRODUCTION

The Least Mean Square (LMS) algorithm is a simple, yet most frequently used algorithm for adaptive finite impulse response (FIR) filters. It is described by the following equations [10]

$$\begin{aligned} y(k) &= \mathbf{x}^T(k)\mathbf{w}(k) \\ e(k) &= d(k) - \mathbf{x}^T(k)\mathbf{w}(k) \\ \mathbf{w}(k+1) &= \mathbf{w}(k) + \mu e(k)\mathbf{x}(k) \end{aligned} \quad (1)$$

where $y(k)$ denotes the filter output at time instant k, $e(k)$ is the instantaneous error at the output of the filter, $d(k)$ is the desired signal, $\mathbf{x}(k) = [x(k-1), \ldots, x(k-N)]^T$ is the input signal vector, N is the length of the filter, $(\cdot)^T$ is the vector transpose operator, and $\mathbf{w}(k) = [w_1(k), \ldots, w_N(k)]^T$ is the filter coefficient (weight) vector. The parameter μ is the step–size (learning rate) which defines how fast the algorithm is converging along the error performance surface defined by a cost function $E(k) = \frac{1}{2}e^2(k)$, and is critical to the performance of LMS. Ideally, we want an algorithm for which [4]

- the speed of convergence is fast and the steady state misadjustment is small when operating in a stationary environment;
- in a nonstationary environment the algorithm should change the learning rate according to the dynamics of the input signal, so as to achieve as good a performance as possible.

To cater for the time–varying nature of the input signal $\mathbf{x}$, and its changing power levels, the normalised LMS (NLMS) algorithm has been introduced [10], for which the time–varying step size is given by

$$\eta(k) = \frac{\mu}{\| \mathbf{x}(k) \|_2^2}, \quad 0 < \mu < 2$$

where $\| \cdot \|_2$ denotes the Euclidean norm.

To preserve stability for close–to–zero input vectors, the optimal NLMS learning rate is usually modified to yield

$$\eta(k) = \frac{\mu}{\| \mathbf{x} \|_2^2} \rightarrow \frac{\mu}{\| \mathbf{x}(k) \|_2^2 + \varepsilon}$$

where ε is a small positive constant. The algorithm with the above defined learning rate is often referred to as the ε–NLMS [4]. It is well known that the regularisation term ε plays a critical role in the preformance of the ε–NLMS, a fact well known in the speech– and audio–processing community.

For a constant regularisation parameter ε, there are conflicting optimisation objectives between the convergence rate and the misadjustment error. In addition, for this class of algorithms, input signals with unknown and possibly very large dynamical range, and coupling between different signal modes can lead to poor performance or divergence.

To cater for these problems, a Generalised Normalised Gradient Descent (GNGD) algorithm was recently introduced [5], for which the adaptive learning rate is given by

$$\eta(k) = \frac{\mu}{\| \mathbf{x}(k) \|_2^2 + \varepsilon(k)} \quad (2)$$

where an additional stabilisation and faster convergence are introduced by making the compensation term ε in the denominator of the NLMS step size gradient–adaptive. The GNGD has been shown to converge extremely fast, and to be robust to perturbations in the regularisation term ε, and the initialisation of the parameter μ.

Despite its stability and very fast convergence, the GNGD does not guarantee that after convergence, for some very small output error of the filter, the update of the ε term in (2) will settle to some fixed value. This is due to the fact, that the filter remains "alert" at all time instants, in order to react quickly to the changes in the environment. This can also lead to an increased staedy state error. Our aim in this paper is to propose simple schemes which will make the GNGD algorithm achieve a lower steady state error, while keeping fast initial convergence.

