An Adaptive Radial Basis Function Network algorithm for nonlinear channel equalization

Nam yong Kim* *Regular Member*

ABSTRACT

The authors investigate the convergence speed problem of nonlinear adaptive equalization. Convergence constraints and time constant of radial basis function network using stochastic gradient (RBF-SG) algorithm is analyzed and a method of making time constant independent of hidden-node output power by using sample-by-sample node output power estimation is derived. The method for estimating the node power is to use a single-pole low-pass filter. It is shown by simulation that the proposed algorithm gives faster convergence and lower minimum MSE than the RBF-SG algorithm.

Key Words : nonlinear; RBF; equalization; stochastic; gradient.

I. Introduction

For several years, neural networks have been studied as a form of nonlinear equalizer[1-7]. These studies have shown that these nonlinear neural network equalizers are much superior to linear equalizers for channels that suffer from nonlinear distortion[1]. However neural equalizers require much longer training periods than linear equalizers, and they are sensitive to the choice of network parameters[8]. RBF equalizers[5-8] have a simple structure and offer some advantages over both linear and multiplayer perceptron (MLP) equalizers. The architecture of RBF network (RBFN) consistsof input, hidden, and output layers. RBF equalizers generally need more hidden nodes and training samples to achieve the performance comparable to that of a well trained MLP. However, learning in the RBF is usually much faster than in a MLP. The basis functions in the hidden layer of RBF produce a localized response to the input and typically uses hidden layer nodes with Gaussian response functions and the outputs of thehidden node lie between 0 and 1. Generalization to network with multiple outputs is straightforward and done by assigning M connection weights for each output nodes.

The performance of RBF network is highly dependent on the choice ofcenters and widths in basis function. For a minimum number of nodes, the selected centers should well represent the training data for acceptable classification. Most of the training algorithms for RBF network have been divided into the two stages of processing. Firstly, a clustering method such as the K-means or fuzzy c-means algorithm is applied to the input training samples in order to determine the centers for hidden layer nodes[9]. After the centers are fixed, the widths are determined in a way that reflects the distribution of the centers and input samples. Once the centers and widths are fixed, the weights between hidden and output layer are trained usually by least-mean-squared (LMS) algorithm. This two-stage method provides some useful solutions in pattern classification problems. However, since the centers and widths are fixed after they are chosen and only the weights are adapted for supervised learning, this method often results in not satisfying performance when input data are not particularly clustered. Also these two

* Samcheok National University, Deprt. of Comm. & Inform., (namyong@samcheok.ac.kr),
논문번호 : KICS2005-02-066, 접수일자 : 2005년 2월 14일

stage block-processing algorithms are generally computation-intensive and not suitable for equalization applications where fast on-line processing is needed. In [10], a simple learning algorithm that simultaneously adapts allthe network parameters- centers, widths, and weights was proposed. The algorithm applies stochastic-gradient (SG) method to the RBF parameter adaptation. This RBF- stochastic-gradient (RBF-SG) algorithm has proved superior to many of the existing algorithms, with less computational requirementsin nonlinear channel equalization applications[11].

The learning speed of the stochastic-gradient descent method is dependent on signal variance. If we use node output variance in adapting the RBF network parameters we can acquire faster convergence. In this paper we introduce a sample-by-sample node-power estimation method and propose a new algorithm which combines the node-power estimation method with the RBF-SG algorithm. Several nonlinear channel models have been simulated to show that the proposed algorithm performs better than the RBF-SG equalizer algorithm.

The organization of this paper is as follows. The RBF-SG algorithm is reviewed in Section II. In Section III, the proposed RBF-SG algorithm using node-power estimation is presented. Simulation results are presented in Section IV, and Conclusions are given in Section V.

Ⅱ. RBF-SG Algorithm

The RBF-SG algorithm adapts all the free parameters of the network using gradient descent of the instantaneous output error power. Let input vector $x^{(n)}=[x(n),x(n-1),...,x(n-M+1)]^T$ and $d^{(n)}$ denote desired output. Let $y^{(n)}$ denote the RBF output for input $x^{(n)}$ and let the error be denoted by $e^{(n)}=d^{(n)}-y^{(n)}$, all at the training time n. For a network parameter ϕ, the RBF-SG algorithm adapts its value $\phi^{(n)}$ at time n according to

