Neural Processing Letters **17:** 1–7, 2002.

A Data-Reusing Nonlinear Gradient Descent Algorithm for a Class of Complex-Valued Neural Adaptive Filters

ANDREW I. HANNA[1] and DANILO P. MANDIC[2]
[1]*School of Information Systems, University of East Anglia, Norwich, NR4 7TJ, Norfolk, U.K. e-mail: aih@sys.uea.ac.uk*
[2]*Communications and Signal Processing Group of the Department of Electrical and Electronic Engineering, Imperial College of Science, Technology and Medicine, London, U.K. e-mail: d.mandic@ic.ac.uk*

Abstract. A complex-valued data-reusing nonlinear gradient descent (CDRNGD) learning algorithm for a class of complex-valued nonlinear neural adaptive filters is introduced and the affinity between the family of data-reusing algorithms and the class of normalised gradient descent algorithms is examined. Error bounds on the class of complex data-reusing algorithms are established and indicate the stability of such algorithms. Experiments on nonlinear inputs show the class of complex data-reusing algorithms outperforming the standard complex non-linear gradient descent algorithms and converging to the normalised complex non-linear gradient descent algorithm without experiencing the stability problems commonly encountered with normalised gradient descent algorithms.

Key words. complex-valued nonlinear adaptive filter, data-reusing, normalised complex non-linear gradient descent

1. Introduction

There has been a recent interest in complex-valued nonlinear neural adaptive filters, which are the extensions of the real-valued nonlinear adaptive filters to the complex plane, $\mathbb{C}$. This class of filter have expanded the application fields in image processing, computer vision, optoelectronic imaging, and communications. The potentially wide applicability yields new aspects of theories required for novel or more effective functions and mechanisms. The complex least mean square (CLMS) algorithm, [1], gave rise to the growing applications of complex-valued linear filters. This led to the development of the complex nonlinear gradient descent (CNGD) algorithm and the complex backpropagation (CBP) algorithm [2–4] for nonlinear filters and neural networks. As with their real-valued counterparts [5, 6], this family of complex-valued algorithms suffer from the same problems of slow convergence and subjection to local minima. A resolution to these problems in the field of real-valued nonlinear adaptive filtering is using the family of data-reusing algorithms to help speed up convergence [7, 8]. The aim of this paper is to develop a complex data-reusing

NEPL758 Pe: 5110663 LE/CP/DISK (LATEX)
11-02-2002 15:40 1st Proof

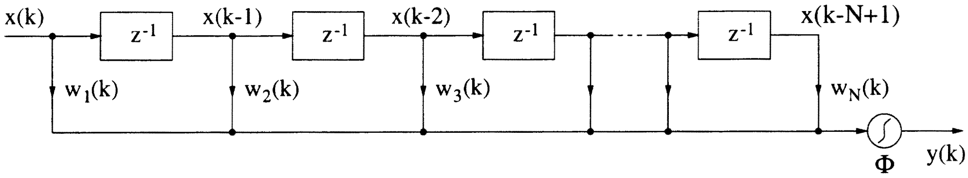

Figure 1. A nonlinear neural FIR filter.

nonlinear gradient descent (CDRNGD) algorithm for use in complex-valued neural adaptive filters.

In this Letter we consider a dynamical feedforward perceptron which is in fact a nonlinear adaptive finite impulse response (FIR) filter with a single neuron, shown in Figure 1.

This architecture is valid for both real valued and complex-valued neural adaptive filters and is a generalisation of the widely used linear FIR adaptive filters. For the class of filters discussed in this paper we derive the direct gradient algorithms.

2. The Nonlinear Complex Gradient Descent Algorithm

The equations that describe the nonlinear complex gradient descent (NCGD) algorithm for a complex-valued dynamical perceptron employed as a nonlinear FIR filter with a single output neuron, shown in Figure 1, are given by

$$e(k) = d(k) - y(k), \qquad y(k) = \Phi(\mathbf{x}^T(k)\mathbf{w}(k)) \tag{1}$$

where $e(k)$ denotes the instantaneous output error, $d(k)$ the desired output, $\Phi(\cdot)$ a complex nonlinear analytic activation function that is bounded almost everywhere in the complex domain, $\mathbb{C}$, $\mathbf{x}(k) \stackrel{\Delta}{=} [x_1(k), x_2(k), \ldots, x_N(k)]^T$ the complex tap input, $\mathbf{w}(k) \stackrel{\Delta}{=} [w_1(k), w_2(k), \ldots, w_N(k)]^T$ the complex weight vector and N denotes the number of tap inputs. For simplicity we state that

