6th Seminar on Neural Network Applications in Electrical Engineering, NEUREL-2002
Faculty of Electrical Engineering, University of Belgrade, Yugoslavia, September 26-28, 2002

PERLUSTRATION OF ERROR SURFACES FOR NONLINEAR STOCHASTIC GRADIENT DESCENT ALGORITHMS

Andrew I. Hanna
School of Information Systems,
University of East Anglia,
Norwich, UK.
aih@sys.uea.ac.uk

Igor R. Krcmar
Faculty of Electrical Engineering,
University ofBanjaluka,
Banjaluka, Bosnia–Herzegovina.
ikrcmar@etf-bl.rstel.net

Danilo P. Mandic
Dept. of Electrical Engineering
Imperial College
London, UK.
d.mandic@ic.ac.uk

ABSTRACT

We attempt to explain in more detail the performance of several novel algorithms for nonlinear neural adaptive filtering. Weight trajectories together with the error surface give a clear understandable representation of the family of least mean square (LMS) based, nonlinear gradient descent (NGD), search-then-converge (STC) learning algorithms and the real-time recurrent learning (RTRL) algorithm. Performance is measured on prediction of coloured and nonlinear input. The results are an alternative qualitative respresentation of different qualitative performance measures for the analysed algorithms. Error surfaces and the adjacent instantaneous prediction errors support the analysis.

1. INTRODUCTION

Stochastic gradient descent is a very well understood algorithm in linear and nonlinear adaptive signal processing, whose variants include least mean square (LMS), nonlinear gradient descent (NGD), backpropagation and many simulated annealing algorithms. Various techniques have been employed in order to speed up convergence of these algorithms, such as momentum terms [10][4], adaptive slopes, β, in the activation function [8], and adaptive learing rates, η, in the weight update algorithm [7][3][9]. Performance of these algorithms can be examined many ways, i.e. prediction gain, convergence curves and Monte Carlo analysis. A straightforward yet effective and insightful method of visually measuring performance of an algorithm is by describing the learning procedure by the trajectory along the error surface. As the prediction gain and Monte Carlo analysis are based upon the instantaneous output error, either logarithmic or averaged, the error performance surface is mathematically equivalent and offers a convienient visualisation of learning. However, due to the requirement of a quadratic error surface we can only visualise filters with two weights, hence the filter may not be of optimal order[1]. An attempt to visualise multidimensional surfaces was presented in [2]. However, that method only performed well in the absence of noise, and will not be used here. Error surfaces are constructed based upon the characteristics of the input data. A classical approach is Wiener filter theory [13], where the correlation matrix of the input signal, $\mathbf{R} = E[\mathbf{x}(k)\mathbf{x}^T(k)]$, and the cross-correlation vector of the desired response and the input signal, $\mathbf{p} = E[\mathbf{x}(k)d(k)]$ give a quadratic equation to the error performance surface of

$$J(\mathbf{w}) = E[d^2(k)] - 2\mathbf{p}^T\mathbf{w} + \mathbf{w}^T\mathbf{R}\mathbf{w}. \quad (1)$$

where $\mathbf{x} = [x_1, x_2, \ldots, x_N]^T$ denotes the tap input vector, $\mathbf{w} = [w_1, w_2, \ldots, w_N]^T$ the weight vector, $d(k)$ the desired response at time instant k and $J(\mathbf{w})$ the objective function, and $E[\cdot]$ is the statistical expectation operator.
The purpose of all the optimisation techniques discussed in this paper aim to find

$$J(\mathbf{w}^*) \leq J(\mathbf{w}(k)), \quad (2)$$

where $J(\mathbf{w}(k)) = \frac{1}{2}\sum e^2(k)$ denotes the objective function, $\mathbf{w}^*$ the set of optimal weights, $\mathbf{w}(k)$ the set of weights at time instant k and $e(k)$ the output error of the filter at time instant k using weights $\mathbf{w}(k)$. The algorithms examined in this paper adapt the weights of the filter according to the gradient descent based method of steepest descent. Since the error surface can be viewed as a paraboloid for two weights, the condition of optimality is

$$\nabla_w J(\mathbf{w}^*) = 0 \quad (3)$$

where ∇ denotes the gradient operator. Ideally, the method of steepest descent adapts the weights according to

$$J(\mathbf{w}(k+1)) \leq J(\mathbf{w}(k)) \quad (4)$$

[1] For instance, if we want to recover a strange attractor, then due to Takens' theorem, the filter order would be twice the order of the attractor.

