IMPROVED NEURAL NETWORK EQUALIZATION BY THE USE OF MAXIMUM COVARIANCE WEIGHT INITIALIZATION

Arto Kantsila, Mikko Lehtokangas and Jukka Saarinen
Tampere University of Technology
Signal Processing Laboratory
P.O.Box 553
FIN-33101 Tampere
FINLAND
Tel: +358-3-365 2140; Fax: +358-3-365 3095
Email: artok@cs.tut.fi

ABSTRACT

In this paper we focus on adaptive equalization of binary telecommunication signals in a baseband digital communication system. We have studied the use of multilayer perceptron (MLP) networks for equalizing binary data bursts in a channel that introduces both intersymbol interference and noise to the transmitted signal. Conventionally the weights of the MLP network are initialized randomly. Here we have studied the use of maximum covariance (MC) initialization scheme in the weight initialization. By applying MC initialization we have been able to speed up the convergence and decrease the total computational load of the system. This is very important in telecommunications, where it is often not possible to use systems that require a lot of computation.

1. INTRODUCTION

Equalizer is a part of the receiver, where the channel influences are being compensated [1]. Successful equalization enables to find out the original information that was transmitted. The channel models have commonly both linear intersymbol interference (ISI) and additive white Gaussian noise. This kind of a channel can distort the original signal so that nonlinear channel equalization is needed in order to be able to find out the original information that was sent. Therefore we have studied the use of multilayer perceptron (MLP) networks [2] for equalizing binary data bursts. The main advantage of MLP is that it can perform nonlinear equalization unlike the conventional linear equalizers. The disadvantage is that the training of MLP requires a lot of computation compared to linear equalizers, if the network weights are initialized randomly [3]. The computational load of MLP network can be decreased, if some computationally efficient weight initialization method is used. Therefore we have studied the use of maximum covariance (MC) initialization method [4]. We shall present results where it can be seen that MLP networks can be used successfully in a case where linear equalizers perform poorly. Also we show that by applying MC initialization method the performance of an MLP network can be improved significantly. The convergence is faster and thus better bit error rates (BER) are achieved faster than with randomly initialized MLP network. Also it is shown that MC initialization method does not require a lot of computation, but makes it actually possible to decrease the number of training iterations and the size of the network. Thus the total computational load of the system can be made smaller.

2. APPLICATION SETUP

The transmitted signal considered in this study is a binary data burst, which has similar features to a GSM-burst [5]. First of all a burst consists of two sequences, training sequence and data sequence, totaling 142 bits. The first 26 bits of the burst are called training bits. They are used to train the equalizers to compensate the channel influences as much as possible. Note, that the training bits are a fixed sequence of bits which are known in the receiver end. The last 116 bits of the burst, called as data bits, contain the information being sent. These are the bits that need to be equalized. Thus they are not known in the receiver. Here, the transmitted data are assumed to be binary taking values of either 1 or -1 with equal probability. Before being transmitted to the channel the burst is oversampled so that for every bit three samples are taken.

The communication channel which we shall consider throughout this paper has both ISI and additive white Gaussian noise. The ISI part of the channel is modeled as a finite impulse response (FIR) discrete-time filter. The channel output y can now be described by the following equation

$$y(t) = \sum_{i=0}^{N} h(i)a(t-i) + \delta(t) \quad , \tag{1}$$

where a is the transmitted signal, $h(i)$ are the channel coefficients, δ represents additive noise, and t represents time.

3. MULTILAYER PERCEPTRON NETWORK EQUALIZER

In this study we have used MLP networks with one output unit and one hidden layer. The output unit was set to be linear, and the activation function in the hidden units was chosen to be hyperbolic tangent function. Therefore the network equation is as follows

$$z(t) = v_0 + \sum_{j=1}^{q} v_j \tanh\left(w_{0j} + \sum_{i=1}^{p} w_{ij} y(t-i+1)\right), \quad (2)$$

where $z(t)$ is the network output, $y(t)$ is the channel output, 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. The training of the networks was done with the RPROP-algorithm [6]. First we have studied a network which has randomly initialized weight values (interval -0.5,...,0.5). Secondly we have used a network, where the weights have been initialized by using maximum covariance (MC) initialization scheme.