2. THE GENERALISED GRADIENT DESCENT ALGORITHM (GNGD)

The NLMS algorithm (3) was originally derived using the so–called "independence assumptions" [4], given by:-

i) sequences $\mathbf{x}(k)$ and $\mathbf{w}(k)$ are zero mean, stationary, jointly normal and with finite moments;

ii) the successive increments of tap weights are independent of one another;

iii) the error and $\mathbf{x}(k)$ sequences are statistically independent of one another.

and is therefore **not optimal** for real–world practical settings. Its weight update is given by

$$\begin{aligned}\mathbf{w}(k+1) &= \mathbf{w}(k) + \frac{\mu}{\| \mathbf{x}(k) \|_2^2 + \varepsilon} e(k)\mathbf{x}(k) \\ &= \mathbf{w}(k) + \eta(k)e(k)\mathbf{x}(k) \end{aligned} \quad (3)$$

To overcome these problems, the GNGD algorithm makes the regularisation paramater ε within the denominator of (3) gradient adaptive, giving

$$\varepsilon(k+1) = \varepsilon(k) - \rho \nabla_{\varepsilon(k-1)} E(k) \quad (4)$$

where ρ is some small constant. Using the chain rule, the gradient $\nabla_{\varepsilon(k-1)} E(k)$ can be evaluated as

$$\begin{aligned}\frac{\partial E(k)}{\partial \varepsilon(k-1)} &= \frac{\partial E(k)}{\partial e(k)} \frac{\partial e(k)}{\partial y(k)} \frac{\partial y(k)}{\partial \mathbf{w}(k)} \frac{\partial \mathbf{w}(k)}{\partial \eta(k-1)} \frac{\partial \eta(k-1)}{\partial \varepsilon(k-1)} \\ &= \frac{e(k)e(k-1)\mathbf{x}^T(k)\mathbf{x}(k-1)}{\left(\| \mathbf{x}(k-1) \|_2^2 + \varepsilon(k-1)\right)^2} \end{aligned} \quad (5)$$

From (5), the update of the regularisation term $\varepsilon(k)$ within the generalised normalised gradient descent algorithm can be expressed as

$$\varepsilon(k) = \varepsilon(k-1) - \rho\, \mu\, \frac{e(k)e(k-1)\mathbf{x}^T(k)\mathbf{x}(k-1)}{\left(\| \mathbf{x}(k-1) \|_2^2 + \varepsilon(k-1)\right)^2} \quad (6)$$

Notice that there is a fundamental difference between GNGD and the variable step size algorithms with a "linear"[1] multiplicative adaptation factor, such as the ones proposed in [7, 2].

The algorithm proposed in [7] is based upon a gradient adaptation of the standard LMS learning rate μ from (1), that is, on $\frac{\partial E(k)}{\partial \mu}$. The step size update in this algorithm is given by

$$\begin{aligned}\mu(k) &= \mu(k-1) - \frac{\rho}{2} \frac{\partial}{\partial \mu(k-1)} e^2(k) \\ &= \mu(k-1) + \rho e(k)e(k-1)\mathbf{x}^T(k)\mathbf{x}(k-1) \end{aligned} \quad (7)$$

and is shown to be very sensitive to the choice of initial values of ρ and $\mu(0)$ [7, 1]. Another, more general, member of the same class of gradient adaptive learning algorithms is the Benveniste algorithm [2], for which the updates are given by

$$\begin{aligned}\mu(k) &= \mu(k-1) + \rho e(k)\mathbf{x}^T(k)\boldsymbol{\psi}(k) \\ \boldsymbol{\psi}(k) &= \left[\mathbf{I} - \mu(k-1)\mathbf{x}(k-1)\mathbf{x}^T(k-1)\right]\boldsymbol{\psi}(k-1) \\ &+ e(k-1)\mathbf{x}(k-1) \end{aligned} \quad (8)$$

The algorithm (8) is based upon the exact derivation of the gradient–adaptive learning rate and is computationaly demanding, since it requires matrix multiplications within the updates of its parameters.