0-7803-7593-9/02/$17.00 © 2002 IEEE

However, in the real world we want to see the effects an error surface has on the performance of a particular algorithm. Firstly, we construct the error performance surface matrix according to the specified filter with no weight adaptation for a set of predefined weights. This is achieved by passing the complete set of input data through each pair of weights and computing the average error. To plot the weight trajectory onto the error surface, we take some starting weight and pass the data through the filter using a sliding window as in traditional adaptive filtering. After each weight update the complete set of input data is then passed through the filter with static weights. The average error is then calculated as in the construction of the error surface. For clarity, in all the experiments in this paper every 10^{th} contour in the weight trajectory is plotted. In all the experiments, two typical input signals were considered, a linear stochastic AR [7] and nonlinear signal. The coloured input is given by the stable filter [3]

$$\begin{aligned} y(k) &= 1.79y(k-1) - 1.85y(k-2) \\ &\quad +1.27y(k-3) - 0.41y(k-4) + u(k), \end{aligned} \quad (5)$$

whereas the nonlinear input is given by the benchmark input [11]

$$y(k) = \frac{y(k-1)}{1+y^2(k-1)} + u^3(k), \quad (6)$$

where $u(k)$ is normally distributed $\mathcal{N}(0,1)$ white noise. In this paper the performance of novel algorithms for nonlinear neural adaptive filtering are demonstrated on error performance surfaces.

2. NONLINEAR GRADIENT DESCENT ALGORITHMS

The family of nonlinear gradient descent (NGD) algorithms is based upon the LMS algorithm, with the addition of a nonlinearity (neuron) denoted by $\Phi(\cdot)$. A nonlinear adaptive FIR filter[2] is shown in Figure 1. The equations that define

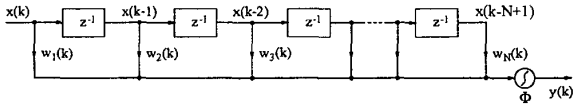

Fig. 1. A nonlinear adaptive FIR filter

the weight update in the NGD algorithms are given by

$$e(k) = d(k) - \Phi(\mathbf{x}^T(k)\mathbf{w}(k)), \quad (7)$$

$$\begin{aligned} \mathbf{w}(k+1) &= \mathbf{w}(k) - \eta\nabla J_{\mathbf{w}}(k) \\ &= \mathbf{w}(k) + \eta e(k)\Phi'(\mathbf{x}^T(k)\mathbf{w}(k))\mathbf{x}(k) \end{aligned} \quad (8)$$

[2]In fact this is a dynamical perceptron; in neural network terminology.

and are closely derived from the standard LMS algorithm. In all the NGD filters considered, the nonlinearity in the output neuron was the hyperbolic tangent function. This model can easily be extended to the recurrent case defined by the equations

$$y(k+1) = \Phi(\mathbf{u}^T(k)\mathbf{w}(k)) \quad (9)$$

where $\mathbf{u}(k) = [y(k-1), \ldots, y(k-N), \xi, x(k-1), \ldots, x(k-M)]^T$ denotes the external and feedback inputs to the filter, $\Phi(\cdot)$ the nonlinear activation function, ξ denotes the constant valued bias input and the weight vector is denoted by $\mathbf{w}(k) = [w_1(k), w_2(k), \ldots, w_{N+M+1}(k)]^T$. A recurrent

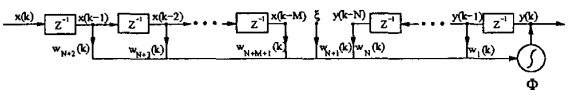

Fig. 2. An adaptive IIR filter

perceptron employed as an adaptive IIR filter is shown in Figure (2).