3.1. Maximum Covariance (MC) Initialization Scheme

In the MC initialization method [4], 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. Finally weights feeding the output unit are calculated with linear regression. The algorithm goes as follows:

1. Create Q candidate hidden units ($Q>>q$, where q is the desired number of hidden units) by initializing the weights with random values. Here we have used $Q = 10q$ and the uniformly distributed random values of the candidate hidden units were chosen from the interval [-4,...,4].
2. Do not connect the candidate units to the output unit yet. At this time, the only parameter feeding the output unit is the bias weight. Set the bias weight to such a value that the network output is the mean of the desired output sequence.
3. Calculate the covariance for each of the candidate unit from the equation

$$C_j = \frac{1}{n}\sum_{t=1}^{n}(o_j(t)-\bar{o}_j)(\varepsilon(t)-\bar{\varepsilon}) \quad , j = 1,...,Q \quad (3)$$

where n is the number of training examples, $o_j(t)$ is the output of the jth hidden unit for the tth example, $\bar{o}_j$ is the mean of the jth hidden unit outputs, $\varepsilon(t)$ is the output error at the network output and $\bar{\varepsilon}$ is the mean of the output errors.
4. Find the maximum absolute covariance $|C_j|$ and connect the corresponding hidden unit to the output unit. Then decrement the value of Q by one.
5. 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.
6. If q candidate units have been connected to the output unit, quit the initialization phase. Otherwise repeat steps 3-5 for the remaining candidate units.

4. SIMULATION RESULTS

In this study we simulated both fixed and altering channel when the MLP networks had two inputs. When simulating fixed channel the channel coefficients were set to have values $\underline{h}$ = [0.5 -0.3 0.6 -0.7 -0.8]. In this case a linear equalizer performs poorly as was found out in our research work earlier [3]. The networks were trained with the training sequence for 150 epochs applying RPROP-algorithm. Each simulation was repeated 10 times. Fig. 1 depicts the average convergence of the data sequence BER for both randomly initialized network and MC initialized network when the number of hidden units varies from four to ten and the signal to noise ratio (SNR) is 20 dB. Table 1 shows the average final bit error rates and the standard deviations achieved with both methods as a function of number of hidden units when the networks were trained for 150 epochs and SNR is 20 dB. From Fig. 1 and Table 1 it can be seen that MC initialized network converges very quickly compared to the randomly initialized MLP network. When the network has four or six hidden units, the MLP network with MC initialization only takes about 50 training epochs to converge, whereas randomly initialized network needs more than 150 training epochs. When the MC initialized network has eight hidden units it only takes a few training epochs to converge and with ten hidden units it achieves BER that is almost equal to zero even before the actual training phase. In all these cases the randomly initialized network needs at least 150 training epochs to converge. It can be seen that the MC initialized network achieves clearly better BER than the randomly initialized network, when they are trained for the same amount of epochs. Inversely, it can be said that the MC initialized network achieves the same BER as the randomly initialized network in a considerably smaller amount of training epochs.

When simulating altering channel we transmitted several bursts consecutively. The channel impulse response $\underline{h}$ = [$h(0)$, $h(1)$,...,$h(4)$] was given uniformly distributed randomly chosen values between -1,...,1 and it was different for every transmitted burst. Also the signal to noise ratio was given a randomly chosen value between 3,...,50 dB for each burst (the average SNR value was 26.7 dB). Fig. 2 depicts the average convergences of the data sequence BER when the number of hidden units in the networks varies from four to ten. Table 2 shows the average BERs achieved when 635 bursts were transmitted and the networks were trained for 150 epochs. As can be seen the MC initialized MLP network performs better than the randomly initialized network in terms of convergence and final bit error rates, when the networks are trained for the same amount of training epochs. In all four cases the BER of the MC initialized network right after initialization is about at same level as the BER of the randomly initialized network after 150 training epochs. For comparison, a linear equalizer with ten weight values achieved an average BER of 0.2868, which is obviously too large.

Finally, Table 3 shows the computational load of MC initialization in training epochs (e.g. if there are 6 hidden units, the computational load of MC initialization is equal to 15 training epochs). From this table it can be seen that the computational load of MC initialization is relatively small. Thus the total computational load of the system can be made much smaller, which is very important in telecommunication systems, where it is often not possible to use systems that require a lot of computation.