On the contrary, GNGD employs a *nonlinear update* of the adaptive learing rate $\eta(k)$, which is based on a time–varying regularisation factor $\varepsilon(k)$, which compensates for the assumptions in the derivation of NLMS. Figure 1 illustrates the benefits of GNGD, which remains stable even when NLMS diverges, that is for $\mu = 2.1$, with the GNGD parameters set to $\varepsilon(0) = 0.15$ and $\rho = 0.1$.

3. PROPOSED APPROACH

At every time instant k, the GNGD algorithm attempts to optimise for ε, even for small values of error $e(k)$ and small power levels of input signal $\| \mathbf{x}(k) \|_2^2$.
Notice that the lower bound for the stability of GNGD with respect to $\varepsilon(k)$ can be derived straightforwardly from the standard LMS–type stability bounds and is given by [5]

$$\varepsilon(k) > -\frac{\| \mathbf{x}(k) \|_2^2}{2} \quad (9)$$

Clearly, the variable ε can assume negative values[2], in order to provide optimal estimation of μ. To improve the performance of GNGD in these critical cases, which is difficult to

[1] By "linear" multiplicative factor we mean algorithms based on $\frac{\partial E}{\partial \mu}$, whereas the GNGD is based on a time varying η for which the parameter ε from (2) is being updated, hence giving a "nonlinear" update of η.

[2] This is obvious from the expression for the denominator of the GNGD learning rate (2).

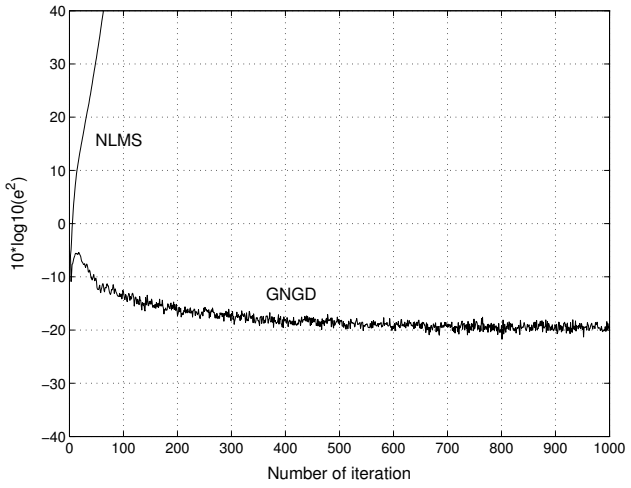


Fig. 1. Performance comparison between NLMS and GNGD on a nonlinear signal [8] for $\mu = 2.1$

control in an automated manner, we propose the following modification of the GNGD:-

- A faster stochastic gradient search [3] by means of a cooling schedule whereby the learning rate decreases with the number of iteration;
- This will help to reduce the relatively high misadjustment of the GNGD, while keeping its excellent initial fast convergence.

This way we combine the virtue of a very fast convergence of GNGD with the additional stabilisation provided by the proposed modification of GNGD.

The cooling schedule can be performed using several well established annealing schemes [9, 3, 4]. These "search then converge" (STC) schemes combined with GNGD offer the ability to avoid local minima within the stochastic gradient optimisation task, together with good performance even when the gradients vanish almost anywhere [9].
Some of the standard STC algorithms adjust their learning rates according to [4]:-

i) The annealing algorithm which is controlled by a "time constant", given by

$$\eta(k) = \frac{\eta_0}{1 + (k/\tau)} \tag{10}$$

where η_0 and τ are some constants;

ii) The Darken and Moody STC algorithm [3], which updates its step size according to

$$\eta(k) = \eta_0 \frac{1 + \frac{c}{\eta_0}\frac{k}{\tau}}{1 + \frac{c}{\eta_0}\frac{k}{\tau} + \tau\frac{k^2}{\tau^2}}, \tag{11}$$

where η_0, τ, and c are some constants.

Accordingly, the learning rate of the proposed modification of GNGD, termed STC GNGD, becomes

$$\eta(k) = \frac{\mu}{\| \mathbf{x}(k) \|_2^2 + \varepsilon(k) + STC(k)} \tag{12}$$

where the term $STC(k)$ refers the cooling schedule from either (10) or (11).

