ESANN'2003 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 23-25 April 2003, d-side publi., ISBN 2-930307-03-X, pp. 179-184

ON RADIAL BASIS FUNCTION NETWORK EQUALIZATION IN THE GSM SYSTEM

A. Kantsila*, M. Lehtokangas and J. Saarinen

Tampere University of Technology
Institute of Digital and Computer Systems
P.O. Box 553, 33101 Tampere, Finland
*arto.kantsila@tut.fi

Abstract. In this paper we have studied adaptive equalization in the GSM (Global System for Mobile communications) environment using radial basis function (RBF) networks. Equalization is here considered as a classification problem, where the idea is to map the received complex-valued signal into desired binary values using RBF network equalizer. Results prove that the RBF network provides very good bit error rates with acceptable computational complexity. Performance comparisons are made to a linear equalizer, a multilayer perceptron (MLP) network equalizer and to a Viterbi equalizer.

1. Introduction

In today's mobile communication systems, the transmitted signal is subject to various corruptions, caused by for example intersymbol interference (ISI) due to multipath propagation, mobile movement and noise [1]. The purpose of equalization is to compensate for these channel influences so that the original information can be found from the received corrupted signal. Adaptive equalization methods are needed, since the channel response is usually not known beforehand and it is often time-varying. Additionally, equalization needs to be performed efficiently both in terms of computational complexity and time consumed.

Neural networks [2] have been studied for equalization purposes with promising results. However, not a lot of studies deal with real-world communication environments, such as GSM [3]. Therefore, we have studied equalization with radial basis function (RBF) networks in GSM environment. Here, instead of formulating equalization as an inverse filtering or deconvolution problem, we have considered equalization as a classification problem (e.g. [4-6]).

This paper is organized as follows. Chapter 2 introduces the GSM simulation system. The studied RBF network is described in Chapter 3 and the obtained results are discussed in Chapter 4. Final conclusions are then made in Chapter 5.

2. Simulation Model

We have modeled the GSM system using Cossap software [7]. At the transmitter, the information bits are first coded and arranged into bursts according to GSM

ESANN'2003 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 23-25 April 2003, d-side publi., ISBN 2-930307-03-X, pp. 179-184

Recommendations [8]. In this paper we have studied the use of GSM normal bursts only. At the middle of each GSM normal burst there is a training sequence of 26 fixed bits, also known as a midamble, which is known at the receiver end. The midamble is surrounded by 58 information bits on it's both sides. Adding three tail bits on both ends result to the total burst length of 148 bits.

In this study we have used three channel models; hilly terrain at mobile velocity 100 km/h (denoted HT100), typical urban at 50 km/h (TU50) and rural area at 250 km/h (RA250) [8]. Each model is presented by six taps, each determined by their time delay and average power, and the Rayleigh distributed amplitude of each tap varying according to a doppler spectrum. Prior to reception, zero-mean white Gaussian noise with varying signal to noise ratio (SNR) is added to the signal.

At the receiver part the transmitted signal is first filtered with an adjacent channel filter and sampled. This filtered complex-valued signal is then fed to the correlator, which prepares the channel output to be equalized and also estimates the channel response, if needed [7]. The resulted signal is now the input to our studied equalizers.

3. RBF Network Equalizer

The radial basis function network considered here, has a following structure. The network has p inputs, meaning that at every time instant t, an input vector $\underline{y}_t = [y(t+d), y(t+d-1),\ldots, y(t-p+1+d)]^T$ is fed to the network. Here, $y(t)$ is the complex-valued output of the correlator and d represents delay. The hidden layer consists of an array of computing nodes. Each node contains a parameter vector, called a centre. A squared distance between the centre and the input vector is computed in each node. This distance is divided by a parameter called a width and the following result is passed through a nonlinear radial basis function, which in our case is Gaussian. The output layer has a single output unit and it performs as a linear combiner with connection weights $\underline{w}$, resulting a following definition to our network:

$$z(t) = w_0 + \sum_{k=1}^{K} w_k \exp\left(-\frac{1}{s_k^2}\left\|\underline{y}_t - C_k\right\|^2\right), \tag{1}$$

where C_k is the kth centre and s_k is the width. The input vector $\underline{y}_t$ is complex-valued and the centers are complex-valued vectors as well. However the output of the *exp*-function is real and thus the connection weights are real-valued as well. The width parameter was given a constant value, which is the same for each center. The weights have been computed using the method of least squares [9]

$$\underline{w} = \Phi^{+}\underline{d}_{tr} \tag{2}$$

where the weight vector $\underline{w} = [w_0, w_1,\ldots, w_K]$, $\underline{d}_{tr}$ is the desired output vector, $^+$ denotes pseudoinverse and

$$\Phi = \begin{bmatrix} 1 & j(\underline{y}_1;C_1) & j(\underline{y}_1;C_2) & \cdots & j(\underline{y}_1;C_K) \\ 1 & j(\underline{y}_2;C_1) & j(\underline{y}_2;C_2) & \cdots & j(\underline{y}_2;C_K) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & j(\underline{y}_N;C_1) & j(\underline{y}_N;C_2) & \cdots & j(\underline{y}_N;C_K) \end{bmatrix} \tag{3}$$

ESANN'2003 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 23-25 April 2003, d-side publi., ISBN 2-930307-03-X, pp. 179-184

in which

$$j(\underline{y}_t; C_k) = \exp(-\frac{1}{s_k^2}\left\|\underline{y}_t - C_k\right\|^2). \quad (4)$$

The centers for each computing node can be found from the midamble of the transmitted burst using a nearest-neighbor-type clustering procedure given e.g. in [10, 11]. Following this procedure we create cluster centers for transmitted +1 and –1 – bits. A distance threshold value D_{th} determines whether we classify the received sample to the nearest cluster center representing the correct binary value or create a new center out of it. Thus, the value of D_{th} also determines the total number of cluster centers.

4. Simulation Results

We have transmitted 1000 consecutive bursts through a GSM simulation model using *Cossap* software [7]. The equalization methods have then been studied using *Matlab* software [12]. For each received burst, the equalizers were first trained using the midamble and the information bits were then equalized without further adaptation. Note, that the training is done separately for each received burst.

Two performance criteria for the studied equalizers were observed. First, the bit error rate (BER), which is computed as the percentage number of incorrect decisions made on the information sequence of each burst after equalization. The second performance measure is the computational complexity, which is given in terms of the number of most important floating point operations (*flops*) used for equalization. *Matlab* counts flops as follows [12]; additions, subtractions, multiplications and divisions are one flop each, if the operands are real. Two flops are noticed in the case of complex operands. It should be noted, that *Matlab* counts for no flops in case of comparison and search functions.

The RBF network equalizer was given five inputs and the delay was set to 2. For all the computing nodes in the network, the width parameter was set to $s_k = 80$ after empirical tests. The weights of the network were computed using linear regression. We made tests with varying values for the D_{th} and also studied the influence of applying the stochastic gradient algorithm [9] for updating the center locations and the width parameters. However, the improvement in BERs was very small compared to the increase in computational complexity.

In fact, the best results were achieved when D_{th} was given value 0 in almost each case. This means that the clustering algorithm creates centers out of each training vector. The network has in this case 26 computing nodes. Since this is the case, we can actually drop the clustering procedure and simply set the centers to be the same as the training sequence vectors. The width parameters and the center locations are not updated during the computation of the weights. This way we can decrease the total computational load of the system and similarly, the actual time spent for computation. The simulation results are given for this type of RBF network equalizer.