5. CONCLUSIONS

We have compared MLP network with randomly initialized weight values and MLP network with MC initialization for equalization purposes in a baseband digital communication system. We have simulated both fixed and altering channel and found out that the network with MC initialization converges much faster than the randomly initialized network. Due to this the MC initialized network achieves better bit error rates than the randomly initialized network, if they are trained with the same amount of training epochs. Inversely we can say that the bit error rates achieved with randomly initialized MLP network can be reached clearly faster and/or with a smaller network by using MLP network with MC initialization. Since the MC initialization scheme is also computationally relatively light, we can actually significantly decrease the amount of computation needed in the network by applying MC initialization.

Table 1. Average bit error rates ber_{av} and standard deviation *std* for the data sequence as a function of number of hidden units. Signal to noise ratio is 20 dB and the networks were trained for 150 epochs.

	rand. initialization		*MC initialization*	
Hidden units	ber_{av}	*std*	ber_{av}	*std*
4	0.1503	0.1131	0.0776	0.0706
6	0.1279	0.0931	0.0121	0.0092
8	0.0618	0.0496	0.0055	0.0061
10	0.0575	0.0295	0.0043	0.0056

Table 2. Average bit error rates for the data sequence for randomly initialized MLP network and MC initialized MLP network as a function of number of hidden units when simulating altering channel. Networks were trained for 150 epochs.

Hidden units	*rand. init.*	*MC init.*
4	0.1742	0.1307
6	0.1419	0.0973
8	0.1363	0.0927
10	0.1201	0.0849

Table 3. The amount of computation needed to perform MC initialization in training epochs (MC cost) as a function of number of hidden units.

Hidden units	4	6	8	10
MC cost (epochs)	12	15	19	23

REFERENCES

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

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

[3] A. Kantsila, M. Lehtokangas and J. Saarinen, “Adaptive equalization of binary data bursts,” *Proc. IASTED International Conference on Signal and Image Processing*, New Orleans, Louisiana, USA, Dec 4-6, 1997, pp. 117-122.

[4] M. Lehtokangas, P. Salmela, J. Saarinen and K. Kaski, “Weight initialization techniques,” in C. Leondes (Ed.), *Algorithms and Architectures, Neural Network Systems Techniques and Applications*, Academic Press, 1998.

[5] M. Mouly and M-B. Pautet, *The GSM System for Mobile Communications*. Palaiseau: Mouly & Pautet, 1992.

[6] M. Riedmiller and H. Braun, “A direct adaptive method for faster backpropagation learning: the RPROP algorithm,” *Proc. IEEE Internat. Conf. Neural Networks*, San Francisco, CA, March 28- April 1, 1993, pp. 586-591.

