On Neural Network Adaptive Equalizers for Digital Communication

Hongrui Jiang*, Kyung-Sup Kwak* *Regular Members*

ABSTRACT

Two decision feedback equalizer structures employing recurrent neural network (RNN) used for non-linear channels with severe intersymbol interference (ISI) and non-linear distortion are proposed in this paper, which skillfully put the traditional decision feedback structure for linear channels equalization into RNN, replace decision feedback signal with training signal in the learning process and adaptively adjust the learning step. Simulative results of the first type of two new equalizer structures have shown that it has better equalization performances than traditional recurrent neural network equalizer (RNNE) under the same condition.

Ⅰ. Introduction

Channel equalization can be regarded as a mode classification problem. Since neural networks have good mode classification properties, different neural networks structures are applied to channel adaptive equalization [1][2][3] with the development of neural networks technology. Various structures and algorithms of equalizers possess their own advantages and shortcomings. For example, provided enough freedom, the multilayer perceptron can be applied to arbitrary complicate non-linear channel equalization. But in project realization, there is always a contradiction between its properties and complexity of realization. The larger the structure, the longer the time needed for computing, and the smaller the data transmitting rate. Recurrent neural network (RNN) has the properties of small size and good performance, and it relieves the contradiction in the channel equalization [3]. Because it is similar to IIR filter, it can get good equalization effects, or complete complicate non-linear map with only a few nodes. But the output of each node will join the feedback, which causes bad stability and consistency of RNNE. The introduction of decision feedback structure can overcome this fault [4].

In this paper, decision feedback structure is put into RNN, and two decision feedback recurrent neural network Equalizer (DFRNNE) structures and their algorithms are proposed, and the learning step is adaptively adjusted. Section II summarizes two DFRNNE structures. Section III gives the learning algorithms of DFRNNE. Section IV shows simulation results. Section V concludes this paper.

Ⅱ. Two DFRNNE Structures

The channel equalizer model is shown in Fig.1. S_k, n_k, X_k and $\hat{s}_{k-d}$ represent signal, noise, input signal of equalizer and estimate signal, respectively. In this paper, the following distortion channel models [6] are used:

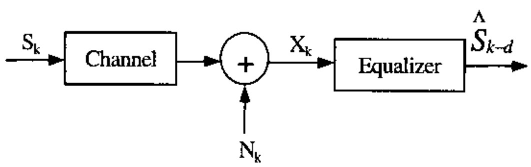


Fig. 1 Channel equalizer model

* 인하대학교 전자공학과 통신공학 연구실
논문번호 : K01086-0219, 접수일자 : 2001년 2월 19일

LCH:

$$X_k = 0.3482\ S_k + 0.8704\ S_{k-1} + 0.3482\ S_{k-2} \qquad (1)$$

$$\text{NLCH:}\ \ Y_k = X_k + 0.2 X_k^2 \qquad (2)$$

where linear distortion channel with severe ISI (LCH) and non-linear distortion channel (NLCH) are investigated. In the following simulative experiments, S_k is a random sequence with equal probability of 1 or +1 with unity power, and Gaussion white noise is added to the equalizers reception end.

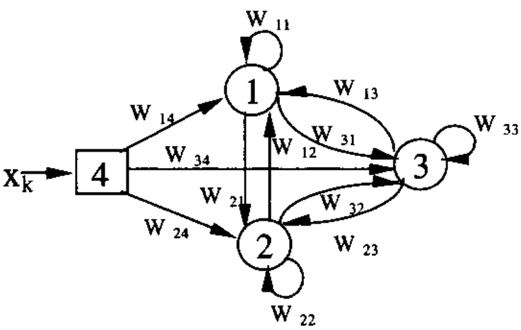


Fig. 2 RNN structure with 3 nodes

Fig.2 is RNN structure with 3 nodes, where all neurons connect each other, each input will be imported into each neuron, and each neuron output may be regarded as the external output of the network.

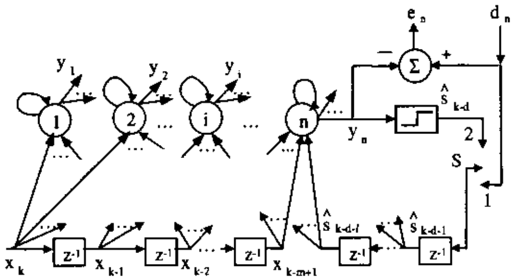


Fig. 3 Structure of S-DFRNNE

Fig.3 is DFRNNE structure with serial outputs (S-DFRNNE). We define that n, m and l are the numbers of inner nodes, the delaying inputs and the feedback delaying inputs of S-DFRNNE.

