Composite Squared-Error Algorithm for Training Feedforward Neural Networks

Dirceu Gonzaga,‡ Marcello L. R. de Campos,‡ and Sergio L. Netto†

‡Departamento de Engenharia Elétrica
Instituto Militar de Engenharia
Praça General Tibúrcio, 80
CEP 22290-270, Rio de Janeiro, RJ - Brazil

†Departamento de Eletrônica/EE
Universidade Federal do Rio de Janeiro
P.O. Box 68564
CEP 21945-970, Rio de Janeiro, RJ - Brazil

Abstract

A new algorithm, the so-called composite squared-error (CSE) algorithm, for training neural networks is presented. The CSE algorithm, whose roots lie on the field of adaptive IIR filtering, is able to avoid suboptimal solutions and associated saddle points, thus achieving lower values of the associated mean-squared-error function in a fewer number of iterations. For that matter, the CSE algorithm can regularly outperform other existing training schemes in most applications where neural networks are employed.

I. Introduction

Neural networks have been used as an efficient tool for solving a wide variety of problems in signal processing and control. Feedforward neural networks trained with the backpropagation algorithm [1] have become popular in numerous applications. Despite its relative success, the rate of convergence of the backpropagation algorithm when used to train multilayer neural networks is often not satisfactory. Even simple classification problems may require long training periods before convergence is achieved. Poor performance is usually attributed to the minimization of a nonquadratic function, possibly multi-modal, by a gradient-type algorithm.

In this article, we explore the great similarities between feedforward neural networks and their counterparts in adaptive filter theory. We propose a composite algorithm that has faster convergence speed than the backpropagation algorithm, may achieve a lower mean-squared output error (MSE) than the fast new (FN) algorithm presented in [2], and has a computational complexity only marginally greater than that of the backpropagation and the FN algorithms. In particular, we establish a parallel between the backpropagation and FN algorithms with the output error (OE) and equation error (EE) schemes, respectively, defined in the field of infinite-duration impulse response (IIR) adaptive filtering [3]. Hence we are able to employ the composite squared-error (CSE) algorithm [4]–[6] for training neural networks.

II. Backpropagation Algorithm

The backpropagation algorithm has gained general acceptance for training feedforward neural networks [1]. The algorithm utilizes a gradient method to update the weights of a neural network by minimizing the output error $e_{p,L,k}$, formed by the desired signal $d_{p,L,k}$ for the corresponding training-pattern signal $x_{p,0,k}$, and the signal at the output layer of the network, $x_{p,L,k}$ (see Figure 1). The error is directly used to update the weights of the output layer and it is propagated back to the hidden layers of the network in order to update these weights. For the sake of simplicity and brevity we only state the equations involved, and we refer the reader interested in a more complete discussion on the method to the many references treating the subject (see, e.g., [1]).

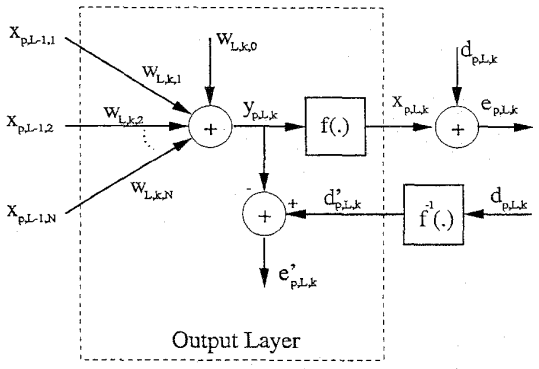

Figure 1: Modified neuron structure for the output layer.

At the n-th iteration and p-th pattern, the i-th weight of the k-th neuron of the L-th (output) layer

is updated as

$$w(n+1)_{L,k,i} = w(n)_{L,k,i} + \mu e_{p,L,k} x_{p,L-1,i} + \eta\,[w(n)_{L,k,i} - w(n-1)_{L,k,i}] \quad (1)$$

where μ is the step-size, η is the momentum gain, $x_{p,L,i}$ denotes the output signal of the i-th neuron of the L-th layer. The output error is calculated as

$$e_{p,L,k} = f'(y_{p,L,k})\,[f(d_{p,L,k}) - x_{p,L,k}] \quad (2)$$

where $y_{p,L,k}$ denotes the signal at the output of the summation node, and $f'(\cdot)$ is the first derivative of $f(\cdot)$. For the j-th (hidden) layer, the i-th weight is updated as

$$w(n+1)_{j,k,i} = w(n)_{j,k,i} + \mu e_{p,j,k} x_{p,j-1,i} + \eta\,[w(n)_{j,k,i} - w(n-1)_{j,k,i}] \quad (3)$$

with the error calculated as

$$e_{p,j,k} = f'(y_{p,j,k}) \sum_{i=1}^{k} e_{p,j+1,i} w_{j+1,i,k} \quad (4)$$

$w_{j+1,i,k}$ refers to the k-th weight of i-th neuron of the $(j+1)$-th layer.