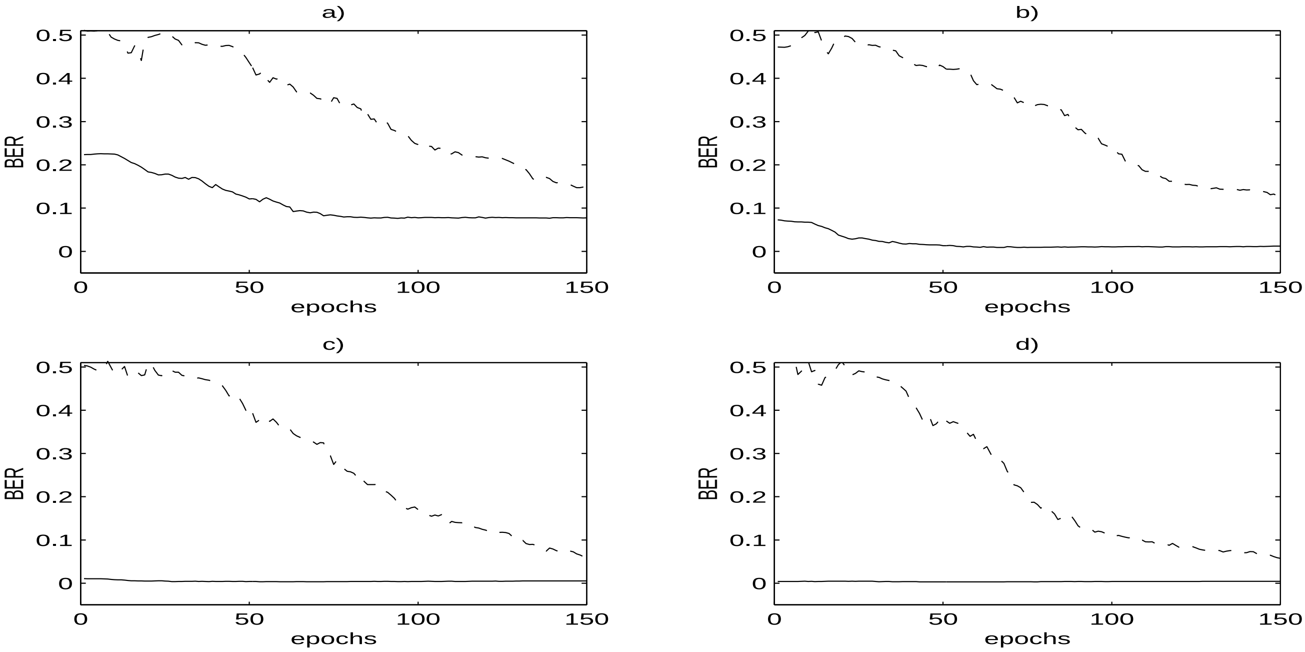

Figure 1. Mean bit error rate convergence of the data sequence for networks that have a) four, b) six, c) eight and d) ten hidden units as a function of number of training epochs when simulating a fixed channel with SNR value 20 dB. The solid line is for the MLP network with MC initialization and the dashed line is for the randomly initialized MLP network.

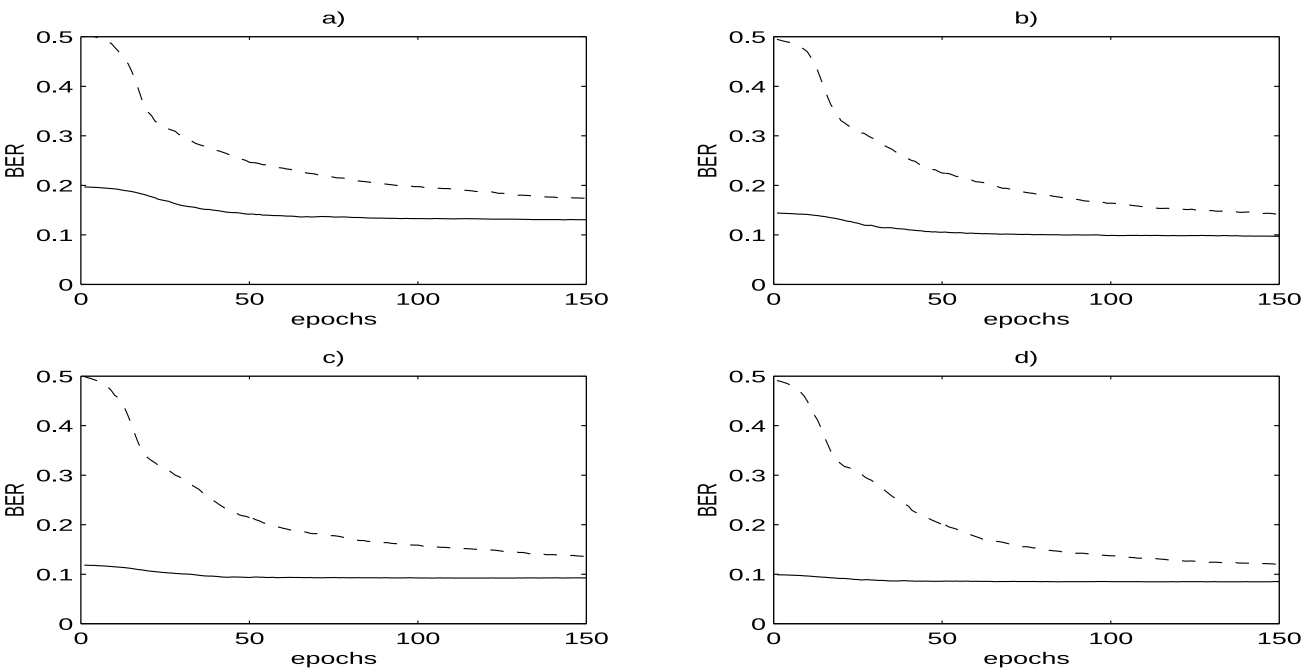

Figure 2. Mean bit error rate convergence of the data sequence for networks that have a) four, b) six, c) eight and d) ten hidden units as a function of number of training epochs when simulating an altering channel. The solid line is for the MLP network with MC initialization and the dashed line is for the randomly initialized MLP network.