NONLINEAR CHANNEL EQUALIZATION WITH MAXIMUM COVARIANCE INITIALIZED CASCADE-CORRELATION LEARNING

Arto Kantsila

Institute of Digital and Computer Systems, Tampere University of Technology
P. O. Box 553, 33101 Tampere, Finland (Europe)
phone: +358 3 3115 2140, fax: +358 3 3115 3095, email: arto.kantsila@tut.fi

ABSTRACT

In this paper we have studied maximum covariance initialization scheme and cascade-correlation learning to improve the performance of a multilayer perceptron network equalizer in nonlinear channel environment. The initialization scheme enables faster convergence and the cascade-correlation learning provides adaptive network size. These methods are compared to a traditional MLP network equalizer and to a simple linear equalizer.

1. INTRODUCTION

In digital wireless communications the transmitted signal is subject to various distortions during its propagation through the communication channel. Multipath propagation causes several delayed and differently attenuated copies of the transmitted signal to arrive to the receiver, hence causing intersymbol interference (ISI). Noise is also always present to some extent in real applications. Furthermore, some nonlinear distortions can occur, e.g., in the amplifiers of the transmitter and the receiver. These disturbances often corrupt the transmitted signal to an extent, that without any compensation, the original information cannot be found from the received signal. The compensation for these disturbancies is called equalization. The simplest technique for equalization is a linear equalizer (LE), i.e., a linear filter [1]. However, sometimes nonlinear equalization methods are needed in order to compensate for the channel distortions. Here we have studied multilayer perceptron (MLP) neural networks for nonlinear channel equalization. MLP networks have been studied for equalization purposes for some time and they have been found to perform very well, e.g., [2, 3]. They do have a drawback in the form of high computational complexity, mainly due to extensive training phase, which can restrict their use in practical applications. However, there exists ways to decrease their complexity. In this paper we have applied two techniques that decrease the required amount of computation. These are the maximum covariance (MC) initialization and cascade-correlation (CC) learning.

Conventional MLP networks use a fixed size, and then train the network to solve the problem. This approach can cause unnecessary computation, if the network size is determined to be too large. Furthermore, it may also lead to overfitting [4]. On the other hand, if the network size is set too small, the network may never learn the problem properly. Since the channel response can also be time-varying, it is often very difficult to determine the optimal fixed size for the MLP network equalizer in advance. In cascade-correlation learning method [5] we start with a network with no hidden units, which corresponds to a linear equalizer, and add hidden units one by one to the network if needed. The CC learning method finds a suitable sized network for each channel response. This can decrease the computational load of the system compared to a fixed size network.

The learning can be also improved by using some computationally efficient weight initialization method instead of random weight initialization. Here, we have applied maximum covariance weight initialization scheme [6] to the network. In the MC initialization method, first a large number of candidate hidden units is created by initializing their weights with random values. Then the desired number of hidden units is selected among the candidates applying MC criterion [6]. MC initialization scheme speeds up the convergence of the network and thus less training is needed. Since MC initialization is itself computationally light, it makes possible to further decrease the computational load of an MLP network as we will show.

2. NONLINEAR CHANNEL MODEL

A widely used model for a linear dispersive channel is given by the finite impulse response (FIR) model as follows

$$s(t) = \sum_{i=0}^{L} h(i)a(t-i)$$

where $h(i)$ is the impulse response of the channel and $a(t)$ denotes the true transmitted symbol at time t. The nonlinearities in the channel can be expressed as a polynomial function of the transmitted signal in the following manner

$$r(t) = s(t) + \sum \lambda_k (s(t))^k \quad , k = 2,3,\ldots$$

where λ_k are appropriate parameters. The received signal at the equalizer can be then written as

