CHANNEL EQUALIZATION BY FEEDFORWARD NEURAL NETWORKS

Biao Lu and Brian L. Evans

Embedded Signal Processing Laboratory
Dept. of Electrical and Computer Engineering
The University of Texas at Austin, Austin, TX 78712-1084 USA

{blu,bevans}@ece.utexas.edu

ABSTRACT

A signal suffers from nonlinear, linear, and additive distortion when transmitted through a channel. Linear equalizers are commonly used in receivers to compensate for linear channel distortion. As an alternative, nonlinear equalizers have the potential to compensate for all three sources of channel distortion. Previous authors have shown that nonlinear feedforward equalizers based on either multilayer perceptron (MLP) or radial basis function (RBF) neural networks can outperform linear equalizers. In this paper, we compare the performance of MLP vs. RBF equalizers in terms of symbol error rate vs. SNR. We design a reduced complexity neural network equalizer by cascading an MLP and a RBF network. In simulation, the new MLP-RBF equalizer outperforms MLP equalizers and RBF equalizers.

1. INTRODUCTION

A transmitted signal suffers from nonlinear, linear, and additive distortion when passing through a channel. The nonlinear and linear distortion cause adjacent pulses to interfere with each other, which is known as intersymbol interference (ISI). At the receiver, the equalizer would compensate for one or more of the distortion effects in the channel. Equalization may either require a training signal or be blind. In digital communications, the training signal is simply a known sequence of symbols sent by transmitter so that the receiver can estimate the channel distortion.

Linear equalizers that employ training sequences are often based on adaptive finite impulse response (FIR) filters. They are easy to implement and track linear distortion in the channel fairly well provided that enough taps are used (using 50–100 taps is common). Some linear equalizers, such as a zero-forcing equalizer, may amplify channel noise [1]. As an alternative, nonlinear equalizers have the potential to compensate for all three sources of channel distortion. A common nonlinear equalizer is the decision-feedback equalizer [2].

Another class of nonlinear equalizers is based on artificial neural networks, e.g. multilayer perceptrons (MLP) and radial basis functions (RBF) feedforward neural networks. Section 2 describes MLP equalizers [3, 4, 5, 6, 7, 8] and RBF equalizers [9, 10]. MLP equalizers [3] and RBF equalizers [9] outperform linear feedforward equalizers in symbol error rate vs. SNR, but at the cost of significantly higher computational complexity. Section 3 describes a new lower complexity neural network equalizer formed by cascading an MLP network and an RBF network to decrease the number of hidden neurons for the same level of performance. Section 4 compares the symbol error rate vs. SNR performance of MLP, RBF, and MLP-RBF equalizers using different channel characteristics, number of input neurons, and number of hidden neurons. It also compares MLP and RBF equalizers vs. an optimal linear equalizer and a zero-forcing equalizer. Section 5 concludes the paper. The key contributions of this paper are

1. a reduced complexity MLP-RBF neural network equalizer (Section 3), and
2. a comparison of the performance of MLP equalizers vs. RBF equalizers (Section 4).

2. FEEDFORWARD NEURAL NETWORKS

Fig. 1 shows the block diagram of feedforward neural networks. Each node is the basic element of a neural network called a neuron. The output, y, of a neuron is given by

$$y = f\left(\sum_{i=1}^{N} w_i x_i\right)$$

where x_i is the ith input to a neuron, w_i is the weight associated with the ith input, and f is the activation function.

2.1. Multilayer Perceptrons

Possible activation functions for multilayer perceptrons are

- linear: $f(v) = k\,v$
- sigmoid: $f(v) = \dfrac{1}{1+e^{-v}}$
- hyperbolic tangent: $f(v) = \tanh(v) = \dfrac{1-e^{-v}}{1+e^{-v}}$

An MLP may have more than one hidden layer. The neurons in the hidden layer may use either sigmoid or hyperbolic tangent activation functions. The activation function for the output layer may be any one of the above.

This research was supported by the US Defense Advanced Research Projects Agency under grant DAAB07-97-C-J007, and a US National Science Foundation CAREER Award under grant MIP-9702707.

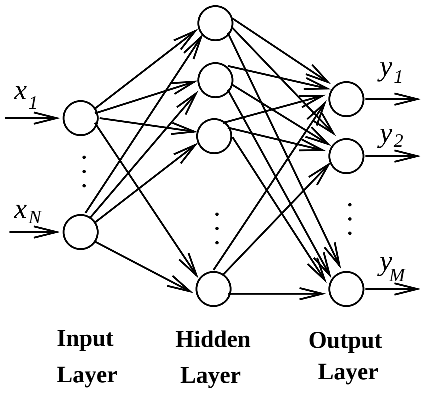


Figure 1: A feedforward neural network. Each circle represents a neuron which sums the inputs and passes the sum through an activation function. Each arc represents multiplication by a scalar weight.