$$\Phi(\mathbf{x}^T(k)\mathbf{w}(k)) = \Phi^r(\mathbf{x}^T(k)\mathbf{w}(k)) + j\Phi^i(\mathbf{x}^T(k)\mathbf{w}(k)) = u(k) + j(v) \tag{2}$$

where the superscripts $(\cdot)^r$ and $(\cdot)^i$ denote the real and imaginary parts of the complex output, and $j = \sqrt{-1}$. The objective function of the filter is given by

$$\mathcal{J}(k) = \frac{1}{2}|e(k)|^2 = \frac{1}{2}[e(k)e^*(k)] \tag{3}$$

where $(\cdot)^*$ denotes the complex conjugate operator and $|\cdot|$ the modulus operator. The weight adaptation in the NCGD algorithm is given by [7]

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \Delta\mathbf{w}(k) \tag{4}$$

$$\Delta\mathbf{w}(k) = \eta[\Phi'(\mathbf{x}^T(k)\mathbf{w}(k))]^*\mathbf{x}^*(k)e(k) \tag{5}$$

where η denotes the step size of the algorithm. Notice that for a linear real-valued filter, (4) and (5) degenerate into the least mean square (LMS) algorithm.

3. The Complex-Valued Data-Reusing Nonlinear Gradient Descent Algorithm

The complex-valued data-reusing nonlinear gradient descent (CDRNGD) algorithm is a general form of the standard data-reusing nonlinear gradient descent algorithms [7, 8]. It is shown in [9] that for real-valued algorithms, a posteriori approach is the first reuse of a general data-reusing algorithm, and as the number iterations in the data-reusing algorithm increase, this leads to the normalised algorithm. The purpose of the family of data-reusing algorithms is to provide an increased convergence rate compared to standard algorithms with minimal increase in computational complexity. The technique of data-reusing is reliant on the $\mathbf{w}(k+1)$ updated weight vector being available before the next input vector $\mathbf{x}(k+1)$. To this cause a new a posteriori output error, $\bar{e}(k)$, and a posteriori output estimate, $\bar{y}(k)$, are calculated as [10]

$$\bar{e}(k) = d(k) - \bar{y}(k), \quad \bar{y}(k) = \Phi(\mathbf{x}^T(k)\mathbf{w}(k+1)). \tag{6}$$

The reusing of the input data can then be repeated and is known as the data-reusing technique, which is expressed for the complex nonlinear gradient descent (CNGD) algorithm as

$$e_l(k) = d(k) - \Phi(\mathbf{x}^T(k)\mathbf{w}_l(k)), \quad 1 \leqslant l \leqslant L \tag{7}$$

$$\mathbf{w}_{l+1}(k) = \mathbf{w}_l(k) + \eta[\Phi'(\mathbf{x}^T(k)\mathbf{w}_l(k))]^*\mathbf{x}^*(k), \quad 1 \leqslant l \leqslant L \tag{8}$$

where L denotes the number of iterations in the data-reusing (DR) algorithm. It can be clearly seen that

$$\mathbf{w}_1(k) = \mathbf{w}(k), \quad \mathbf{w}_{L+1}(k) + \mathbf{w}(k+1) \tag{9}$$

and if $L = 1$ the CDRNGD algorithm reduces to the standard complex nonlinear gradient descent (CNGD) algorithm, (4) and (5). Using (9) and (5) we can then state

$$\begin{aligned}
\mathbf{w}(k+1) &= \mathbf{w}_{L+1}(k) \\
&= \mathbf{w}_L(k) + \eta[\Phi'(\mathbf{x}^T(k)\mathbf{w}_L(k))]^*\mathbf{x}^*(k)e_L(k) \\
&= \mathbf{w}_{L-1}(k) + \eta[\Phi'(\mathbf{x}^T(k)\mathbf{w}_{L-1}(k))]^*\mathbf{x}^*(k)e_{L-1}(k) + \\
&\quad + \eta[\Phi'(\mathbf{x}^T(k)\mathbf{w}_L(k))]^*\mathbf{x}^*(k)e_L(k) \\
&= \mathbf{w}_1(k) + \sum_{i=1}^{L} \Delta\mathbf{w}_i(k) = \mathbf{w}(k) + \sum_{i=1}^{L} \Delta\mathbf{w}_i(k)
\end{aligned} \tag{10}$$

