Fast Adaptive Digital Equalization by Recurrent Neural Networks

Raffaele Parisi, Elio D. Di Claudio, Gianni Orlandi, and Bhaskar D. Rao

***Abstract*—In recent years, neural networks (NN's) have been extensively applied to many signal processing problems. In particular, due to their capacity to form complex decision regions, NN's have been successfully used in adaptive equalization of digital communication channels. The mean square error (MSE) criterion, which is usually adopted in neural learning, is not directly related to the minimization of the classification error, i.e., bit error rate (BER), which is of interest in channel equalization. Moreover, common gradient-based learning techniques are often characterized by slow speed of convergence and numerical ill conditioning. In this paper, we introduce a novel approach to learning in recurrent neural networks (RNN's) that exploits the principle of *discriminative learning*, minimizing an error functional that is a direct measure of the classification error. The proposed method extends to RNN's a technique applied with success to fast learning of feedforward NN's and is based on the descent of the error functional in the space of the linear combinations of the neurons (the *neuron space*); its main features are higher speed of convergence and better numerical conditioning w.r.t. gradient-based approaches, whereas numerical stability is assured by the use of robust least squares solvers. Experiments regarding the equalization of PAM signals in different transmission channels are described, which demonstrate the effectiveness of the proposed approach.**

I. Introduction

ADAPTIVE channel equalization is a major issue in digital communications [2], [4], [24]. Fig. 1 depicts the typical digital baseband transmission system; the channel model takes into account the effects of the transmitter, the transmission medium, and the receiver and is usually represented by a finite impulse response (FIR) filter. The input to the channel is assumed to be a sequence $\{s(k)\}$ of independent symbols extracted from a specified alphabet; the channel output $\{\hat{u}(k)\}$ is corrupted by noise $\{n(k)\}$ and is usually modeled as an additive Gaussian white process. The transmission channel can be affected by both linear and nonlinear distortion; in the first case, intersymbol interference (ISI) occurs as a consequence of the limited bandwidth of the channel and consists of spreading of the received symbol energy through several time intervals. In the second case, the channel cannot be considered linear due to the presence of nonlinear devices (amplifiers working in saturation, converters, ...). The objective of equalization is to reconstruct the transmitted sequence and combat the effects of ISI and noise.

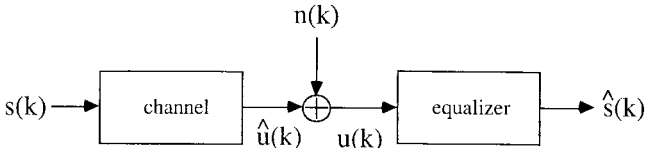

Fig. 1. Schematic of data transmission system.

The equalization problem can be viewed from two different viewpoints. Traditionally, equalization has been considered equivalent to inverse filtering of the channel; this corresponds to deconvolving the received sequence in order to reconstruct the original message; therefore, the combination of channel and equalizer should be as close as possible to an ideal delay function [2], [24].

A different approach considers equalization as a *classification* problem [15], in which the objective is the separation of the received symbols in the output signal space. In this case, the full inversion of the channel is not required, and the problem can be cast in the general framework of classification techniques.

From both points of view, the NN approach to equalization is well justified: in the first case, NN capability as universal function approximators [9] could be exploited; in the second, it is the well-known NN ability to perform classification tasks by forming complex nonlinear decision boundaries. In particular, it can be shown that feedforward NN's [25] can implement the maximum *a posteriori* probability (MAP) symbol decision equalizer [22]. For all these reasons, in recent years, NN's have been successfully applied to the equalization problem. In particular, recurrent NN's (RNN's) [25] are attractive for the presence of feedback and their small size [26].

The choice of the particular equalizer involves both the architecture and the training algorithm. Linear equalizers have been used for long time, mainly due to their simplicity and theoretical tractability; for many typical situations, the use of some form of nonlinearity is more appropriate [15].

NN equalizers work by processing linear combinations of received samples passed through nonlinear "activation" functions; learning consists in the determination of the "optimal" weights through the minimization of a specified error functional, whose choice should be related to the particular task considered. Most NN approaches use the mean square error (MSE), which is not directly related to the classification

Manuscript received June 17, 1997. The work of R. Parisi, E. D. Di Claudio, and G. Orlandi was supported in part by the Italian Ministery for University and Scientific and Technological Research (M.U.R.S.T.). The associate editor coordinating the review of this paper and approving it for publication was Prof. Jenq-Neng Hwang.

R. Parisi, E. D. Di Claudio, and G. Orlandi are with the INFOCOM Department, University of Rome, "La Sapienza," Rome, Italy.

B. D. Rao is with the Department of Electrical and Computer Engineering, University of California, La Jolla, CA 92037 USA.

Publisher Item Identifier S 1053-587X(97)08058-6.

error but rather to the quality of system identification. In [19], a new error functional has been proposed that takes directly into account the classification error; the new method has been successfully applied to general classifier structures and, in particular, to NN's. Authors called this approach *discriminative learning*.

The determination of equalizer weights is essentially an optimization issue. Today, high-speed data transmission over distorting channels is a commonly encountered situation; fast optimization methods for the design of the "optimal" equalizer are thus required. In the case of a "neural equalizer," due to its inherently nonlinear nature, the need of fast and stable training algorithms is particularly important. Many different approaches to NN learning exist. The main problems are usually the slow rate of convergence and the occurrence of local minima; both these drawbacks are essentially due to the high degree of nonlinearity of the error surface. Moreover, a recent analysis [23] has demonstrated that learning in NN's is very often an ill-conditioned problem since the Hessian matrix [6] is badly ill conditioned; this implies that a maximum likelihood (ML) weight identification problem may suffer from lack of information (e.g., the *Fisher's information matrix* may be nearly singular [3], [17]).

Most common learning techniques involve the use of the gradient of the specified error functional (like *backpropagation* [5] in feedforward NN's). Gradient-based approaches, even if they are computationally simpler, are characterized by low rates of convergence and may not be suitable in applications where fast convergence is required. In analogy with the signal processing field, least squares (LS) methods could be envisaged to speed up convergence; in any event, the application of LS concepts to highly nonlinear structures like NN's requires an appropriate treatment.