$$\phi^{(n+1)}=\phi^{(n)}-\mu_\theta(\frac{\partial e^{(n)2}}{\partial\phi^{(n)}}) \tag{1}$$

where μ_θ is the convergence parameter. Among localized basis functions, the Gaussian is the most popular choice for RBF-SG. The output of RBF-SG with M Gaussian basis functions is

$$f(x)=\sum_{j=1}^{j=M}w_j^{(n)}\exp(-\frac{\left\|x^{(n)}-c_j^{(n)}\right\|^2}{(\sigma_j^{(n)})^2}) \tag{2}$$

where $c_j^{(n)}$ is the center associated with hidden node j at time n and $\sigma_j^{(n)}$ is the width parameter at time n, which represent a measure of the spread of data. $w_j^{(n)}$ is the weight from hidden unit j at time n and M is the number of the hidden nodes.

The RBF-SG algorithm adapts the network parameters according to the following equations[10].

$$w_j^{(n+1)}=w_j^{(n)}+ \mu_w e^{(n)}\exp(\frac{-\left\|x^{(n)}-c_j^{(n)}\right\|^2}{(\sigma_j^{(n)})^2}) \tag{3}$$

$$c_j^{(n+1)}=c_j^{(n)}+ \mu_c e^{(n)}w_j^{(n)}\exp(\frac{-\left\|x^{(n)}-c_j^{(n)}\right\|^2}{(\sigma_j^{(n)})^2})\frac{(x^{(n)}-c_j^{(n)})}{(\sigma_j^{(n)})^2} \tag{4}$$

$$\sigma_j^{(n+1)}=\sigma_j^{(n)}+ \mu_s e^{(n)}w_j^{(n)}\exp(\frac{-\left\|x^{(n)}-c_j^{(n)}\right\|^2}{(\sigma_j^{(n)})^2})\frac{\left\|x^{(n)}-c_j^{(n)}\right\|^2}{(\sigma_j^{(n)})^3} \tag{5}$$

The weights can be initialized to be either small random values or zeros. The initial centers can be determined, for example, by means of a nearest-neighbor clustering of a number of input samples in the training set. If no delay in processing isdesired, the centers can be simply initialized by forming them with the first few inputs. The values of width parameters also can be initialized in different ways. One simple method is to use a common value for all widths by using the average of the nearest-neighbor distances among initialized centers.

The RBF-SG algorithm has certain advantages over existing methods. All free network parameters of the RBF-SG are adapted simultaneously usually yielding improved overall solutions. The method can provide greater robustness to poor initial choices of parameters, especially the centers. Also the algorithm is well suited for on-line adaptive signal processing unlike block processing algorithms. As a localized basis function, Gaussian is fast decaying function. It can be assumed that not all the basis function units contribute significantly to the network output values. Hence, instead of training all the hidden nodes, one could train only a selected number of basis function nodes with the largest output values. This means each node out power widely fluctuates in time. It is desirable that we take into account the fluctuations of each node output power when we adapt the network parameters. In this paper, we apply simultaneous power estimation at each node to the RBF-SG algorithm which uses localized Gaussian basis function and fixed convergence factors.

Ⅲ. The proposed algorithm using node-power estimation

The convergence speed of the stochastic gradient method is dependent on signal power[12]. If we use node output power in adapting the network parameters we can acquire fast convergence speed. Each node output power fluctuates and it can be considered not much correlated with other node outputs due to its localized basis function. Defining $X_j^{(n)} = \exp(\frac{-\|x^{(n)} - c_j^{(n)}\|^2}{(\sigma_j^{(n)})^2})$, equation (3) becomes

$$w_j^{(n+1)} = w_j^{(n)} + \mu_w e^{(n)} X_j^{(n)} \tag{6}$$

The auto-correlation matrix $R_{XX}^{(n)}$ of node output $X_j^{(n)}$ is defined as

$$R_{XX}^{(n)} = E[X^{(n)} X^{(n)T}] \tag{7}$$

where $X^{(n)} = [X_1^{(n)}, X_2^{(n)}, \ldots, X_M^{(n)}]^T$

If the correlation matrix $R_{XX}^{(n)}$ is positive definite, it can be expressed as $Q \Lambda_{XX} Q^{-1}$, where Q is the eigen vector matrix of $R_{XX}^{(n)}$ whose columns are the eigen vectors of $R_{XX}^{(n)}$ and Λ_{XX} is a diagonal matrix of eigenvalues $\Lambda_{XX} = diag[\lambda_1, \lambda_2, \ldots, \lambda_M]$[12]. Then equation (6) can be expressed as

$$w_j^{(n+1)} = (1 - \mu_w \lambda_j) w_j^{(n)} + \mu_w \lambda_j w_j^o \tag{8}$$

where w_j^o is j-th optimum weight.