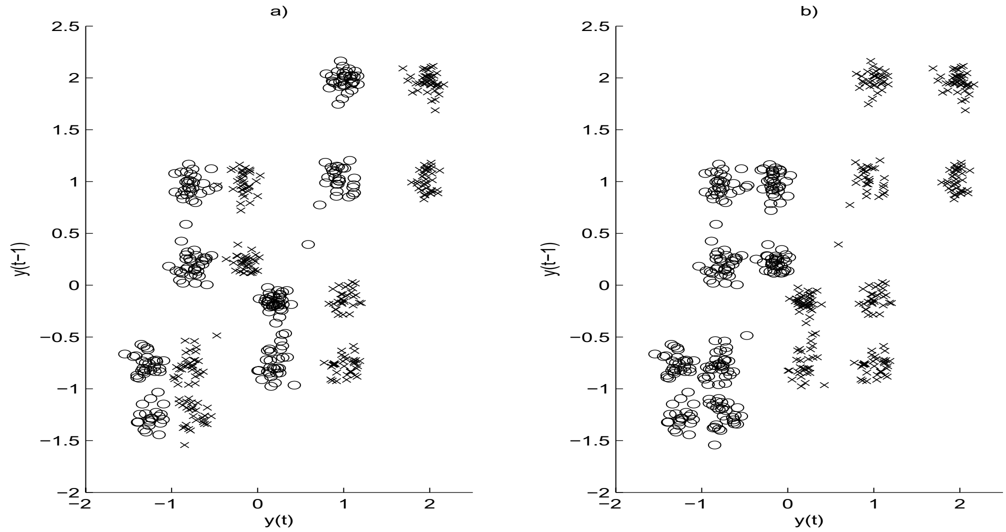

Figure 1. Two-dimensional scatter plot of a channel output with SNR = 20 dB, where ‘x’ denotes transmitted +1 and ‘o’ denotes transmitted -1. In plot a) the delay is 0 and in plot b) the delay is 1.

$$y(t) = r(t) + \eta(t)$$

where η(t) denotes additive noise.

In our simulations, we have studied a channel impulse response H = [0.3482 0.8704 0.3482], which has been widely used in the literature, e.g., [3]. The channel nonlinearities are given by the following equation [7]

$$r(t) = s(t) + 0.15(s(t))^2 + 0.01(s(t))^3 .$$

Equalization can also be seen as a classification problem, which gives the advantage, that we do not need a model for the channel or the interferences. This approach is depicted in Fig. 1. There we have considered a case, where the equalizer has two inputs, the channel output at time t and time t-1 $[y(t)\ y(t\text{-}1)]^T$ and the received sample is marked with ‘x’, if the transmitted bit at time t was +1, and with ‘o’, if the transmitted bit was -1. Equalization can now be seen as a task, where we need to separate the clusters representing transmitted +1 from clusters representing transmitted -1. One can see, that in the first case (Fig. 1a.) the formed clusters are not linearly separable. Therefore, nonlinear equalization methods are needed. By inserting a delay of one symbol period (Fig. 1b.), the formed clusters are closer to being linearly separable. However, nonlinear methods can provide better bit error rates (BER) in this case too, as will be shown.

3. MAXIMUM COVARIANCE INITIALIZED CASCADE-CORRELATION LEARNING

The studied MLP network equalizer had one hidden layer and one output unit. The activation function in the hidden layer was chosen to be hyperbolic tangent (tanh) function, whereas the output unit was set to be linear. The output of the network can now be written as follows

$$\boldsymbol{z(t)} = \boldsymbol{v}_0 + \sum_{\boldsymbol{j}=1}^{\boldsymbol{q}} \boldsymbol{v_j} \tanh(\boldsymbol{w}_{0\boldsymbol{j}} + \sum_{\boldsymbol{i}=1}^{\boldsymbol{p}} \boldsymbol{w_{ij}}\, \boldsymbol{y(t-i}+1))$$

where w_{ij} is the weight between ith input and jth hidden unit and v_j is the weight between jth hidden unit and the output unit.

We have studied three MLP networks. First, a conventional network, with fixed size and random weight initialization. Secondly, an MLP network with fixed size, which uses MC initialization scheme. Finally, the third MLP network applies MC initialization and CC learning combined. All the networks apply RPROP-algorithm for training [8]. The MC initialization scheme can be found in [6]. Here we shall give details of the combined MC-CC learning algorithm, which goes as follows:

1. First there are no hidden units in the network. At this point the only weights that feed the output unit are the bias weight and the network inputs. This corresponds to linear equalization.
2. Optimize the weights feeding the output unit by minimizing the cost function E (sum of squared output error) using pseudo-inverse method of linear regression.
3. If the desired learning performance was reached, quit the cascade-correlation training. Otherwise, proceed to step 4.
4. Create Q candidate hidden units ($Q >> q_m$, where q_m is the maximum number of hidden units) by initializing the weights with random values. Here we have used $Q = 10q_m$ and the uniformly distributed random values of the candidate hidden units were chosen from the interval [-4,...,4]. Do not connect the candidate units to the output unit yet.

5. Compute the covariance for each of the candidate unit from the equation

$$C_{MC,j} = \frac{1}{N}\sum_{t=1}^{N}(o_j(t) - \overline{o}_j)(\varepsilon(t) - \overline{\varepsilon}) \ \ , j = 1,\ldots,Q$$

where N is the number of training examples, $o_j(t)$ is the output of the jth hidden unit for the tth example, $\overline{o}_j$ is the mean of the jth hidden unit outputs, $\varepsilon(t)$ is the output error at the network output and $\overline{\varepsilon}$ is the mean of the output errors.