In the present paper, we introduce a novel LS-based learning method for fully recurrent networks that minimizes the classification error through application of the discriminative learning criterion. The proposed approach is able to provide higher speed of convergence w.r.t. gradient-based solutions; moreover, it overcomes the difficulties related with ill conditioning that is typical of NN learning, giving learning procedures that are numerically stable and robust. The resulting *discriminative least squares* (DLS) learning approach can be successfully applied to the problem of digital equalization. In the following, we review some neural approaches to equalization present in the literature (Section II), and we introduce the new method, which takes into account both the requirements of minimum classification error and high speed of convergence (Section III). Finally, we apply it to the equalization of PAM signals in some typical transmission channels (Section IV).

II. Neural Approaches to Equalization

Traditional approaches consider equalization to be an inverse filtering problem, and the equalizer should approximate the inverse of the distorting channel. This approach in digital communications can be more complex than necessary; due to the quantized nature of the transmitted symbols, in order to equalize the channel, it is sufficient to ensure that the decision on the equalizer output is correct. This means that equalization can be viewed also as a *geometrical* problem, consisting of correctly establishing the boundaries of the decision regions in the output signal space. This interpretation was first pointed out in [13] and [15] and corresponds to considering equalization as a classification problem. The use of NN's is justified by noting that in most cases, the boundaries of the optimal decision regions are highly nonlinear, thus requiring the use of nonlinear classifiers, even with linear channels.

Many possible approaches to neural equalization have been developed in the last few years. Kirkland *et al.* [20] applied feedforward NN's to equalize the digital microwave radio channel in the presence of multipath fading.

Peng *et al.* [21] modified the nonlinear activation function of the classical multilayer perceptron in order to take into account signals typically encountered, namely, PAM and QAM.

Kechriotis *et al.* [26] applied fully recurrent NN's trained with the real-time recurrent-learning algorithm (RTRL) [25] to the equalization of nonminimum phase, partial response, and nonlinear channels; they compared their neural equalizer to linear FIR equalizers trained with the Kalman algorithm. Moreover, with a proper modification of the error functional, they extended their analysis to the case of blind equalization [17].

Chang *et al.* [27] introduced a neural-based decision feedback equalizer to perform equalization of indoor radio channel. The new structure advantageously compares to the classical decision feedback equalizer [24].

In [28], a wavelet NN [18] trained with the recursive least squares (RLS) algorithm [17] was used to equalize a nonlinear transmission channel. Later, the same authors successfully applied their idea to satellite channels [30].

Al-Mashouq *et al.* [29] used a feedforward NN to perform both equalization and decoding in the presence of severe ISI conditions; their equalizer outperforms classical structures formed by cascading a linear equalizer and a decoder.

All these papers showed that NN's can be successfully applied to the problem of equalization; in particular, recurrent NN's are characterized by feedback, which makes them attractive in the presence of channels with deep spectral nulls [26].

As a matter of fact, most training algorithms for RNN's are gradient-based, and this is in contrast with the requirements of fast equalization. For an extensive review of gradient-based approaches to the training of dynamic RNN's, see [31].

The interpretation of channel equalization as classification in symbol space [22] also enables the use of neural architectures that make use of explicit clustering of input patterns during learning, such as radial basis function (RBF) and wavelet networks [18], [25].

The signal processing field has inspired a number of neural learning techniques; in particular, the analogy with adaptive filters has led researchers to consider the use of LS concepts to speed up learning in feedforward architectures. Several approaches of this kind have been presented in the literature, and a review can be found in [33].

In the following, a new LS-based fast approach to the training of recurrent NN's is introduced; the described algorithm extends to recurrent structures the concept of *neuron space* descent proposed in [33]. Taking into account the objective of direct minimization of the classification error, the new approach is interpreted as a supervised symbol clustering procedure, coupled with a statistically robust LS fitting [1], that can be successfully applied to the digital equalization problem.

III. Novel Approach: Discriminative Least Squares (DLS) Learning

As already pointed out, two main issues are considered in this paper: the choice of a proper error functional, which takes directly into account the objective of minimum error classification, and the need of fast convergence procedures for high-rate digital equalization. Let us consider these aspects separately.

A. Discriminative Learning

In this section, we briefly recall the fundamental concepts of discriminative learning introduced in [19] and which will be used in the following; interested readers can refer to [19] for more details.

Learning in NN's consists of the minimization of a specified error functional, whose choice depends on the particular task under consideration; the most common choice is the MSE, which offers desirable properties of smoothness and mathematical tractability. For classification purposes, the minimization of the MSE can be inconsistent with the objective of minimum error probability [19]. The need for a smooth and differentiable error functional, depending on the probability of misclassification involved in the decision process, can be satisfied in the following way.

Suppose that the aim of training is to associate an input pattern $\mathbf{z}$ to one of M possible classes. As a first step, M discriminant functions $g_i(\mathbf{z}, \mathbf{w})$, depending on the parameters $\mathbf{w}$, are introduced; they can be, for example, the outputs of a NN, whereas $\mathbf{w}$ can be the network weight vector.

The second step is the choice of an appropriate misclassification measure, which is continuous with respect to the weights $\mathbf{w}$; a possible definition is

$$d_i(\mathbf{z}) = -g_i(\mathbf{z}, \mathbf{w}) + \left\{ \frac{1}{M-1} \sum_{j,\, j \neq i} g_j^{\mu}(\mathbf{z}, \mathbf{w}) \right\}^{1/\mu} \tag{1}$$

where μ is a positive number. Equation (1) gives a measure of the classification error when the input belongs to the ith class; in the simple case of two classes, it reduces to the difference between the outputs; therefore, $d_i(\mathbf{z}) > 0$ means misclassification, whereas $d_i(\mathbf{z}) \leq 0$ implies a correct decision.