4. SIMULATION RESULTS

Simulations were conducted in the prediction setting, where the analysed signals were [6]:-

i) Coloured noise (AR(4) model) given by

$$\begin{aligned} y(k) &= 1.79y(k-1) - 1.85y(k-2) \\ &\quad +1.27y(k-3) - 0.41y(k-4) + x(k) \end{aligned} \tag{13}$$

where $\{x(k)\}$ denotes a driving white Gaussian noise with zero mean and unit variance;

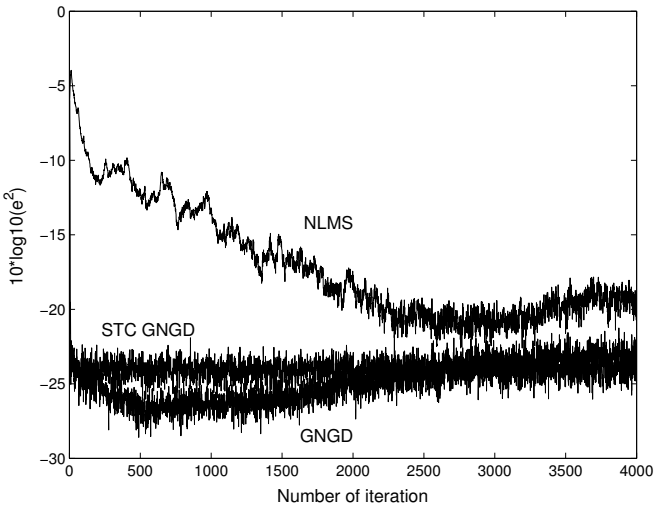
ii) A nonlinear signal [8], derived from its linear counterpart (13), and given by

$$y(k+1) = \frac{y(k)}{1 + y^2(k)} + x^3(k) \tag{14}$$

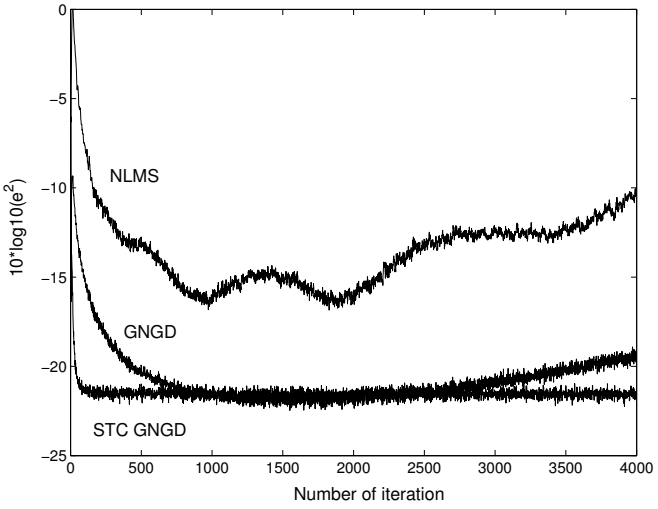
Learning curves were averaged over 1000 independent trials.

The performances of NLMS, GNGD, and proposed STC GNGD algorithms are shown respectively in Figure 2a) and Figure 2b). The proposed STC GNGD approach exhibits much better steady state characteristis than the GNGD and NLMS, as indicated by monotonic convergence curves, unlike the NLMS and GNGD.
In the case of nonlinear signal (14), the NLMS did not converge, the GNGD exhibitted inconsistency in its convergenc, whereas STC GNGD had no problems in its convergence, as illustrated in Figure 2b). In addition, the initial convergence rates of STC GNGD were not worse than those of GNGD. In both cases, the STC schedule was the second–order one, given in (11).