The weighted sum of the pth node's inputs is

$$v_p(t+1) = \sum_{i=1}^{n} w_{pi}\, y_i(t) + \sum_{i=1}^{m} w_{p,n+i}\, x_{k-i+1}(t) + \sum_{i=1}^{l} w_{p,n+m+i}\, \hat{s}_{k-d-1} \quad p=1,\ 2,\ \ldots,\ n \qquad (3)$$

where x_{k-i+1} (i=1, 2, ..., n) represents the delaying input signal, y_i the output of the ith (i=1, 2, ..., n) node, $\hat{s}_{k-d-i}$ (d is the channel delay) the feedback delaying signal, w_{pi} the weight from the ith (i=1, 2, ..., n) node to the pth node, w_{pj} the weight from each delaying input signal (j=n+1, ..., n+m) to the pth node, and w_{ph} the weight from each decision feedback delaying signal (h=n+m+1, ..., n+m+l) to the pth node.

Let

$$y_i(t+1) = f[v_{i+1}],\quad i=1,\ 2,\ \ldots,\ n \qquad (4)$$

where $f(x)$ is the active function, and

$$f(x) = \frac{1-\exp(-2x)}{1+\exp(-2x)} \qquad (5)$$

and

$$\hat{s}_{k-d} = SGN(\ y_n(t+1)) \qquad (6)$$

In training process (switch S points to 1) training signal is regarded as the delaying input of each decision feedback signal so that effective information can be taken full use and false propagation be prevented. When signals propagate (switch S points to 2), equalizer's decision output $SGN(\ y_n)$ becomes delaying feedback input.

Fig. 4 is DFRNNE structure with parallel outputs (P-DFRNNE), whose formulas are the same as those of S-DFRNNE except $v_p(t+1)$. The $v_p(t+1)$ of P-DFRNNE is expressed as

$$v_p(t+1) = \sum_{i=1}^{n} w_{pi}\, y_i(t) + \sum_{i=1}^{m} w_{p,n+i}\, x_{k-i+1}(t) + \sum_{i=1}^{l} w_{p,n+m+i}\, \hat{s}_{k-d-1,i} \qquad (7)$$

where $\hat{s}_{k-d-1,i}$ is the decision feedback input of the ith node with d+1 time delay (i=1, 2, ..., n) .

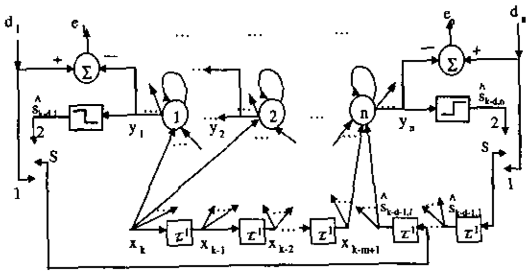


Fig. 4 Structure of P-DFRNNE

Ⅲ. Learning Algorithms of DFRNNE

3.1 Modified RTRL Algorithms

Real-time recurrent learning (RTRL) algorithm is modified for adjusting the equalizers weights of the two structures, and the algorithm expressions for each of which are a little different. d_h (h=1, 2, ..., n) is the expected response or training signal, and error of the hth neuron is defined as

$$e_h(t+1) = d_h(t+1) - y_h(t+1) \quad h=1,\ 2,\ \ldots,\ n \tag{8}$$

The networks instantaneous total error is given by

$$J(t+1) = \frac{1}{2}\sum_{h=1}^{n} e_h^2(t+1) \tag{9}$$

The objective of algorithm updating the connecting weight w_{ij} is to minimize $J(t+1)$.