6. Find the maximum absolute covariance $\left|C_{MC,j}\right|$ and add the corresponding hidden unit to the network. Do not yet connect it to the output unit. Remove the corresponding hidden unit from the candidate units and set $Q = Q$-1.

7. Maximize the absolute correlation between the output of the new unit and the errors of the network output by maximizing the cost function C_{CC},

$$C_{CC} = \left|\sum_{t=1}^{N}(o(t) - \overline{o})(\varepsilon(t) - \overline{\varepsilon})\right|$$

where $o(t)$ is the output of the new unit at time t and $\overline{o}$ is the average of the outputs. The maximization is done by adjusting only the weights feeding the new hidden unit. As the hidden units have nonlinear activation function, gradient methods must be used. Only after the new hidden unit has been trained, its output is connected to the output unit.

8. Optimize the currently existing weights that feed the output unit with linear regression. Note that the number of these weights is increased by one every time a new candidate unit is connected to the output unit and due to the optimization the output error changes each time.

9. If the desired learning performance has been reached or q_m candidate units have been connected to the output unit, quit. Otherwise, go back to step 5 for the remaining candidate units.

To determine the desired learning performance in step 9, we observe the training mean square error (MSE). After we have added and trained the nth hidden unit, its training MSE is computed. This value is then subtracted from the training MSE value of the (n-1)th hidden unit. If this value is smaller than a given threshold, which was set to 0.01 in our simulations, no more hidden units are added. Otherwise (n+1)th hidden unit is added and trained, unless the maximum network size has been reached, and the same steps are taken as with the nth hidden unit.

4. SIMULATION RESULTS

The transmitted signal considered here is a binary data burst, which consists of 500 bits. The first 100 bits are used for training the equalizers, thus they are known at the receiver. The final 400 bits carry the information, which is not known beforehand. We have transmitted 100 consecutive bursts through the described nonlinear channel with varying SNR. For each received burst, the equalizer is trained with the known training sequence of the burst and the information sequence is then equalized without further adaptation. The average bit error rate (BER) is then computed over the 100 transmitted bursts.

The MLP network equalizers with fixed size were given eight hidden units and two inputs. The MC initialized MLP network required approximately 80 training epochs for each burst, whereas the randomly initialized MLP network needed at least 500 training epochs. Each added hidden unit of the MC-CC MLP network was trained for 40 epochs, which seemed to be enough for convergence. In addition, we have also given results for a linear equalizer, which uses pseudo-inverse method of linear regression to determine its weights by the following equation

$$\boldsymbol{v}^{\boldsymbol{T}} = (\boldsymbol{R}\boldsymbol{R}^{\boldsymbol{T}})^{-1}\boldsymbol{R}\boldsymbol{a}^{\boldsymbol{T}}$$

where $\boldsymbol{R}$ is a (p+1)*N-dimensional input matrix and $\boldsymbol{a}$ is an 1*N target output vector [4]. This matrix computation provides fast and efficient way to determine the weights for the linear equalizer. Furthermore, there is no need for iterations nor any stopping criteria and since there are no need for user-defined parameters, the final solution cannot be deteriorated because of poor parameter selection.

Fig. 2 shows the results achieved. In plot 2a. the delay is 0, thus the equalization problem is similar to the one depicted in Fig. 1a., except that the clusters appearing on the scatter plot may overlap differently, depending on the SNR. However, it is clear that the LE is not able to provide satisfactory bit error rates, whereas all the MLP network equalizers can. The MC-CC MLP network seems to provide the best BERs, even though its average size is smaller than the size of the MLP networks with fixed size, as can be seen in Fig. 2b. Also, the MC initialized MLP network provides slightly better BERs than the randomly initialized network. It also needs clearly less computation, since it requires only 80 training epochs, whereas the randomly initialized network needs 500 training epochs. When comparing the time spent for computation, we observed the used CPU-time to run the equalization operation using Matlab 6.5. with Pentium III 600 MHz processor. The LE spends approximately 0.3 seconds in each case for equalizing 100 bursts, whereas the randomly initialized MLP network and MC initialized network spend 82.1 and 15.8 seconds, respectively. The time spent with MC-CC network varies, depending on the incoming signal and hence, the size of the network to be grown. Given the delay is 0, MC-CC MLP network spends 26.3 seconds, if the SNR = 10 dB, and 23.1 seconds, if the SNR = 20 dB. When the delay is 1, it spends 12.6 seconds, if the SNR = 10 dB, and 7.0 seconds, if the SNR = 20 dB.