As a third step, the following error functional is defined as a function of the misclassification measure.

$$l_i(\mathbf{z}, \mathbf{w}) = l_i[d_i(\mathbf{z})] \tag{2}$$

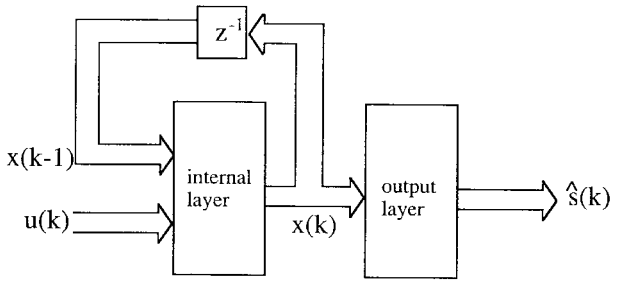

Fig. 2. Scheme of the RNN used as equalizer.

where l_i is a differentiable zero-one function, like a sigmoid or an exponential. The objective of learning is thus the minimization of the error functional l_i w.r.t. the weights, which can be performed by applying well-known methods in optimization theory.

This formulation allows us to express the minimum classification error (or *Bayes minimum risk*) directly in terms of the functionals l_i when the discriminant functions $g_i(\mathbf{z}, \mathbf{w})$ give exactly the *a posteriori* probability of the ith class given $\mathbf{x}$; this means that the minimum classification probability objective is conditioned on the choice of the correct discriminant functions. Due to their function approximation capabilities, NN's with the proper number of units are potentially able to converge to the true minimum Bayes risk [19]. The application of this new approach to multilayer perceptrons has led to positive results in classification and speech recognition experiments, solving the inconsistency between an MSE-based learning and the desired minimization of the misclassification probability [19].

B. Least Squares Learning Algorithm

Learning in NN's is a nonlinear optimization problem. The determination of the optimal weights for the task of interest is performed by a proper algorithm of descent on the surface defined by the specified error functional E; in this work, we will adopt for E the expression given by (2). The learning method proposed in this paper is based on the separability of the neuron model in a linear and a nonlinear part. This particular feature allows to apply linear LS techniques with a proper preliminary treatment of the nonlinearities: the *neuron space approach*. Extensive description of this general method in the case of feedforward NN's has been given in [33]; in particular, it has been demonstrated that the neuron space approach is equivalent to a *modified Newton's method* [6] in weight space, where a proper well-conditioned approximation of the Hessian is used. In this section, we show how this approach can be extended to RNN's, giving the high convergence rates needed in fast equalization. A preliminary description of the algorithm can be found in [32].

1) The Network Model: The structure being considered is depicted in Fig. 2 and consists of a fully connected RNN in cascade to a feedforward network. In the following, we will refer to the case in which both the recurrent and the feedforward parts consist of a single layer of memoryless nonlinearities. This structure is similar to the Elman net [12], with the addition of the feedforward part having the role of decisor.

The network implements the following nonlinear discrete-time dynamical system.

$$\begin{cases} \mathbf{x}(k) = \phi[\mathbf{x}(k-1), \mathbf{u}(k)] \\ \hat{\mathbf{s}}(k) = \psi[\mathbf{x}(k)] \end{cases} \tag{3}$$

where ϕ and ψ are the functionals modeled by the internal and the output sections, respectively, and $\mathbf{x}(k)$, $\mathbf{u}(k)$, and $\hat{\mathbf{s}}(k)$ are column vectors representing, respectively, the *state* of the net, the external input (e.g., channel output + noise), and the output (e.g., estimated symbol) at time k ($k = 1, 2, \cdots$).

Using a matrix formulation, forward propagation through the internal and the output blocks can be represented by

$$\begin{cases} [\mathbf{x}^T(k-1) \quad 1 \quad \mathbf{u}^T(k)]\,\mathbf{W}_i = \mathbf{y}_i^T(k) \\ \mathbf{x}(k) = f[\mathbf{y}_i(k)] \end{cases} \tag{4}$$

and

$$\begin{cases} [\mathbf{x}^T(k) \quad 1]\,\mathbf{W}_o = \mathbf{y}_o^T(k) \\ \hat{\mathbf{s}}(k) = f[\mathbf{y}_o(k)] \end{cases} \tag{5}$$

respectively. In the preceding formulas, $\mathbf{W}_i$ and $\mathbf{W}_o$ are the weight matrices of the internal and the output sections, $\mathbf{y}_i$ and $\mathbf{y}_o$ are the outputs of the linear combinations (e.g., the inputs of the nonlinearities), 1 is the bias input, and f is a place holder for the selected neuron activation sigmoidal-type function.

Learning in recurrent networks can be performed following several possible approaches. Among them the Real-Time Recurrent-Learning (RTRL) is probably the most popular [8]; it consists in the minimization of the error functional based on an instantaneous estimate of the gradient, and gives a structure operating in real time.

The approach herein described gets inspiration from the *time unfolding* technique [25], which expands the network through a number of subsequent time steps. Learning on the unfolded network could be performed by the *backpropagation through time* (BPTT) approach [11], which is an extension of the classical backpropagation algorithm to recurrent architectures; this method is in contrast with the on-line requirements of the training process. We propose a different solution that, although it is on-line, has been demonstrated to provide higher rate of convergence and better numerical properties w.r.t. gradient-based solutions [33].

2) Definitions: We first describe the epochwise form of the algorithm; later, we will show how the update of the weights can be made in real time. Referring to a single epoch of length h, we introduce the following matrices:

1-matrix $\mathbf{U}$, containing the external inputs

$$\mathbf{U}(k : k+h-1) = \begin{bmatrix} \mathbf{u}^T(k) \\ \mathbf{u}^T(k+1) \\ \cdots \\ \mathbf{u}^T(k+h-1) \end{bmatrix}; \tag{6}$$

2-matrix $\mathbf{X}$ containing the internal states

$$\mathbf{X}(k-1 : k+h-2) = \begin{bmatrix} \mathbf{x}^T(k-1) \\ \mathbf{x}^T(k) \\ \cdots \\ \mathbf{x}^T(k+h-2) \end{bmatrix}; \tag{7}$$

3-matrix $\mathbf{Y}$ containing the linear outputs of the generic layer

$$\mathbf{Y}(k : k+h-1) = \begin{bmatrix} \mathbf{y}^T(k) \\ \mathbf{y}^T(k+1) \\ \cdots \\ \mathbf{y}^T(k+h-1) \end{bmatrix}. \tag{8}$$

$\mathbf{Y}$ can refer either to the internal or to the output layer.