Many different training algorithms exist for MLPs. Although batch backpropagation [11] is widely used, it suffers from slow convergence and can be trapped in local minima. Modified backpropagation algorithms attempt to overcome both drawbacks [11]. To increase the convergence rate, the Levenberg-Marquardt (LM) algorithm and second-order optimization techniques, such as the conjugate gradient method, scaled conjugate gradient method, and the quasi-Newton method, can be used [11]. Hybrid linear-nonlinear training [12], natural gradient learning [13], and simulated annealing [14] escape from local minima during training. All of these training algorithms randomly initialize the weights.

2.2. Radial Basis Function Networks

Since there is no guarantee that an MLP would converge to a global minimum, radial basis function (RBF) networks are a key alternative. RBFs have only three layers (one input, one hidden, and one output). The kth output is given by

$$y_k = \sum_{i=1}^{N_h} w_{k,i}\ \phi_i(\mathbf{x})$$

where N_h is the number of neurons in the hidden layer and $\phi_i(\mathbf{x})$ is a radially symmetric scalar function with N_h centers of the radial basis function. A commonly used radial basis function $\phi_i(\cdot)$ is a Gaussian function

$$\phi_i(\mathbf{x}) = \exp\left(-\frac{\| \mathbf{x} - \mathbf{c}_i \|^2}{2\sigma_i^2}\right)$$

where $\| \cdot \|$ is a norm (usually Euclidean). A radial basis function is local in character— its response to the input $\mathbf{x}$ drops off quickly for input values that are away from the center of the activation function's receptive field, $\mathbf{c}_i$.

Training takes two steps [11]. First, the σ_i's and $\mathbf{c}_i$'s are calculated. We calculate them by using expectation maximization (EM). Second, the weights between the hidden and output layers are determined. We calculate them by using least mean squares (LMS).

3. A NEW MLP-RBF EQUALIZER

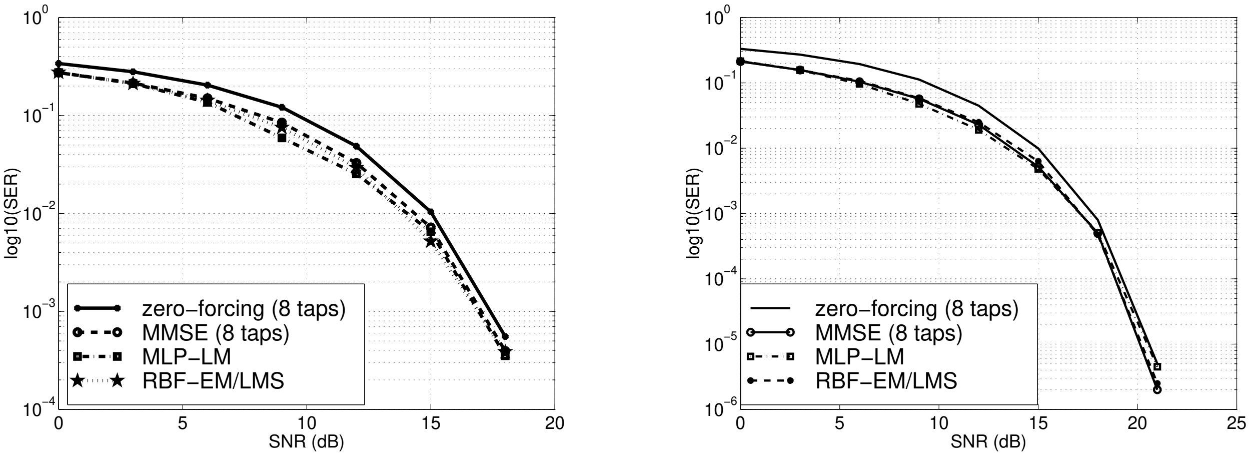
Among feedforward equalizers, RBF [9, 10] and MLP [3, 4, 5, 6] equalizers outperform linear equalizers in symbol error rate vs. SNR. RBF equalizers estimate the probability density function of the incoming signal to approximate the optimal Bayesian equalizer [15]. MLP equalizers can approximate a Bayesian discriminant function. The size and structure of the MLP limit network the approximation accuracy [16]. MLP and RBF equalizers are insensitive to the channel phase response, as demonstrated in Section 4.