Taking the Z transform of the above equation (8), the transfer function of the system A(z) with input $\lambda_j w_j^o$ and output $w_j^{(n)}$ becomes

$$A(z) = \frac{\mu_w z^{-1}}{1 - (1 - \mu_w \lambda_j) z^{-1}} \tag{9}$$

This system is recursive low pass filter with time constant τ_j given by

$$\tau_j = \frac{-1}{\ln(1 - \mu_w \lambda_j)} \approx \frac{1}{\mu_w \lambda_j} \tag{10}$$

And the system is stable if the poles are within the unit circle.

$$0 < \mu_w < \frac{1}{\lambda_j} \tag{11}$$

Average time constant can be depicted as

$$\tau_{av} = \frac{1}{\mu_w \lambda_{av}} \tag{12}$$

where

$$\lambda_{av} = 1/M \sum_{j=1}^{M} \lambda_j = 1/M \sum_{j=1}^{M} p_j^{(n)} = p_{av} \tag{13}$$

and $p_j^{(n)} = E[X_j^{(n)} X_j^{(n)}]$, j-th node output variance .

For some applications we might need the RBF-SG algorithm whose time constant is independent on the node output power. Now if we define the convergence parameter as $\mu_w = \alpha_w / p_{av}$, α_w is a small constant, the time constant τ_{av} becomes independent of λ_j, i.e.,

$$\tau_{av} = \frac{1}{\alpha_w} \tag{14}$$

Assuming that the node output power changes slowly, one common method for estimating the node power for the j-th node is to use a single-pole low-pass filter,

$$p_j^{(n+1)} = \theta \cdot p_j^{(n)} + (1-\theta)\cdot[X_j^{(n)}]^2 \quad (15)$$

where $0<\theta<1$ is a smoothing parameter which controls the bandwidth and time constant of the power- estimation system $S(z)$ with its input $[X_j^{(n)}]^2$.

$$S(z) = (1-\theta)\frac{z}{z-\theta} \quad (16)$$

Weight adaptation by node-power estimation RBF-SG can be written as

$$w_j^{(n+1)} = w_j^{(n)} + \frac{\alpha_w}{p_j^{(n)}} e^{(n)} X_j^{(n)} \quad (17)$$

Similarly, we can apply node-power estimation RBF-SG to adaptation of centers and widths.

$$Y_j^{(n)} = w_j^{(n)} \exp\left(\frac{-\left\|x^{(n)} - c_j^{(n)}\right\|^2}{(\sigma_j^{(n)})^2}\right)\frac{(x^{(n)} - c_j^{(n)})}{(\sigma_j^{(n)})^2} \quad (18)$$

$$p_{c,j}^{(n+1)} = \theta \cdot p_{c,j}^{(n)} + (1-\theta)\cdot[Y_j^{(n)}]^2 \quad (19)$$

$$c_j^{(n+1)} = c_j^n + \frac{\alpha_c}{p_{c,j}^{(n)}} e^{(n)} Y_j^{(n)} \quad (20)$$

$$Z_j^{(n)} = w_j^{(n)} \exp\left(\frac{-\left\|x^{(n)} - c_j^{(n)}\right\|^2}{(\sigma_j^{(n)})^2}\right)\frac{\left\|x^{(n)} - c_j^{(n)}\right\|^2}{(\sigma_j^{(n)})^3} \quad (21)$$

$$p_{s,j}^{(n+1)} = \theta \cdot p_{s,j}^{(n)} + (1-\theta)\cdot[Z_j^{(n)}]^2 \quad (22)$$

$$\sigma_j^{(n+1)} = \sigma_j^{(n)} + \frac{\alpha_s}{p_{s,j}^{(n)}} e^{(n)} Z_j^{(n)} \quad (23)$$

Computation complexity of the RBF-SG algorithm is beyond this work. But the proposed method requires 4 more multiplications for the calculation $p_j^{(n+1)} = \theta \cdot p_j^{(n)} + (1-\theta)\cdot[X_j^{(n)}]^2$ in (15) and 3 more divisions for $\frac{\alpha_w}{p_j^{(n)}}$, $\frac{\alpha_c}{p_j^{(n)}}$ and $\frac{\alpha_s}{p_j^{(n)}}$ in (17), (20) and (23), respectively, at the training time n than the RBF-SG algorithm does.