At the beginning of each epoch, it is supposed that weight matrices $\mathbf{W}_i$ and $\mathbf{W}_o$ were either set by a previous iteration of the algorithm or properly initialized from scratch. The initial state $\mathbf{x}(k-1)$ and the input matrix for the present epoch $\mathbf{U}(k : k+h-1)$ are also assumed known. Forward equations (4) and (5) are then used to compute matrices $\mathbf{X}(k : k+h-1)$ and $\mathbf{Y}(k : k+h-1)$, thus establishing a consistent set of input-state and state-output relationships.

3) The Neuron Space Approach: The neuron space approach consists of two steps. The first step estimates the "optimal" $\mathbf{Y}$ for each layer by performing a descent of the error surface in the space of the $\mathbf{y}$'s (the *neuron space*). This can be accomplished by introducing a proper *direction matrix* $\mathbf{D}$, as described by

$$\hat{\mathbf{Y}}(k : k+h-1) = \mathbf{Y}(k : k+h-1) + \eta \mathbf{D}(k : k+h-1) \tag{9}$$

where η is the *step-size* [6], which is also called the *learning rate* in the neural field. Different choices of $\mathbf{D}$ are possible, as is known from optimization theory [6]. The simplest choice is the opposite of the *gradient matrix*[1] $\nabla_{\mathbf{Y}} E$; in this case, we get

$$\hat{\mathbf{Y}}(k : k+h-1) = \mathbf{Y}(k : k+h-1) - \eta \nabla_{\mathbf{Y}} E. \tag{10}$$

At each iteration, the gradient of the error is subtracted to the actual $\mathbf{Y}$, giving the estimate $\hat{\mathbf{Y}}$ (*gradient descent* in neuron space). The expressions for the partial derivatives of E can be obtained by applying the chain rule of derivatives [5], [25].

With respect to feedforward NN's [33], a proper treatment is required by the state variables $\mathbf{x}$'s, which in the next step are fed back to the input. Namely, we compute a set of perturbed $\mathbf{x}$'s from

$$\hat{\mathbf{X}}(k : k+h-1) = f[\hat{\mathbf{Y}}(k : k+h-1)]. \tag{11}$$

The second step is the computation of the new weights; after an entire epoch, the following systems are solved in the LS sense for the weight matrices of the internal and output sections.

$$[\hat{\mathbf{X}}(k-1 : k+h-2) \quad \mathbf{1} \quad \mathbf{U}(k : k+h-1)]\,\mathbf{W}_i^{\text{new}} = \hat{\mathbf{Y}}_i(k : k+h-1) \tag{12}$$

$$[\mathbf{X}(k : k+h-1) \quad \mathbf{1}]\,\mathbf{W}_o^{\text{new}} = \hat{\mathbf{Y}}_o(k : k+h-1). \tag{13}$$

After weights have been computed, a new input is presented, and the algorithm proceeds. At the beginning, both the weights and the state are initialized to small random values; this has been proven effective during simulations.

[1] The *gradient matrix* $\nabla_{\mathbf{Y}} E$ is defined by $\{\nabla_{\mathbf{Y}} E\}_{ij} = \partial E / \partial y_{ij}$, where y_{ij} are the elements of the matrix $\mathbf{Y}$.

4) Remarks: The perturbation of the state expressed by (11) is consistent with the approach followed for the summation variables $\mathbf{y}$'s. It can be viewed as an example of *coordinate descent* method [6], which sequentially minimizes the error with respect to different subsets of unknowns; in this sense, the learning process is a sequence of steps in which weights and state variables are alternately corrected until convergence is reached.

The neuron space method is based on the proper perturbation of an *epoch* of consistent system equations [see (10)], followed by a LS fitting procedure [(12) and (13)], which aims to restore consistency after the perturbation by changing neuron weights. Systems (12) and (13) can be solved by any LS algorithm, like the QR or the singular value decompositions [7]. Readers interested in local convergence issues may refer to [33]. Here, we make several remarks.

1) Any local minimum of the chosen error functional w.r.t. weights is also a *stable point* for the descent equations in the neuron space since the sequence of instantaneous gradient estimates in the neuron space becomes statistically orthogonal to the columns of system matrices in LS equation sets [1], [17].
2) The descent in the neuron space is mathematically equivalent to a *modified Newton's method* with a block-shaped positive semidefinite (and almost always positive definite) matrix playing the role of the Hessian [33].
3) The mathematical properties of this matrix near a local minimum are directly determined by the *sensitivity* of the error functional w.r.t. weights, as expected asymptoptically in a well-posed ML identification problem [3].

5) Block Recursive Least Squares (BRLS) Solution: In this section, we show how by use of a QR-based RLS solution [17], the algorithm can be rendered on-line (e.g., $h = 1$ can be chosen). Suppose that at the generic $(k-1)$th step, the QR decomposition [7] of the solving system has been computed; then, the new input at time k can be appended to the triangular factor $\mathbf{R}$ and the following system formed for the generic layer.

$$\begin{bmatrix} \lambda^{1/2}\mathbf{R}(k-1) \\ (1-\lambda)^{1/2}\mathbf{z}^T(k) \end{bmatrix} \mathbf{W}^{\text{new}} = \begin{bmatrix} \lambda^{1/2}\mathbf{C}(k) \\ (1-\lambda)^{1/2}\hat{\mathbf{y}}^T(k) \end{bmatrix}. \quad (14)$$

In preceding formula, vector $\mathbf{z}(k)$ is defined as $\mathbf{z}(k) = [\hat{\mathbf{x}}^T(k-1) \quad 1 \quad \mathbf{u}^T(k)]^T$ for the internal layer and as $\mathbf{z}(k) = [\mathbf{x}^T(k) \quad 1]^T$ for the output layer; λ is a proper *forgetting factor*. Initially, $\mathbf{R}(0) = \text{diag}\{\epsilon\}$, where ϵ is a small number, and $\mathbf{C}(1) = \mathbf{0}$.