which is the total weight update along the data-reusing iterations. Our aim is to make the consecutive errors $e_l(k) = 1, 2, \ldots, L$ decrease in magnitude along the iterations, i.e. $|e_l(k)| < |e_{l-1}(k)|$. The instantaneous output error for the second iteration of the DR loop then becomes

$$\begin{aligned} e_2(k) &= d(k) - \Phi(\mathbf{x}^T(k)\mathbf{w}_2(k)) \\ &= d(k) - \Phi(\mathbf{x}^T(k)[\mathbf{w}_1(k) + \eta[\Phi'(\mathbf{x}^T(k)\mathbf{w}_1(k))]^*\mathbf{x}^*(k)e_1(k)]) \\ &= d(k) - \Phi(\mathbf{x}^T(k)\mathbf{w}_1(k)) - \Phi(\mathbf{x}^T(k)[\mathbf{w}_1(k)+ \\ &\quad + \eta[\Phi'(\mathbf{x}^T(k)\mathbf{w}_1(k))]^*\mathbf{x}^*(k)e_1(k)]) + \Phi(\mathbf{x}^T(k)\mathbf{w}_1(k)) \\ &= e_1(k) - \Phi(\mathbf{x}^T(k)[\mathbf{w}_1(k) + \eta[\Phi'(\mathbf{x}^T(k)\mathbf{w}_1(k))]^*\mathbf{x}^*(k)e_1(k)])+ \\ &\quad + \Phi(\mathbf{x}^T(k)\mathbf{w}_1(k)) \end{aligned} \tag{11}$$

giving a general algorithm for any iteration of the DR loop of

$$e_L(k) = e_{L-1}(k) - \Phi\left(\mathbf{x}^T(k)\left[\mathbf{w}_1(k) + \sum_{l=1}^{L} \Delta\mathbf{w}_l(k)\right]\right) + \Phi(\mathbf{x}^T(k)\mathbf{w}_{L-1}(k)). \tag{12}$$

On the other hand the normalised complex nonlinear gradient descent (NCNGD) algorithm calculates an adaptive learning rate according to $e(k+1) = 0$ using the dynamics of the input signal. The equations that define the NCNGD algorithm can be derived from [11]

$$e_N(k) = d(k) - \Phi(\mathbf{x}^T(k)\mathbf{w}_N(k)), \tag{13}$$

$$\mathbf{w}_N(k+1) = \mathbf{w}_N(k) + \eta(k)[\Phi'(\mathbf{x}^T(k)\mathbf{w}(k))]\mathbf{x}^*(k)e(k) \tag{14}$$

where the subscript N denote the variables used in the NCNGD algorithm and $\eta(k)$[1] denotes the adaptive learning rate in the NCNGD algorithm.

3.1. CONVERGENCE OF DATA-REUSING ALGORITHMS

For the data reusing algorithm to converge to the normalised algorithm the condition $\lim_{L\to\infty} e_L(k) = e_N(k)$ must bold. For simplicity, the limit is ignored to give

$$\begin{aligned} &e_{L-1}(k) - \Phi\left(\mathbf{x}^T(k)\left[\mathbf{w}_{L-1}(k) + \sum_{l=1}^{L} \Delta\mathbf{w}_l(k)\right]\right) + \Phi(\mathbf{x}^T(k)\mathbf{w}_{L-1}(k)) \\ &\quad = d(k) - \Phi(\mathbf{x}^T(k)\mathbf{w}_N(k)) \end{aligned} \tag{15}$$

It has already been stated in (7), that $e_l(k) + \Phi(\mathbf{x}^T(k)\mathbf{w}_l(k)) = d(k)$, thus the equality (15) reduces to

$$\Phi\left(\mathbf{x}^T(k)\left[\mathbf{w}_1(k) + \sum_{i=1}^{L} \Delta\mathbf{w}_l(k)\right]\right) = \Phi(\mathbf{x}^T(k)\mathbf{w}_N(k)). \tag{16}$$

In order to preserve (16), it stands that

[1] The adaptive learning rate in the NCNGD algorithm is calculated via a Taylor series expansion on the instantaneous output error, yielding $\eta(k) = \frac{1}{|\Phi'(\mathbf{x}^T(k)\mathbf{w}(k))|^2 ||\mathbf{x}(k)||^2 + C}$, where C is added to compensate the exclusion of second and higher order derivatives [5].