Ⅳ. Simulation results

Performance simulation results are presented and compared. For the RBFN's the proposed training method was compared to the RBF-SG and linear TDL equalizers with LMS training. In the simulations, performance was measured by the mean-squared-error (MSE) between the equalizer output and the correct symbols. The transmitted training symbol is a random sequence of bipolar signals (+1, -1). The initial centers were formed from the first few successive channel output samples of the training set. The RBFN weights were initialized to zeros. The spread parameters were initially set to one common value for all the basis function units of the RBFN's. The values of adaptation parameters for training algorithms and the size of the RBFN's used in the simulations were chosen to result in good minimum MSE performance. The additive white Gaussian noise v_n has zero mean, variance 0.001. The equalizer input dimension was set to $f=2$ and the initial common value of the spread parameter was set to $\sigma=2$. The RBF equalizer had 11 hidden nodes. The adaptation coefficients for RBF-SG and the proposed were $\mu_c=0.03$, $\mu_s=0.03$ and $\mu_w=0.02$ with no momentum. The smoothing parameter θ was set to 0.9999. The linear TDL equalizer had 11 taps. The LMS convergence parameter was also set to $\mu_w=0.02$ for the linear equalizer. The MSE curves were acquired by averaging 500-independent running.

Example 1: In this example, we considered the performance of RBFN equalizers in the following nonlinear channel environment [9]. The overall channel output was given by

$$y_n = h_n - 0.9h_n^3 + v_n \quad (24)$$

$$h_n = d_n + 0.5d_{n-1} \quad (25)$$

Figure 1 shows the MSE convergence of the

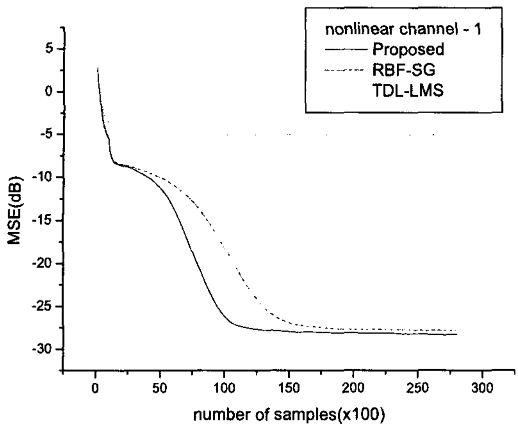


Figure 1. MSE performance in channel-1

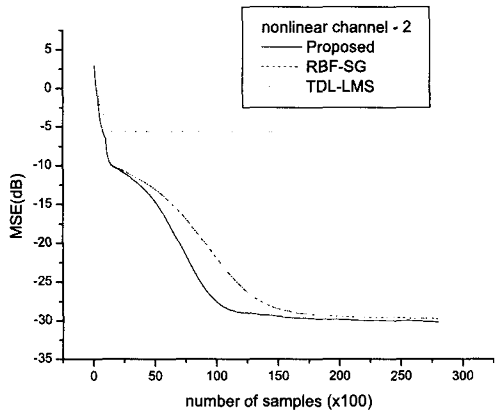


Figure 2. MSE performance in channel-2

equalizers. We can see that the TDL linear equalizer performed very poorly due to the nonlinear channel characteristics. All the RBFN equalizers gave better performance than the linear equalizer. The MSE of linear equalizer decreased no more than -5 dB. The RBF-SG converged in about 24000 samples with minimum MSE, 27.8 dB. The proposed algorithm converged in about 12000 samples compared to the RBF-SG performance. Furthermore, the proposed algorithm had lower minimum MSE, -28.4 dB.

Example 2: This example shows equalization performance in another nonlinear channel environment [13]. The channel output was given by

$$y_n = h_n + 0.1h_n^2 + 0.05h_n^3 + v_n \qquad (26)$$

$$h_n = 0.5d_n + d_{n-1} \qquad (27)$$

The MSE convergence performance of the equalizers in the nonlinear channel is shown in Figure 2. The proposed algorithm's performance was better than that of RBF-SG. The linear FIR equalizer performed also very poorly. The linear equalizer had minimum MSE as -5.6 dB. The RBF-SG converged in about 22000 samples with minimum MSE, 29.5 dB. The proposed algorithm converged in about 16000 samples to the RBF-SG's minimum MSE, 29.5 dB. The proposed algorithm's minimum MSE was -30 dB. This example again shows the superiority of the proposed algorithm to the linear TDL and nonlinear RBF-SG.