III. Fast New Algorithm

The nonquadratic objective function minimized by the backpropagation algorithm arises due to the nonlinear operator, $f(\cdot)$, placed after each summation node. This nonlinearity may generate local minima and associated saddle points that cause slow convergence. As shown in Fig. 1, by means of using the inverse of the nonlinear operator, a linear error $e'_{p,L,k}$ with respect to the weights of the output layer may be constructed yielding a quadratic objective function to be minimized. If, for simplicity, momentum is not considered, the output-layer weights are updated using this linearized error as [2]

$$w(n+1)_{L,k,i} = w(n)_{L,k,i} + \mu(d_{p,L,k} - y_{p,L,k}) x_{p,L-1,i} \quad (5)$$

whereas the hidden-layer weights are updated using the nonlinear error backpropagated according to (3) and (4). Although one may argue that the error minimized is still nonlinear with respect to the weights in the hidden layers, several simulations have shown prospective improvement in convergence speed if (5) is used in conjunction with (3) and (4). This may suggest that the influence of the hidden layers on the shape of the objective function minimized by the network is not as significant as the influence of the output layer.

In [2], the authors claim that once the error at the output of the neural network has been reduced to zero, the modified structure has converged to the global minimum. Although indisputably true, zero output error may not be achievable due to noise, incorrect modeling, or too-short training periods. In this case, the solution obtained after completing the training period is biased and a low linearized error $e'_{p,L,k}$ may not correspond to a low output error $e_{p,L,k}$, as originally claimed.

IV. Composite Squared-Error Algorithm for Adaptive Filtering

An adaptive filter is similar to a neural network in the sense that both systems rely on a numerical algorithm to adjust their corresponding coefficients in order to satisfy some prescribed optimization criterion. The main difference between the two approaches relies on the basic structure being updated. In fact, while a neural network consists of nonlinear neurons interconnect through a series of adders, an adaptive filter consists of a linear digital filter, the coefficients of which are made variable in time. In its most general form, an adaptive filter is described by a time-varying input-output relationship of the form:

$$\hat{H}_n(q) = \frac{\hat{B}_n(q)}{\hat{A}_n(q)} = \frac{b_{0,n} + \ldots + b_{N,n} q^{-N}}{1 + a_{1,n} q^{-1} + \ldots + a_{N,n} q^{-N}} \quad (6)$$

where N is the filter order, $q[.]$ is the unit-delay operator defined by $q[x_n] = x_{n-1}$, and $b_{0,n}, \ldots, b_{N,n}, a_{1,n}, \ldots, a_{N,n}$ are the adaptive coefficients. For this type of adaptive filter, the two most widely known adaptation algorithms are the output error (OE) and equation error (EE) algorithms.

The OE algorithm is based on the error signal, $e_{OE,n}$, between the filter output and the refence signal. For that matter, the OE scheme represents the adaptive filter counterpart of the backpropagation algorithm for neural networks.

On the other hand, the EE algorithm is based on an alternative error signal, $e_{EE,n}$, defined as

$$e_{EE,n} = \hat{A}_n(q)[e_{OE,n}] \quad (7)$$

where $\hat{A}_n(q)$ is the adaptive filter's denominator polynomial. Then we may consider the EE algorithm as the adaptive filter analogous of the FN algorithm for neural networks.

With respect to convergence properties, the MSE funtion associated to the OE algorithm is characterized by the possible existence of local minima and an unbiased global minimum. Meanwhile, the MSE function for the EE algorithm represents a quadratic function whose unique optimal solution may be biased, when

compared to the OE global solution, due to the presence of modeling/measurement noise in the desired output signal. The convergence properties for the OE and EE schemes seem to indicate that an ideal approach approximates the EE convergence behavior in the initial part of the adaptation process and progressively converts itself to become identical to the OE scheme. This methodology is achieved, for instance, with the so-caled composite squared-error (CSE) algorithm which is based on the alternative error signal defined as:

$$e^2_{CSE,n} = \gamma e^2_{EE,n} + (1-\gamma) e^2_{OE,n} \tag{8}$$

where γ is the composition parameter restrained to the interval $\gamma \in [0,1]$.