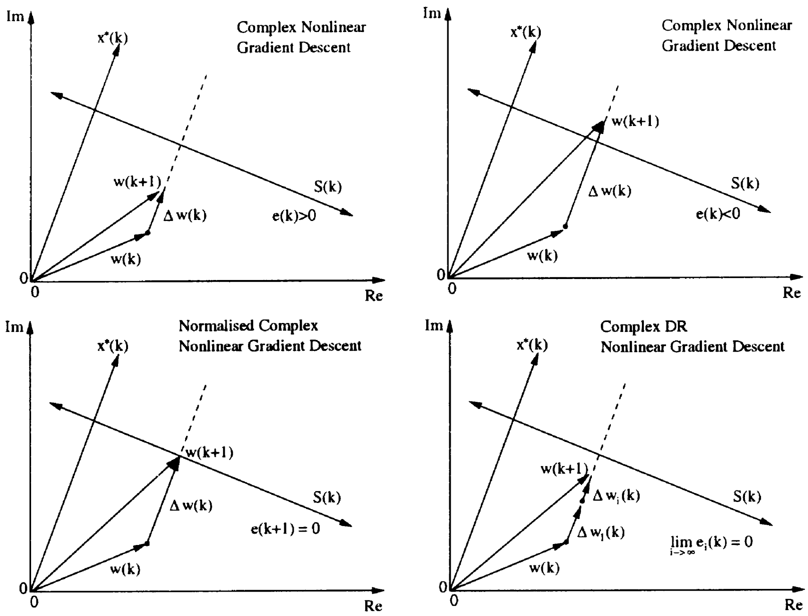

Figure 2. Geometric interpretation of CNGD, NCNGD and CDRNGD algorithms.

$$\lim_{L\to\infty}\left\{\mathbf{w}_1(k)+\sum_{l=1}^{L}\Delta\mathbf{w}_l(k)\right\}=\mathbf{w}_N(k), \tag{17}$$

which has already been proved for linear filters [8].

To illustrate the approach, Figure 2 shows simplified weight update for the complex nonlinear gradient descent (CNGD), normalised complex nonlinear gradient descent (NCNGD), and complex data-reusing nonlinear gradient descent (CDRNGD) algorithms. The hypersurface $S(k)=0$ is the solution space for which the output error is zero. From (5), it is clear that the weight update term has the same direction as the conjugate input vector for the CDRNGD algorithm and as the number of iterations in the data-reusing loop tends to infinity, the weight vector approaches $S(k)$. Figure 2 shows that the NCNGD algorithm minimizes the a posteriori instantaneous output error $e(k+1)=0$ based upon the dynamics of the current input signal [11], and hence the data-reusing algorithm approaches the NCNGD algorithm for large L.

4. Experimental Results

To investigate the performance of the CDRNGD and NCNGD algorithms, they were applied to the problem of time-series prediction, by averaging the performance curves of 300 independent iterations on a benchmark nonlinear input given by [12],

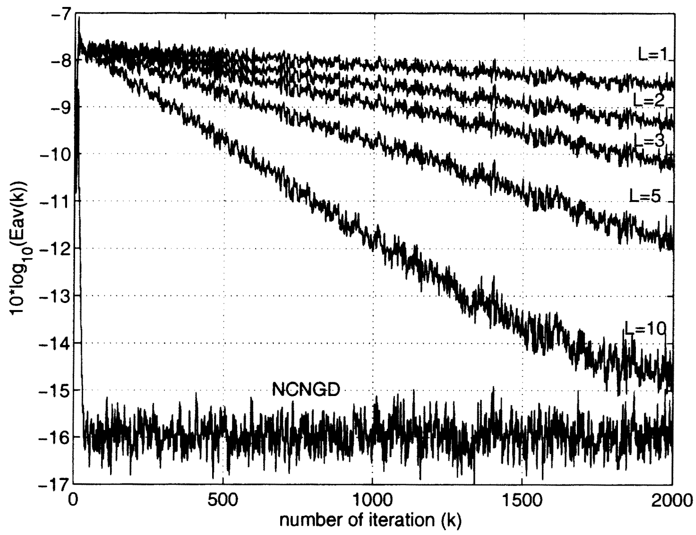

Figure 3. Comparison of performance curves for CDRNGD and NCNGD algorithms.