During learning, matrix $\mathbf{C}(k)$ is computed by multiplication

$$\begin{bmatrix} \mathbf{C}(k) \\ \cdots \end{bmatrix} = \mathbf{Q}^T(k-1) \cdot \begin{bmatrix} \lambda^{1/2}\mathbf{C}(k-1) \\ (1-\lambda)^{1/2}\hat{\mathbf{y}}^T(k-1) \end{bmatrix} \quad (15)$$

where matrix $\mathbf{Q}(k-1)$ comes from the QR decomposition of the coefficient matrix at the preceding step.

C. Neural Classification as Multiple Robust Least Squares Fitting

In this section, we will briefly show how the proposed neural approach to digital equalization can be embodied in the framework of *robust LS fitting* [1].

The neuron space approach fits in a LS manner the back-propagated residuals over the forward propagated inputs at each layer in order to update the weights [5], [33].

However, the classification problem also has a striking resemblance also with the robust LS fitting problem. In robust LS, "target signals" belong to a mixture of two different distributions, only one of which is of interest for modeling, whereas the other is a *contaminating*, unknown distribution within a *gross-error* assumption [1].

In equalization problems, we are dealing instead with a mixture of *several* distributions, each generated by one symbol sequence projected onto the output signal space. Only one distribution at a time is present at the equalizer output and is of interest for the definition of the optimal decision boundaries [19], [22]. At the beginning of learning, the neuron weights are far from optimal values, and large errors are present at the network outputs, which can fool or slow down the descent when using a Newton-type algorithm [6], [33]. Explicit clustering of the input space (like in classical RBF network training [25]) may help to attain convergence, but it can be slow.

A well-known alternative approach for quickly fitting (in the LS sense) data drawn from a mixture is to recognize well-behaved equations having relatively small fitting errors and use them only for optimization; the others are discarded by means of adaptive *underweighting* of error residuals (iteratively reweighted LS, IRLS) [1], [10], [14]. Neuron nonlinearities and the backpropagation formula [5], in fact, play together the role of the *influence functions* used in robust LS fitting. The derivative of a sigmoidal-type activation function is bell-shaped around the bias term and underweights any backpropagated error that exceeds the range of the nonlinearity. Outputs belonging to different but statistically separable distributions produce significant backpropagated errors (and weight changes) only in those neurons that are involved in the determination of the decision boundaries.

IV. Experimental Results

In this section, we describe the results obtained by applying the proposed approach to the equalization of typical linear and nonlinear channels. In particular, we considered some test channels described in [26], where a gradient-based approach—real time recurrent learning (RTRL)—was used. Simulations have been set up so that results can be compared directly with those reported in [26]. We will consider for simplicity the case of 2-PAM signals where symbols are randomly extracted from the alphabet $\{-1, 1\}$; extension to higher signal constellations is possible by using the complex model for the neuron [16]. In all the experiments, a network with three recurrent units and two outputs and the values $\eta = 20$ and $\lambda = 0.999$ of the learning parameters were used.

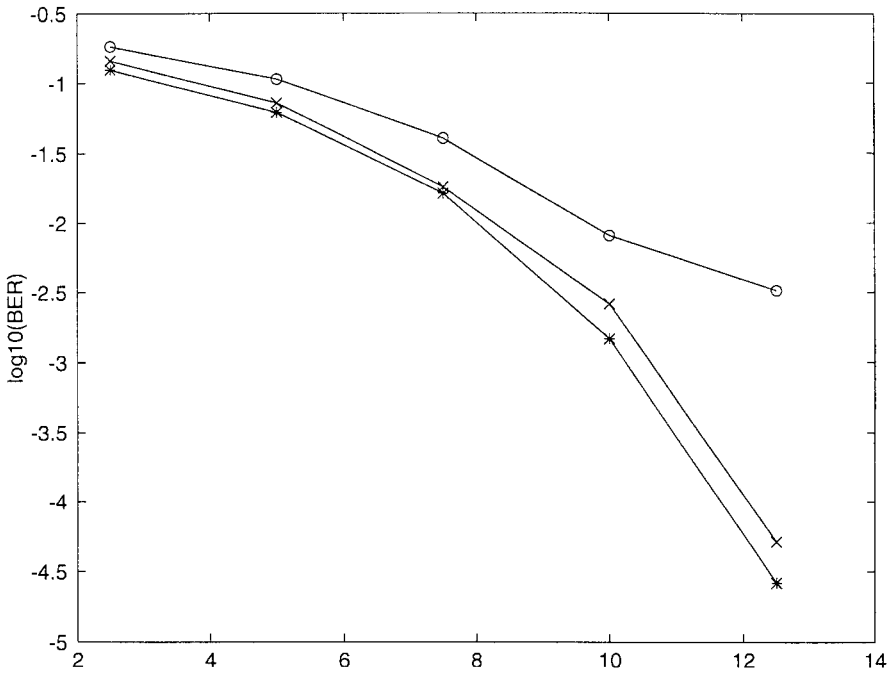

Fig. 3. Channel $H_1(z) = 1 + 0.7z^{-1}$. Plot of the decimal logarithm of the BER versus the SNR. o: after 20 samples in the learning phase. x: after 50 samples. $*$: after 100 samples.