Figure 3 shows a comparison between the performances of STC GNGD algorithms for the annealing schedules given respectively in (10) and (11). Both the SCT GNGD algorithms exhibitted better and more consistent performance than NLMS and GNGD. The first order STC algorithm STC GNGD(10) had a slower rate of convergence but better overall performance than the second order STC algorithm STC GNGD(11).



(a) Convergence curves for linear signal



(b) Convergence curves for nonlinear signal

Fig. 2. Convergence of the NLMS, GNGD and the proposed STC GNGD algorithm

5. CONCLUSIONS

A modification of the General Normalised Gradient Descent (GNGD) algorithm has been proposed which improves the steady state performance of this algorithm. This has been achieved by combining a cooling schedule within the "search then converge" framework [3], with the standard GNGD which uses a gradient adaptive regularisation parameter within its learning rate. Such as search the converge generalised normalised gradient descent (STC GNGD) algorithm, therefore, combines a very fast convergence of GNGD with additional stabilisation introduced through simulated annealing. Simulations on linear and linear benchmark signals support the approach.

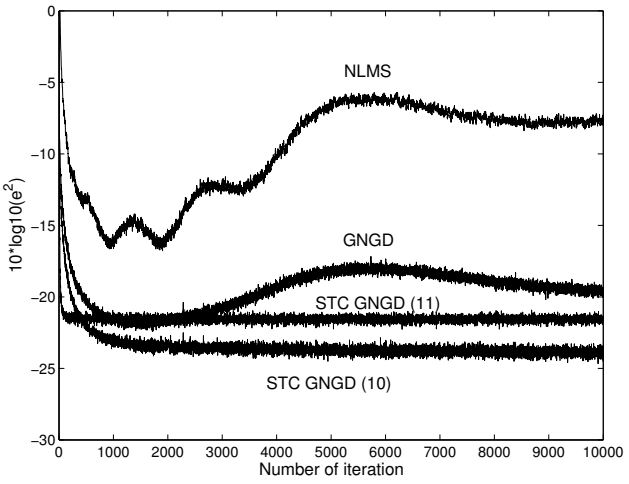


Fig. 3. Performance comparison between NLMS, GNGD, first– (10) and second–order (11) STC GNGD algorithms for nonlinear signal (14), with $\mu = 1.95$ and $\tau = 100$.

6. REFERENCES

[1] W.-P. Ang and B. Farhang-Boroujeny. A new class of gradient adaptive step–size LMS algorithms. *IEEE Transactions on Signal Processing*, 49(4):805–810, 2001.

[2] A. Benveniste, M. Metivier, and P. Priouret. *Adaptive Algorithms and Stochastic Approximation*. Springer–Verlag: New York, 1990.

[3] C. Darken and J. Moody. Towards faster stochastic gradient search. In J. E. Moody, S. J. Hanson, and R. P. Lippmann, editors, *Neural Information Processing Systems 4*, pages 1009–1016. Morgan Kaufmann, 1992.

[4] S. Haykin. *Adaptive Filter Theory*. Prentice-Hall, third edition, 1996.

[5] D. P. Mandic. A general normalised gradient descent algorithm. *IEEE Signal Processing Letters*, 11(2):115–118, 2004.

[6] D. P. Mandic and J. A. Chambers. *Recurrent Neural Networks for Prediction: Architectures, Learning Algorithms and Stability*. Wiley, 2001.

[7] V. J. Mathews and Z. Xie. A stochastic gradient adaptive filter with gradient adaptive step size. *IEEE Transactions on Signal Processing*, 41(6):2075–2087, 1993.

[8] K. S. Narendra and K. Parthasarathy. Identification and control of dynamical systems using neural networks. *IEEE Transactions on Neural Networks*, 1(1):4–27, 1990.

[9] K. Rose. Deterministic annealing for clustering, compression, classification, regression, and related optimization problems. *Proceedings of the IEEE*, 86(11):2210–2239, 1998.

[10] B. Widrow and S. D. Stearns. *Adaptive Signal Processing*. Prentice–Hall, 1985.