For comparison purposes we have also studied the use of a linear equalizer (LE), an MLP network equalizer and a Viterbi equalizer. The weights of the complex-

ESANN'2003 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 23-25 April 2003, d-side publi., ISBN 2-930307-03-X, pp. 179-184

valued linear equalizer have been computed using linear regression. Five inputs with delay of 2 was used here as well. The studied MLP network employs one hidden layer with six hidden units and a single linear output unit. In the hidden layer split complex activation function tanh was used [13]. Although the split complex activation function can never be analytic, it avoids the unboundedness of fully complex activation function [14-15]. Furthermore, our previous studies have shown that it seems to work quite well in our application [13]. The training has been carried out using the backpropagation algorithm. The input vector is the same as for the RBF network.

Another benchmarking method that we have considered is based on Viterbi algorithm (VA) [16]. Unlike the previously described methods, this Viterbi equalizer needs to have an estimate of the channel response. This is done using a correlation approach given in [7]. For each burst five channel coefficients are computed for estimating the response. The VA models the channel as a finite state machine having 16 states and to each state one can arrive from two states. For each arrived sample, we compute the Euclidean distance between the received signal and all the possible reference signals, which represent here a five symbol sequence (4 bits for the states and 1 bit for the transition) multiplied with the estimated channel coefficients. For each of the 16 new states, the VA selects the most likely transition to the new state by comparing the accumulated path metrics of the two predecessor states plus the so called branch (transition) metrics. After all the 148 received samples of the burst are processed as described above, the Viterbi equalizer has stored one decision and one metric difference for each of the 16 states at each sample time. The VA then traces back the best path. Notice, that the achieved results with the Viterbi equalizer are obtained without any use of decoding.

Tables 1.-3. show the BERs obtained with each method in the studied three channel models with SNR 5, 10 and 20 dB. As can be seen, the RBF network equalizer provides the best BERs when there is a lot of noise present. However, when the noise decreases, the Viterbi equalizer provides the smallest BERs. The performance of the MLP network equalizer is quite close to the Viterbi equalizer when SNR is 5-10 dB. The linear equalizer is clearly the worst performer.

Finally, Table 4 summarizes the computational complexities of the studied methods. The figures in the first column of this Table are average number of flops used over 1000 bursts. The linear equalizer is clearly the most efficient one in terms of computational complexity, but as was shown, it also provided the worst bit error rates. The RBF network seems to require significantly less computation than the MLP network. This is mainly due to the extensive training required by the MLP network, whereas linear regression can be applied for the RBF network. The applied Viterbi equalizer required surprisingly little computation in terms of flops, even though it is often described as computationally demanding in publications. However, one must notice here, that Matlab does not count any flops for operations like comparison and search, which are often used in VA. Therefore, we have also added another row to Table 4, which gives the actual time spent for performing the equalization task for 1000 bursts in TU50-channel with SNR = 5 dB, using Matlab version 5.3 with Pentium III 600 MHz processor. There we can see a clear difference between the complexity of the LE and the VA.

We have also made some preliminary tests on using the channel estimates in RBF network equalization. Since the channel estimates are available in GSM, it seems

ESANN'2003 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 23-25 April 2003, d-side publi., ISBN 2-930307-03-X, pp. 179-184

worthwhile to take advantage of them, when locating the centers in RBF network. However, we are still working on this algorithm.

TABLE 1. Bit error rates for the studied methods in HT100-channel.

SNR/Method	LE	MLP	VA	RBF
5 dB	13.52 %	12.23 %	11.94 %	11.00 %
10 dB	7.01 %	5.74 %	4.86 %	5.14 %
20 dB	2.72 %	2.02 %	0.74 %	1.76 %

TABLE 2. Bit error rates for the studied methods in TU50-channel.

SNR/Method	LE	MLP	VA	RBF
5 dB	12.50 %	10.64 %	10.43 %	9.61 %
10 dB	5.61 %	4.25 %	3.76 %	3.92 %
20 dB	1.15 %	0.95 %	0.58 %	0.88 %

TABLE 3. Bit error rates for the studied methods in RA250-channel.