2.1. Algorithms

For the NGD algorithm the learning rate is chosen to be some constant. For the normalised NGD algorithm the learning rate is adapted according to a Taylor series expansion of the instantaneous output error to give

$$\begin{aligned} e(k+1) = e(k) &+ \sum_{i=1}^{N} \frac{\partial e(k)}{\partial w_i(k)} \Delta w_i(k) \\ &+ \frac{1}{2!} \sum_{i=1}^{N}\sum_{j=1}^{N} \frac{\partial^2 e(k)}{\partial w_i(k)\partial w_j(k)} \Delta w_i(k)\Delta w_j(k) + h.o.t. \end{aligned} \quad (10)$$

For simplicity we truncate the second and higher order terms of (10). We want the output error at the next time instant to be zero, therefore the term on the right hand side must be zero allowing us to solve for $\eta(k)$ [7].

$$\begin{aligned} e(k+1) &= e(k) + \sum_{i=1}^{N} \frac{\partial e(k)}{\partial w_i(k)} \Delta w_i(k) \\ &= e(k)\Big[1 - \eta(k)[\Phi'(\mathbf{x}^T(k)\mathbf{w}(k))]^2 ||\mathbf{x}(k)||_2^2\Big] \end{aligned} \quad (11)$$

therefore the term in the square brackets must equal zero, giving

$$\eta(k) = \frac{1}{[\Phi'(\mathbf{x}^T(k)\mathbf{w}(k))]^2 ||\mathbf{x}(k)||_2^2 + C} \quad (12)$$

as the adaptive learning rate for the NNGD algorithm. Notice the inclusion of the constant C, added to balance the

exclusion of the second and higher order terms from (10). Included in the family of NGD algorithms are the fully adaptive normalised nonlinear gradient descent algorithms. In these algorithms, the added constant C in (12) is made adaptive to compenstate for the truncation of the higher order terms in the Taylor series expansion (10). The error adaptive NNGD (EANNGD) algorithm adjusts $C(k)$ according to the variance in the instantaneous output error [5]

$$C(k) = C(k-1) + \mu e^2(k), \tag{13}$$

where μ is chosen to be some small positive constant. The fully adaptive normalised nonlinear gradient descent (FANNGD) algorithm adjusts the parameter $C(k)$ according to a gradient descent based approach [7].

$$C(k) = C(k-1) - \rho \nabla_{C(k-1)}[\frac{1}{2}e^2(k)] \tag{14}$$

where $\nabla_{C(k-1)}[\frac{1}{2}e^2(k)]$ is the gradient of the cost function, $J(k)$, with respect to $C(k-1)$ and ρ denotes the step size of the algorithm. For simplicity, we let $\Phi(\mathbf{x}^T(k)\mathbf{w}(k)) = \Phi(k)$, giving [7]

$$C(k) = C(k-1) - \rho \frac{\Phi'(k)\Phi'(k-1)\mathbf{x}^T(k)\mathbf{x}(k-1)e(k)e(k-1)}{\left([\Phi'(k-1)]^2 \parallel \mathbf{x}(k-1) \parallel_2^2 + C(k-1)\right)^2}. \tag{15}$$

The learning rate for the set of fully adaptive normalised nonlinear gradient descent algorithms can then be stated as

$$\eta_{opt}(k) = \frac{1}{[\Phi'(k)]^2 \parallel \mathbf{x}(k) \parallel_2^2 + C(k)} \tag{16}$$

2.2. Simulations on Coloured Input

The correlated weight contours and trajectories were plotted for the NGD, normalised NGD (NNGD), STC, and DM algorithms on coloured input, (5). Traditional STC algorithms adjust the learning rate according to [4],

$$\eta(k) = \frac{\eta_0}{1 + (k/\tau)}, \tag{17}$$

and the Darken and Moody STC (DM) algorithm [1], adjusts the learning rate according to

$$\eta(k) = \eta_0 \frac{1 + \frac{c}{\eta_0}\frac{k}{\tau}}{1 + \frac{c}{\eta_0}\frac{k}{\tau} + \tau\frac{k^2}{\tau^2}}, \tag{18}$$

where c and τ are some chosen constants. Figures 3 and 4 show that the STC algorithm marginally outperformed the NGD algorithm, but the best performance was by the NNGD algorithm. The output error for the traditional STC and NNGD algorithms converge after approximately 50 iterations. The output error in both the NGD and DM algorithms did not converge to zero.