In the next section, we make use of this composition concept, by combining the backpropagation and FN algorithms, thus deriving the CSE algorithm.

V. Composite Squared-Error Algorithm for Neural Networks

We propose the use of a composite algorithm [4]–[6] which employs the linear error during the initial part of the training period, after which the nonlinear error is used. This approach has the advantage of improving convergence speed and of reducing the risk of convergence to a biased solution.

In the proposed method, the structures shown in Figs. 1 and 2 are used in the first phase of the training period and the linear error $e'_{p,L,k}$ is used to update the weights of the output layer. In the second and final phase of the training period, the nonlinear error $e_{p,L,k}$ is used to update the weights, characterizing an ordinary feedforward neuron structure for hidden and output layers. Convergence in the first phase is usually fast even when the backpropagation algorithm is used. The second part of the training is necessary to remove bias eventually caused by measurement noise, inadequacy of the network employed, or both.

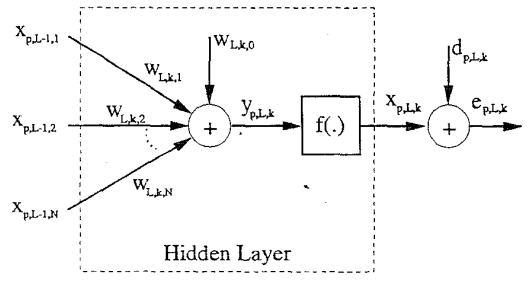

Figure 2: Neuron structure for the hidden layer.

The switch from one updating scheme to the other may be abrupt resulting in two distinct phases, or the updating scheme may employ the composite squared error given by [4]–[6]

$$\bar{e}^2_{p,L,k} = \gamma (e')^2_{p,L,k} + (1-\gamma) e^2_{p,L,k} \tag{9}$$

which results in an updating equation for the weights of the output layer of the form

$$w(n+1)_{L,k,i} = w(n)_{L,k,i} + \gamma \Delta_{lin} w(n)_{L,k,i} + (1-\gamma)\Delta_{non} w(n)_{L,k,i} \tag{10}$$

where

$$\Delta_{lin} w(n)_{L,k,i} = \mu (d_{p,L,k} - y_{p,L,k}) x_{p,L-1,i} \tag{11}$$

$$\Delta_{non} w(n)_{L,k,i} = \mu e_{p,L,k} x_{p,L-1,i} \tag{12}$$

Parameter γ is changed from the initial value $\gamma = 1$ to the final value $\gamma = 0$, according to, for instance, a gradient-type algorithm [4]–[6]

$$\gamma(n+1) = \begin{cases} \gamma(n) - \alpha \left| (e'_{p,L,k})^2 - e^2_{p,L,k} \right|, \\ \qquad \text{if } \gamma(n+1) > 0 \\ 0, \qquad \text{otherwise} \end{cases} \tag{13}$$

VI. Simulation Results

Several simulations were performed for classical problems in function approximation and system identification employing the backpropagation algorithm, the NF algorithm, and the CSE algorithm proposed here. In all simulations we used equations (10)–(13) to gradually change from adptation using the linear error to adaptation using the nonlinear error. We also performed training in batch mode without momentum update and used the hyperbolic tangent function as the nonlinear activation function $f(\cdot)$.

Example 1: XOR Problem

In this example a neural network consisting of 2 neurons in the first layer and 1 neuron in the output layer was used to implement the classical problem where the output signal must be an exclusive-OR operation onto the inputs. Adaptation parameters were set as $\mu = 0.1$ for all algorithms compared. Adaptation of γ for the CSE, following equation (13), used a step-size $\alpha = 0.002$. Fig. 3 shows the MSE for 1000 epochs for the backpropagation, FN, and CSE algorithms. We can clearly verify the superior performance of the proposed method with respect to the other two methods in term of speed of convergence and MSE.