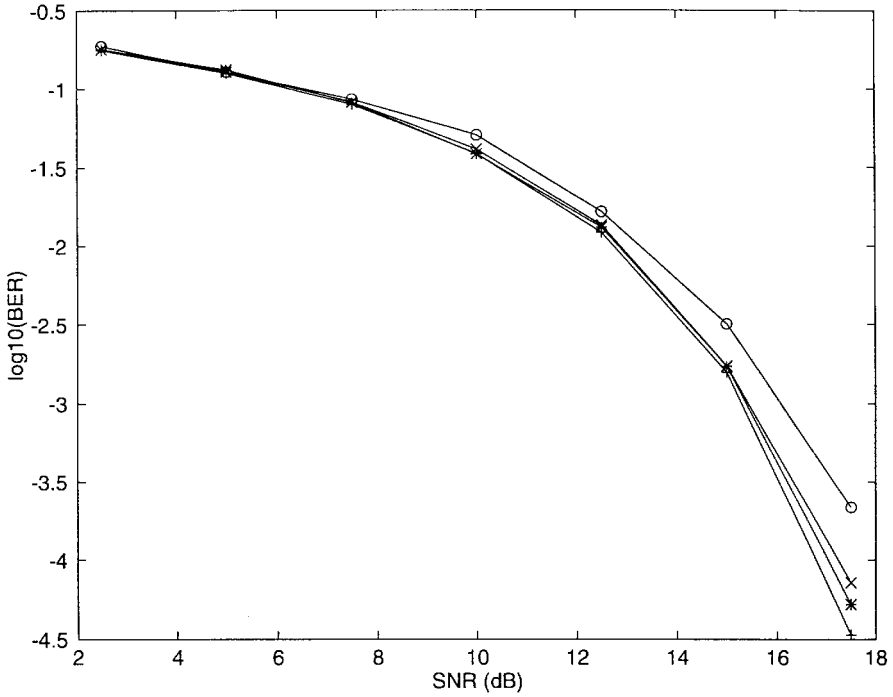

Fig. 4. Channel $H_2(z) = 0.3482 + 0.8704z^{-1} + 0.3482z^{-2}$. Plot of the decimal logarithm of the BER versus the SNR. o: after 100 samples in the learning phase. x: after 200 samples. $*$: after 300 samples. $+$: after 500 samples.

Experiment 1: In the first test, we considered a channel with transfer function $H_1(z) = 1 + 0.7z^{-1}$; this is a simple minimum-phase channel that can be used as a preliminary test. The networks were trained with sequences of 20, 50 and 100 symbols at different noise levels; the BER for each value of signal-to-noise ratio (SNR) was evaluated on 10^5 more received symbols and averaged over 20 realizations. Fig. 3 shows the curves of the average BER versus the SNR while varying the length of the training phase. These curves can be directly compared with those presented in [26], where 2000 symbols were used for learning; it can be seen that the new approach is able to provide the same performance with only 100 samples. No cases of ill-convergence were observed.

Experiment 2: The second example is a linear nonminimum-phase channel with transfer function $H_2(z) = 0.3482 + 0.8704z^{-1} + 0.3482z^{-2}$; as pointed

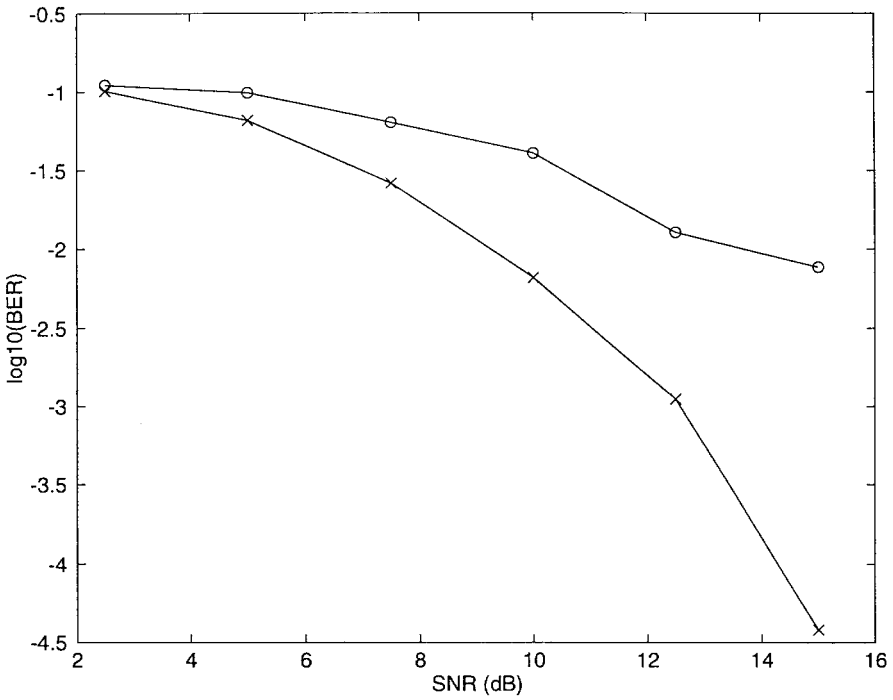

Fig. 5. Channel $H_3(z) = 1 - 2z^{-1} + z^{-2}$. Plot of the decimal logarithm of the BER versus the SNR. o: after 50 samples in the learning phase. x: after 100 samples.

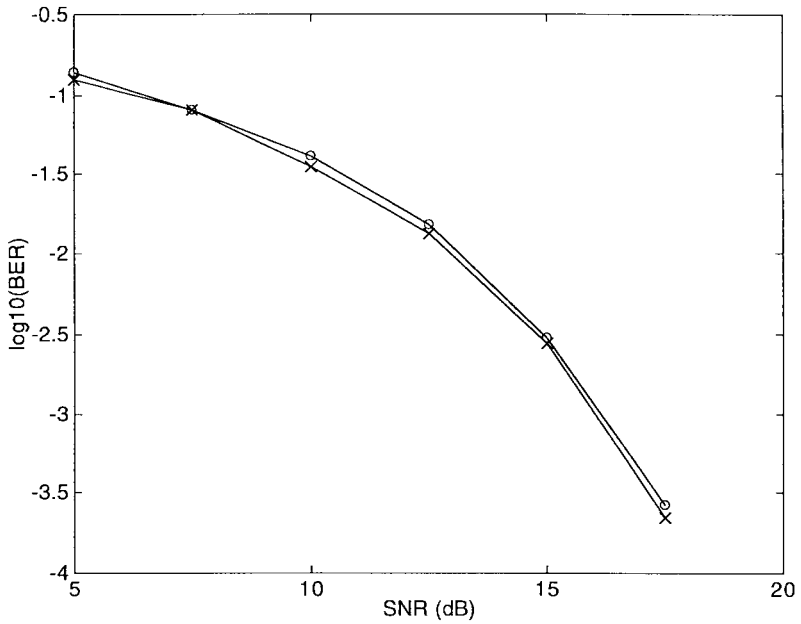

Fig. 6. Nonlinear channel. Plot of the decimal logarithm of the BER versus the SNR. o: after 500 samples in the learning phase. x: after 1000 samples.

out in [26], this type of channel is closer to those encountered in real communication systems. Learning was performed on sequences of 100, 200, 300, and 500 samples. Fig. 4 depicts the curves of the mean BER versus the SNR. With respect to the gradient approach of [26], the proposed method is able to substantially reduce the number of training samples required to get the same BER. In addition, in this case, convergence was reached in all trials, demonstrating the robustness of the proposed method.