MLP training algorithms are either fast but get trapped in local minima (such as the Levenberg-Marquardt algorithm) or slow but converge to a global minimum (such as simulated annealing). Here, "slow" can be several orders of magnitude slower than "fast." When using a "fast" algorithm, an MLP equalizer is trained several times and the best network is chosen. For MLP equalizers, we suspect that the number of hidden neurons is a polynomial function of the length of FIR model of the channel.

In RBF equalizers, the number of hidden neurons increases exponentially with the length of FIR model of the channel [11]. To obtain a similar symbol error rate vs. SNR performance as MLP equalizers, RBF equalizers must use an increasingly larger number of hidden nodes than the MLP equalizer as SNR decreases. In order to reduce the number of neurons in an RBF equalizer, a modified k-mean algorithm [10] or a self-organizing map [17] can be used in the first step of RBF training to compute σ_i's and $\mathbf{c}_i$'s.

When the SNR is low, the transmitted data has been scattered by the addition of strong noise. The data far away from each cluster is considered *irrelevant*. MLP reduces irrelevant data because the hidden layer first calculates the weighted sum of the inputs. The linear combinations of the inputs confine the networks' attention to the linear subspace spanned by the weight vectors. When the data is scattered due to low SNR, the number of hidden units in MLP is not required to increase. In RBF networks, the irrelevant data significantly degrades performance because activation functions have local receptive fields. The local receptive fields can be adapted to a local pattern in the data. RBF networks do not suffer unwanted side effects in other regions, whereas MLP can cause spurious side effects in other parts of the input space while it tries to fit a local pattern.

To improve performance and reduce complexity of MLP and RBF equalizers, we cascade an MLP network and RBF network. We set the number of MLP input neurons, MLP output neurons, and RBF input neurons equal to the number of taps M in the FIR channel model. The inputs to the MLP-RBF equalizer are the current received sample and the previous $M-1$ samples. The MLP stage suppresses the irrelevant data (noise) and outputs cleaned values of the current received sample and previous $M-1$ samples. The RBF stage takes these MLP outputs, performs a best-fit, and outputs the symbol decision on its single output. Using the training sequence, we first train the MLP network using LM, then feed the trained MLP output into the RBF network, and finally train the RBF network using EM/LMS (see Section 2.2). The MLP-RBF equalizer requires far fewer neurons for the same symbol error rate vs. SNR performance, as demonstrated next.



(a) Minimum phase channel
MLP-LM has 6 inputs, 8 hidden units, 1 output.
RBF-EM/LMS has 6 inputs, 20 hidden units, 1 output.

(b) Linear phase channel
MLP-LM has 6 inputs, 16 hidden units, 1 output.
RBF-EM/LMS has 3 inputs, 40 hidden units, 1 output.

Figure 2: Performance analysis of four equalizers. Zero-forcing and minimum mean square error (MMSE) equalizers know the channel coefficients, whereas the neural network equalizers do not. The MMSE equalizer is the optimal linear equalizer.

4. SIMULATION RESULTS

We compare symbol error rate vs. SNR of the MLP and RBF equalizers for the following settings: 6 training algorithms for MLP, 2 channel responses, 3–15 neural network input nodes for MLP and 3–8 inputs for RBF, and different numbers of hidden units. The channel responses

$$H_{min}(z) = 0.6963 + 0.6964z^{-1} + 0.1741z^{-2}$$
$$H_{linear}(z) = 0.3482 + 0.8704z^{-1} + 0.3482z^{-2}$$

are minimum and linear phase, respectively, and have the same magnitude response. The transmitted 2-PAM signals are chosen from $\{-1, 1\}$ with equal probability. They are independent and identically distributed. We model additive distortion in the channel as white Gaussian noise and vary the SNR from 0 dB to 30 dB SNR in increments of 3 dB. The training sequence is 1000 symbols (one symbol/bit). After training, we test the equalizer using 2×10^6 symbols.

Because the probability that MLP networks fall into local minima during training increases with the number of hidden layers, we use only one hidden layer [18]. For activation functions in the MLP equalizer, the hidden layer uses the hyperbolic tangent function and the output layers uses an identity function (linear function with $k = 1$). The RBF equalizer uses a Gaussian radial basis function. The neural network inputs are delayed versions of the received signal. The number of inputs is 3–15 for MLP and 3–8 for RBF. A minimum of 3 inputs is used because each channel has 3 taps. The maximum number of inputs is chosen so that training would complete in a reasonable amount of time and diminishing performance returns are observed. We use 1, 2, 4, 8, and 16 hidden units for MLP equalizers, and 10, 20, and 40 hidden units for RBF equalizers.