In Fig. 2c. we can see that, since the equalization problem is close to being linearly separable (similar to Fig. 1b.), even the LE provides better results. However, the MLP networks

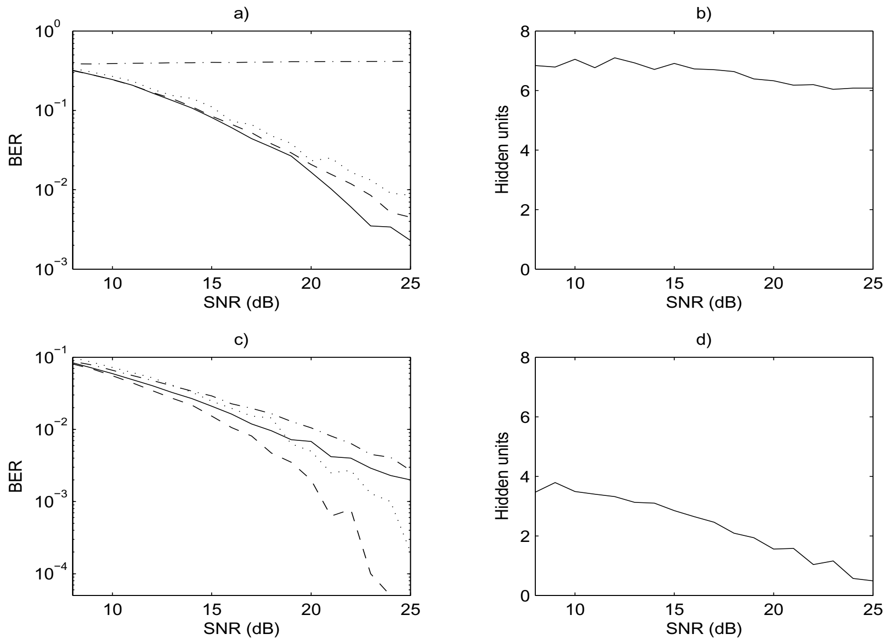

Figure 2. Plots a) and c) show the BERs for the LE (dash-dotted), randomly initialized MLP network (dotted), MC initialized MLP network (dashed) and MC-CC MLP network (solid line) when delay is 0 and 1, respectively. Plots b) and d) show the average number of hidden units in the MC-CC MLP network when delay is 0 and 1, respectively.

still achieve even better BERs. Now, the MC initialized MLP network provides the smallest BERs, wheras the MC-CC trained network seems to grow a bit too small networks as can be seen in Fig. 2d. It saves computation by equalizing several bursts without any hidden units, whereas it should grow a bit larger network in some cases to achieve comparable results with the fixed-size networks. This could be achieved by more careful selection of the desired learning performance in the algorithm.

5. CONCLUSIONS

The studied initialization and learning methods have provided significant improvements on the performance of an MLP network equalizer over nonlinear channels. The studied MC initialization provides much faster convergence than random weight initialization enabling to use less training iterations and hence, less computation. The cascade-correlation learning method can further decrease the need for computation, especially in channels, that have varying conditions. In these cases, there is no need to set some fixed size to the network, but the CC learning can grow the network from scratch, thus providing suitable sized network for each channel case.

REFERENCES

[1] J. G. Proakis, *Digital Communications*. New York: McGraw-Hill, 1995.

[2] 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, no. 2, pp. 107–119, Jun. 1990.

[3] G. J. Gibson, S. Siu and C. F. N. Cowan, “The application on nonlinear structures to the reconstruction of binary signals,” *IEEE Trans. on Signal Processing*, vol. 39, no. 8, pp. 1109–1118, Aug. 1991.

[4] S. Haykin, *Neural Networks, a Comprehensive Foundation*, Macmillan, New York, NY, 1994.

[5] S. Fahlman and C. Lebiere, “The cascade-correlation learning architecture,” Research report CMU-CS-90-100, School of Computer Science, Carnegie Mellon University, Pittsburgh, USA, 1991.

[6] M. Lehtokangas, P. Korpisaari and K. Kaski, ”Maximum covariance method for weight initialization of multilayer perceptron network,” in *Proc. ESANN 1996*, Bruges, Belgium, Apr. 1996, pp. 243–248.

[7] K. Georgoulakis and S. Theodoridis, “Efficient clustering techniques for channel equalization in hostile environments,” *Signal Processing*, vol. 58, no. 2, pp. 153–164, Apr. 1997.

[8] M. Riedmiller and H. Braun, “A direct adaptive method for faster backpropagation learning: the RPROP algorithm,” in *Proc. IEEE Int. Conference on Neural Networks*, San Francisco, CA, USA, Mar.-Apr. 1993, pp. 586–591.