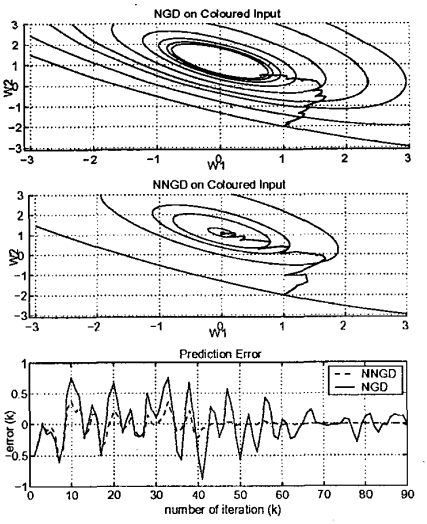

Fig. 3. Error surface of NGD and NNGD algorithms on coloured input

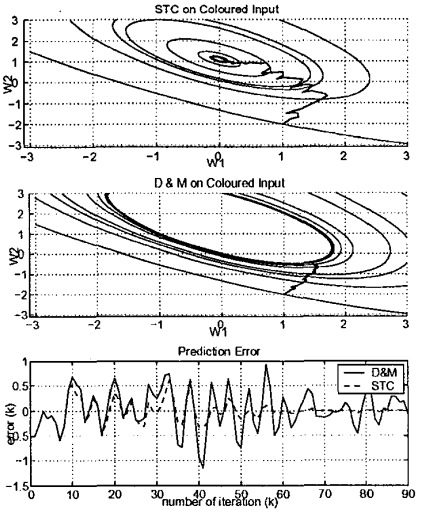

Fig. 4. Error surface of search-then-converge algorithms on coloured input

2.3. Simulations on Nonlinear Input

Due to the nonlinearity in the class of NGD filters, we test the two representative filters (NGD and NNGD) from the experiments on coloured input and test them on nonlinear input, together with the error adaptive NNGD and the fully adaptive NNGD. Both the EANNGD and the FANNGD employ an adaptive C term as given in (12). Figures 5 and 6 show the fully adaptive NNGD algorithms outperforming

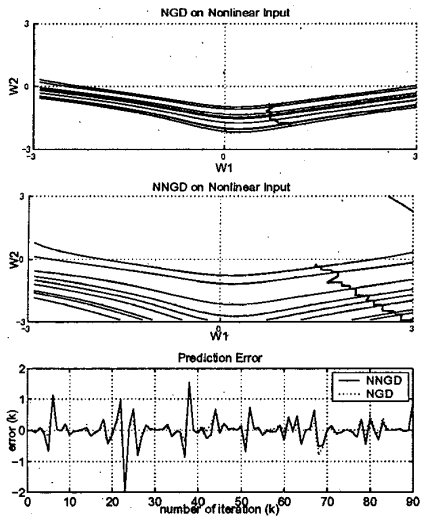

Fig. 5. Error surface of standard and normalised nonlinear gradient descent algorithms on nonlinear input

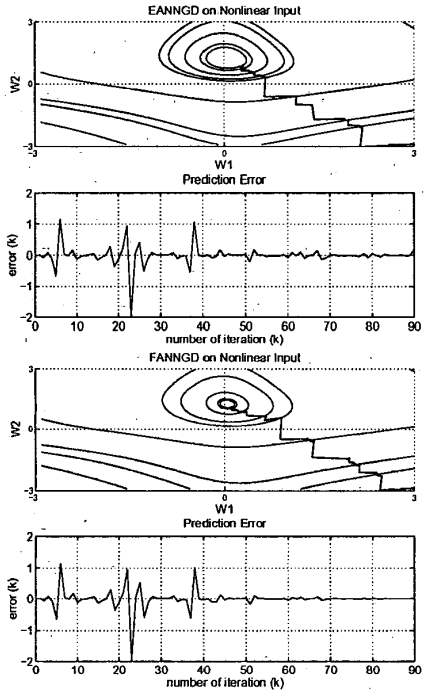

Fig. 6. Error surface of fully adaptive nonlinear gradient descent algorithms on nonlinear input

the standard NGD and NNGD algorithms. The output errors of NGD did not converge to zero, and the weight trajectories show the NGD moving slowly down the error surface. The FANNGD and EANNGD have very similar performance erros, however the FANNGD output error converged to zero in a very short time. The best performance was by the FANNGD algorithm, which converges to the optimal state in minimal time compared to the other algorithms in this class.