Case 1: S-DFRNNE:

Define

$$\delta_{i,p} = \begin{cases} 1 & i=p \\ 0 & i \neq p \end{cases} \tag{10}$$

and the derivative of the active function

$$f'(x) = \frac{4}{[\exp(x)+\exp(-x)]} \tag{11}$$

The sensitivity is defined as

$$p_{ij}^{p}(t+1) = f'(v_p(t+1))\left[\sum_{k=1}^{n} w_{pk}(t)\, p_{ij}^{k}(t) + \delta_{ip}\, z_j(t)\right] \tag{12}$$

$$w_{ij}(t+1) = w_{ij}(t) + \alpha\sum_{k=1}^{n} e_k(t+1)\, p_{ij}^{k}(t+1) + u(w_{ij}(t) - w_{ij}(t-1))\ \mathrm{n} \tag{13}$$

where α is the learning step of the adaptive equalizer. $\alpha>0$; p=1, 2, ..., n; i=1, 2, ..., n; j=1, 2, ..., n, n+1, ..., n+m, n+m+1, ..., n+m+*l*. $z_j(t)$ represents the output of S-DFRNNE inner nodes (j=1, 2, ..., n), the external input signal (j=n+1, ..., n+m) or feedback input signal (j=n+m+1, ..., n+m+*l*). Moreover, momentum factor u (0<u<0.001) is introduced to help updating the weights.

Case 2: P-DFRNNE:

Its expressions are the same as S-DFRNNE except j which is defined to be 1, 2, ..., n, n+1, ..., n+m, n+m+1, ..., n+m+n.

3.2 Adaptive adjusting of the learning step

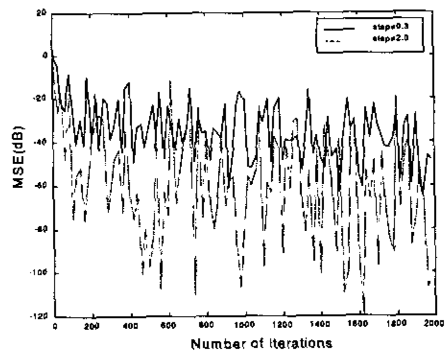


Fig. 5 The RNNE learning curves with different α (LCH, SNR=22dB)

The learning step α has certain effect on the convergency speed of algorithm. Fig.5 is RNNEs learning curves with different α where MSE and SNR represent mean square error and signal-to-noise ratio, respectively. As shown in Fig. 5, within certain range of values, the convergency speed of algorithm increases with the increase of α. For larger α, the algorithm converges quickly, but is subject to vibration and instability; For smaller α, the algorithm converges slowly, and is subject to trapping into local minimization. So the selection of α is very important. The methods of adaptive adjusting α are proposed according to the above analysis. Its basic idea is to continuously adjust α after each iteration, which possesses two objectives: one is to let algorithm skip out of the local minimization and speed up the convergency process, the other is to try to avoid the instability of algorithm.

Firstly define the system's total error E_n which is equal to $\sum J(t+1)$. Use a exponential function to realize the adaptive adjusting of α. This function regards E_n as independent variable. Set

$$\alpha = \alpha_0 \exp(-E_n) \ (0.1 < \alpha_0 < 1.5) \tag{14}$$

The algorithm will adaptively adjust α during the process of the iterations. If the total error is large, α will decrease; if the total error is small, α will increase. The total error will become small with the increase of the iterations, and α gradually holds a certain level.

Ⅳ. Simulation Study

In accordance with S-DFRNNE, the simulative experiments are done to compare it with the traditional RNNE under the same condition.

4.1 Comparison of learning performance

In every experiment, initial weights w_{ij}, initial sensitivity p_{ij}^{k} and initial output y_i of each node are random numbers whose absolute values are less than or equal to 10^{-3}. The parameters of traditional RNNE and S-DFRNNE are respectively m=2, n=1, $\alpha=0.5$ and m=2, n=1, l=3, $\alpha_0=0.5$. Therefore, two groups of learning curves are obtained in accordance with two different distortion channels.

In Fig.6, the distortion channel is LCH, SNR=16dB. In Fig.7, the distortion channel is LCH cascade-connected with NLCH, SNR =18dB. According to Fig.6, S-DFRNNE converges at about -30 dB, while RNNE at about -10dB with greater vibration. According to Fig.7, S-DFRNNE converges at about -30 dB, while RNNE doesnt converge at all. In short, S-DFRNN has better learning performance than the traditional RNNE.