$$z(k) = \frac{z(k-1)}{(1 + z^2(k-1))} + r^2(k), \tag{18}$$

where $r(k)$ was a normally distributed $\mathcal{N}(0, 1)$ white noise, $n(k)$, passed through a stable AR filter given by

$$r(k) = 1.79r(k-1) - 1.85r(k-2) + 1.27r(k-3) - 0.41r(k-4) + n(k). \tag{19}$$

In all the experiments the order of the filter was $N = 10$, and the nonlinearity was the logistic sigmoid function defined as

$$\Phi(z, \beta) = \frac{1}{1 + \mathrm{e}^{-\beta z}} \tag{20}$$

where $z \in \mathbb{C}$ and $\beta = 1$ controls the slope of the nonlinearity. To be able to clearly visualize the merit of the algorithm, we utilise a small learning rate, for instance $\eta = 0.001$ Figure 3 shows the performance curves for the complex data-reusing nonlinear gradient descent (CDRNGD) algorithm using a contractive activation function on $L = 1$, $L = 2$, $L = 3$, $L = 5$ and $L = 10$ as the parameter for the data-reusing loop. The CDRNGD algorithm outperformed the complex nonlinear gradient descent (CNGD) algorithm ($L = 1$). As the number of iterations in the data-reusing loop increased, the speed of convergence improved illustrating connection with the normalised complex nonlinear gradient descent (NCNGD) algorithm in the limit, as shown analytically above.

5. Conclusions

Affinity between the class of complex data-reusing nonlinear gradient descent (CDRNGD) algorithms and the normalised complex nonlinear gradient descent

(NCNGD) algorithm has been established. The error bounds of such data-reusing algorithms have been established, confirming the stability of such algorithms. Monte Carlo simulations show the CDRNGD algorithm outperforming the standard complex nonlinear gradient descent (CNGD) algorithm and converging to the NCNGD algorithm in the limit.

References

1. Widrow, B., McCool, J. and Ball, M.: 'The complex LMS algorithm,' *Proceedings of the IEEE* **63** (1975), 719–720.
2. Hirose, A.: 'Continuous complex-valued backpropagation learning,' *Electronics Letters* **28**(20) (1992), 1854–1855.
3. Georgiou, G. M. and Koutsougeras, C.: 'Complex domain backpropagation,' *IEEE Transactions on Circuits and Systems – II: Analog and Digital Signal Processing* **39**(5) (1992), 330–334.
4. Benvenuto, N. and Piazza, F.: 'On the complex backpropagation algorithm,' *IEEE Transactions on Signal Processing* **40**(4) (1992), 967–969.
5. Hanna, A. I., Mandic, D. P. and Razaz, M.: 'A normalised backpropagation learning algorithm for multilayer feed-forward neural adaptive filters,' *Proceedings of the XI IEEE Workshop on Neural Networks for Signal Processing*, 2001, pp. 63–72.
6. Mandic, D. P., Hanna, A. I. and Razaz, M. 'A normalized gradient descent algorithm for nonlinear adaptive filters using a gradient adaptive step size,' *IEEE Signal Processing Letters*, **8**(11) (2001), 1–3.
7. Mandic, D. P. and Chambers, J. A.: *Recurrent Neural Networks for Prediction*, John Wiley and Sons, 2001.
8. Roy, S. and Shynk, J.: 'Analysis of the data-reuse LMS algorithm,' *Proceedings of the 32nd Midwest Symposium on Circuits and Systems*, **2**, 1989, pp. 1127–1131.
9. Mandic, D. P. and Chambers, J. A.: 'A posteriori real time recurrent learning schemes for a recurrent neural network based non-linear predictor,' *IEE Proceedings – Vision, Image and Signal Processing*, **145**(6), 1998, pp. 365–370.
10. Mandic, D. P. and Chambers, J. A.: 'Relations between the a priori and a posteriori errors in nonlinear adaptive neural filters,' *Neural Computation*, **12**(6) (2000), 1285–1292.
11. Douglas, S.: 'A family of normalized LMS algorithms,' *IEEE Signal Processing Letters*, **1**(3) (1994), 49–51.
12. Narendra, K. S. and Parthasarathy, K.: 'Identification and control of dynamical systems using neural networks,' *IEEE Transactions on Neural Networks*, **1**(1) (1990), 4–27.