Table 1 compares six training algorithms [11] for the MLP equalizer: conjugate gradient, scaled conjugate gradient, quasi-Newton, Levenberg-Marquardt (LM), hybrid linear-nonlinear, and batch backpropagation. The LM algorithm gives the best symbol error rate vs. SNR performance and requires the lowest computational complexity. Table 1 shows the results of using the EM/LMS method [11] for the RBF equalizer. Throughout the rest of the paper, we use MLP-LM and RBF-EM/LMS equalizers.

Fig. 2 compares the simulation performance of the nonlinear MLP and RBF equalizers and the linear zero-forcing and minimum mean square error (MMSE) equalizers. In the simulation, the two linear equalizers know the channel coefficients, whereas the nonlinear equalizers do not. The MLP and RBF equalizers obtain almost the same performance as the MMSE equalizer even though MLP and RBF equalizers have no knowledge on the coefficients of channels.

Fig. 3 compares the simulation performance of MLP, RBF, and MLP-RBF equalizers. For the MLP-RBF equalizer, we set the number of inputs to be the length of the FIR channel model. We use an MLP with 3 inputs, 4 hidden units, and 3 outputs, and an RBF network with 3 inputs, 4 hidden units, and 1 output. The MLP-RBF equalizer outperforms both MLP and RBF equalizers. Since the RBF equalizer in Fig. 3 has the same structure as that in Fig. 2(b), this MLP-RBF equalizer also outperforms linear feedforward equalizers. In Matlab 5, the training time was 3.59 s for the MLP (3-4-1), 170.20 s for the RBF (3-40-1), and 14.95 s for the MLP (3-4-3)–RBF (3-4-1) equalizers. We ran the simulations on a 167 MHz Ultra-2 workstation.

5. CONCLUSION

We compare the performance of two neural network equalizers (MLP and RBF) and two "best-case" linear equalizers (zero-forcing and MMSE). The linear equalizers have precise knowledge of the channel coefficients, which are unknown to the neural network equalizers. The MMSE equalizer is the optimal linear equalizer in the least squares sense. The order of symbol error rate vs. SNR performance from best to worst is MLP, RBF, MMSE, and zero-forcing, ac-

Method	Training		Testing
	CPU Time	GFLOPS	SER
Batch back-propagation	2608.0 s	31.14	0.0084
Levenberg-Marquardt (LM)	65.3 s	2.50	0.0083
Conjugate Gradient	2213.0 s	39.20	0.0114
Scaled Conjugate Grad.	14280.8 s	8.38	0.0120
Quasi-Newton	9770.0 s	27.20	0.0121
Hybrid linear/ nonlinear	More than a day	n/a	0.0085
RBF-EM/LMS (5-20-1)	21.1 s	0.16	0.0074
MLP (3-4-3): RBF (3-4-1)	24.4 s	0.47	0.0033

Table 1: Training time and symbol error rate (SER) for MLP, RBF, and MLP-RBF equalizers for the minimum phase channel $H_{min}(z) = 0.6963 + 0.6964z^{-1} + 0.1741z^{-2}$. The MLP equalizer has 8-6-1 input-hidden-output units.

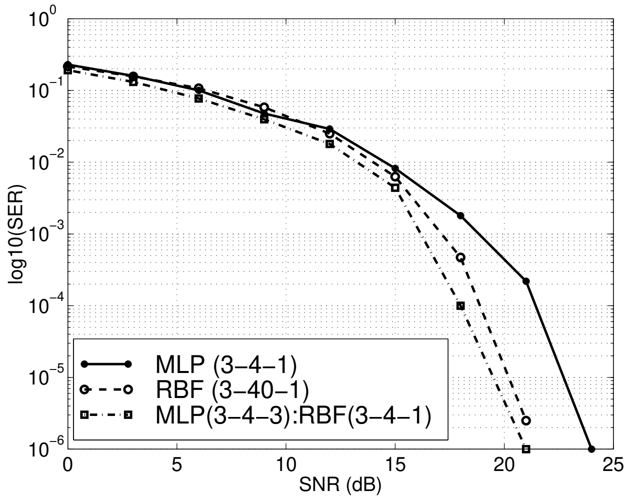


Figure 3: Comparison of MLP, RBF, and MLP-RBF equalizers for the linear phase channel $H_{linear}(z) = 0.3482 + 0.8704z^{-1} + 0.3482z^{-2}$. The MLP-RBF equalizer outperforms MLP and RBF equalizers yet has lower complexity.

cording to Fig. 2. For some SNR values, the RBF equalizer outperforms the MLP equalizer in SER in Fig. 3. The MLP network has to be trained several times, whereas the other equalizers are trained in one pass over the training data.