Example 2: Function Approximation

A hypothetical function was approximated by a neural network with 1 layer, with 1 neuron and 1 weight,

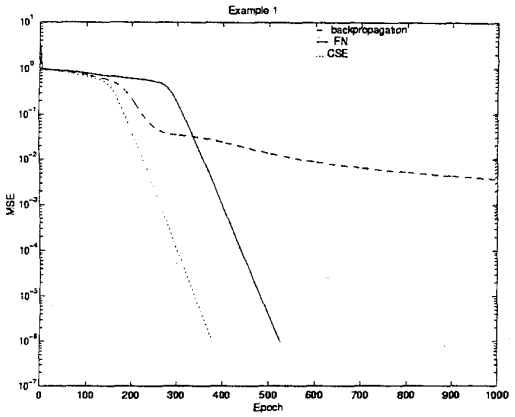

Figure 3: MSE convergence for the XOR problem.

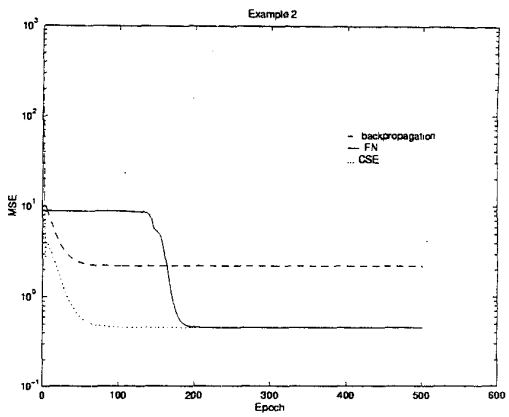

Figure 4: MSE convergence for the function-approximation problem.

and bias. Abscence of hidden layers implies a purely quadratic objective function for the first phase of adaptation, i.e., when $\gamma = 1$. The adaptation parameters were set as $\mu = 0.005$ for all algorithms and $\alpha = 0.002$ for the adaptation of γ in the CSE algorithm. Fig. 4 shows the MSE for 600 epochs of the backpropagation, FN, and CSE algorithms. Clearly, the FN algorithm presented fast initial convergence, but a high steady-state MSE, which probably characterizes convergence to a local minima. The backpropagation algorithm presented low stady-state MSE, but a very slow initial convergence, likely due to a flat region in the objective function. Meanwhile, the CSE algorithm presented very fast convergence and a low final MSE. Figures 5–8 show the MSE contour with respect to the weight and the bias for different epochs, which clearly shows the gradual modification of the objective function minimized by the CSE algorithm, starting as a quadratic function at epoch 0 and finishing as a very nonlinear function at epoch 600.

Example 3: System Identification

In this example a network with 11 neurons in the first layer, 3 neurons in the second layer, and 1 neuron in the last layer was used to identify an unknown plant described by the following transfer function:

$$H(z) = \frac{0.05 - 0.4z^{-1}}{1 - 0.0003z^{-1} - 0.68915z^{-2}}$$

A step-size $\mu = 0.001$ was used for the backpropagation algorithm and for the CSE algorithm, whereas $\mu = 0.0001$ was used for the FN algorithm. Adaptation of γ employed a step-size $\alpha = 0.005$ for the CSE algorithm. The parameters were empirically chosen for the fastest convergence of each algorithm. Fig. 9 shows clearly the superior performance of the CSE algorithm when

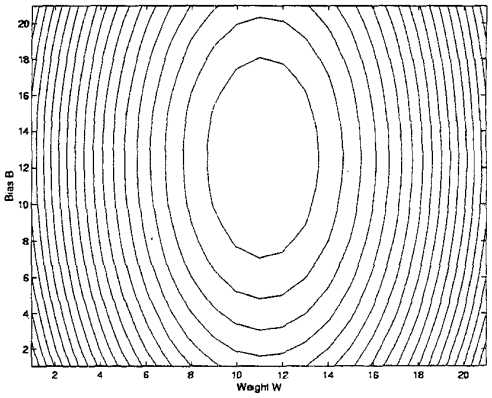

Figure 5: Contour for the function approximation-problem at the start of adaptation.

compared with the other two algorithms, even for a relatively complex neural network. As a matter of fact, the CSE converged faster and attained a lower level of MSE after convergence than the two other algorithms considered.