2.4. Simulations on Chaotic Input

To further test the ability of the NGD and normalised NGD algorithms to deal with complex nonlinear signals, simulations on chaotic input were carried out next. The chaotic signal chosen was then Henon Map

$$x(k+1) = 1 - ax^2(k) + bx(k-1) \qquad (19)$$

Figures 8 and 9 show the fully adaptive NNGD algorithms

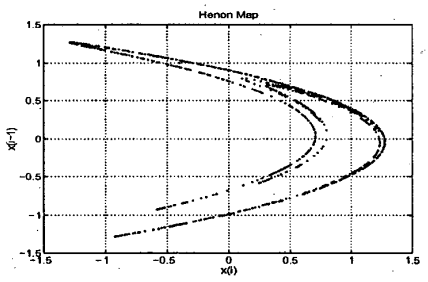

Fig. 7. The Henon map

outperforming the standard NGD and NNGD algorithms. The output errors of the NNGD algorithm converged to zero near the end of training. However, the choice of C in the NNGD algorithm was near the optimal value giving near optimal performance. In the fully adaptive normalised NGD algorithms, the output error converge around zero during training. This is visually prounounced by the weight trajectories convering to the optimal value on the error surface.

2.5. Simulations Using Recurrent Perceptrons

Due to an increasing interest into the use of recurrent perceptrons employed as infinite impulse response (IIR) adaptive filters [6] we now look at the error surfaces produced by such filters and how the weight trajectories traverse the surface [12]. In our experiment, $\Phi(\cdot)$ was chosen to be the hyperbolic tangent function and the external inputs were coloured noise produced by the AR filter described in (5). Figure 2 shows a block diagram structure of a recurrent perceptron with a single input and a single feedback and a bias input $\xi = 0$ in order to preserve the updating of two weights in the algorithm. Figure 10 shows the error surfaces for the recurrent perceptron and the corresponding weight trajectories on coloured and nonlinear inputs. It can be clearly seen that two valleys appear in the surface which were not

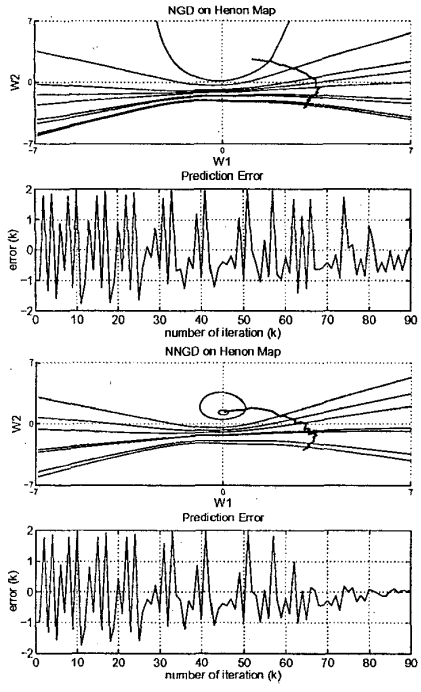

Fig. 8. Error surface of fully adaptive nonlinear gradient descent algorithms on Henon map

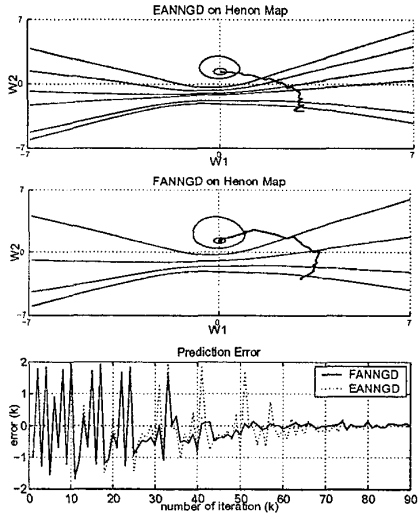

Fig. 9. Error surface of fully adaptive nonlinear gradient descent algorithms on Henon map

apparant in the feedforward perceptrons. This is due to the feedback component distorting the surface and making the search for a single global minimum increasing difficult.