V. Conclusions

In this paper, a method of improving MSE convergence performance of the RBF-SG algorithm is investigated. We analyzeconstraints and time constant of RBF-SG algorithm, and derive a method of making time constant independent of hidden-node output power by using sample-by-sample node output power estimation. The method for estimating the node power is to use a single-pole low-pass filter. All the RBFN equalizers gave better performance than the linear equalizer. The proposed algorithm converged as about two times faster as the RBF-SG. Also, the proposed algorithm had lower minimum MSE than the RBF-SG. This shows the proposed algorithmbe applied more effectively to nonlinear channel equalization.

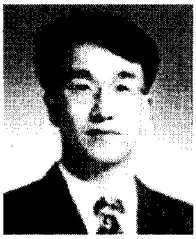
References

[1] S. Chen, G. J. Gibson, C. F. N. Cowan and P. M. Grant, "Adaptive equalization of finite nonlinear channels using multiplayer perceptrons," Signal Processing, vol. 20, pp. 107-119, 1990.

[2] M. Peng, C. L. Nikias and J. G. Proakis, "Adaptive equalization with neural networks; new multiplayer perceptron structures and their evaluation," in Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing, vol. 2.

pp. 301-304, 1992.
[3] N. W. K. Lo and H. M. Hafez, "Neural Network Channel Equalization,"in Proc. IEEE Int. Conf. Neural Networks, vol. 2. pp. 981-986, 1992.
[4] T. Kohonen, K. Ravio, O. Simula, O. Venta and L. Henriksson, "Combining linear equalization and self organizing adaptation in dynamic discrete-signal detection," in Proc. IEEE Int. Joint Conf. Neural Networks, vol. 1. pp. 223-228, 1990.
[5] S. Chen, G. J. Gibson and P. M. Grant, "Reconstruction of binary signals using an adaptive radial basis function equalizer," Signal Processing, vol. 22, pp. 77-93, 1991.
[6] S. Chen, C. F. N. Cowan and P. M. Grant, "Orthogonal least squares learning algorithm for radial basis function networks," IEEE Trans. Neural Networks, vol. 2, pp. 302-309, Mar. 1991.
[7] S. Chen, B. Mulgrew and S. Mclaughlin, "Adaptive Bayesian decision feedback equalizer based on a radial basis function networks," in Proc. IEEE Int. Conf. Commun., pp. 1267-1271, 1992.
[8] 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. pp. 570-579, 1993.
[9] B. J. Oh, K. C. Kwak, S. S. Kim and J. W. Ryu, "A method of designing nonlinear channel equalzier using conditional fuzzy C-means clustering" in Proc. IEEE Int. Conf. Signal Processing, vol. 2. pp. 1355-1358, 2002.
[10] I. Cha and S. A. Kassam, "Time-series prediction by adaptive radial basis function networks,"in Proc. IEEE Twenty-Sixth Annu. Conf. Inform. Sci., Syst., pp. 818-823, Mar. 1993.
[11] I. Cha and S. A. Kassam, "Channel equalization using adaptive complex radial basis function networks," IEEE Journal on Selected Areas in Communications, vol. 13., pp. 122-131, Jan. 1995.
[12] B. Widrow and S. D. Stearns, *Adaptive signal processing,* Prentice-Hall, 1985.
[13] D. J. Sebald and J. A. Bucklew, "Support vector machine techniques for nonlinear equalization," IEEE Trans. Signal Processing, vol. 48. pp. 3217-3226, Nov. 2000

김 남 용(Nam yong Kim) 정회원



1986년 2월 연세대학교 전자공학과 졸업
1988년 2월 연세대학교 전자공학과 석사
1991년 8월 연세대학교 전자공학과 박사
1992년 8월~1998년 2월 관동대학교 전자통신공학과 부교수
1998년 3월~현재 삼척대학교 정보통신공학과 부교수
<관심분야> adaptive equalization, RBFN algorithm, odor sensing systems.