We have designed a new reduced complexity neural network equalizer by cascading an MLP and an RBF network. In the MLP-RBF equalizer, the MLP network suppresses noise and the RBF network performs the equalization. Our new MLP-RBF equalizer outperforms the MLP and RBF equalizers in terms of symbol error rate vs. SNR.

6. REFERENCES

[1] S. U. H. Qureshi, "Adaptive equalization," *Proc. IEEE*, vol. 73, pp. 1349–1387, Sept. 1985.

[2] C. A. Belfiore and J. J. H. Park, "Decision feedback equalization," *Proc. IEEE*, vol. 67, pp. 1143–1156, Aug. 1979.

[3] G. J. Gibson, S. Siu, and C. F. N. Cowan, "Multilayer perceptron structures applied to adaptive equalizer for data communications," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Proc.*, vol. II, (Glasgow, Scotland), pp. 1183–1186, May 1989.

[4] G. J. Gibson, S. Siu, and C. F. N. Cowan, "The application of nonlinear structures to the reconstruction of binary signals," *IEEE Trans. Signal Proc.*, vol. 39, pp. 1877–1884, Aug. 1991.

[5] S. Chen, G. J. Gibson, C. F. N. Cowan, and P. M. Grant, "Adaptive equalization of finite non-linear channels using multilayer perceptrons," *Signal Processing*, vol. 20, pp. 107–119, 1990.

[6] M. Peng, C. L. Nikias, and J. Proakis, "Adaptive equalization with neural networks: new multilayer perceptron structure and their evaluation," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Proc.*, vol. II, (San Francisco, CA), pp. 301–304, Mar. 1992.

[7] M. Peng, C. L. Nikias, and J. G. Proakis, "Adaptive equalization for PAM and QAM signals with neural networks," in *Proc. IEEE Asilomar Conf. on Signals, Systems, Comp.*, pp. 496–500, Nov. 1991.

[8] M. Peng, H. Lev-Ari, C. L. Nikias, and J. G. Proakis, "Performance improvement of neural network equalizers," in *Proc. IEEE Asilomar Conf. on Signals, Systems, Comp.*, pp. 396–400, Nov. 1993.

[9] S. Chen, G. J. Gibson, C. F. N. Cowan, and P. M. Grant, "Reconstruction of binary signals using an adaptive radial-basis-function equalizer," *Signal Processing*, vol. 22, pp. 77–93, 1991.

[10] J. S. Lee, C. D. Beach, and N. Tepedelenlioglu, "Channel equalization using radial basis function network," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Proc.*, vol. III, (Atlanta, GA), pp. 1719–1722, May 1996.

[11] C. M. Bishop, *Neural Networks for Pattern classification.* New York: Oxford University Press, 1995.

[12] S. McLoone, M. D. Brown, G. Irwin, and G. Lightbody, "A hybrid linear/nonlinear training algorithm for feedforward neural networks," *IEEE Trans. Neural Networks*, vol. 9, pp. 669–684, July 1997.

[13] S. Amari, "Natural gradient works efficiently in learning," *Neural Computation*, vol. 10, pp. 251–276, 1998.

[14] B. J. Sheu and D. C. Chen, "1-D compact neural networks for wireless communication and mobile computing," in *Proc. IEEE Int. Symp. Circ. Syst.*, vol. III, (Atlanta, GA), pp. 551–554, May 1996.

[15] B. Mulgrew, "Applying radial basis functions," *IEEE Signal Proc. Magazine*, vol. 13, pp. 50–65, Mar. 1996.

[16] D. W. Ruck, S. K. Rogers, M. Kabrisky, M. E. Oxley, and B. W. Suter, "The multilayer perceptron as an approximation to a Bayes optimal discriminant function," *IEEE Trans. Neural Networks*, vol. 1, pp. 296–298, Dec 1990.

[17] S. Bouchired, M. Ibnkahla, D. Roviras, and F. Castanie, "Equalization of satellite mobile communication channels using combined self-organizing maps and RBF networks," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Proc.*, vol. VI, (Seattle, WA), pp. 3377–3379, May 1998.

[18] J. de Villiers and E. Barnard, "Backpropagation neural nets with one and two hidden layers," *IEEE Trans. Neural Networks*, vol. 4, pp. 136–141, Jan 1993.