Experiment 3: As a third test, we considered the partial response channel described by $H_3(z) = 1 - 2z^{-1} + z^{-2}$. This channel has a double zero on the unit circle; the problem is badly ill conditioned due to the minimax property of the eigenvalues of the correlation matrix of received signals and the small energy available at frequencies near the nulls of the transfer function [17]. It is well known that gradient-based methods have in this case serious problems of convergence [33]. The robustness of the proposed approach in this situation is instead confirmed by Fig. 5, showing the average BER curves obtained after 50 and 100 samples in the training phase. With respect to the curves depicted in [26], the new method is able to get superior performance in terms of both the number

of samples necessary and the minimum SNR to get a prefixed BER (12.5 dB against 20 dB to obtain BER $= 10^{-3}$). The performance improvement may be explained by the higher speed of convergence of second-order methods and the good numerical conditioning of the proposed approach [6] [33].

Experiment 4: The last experiment deals with a typical nonlinear channel (see [26]), The network input $u(k)$ is described by $u(k) = \hat{u}(k) + 0.2\hat{u}^2(k) + n(k)$, where $\hat{u}(k)$ is the output of the linear channel $H_2(z) = 0.3482 + 0.8704z^{-1} + 0.3482z^{-2}$. In addition, in this case, the BRLS algorithm requires a lower number of iterations (500 versus 2000) to get about the same performance of RTRL in terms of BER (Fig. 6).

Some performance issues should be addressed in detail, namely, the practical requirements for fast learning, the computational cost per training and the regularity of the algorithm dependence graph. Most equalizers work in real time in the presence of nonstationary channels (mobile radio, cellular telephone, ...). Efficiency in data transmission also requires that a small percentage of symbols is used as a preamble for training; therefore, high rate of convergence is a prerequisite for the functionality of the equalizer. The proposed architecture exhibits a convergence speed of the same order as traditional linear equalizers so that message structure and communication efficiency can be both preserved even in demanding packet and mobile radio applications. The computational cost per iteration of a Newton-type algorithm is higher than that of gradient-based approaches. Nevertheless, as shown in [33], since the total number of iterations necessary for the convergence is much lower, the overall cost for the training is reduced. In addition, the RTRL algorithm is known to be relatively expensive [31]; in fact, given a fully connected neuron layer having N inputs and M outputs, at each time step, the forward propagation, the gradient computation, the weight update, and *each relaxation step* require all $O(MN)$ operations. The discriminative LS algorithm computes instead the new weights by QR updating and back substitution, each having $O(MN^2)$ complexity [7]. We remark that RTRL requires $O(N)$ average iterations for each weight computation; therefore, its order of complexity reaches just that of the LS approach, without giving the benefits of super linear convergence rate [33]!

A disadvantage of Newton-type algorithms is the peak computing power required to the processor during learning. However, the dependence graph of the proposed algorithm, which is based on QR decomposition and back substitution, is amenable to regular implementation on parallel array processors [17]. In addition, the demodulation task after learning is simply the forward propagation pass through the network and does not require dedicated hardware. It can be also noted that all experimental tests were performed with a fixed learning rate, thus enabling an effective parallel processing.

V. Conclusion

This paper has introduced a new approach to adaptive digital channel equalization that makes use of recurrent neural networks. Previous approaches were based on the minimization of the mean square error (MSE) performed by a gradient descent procedure. The MSE is not necessarily related with the classification error—bit error rate (BER)—that is considered in equalization problems; moreover, the use of gradient-based learning techniques is often hampered by slow speed of convergence and numerical ill conditioning. Overcoming these difficulties, the proposed method minimizes an error functional that is a direct measure of the BER. Moreover, the determination of the optimal weights is performed by a procedure (*gradient descent in neuron space*) that is faster and numerically more stable and robust with respect to traditional gradient-based schemes. Experimental tests conducted on 2-PAM signals for different channels have confirmed the better performance of the novel algorithm.

References

[1] P. J. Huber, *Robust Statistics.* New York: Wiley, 1981.
[2] J. G. Proakis, *Digital Communications.* New York: McGraw-Hill, 1983.
[3] A. Papoulis, *Probability, Random Variables, and Stochastic Processes,* 2nd ed. New York: McGraw-Hill, 1984.
[4] S. U. H. Qureshi, "Adaptive equalization," *Proc. IEEE,* vol. 73, Sept. 1985.
[5] D. E. Rumelhart, G. E. Hinton, and R. G. Williams, "Learning internal representations by error propagation," in *Parallel Distributed Processing: Exploration in the Microstructure of Cognition,* D. E. Rumelhart and J. L. McLelland, Eds. Cambridge, MA: MIT Press/Bradford, 1986, vol. 1.
[6] D. G. Luenberger, *Linear and Nonlinear Programming.* 2nd ed. Reading, MA: Addison-Wesley, 1989.
[7] G. H. Golub and C. F. Van Loan, *Matrix Computations.* Baltimore, MD: Johns Hopkins Univ. Press, 1989.
[8] R. J. Williams and D. Zipser, "A learning algorithm for continually running fully recurrent neural networks," *Neural Comput.*, vol. 1, pp. 270–280, 1989.
[9] G. Cybenko, "Approximations by superpositions of a sigmoidal function," *Math. Contr. Signals Syst.*, vol. 2, no. 4, 1989.
[10] L. P. Ammann, "Statistically robust signal subspace identification," in *Proc. Int. Conf. Acoust., Speech, Signal Process., ICASSP*, 1991, pp. 2711–2714.
[11] R. J. Williams and J. Peng, "An efficient gradient-based algorithm for on-line training of recurrent network trajectories," *Neural Comput.*, vol. 1, pp. 270–280.
[12] J. L. Elman, "Finding structure in time," *Cognitive Sci.*, vol. 14, pp. 179–211, 1990.
[13] S. Chen, G. J. Gibson, C. F. N. Cowan, and P. M. Grant, "Adaptive equalization of finite nonlinear channels using multilayer perceptrons," *Signal Process.*, vol. 20, pp. 107–119, 1990.
[14] E. D. Di Claudio, G. Orlandi, F. Piazza, and A. Uncini, "Optimal weighted LS AR estimation in presence of impulsive noise," in *Proc. Int. Conf. Acoust., Speech, Signal Process., ICASSP*, May 1991, pp. 3149–3152.
[15] G. J. Gibson, S. Siu, and C. F. N. Cowan, "The application of nonlinear structures to the reconstruction of binary signals," *IEEE Trans. Signal Processing*, vol. 39, Aug. 1991.
[16] H. Leung and S. Haykin, "The complex backpropagation algorithm," *IEEE Trans. Signal Processing,* vol. 39, no. 9, Sept. 1991.
[17] S. Haykin, *Adaptive Filter Theory,* 2nd ed. Englewood Cliffs, NJ: Prentice-Hall, 1991.
[18] Q. Zhang and A. Benveniste, "Wavelet networks," *IEEE Trans. Neural Networks*, vol. 3, Nov. 1992.
[19] B. H. Juang and S. Katagiri, "Discriminative learning for minimum error classification," *IEEE Trans. Signal Processing,* vol. 40, Dec. 1992.
[20] W. R. Kirkland and D. P. Taylor, "On the application of feedforward neural networks to channel equalization," in *Proc. IJCNN Int. Joint Conf. Neural Networks*, New York, 1992.
[21] M. Peng, C. L. Nikias, and J. G. Proakis, "Adaptive equalization with neural networks: New multilayer perceptron structures and their evaluation," in *Proc. ICASSP'92 IEEE Int. Conf. Acoust., Speech Signal Processing*, New York, 1992.
[22] S. Chen, B. Mulgrew, and S. McLaughlin, "Adaptive bayesian equalizer with decision feedback," *IEEE Trans. Signal Processing,* vol. 41, Sept. 1993.