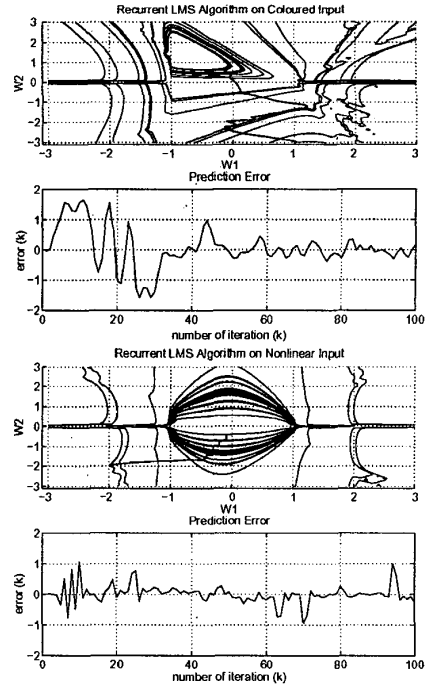

Fig. 10. Recurrent perceptron and the error surface on coloured input

3. CONCLUSIONS

Performance analysis via contour plots of error surfaces for the nonlinear descent (NGD), normalised NGD (NNGD), error adaptive NNGD (EANNGD), fully adaptive NNGD (FANNGD) algorithms and the real time recurrent learning (RTRL) algorithm have been undertaken for feedforward and recurrent neural adaptive filters. This has been achieved on coloured and nonlinear input. A qualitative insight into performance of novel nonlinear gradient algorithms for neural adaptive filtering has been provided using error performance surfaces, which is supported by the quantitative measure via the instantaneous output error of the filters.

4. REFERENCES

[1] C. Darken and J. E. Moody. Towards faster stochastic gradient search. In John E. Moody, Steve J. Hanson, and Richard P. Lippmann, editors, *Advances in Neural Information Processing Systems*, volume 4, pages 1009–1016. Morgan Kaufmann Publishers, Inc., 1992.

[2] M. Fisher, D. P. Mandic, J. A. Bangham, and R. Harvey. Visualising Error Surfaces for Adaptive Filters and Other Purposes. *In Proceedings of the Interna-*

tional Conference on Acoustics, Speech and Signal Processing, ICASSP, 4:3522–3525, 2000.

[3] A. I. Hanna, D. P. Mandic, and 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*, pages 63–72, 2001.

[4] S. Haykin. *Neural Networks: A Comprehensive Foundation*. Prentice Hall, second edition, 1999.

[5] I. Krcmar and D. P. Mandic. A Fully Adaptive Normalized Nonlinear Gradient Descent Algorithm for Nonlinear System Identification. *in Proceedings of ICASSP 2001*, 6:3493–3496, 2001.

[6] D. Mandic and J. Chambers. *Recurrent Neural Networks for Prediction*. John Wiley and Sons, 2001.

[7] D. P. Mandic, A. I. Hanna, and M. Razaz. A Normalised Gradient Descent Algorithm for Nonlinear Adaptive Filters Using a Gradient Adaptive Step Size. *IEEE Signal Processing Letters*, 8(11):295–297, 2001.

[8] D. P. Mandic and 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)*, pages 42–45, 2000.

[9] V. J. Mathews and Z. Xie. Stochastic Gradient Adaptive Filter with Gradient Adaptive Step Size. *IEEE Transactions on Signal Processing*, 41(6):2075–2087, 1993.

[10] M. Moreira and E. Fiesler. Neural Networks with Adaptive Learning Rate and Momentum Terms. *IDIAP Technical Report*, 04(95), 1995.

[11] K. S. Narendra and K. Parthassarathy. Identification and Control of Dynamical Systems Using Neural Networks. *IEEE Transactions on Neural Networks*, 4(1):4–27, 1990.

[12] S. D. Stearns. Error surfaces of recursive adaptive filters. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 29(3):763–766, 1981.

[13] B. Widrow and S. Stearns. *Adaptive Signal Processing*. Prentice Hall, 1995.