VII. Conclusions

In this article, a new algorithm for neural network training was presented. In the several simulations carried out, the proposed algorithm converged faster than the conventional backpropagation algorithm with similar computational complexity. This seems to indicate that saddle points are avoided along with the accompanying flat regions often responsible for slow convergence. Convergence to a global minimum was also regu-

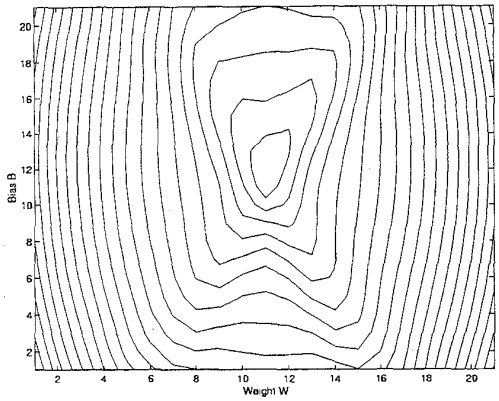

Figure 6: Contour for the function approximation-problem during adaptation.

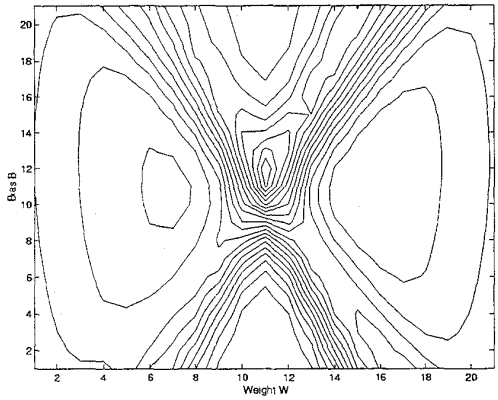

Figure 7: Contour for the function approximation-problem durig adaptation.

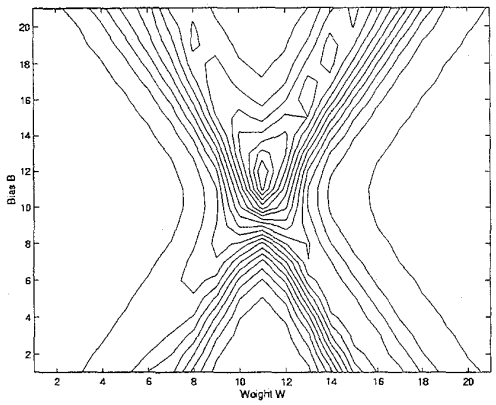

Figure 8: Contour for the function approximation-problem at the final stage of adaptation.

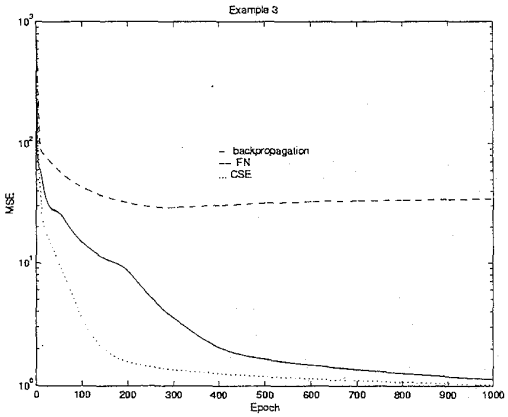

Figure 9: MSE convergence for the system identification problem.

larly verified, thus illustrating an evident improvement over other training methods found in the neural network literature.

References

[1] S. Haykin, *Neural Networks — A Comprehensive Foundation.* New York: IEEE Press, 1994.

[2] R. S. Scalero and N. Tepedelenlioglu, "A fast new algorithm for training feedforward neural networks," *IEEE Trans. Signal Processing*, vol. 40, pp. 202–210, Jan. 1992.

[3] J. J. Shynk, "Adaptive IIR filtering," *IEEE Signal Processing Magazine*, vol. 6, pp. 4–21, Apr. 1989.

[4] S. L. Netto, *On Algorithms, Structures, and Implementation of Adaptive IIR Filters.* PhD thesis, University of Victoria, Victoria, Canada, Feb. 1996.

[5] S. L. Netto and P. Agathoklis, "A new composite adaptive IIR algorithm," *Proc. 28th Asilomar Conf. Signals, Syst., Computers*, Pacific Grove, CA, pp.1506–1510, Oct./Nov. 1994.

[6] S. L. Netto and P. Agathoklis, "On the composite squared error algorithm for adaptive IIR filters," *Proc. 31st Asilomar Conf. Signals, Syst., Computers*, Pacific Grove, CA, 1997.