SNR/Method	LE	MLP	VA	RBF
5 dB	10.59 %	8.83 %	8.93 %	7.89 %
10 dB	4.64 %	3.54 %	3.45 %	3.10 %
20 dB	0.83 %	0.74 %	0.80 %	0.77 %

TABLE 4. Computational complexities given as the average number of flops required to equalize one burst and as actual time spent for equalizing 1000 bursts.

Method	LE	MLP	VA	RBF
Flops	$2.9*10^4$	$1.5*10^7$	$4.0*10^4$	$4.7*10^5$
Secs	11.9	237.3	48.3	66.2

5. Conclusions

We have studied adaptive equalization in the GSM environment. Since equalization can be seen as a classification problem, neural networks offer one possibility to this task. RBF networks have been used for many classification tasks and they also proved to work very well in this application. Their performance in terms of bit error rates was found to even outperform the applied Viterbi equalizer in cases where there was a lot of noise present. In addition, the computational complexity of the RBF network equalizer was clearly smaller than that of the studied MLP network equalizer. However, the applied Viterbi equalizer outperforms the RBF network in both floating point operations required and actual time spent for computation. This

issue needs to be addressed before the RBF network can be truly considered as a realistic alternative to Viterbi. Future studies will be made on applying the channel estimates for the RBF network. Nevertheless, RBF networks seem to provide interesting results for this application.

6. Acknowledgements

Special thanks to Dr. Jorma Lilleberg for his constructive ideas and comments.

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] M. Mouly and M-B. Pautet, *The GSM System for Mobile Communications*. Palaiseau: Mouly & Pautet, 1992.
[4] B. Mulgrew, "Applying radial basis functions," *IEEE Signal Processing Magazine*, March 1996, pp. 50-65.
[5] S. Chen, B. Mulgrew and P. M. Grant, "A clustering technique for digital communications channel equalization using radial basis function networks," *IEEE Trans. Neural Networks*, Vol. 4, No. 4, Jul. 1993, pp. 570-579.
[6] K. Georgoulakis and S. Theodoridis, "Efficient clustering techniques for channel equalization in hostile environments," *Signal Processing*, Vol. 58, No. 2, Apr. 1997, pp. 153-164.
[7] *COSSAP v. 1998.08 Model Library gsmeq*, Synopsys Inc., USA, 1998, 31 pages.
[8] GSM Technical Specification, ETSI, 1997.
[9] S. Haykin, *Adaptive Filter Theory*, 3rd Edition, Prentice-Hall, 1996.
[10] A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data*, Prentice-Hall, 1988.
[11] A. Kantsila, M. Lehtokangas and J. Saarinen, "Adaptive equalization of binary data bursts," *Proc. IASTED Int. Conf. Signal and Image Processing (SIP´97)*, Dec 4-6 1997, pp. 117-122.
[12] *Using MATLAB - Version 5*, The Mathworks Inc., USA, 1999.
[13] A. Kantsila, T. Haverinen, M. Lehtokangas and J. Saarinen, "On equalization with complex MLP network in the GSM environment," *Proc. IFSA World Congress and 20th NAFIPS International Conference*, 2001, Vol. 5, pp. 2687-2692.
[14] N. Benvenuto and F. Piazza, "On the complex backpropagation algorithm," *IEEE Trans. on Signal Processing*, Vol. 40, No. 4, Apr. 1992, pp. 967-969.
[15] T. Kim and T. Adali, "Fully complex backpropagation for constant envelope signal processing," *Proc. of the 2000 IEEE Signal Processing Society Workshop, Neural Networks for Signal Processing X*, Vol. 1, 2000, pp. 231-240.
[16] G. D. Forney, "Maximum-likelihood sequence estimation of digital sequences in the presence of intersymbol interference," *IEEE Trans. on Inform. Theory*, Vol. IT-18, No. 3, May 1972, pp. 363-378.