[23] S. Saarinen, R. Bramley, and G. Cybenko, "Ill-conditioning in neural network training problems," *SIAM J. Sci. Comput.*, vol. 14, no. 3, pp. 693–714, May 1993.
[24] E. A. Lee and D. G. Messerschmitt, *Digital Communication.* Boston, MA: Kluwer, 1994.
[25] S. Haykin, *Neural Networks—A Comprehensive Foundation*. New York: IEEE, 1994.
[26] G. Kechriotis, E. Zervas, and E. S. Manolakos, "Using recurrent neural networks for adaptive communication channel equalization," *IEEE Trans. Neural Networks*, vol. 5, Mar. 1994.
[27] P. R. Chang, B. F. Yeh, and C. C. Chang, "Adaptive packet equalization for indoor radio channel using multilayer neural networks," *IEEE Trans. Veh. Technol.*, vol. 43, Aug. 1994.
[28] P. R. Chang and B. F. Yeh, "Nonlinear communication channel equalization using wavelet neural networks," *Proc. IEEE Int. Conf. Neural Networks*, New York, 1994.
[29] K. A. Al-Mashouq and I. S. Reed, "The use of neural nets to combine equalization with decoding for severe intersymbol interference channels," *IEEE Trans. Neural Networks*, vol. 5, Nov. 1994.
[30] P. R. Chang and B. C. Wang, "Adaptive decision feedback equalization for digital satellite channels using multilayer neural networks," *IEEE J. Select. Areas Commun.*, vol. 13, Feb. 1995.
[31] B. A. Pearlmutter, "Gradient calculations for dynamic recurrent neural networks: a survey," *IEEE Trans. Neural Networks*, vol. 6, Sept. 1994.
[32] R. Parisi, E. D. Di Claudio, A. Rapagnetta, and G. Orlandi, "Recursive least squares approach to learning in recurrent neural networks," in *Proc. Int. Conf. Neural Networks ICNN*, Washington, D.C., June 3–6, 1996.
[33] R. Parisi, E. D. Di Claudio, G. Orlandi, and B. D. Rao, "A generalized learning paradigm exploiting the structure of neural networks," *IEEE Trans. Neural Networks*, vol. 7, Nov. 1996.

Raffaele Parisi received the "Laurea" in electrical engineering with honors in 1991 and the Ph.D. degree in 1995, from the University of Rome "La Sapienza," Rome, Italy.

In 1994, he was a visiting student in the Department of Electrical and Computer Engineering, University of California, San Diego. He is currently with the University of Rome "La Sapienza," where he is a Researcher in the Department of Information and Communication. His research interests are in the fields of neural networks, optimization theory, and array processing.

Elio D. Di Claudio received the degree with honors in electrical engineering from the University of Ancona, Italy, in 1986.

He is currently a researcher at the INFOCOM Department, University of Rome, "La Sapienza," Rome, Italy. From 1986 to 1990, he was with Telettra S.p.A. Company, Chieti, Italy, where he worked on spread-spectrum telecommunication equipment and digital signal processing algorithms. From 1990 to 1991, he was with Elasis S.c.p.A. Company. His current interests are in the fields of parallel algorithms for signal processing, parallel architectures for VLSI, spectral estimation, neural networks, and array processing.

Gianni Orlandi received the degree in electrical engineering from the University of Rome "La Sapienza," Rome, Italy, in 1972.

Since 1978, he has been with the University of Rome "La Sapienza," first as an Assistant Professor and, from 1983, as an Associate Professor of Electrical Engineering. From 1986 to 1989, he was Full Professor with the Department of Electronics and Automation, University of Ancona, Italy, and since 1989, he has been Full Professor of Electrical Engineering with the Department of Information and Communication, University of Rome "La Sapienza." His research interests are in the areas of circuit theory, spectral estimation, array processing, parallel algorithms, VLSI parallel architectures and neural networks.

Prof. Orlandi is President of the Italian Neural Networks Society (SIREN).

Bhaskar D. Rao received the B. Tech. degree in electronics and electrical communication engineering from the Indian Institute of Technology, Kharagpur, in 1979 and the M.S. and Ph.D. degrees from the University of Southern California, Los Angeles, in 1981 and 1983, respectively.

Since 1983, he has been with the University of California, San Diego, where he is currently a Professor in the Department of Electrical and Computer Engineering. His interests are in the areas of digital signal processing, estimation theory, and optimization theory, with applications to communications, biomedical imaging, and speech.