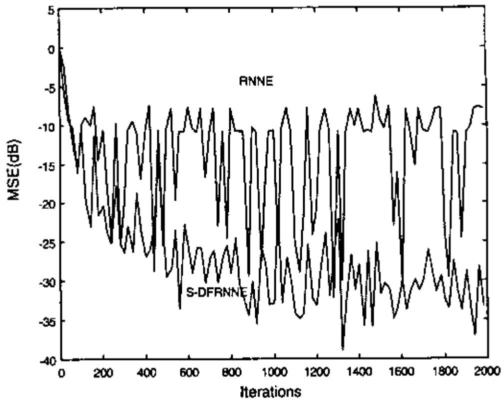


Fig. 6 Learning curves (LCH)

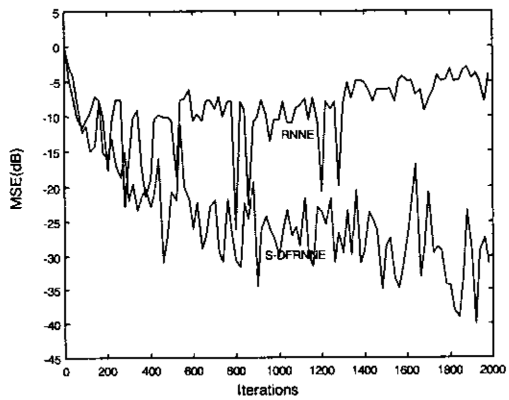


Fig. 7 Learning curves (LCH+NLCH)

4.2 Comparison of Bit Error Rate (BER)

In order to research the BER performance of S-DFRNNE, two groups of BER curves (Fig.8 and Fig.9) are obtained according to above distortion channels.

According to these curves, we note that S-DFRNNE has better BER performance than RNNE. During the simulation process, we find that, for the two given distortion channels, the BER of RNNE always presents inconsistency and possesses certain vibration, and that the BER performances are even deteriorated with the increase of SNR, which is the fault of RNNE applied to channel equalization, while S-DFRNNE makes up for the fault and its BER possesses consistency.

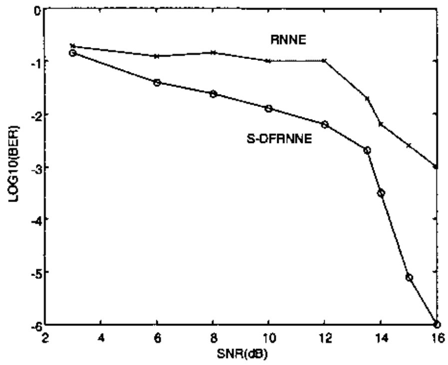


Fig. 8 BER curves (LCH)

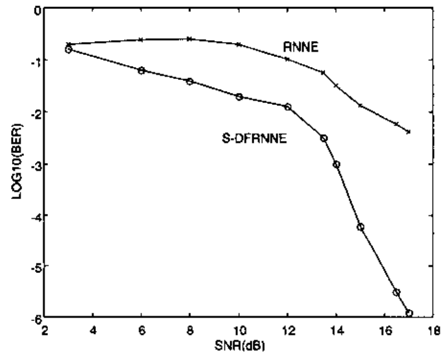


Fig. 9 BER curves (LCH+NLCH)

V. Conclusion

RNNE is very sensitive to the setting of various initial values. How to select initial values has still no conclusion, which is the existent problem for most of neural networks adaptive equalizers, especially for RNNE. However, S-DFRNNE is easy to be real-time disposed since it is not sensitive to various initial values. In adaptive training, it only needs to set any initial value as smaller random data and properly select adaptive step, which will gain better equalization effects that are sufficiently supported by the consistency of the learning and BER curves.

From the above discussion we can draw a conclusion that S-DFRNNE used for non-linear channels with severe ISI and non-linear distortion has better and more stable equalization properties. It not only possesses the advantage of only a few adjusting parameters, but also exerts the predominance of the decision feedback structures that can refrain from ISI. And it is a feasible scheme for the project implementation. In addition, the structure and algorithm of P-DFRNNE are proposed in the paper. P-DFRNNE can produce signals in the output end in parallel and has the approximate same hardware cost as that of S-DFRNNE, which can save time and improve the availability of hardware. In conclusion, both S-DFRNNE and P-DFRNNE are preferentially selected for adaptive equalizers in practical application.

References

[1] G. J. Gibson, S. Siu and C. F. N. Cowan, "Application of multilayer perceptrons as adaptive channel equalizers," *Proc. IEEE Internat. Conf. Acoust. Speech Signal Process, Glasgow, Scotland*, pp. 1183-1186, May 1989.

[2] Inhyok Cha and Saleem A. Kassam, "Channel equalization using adaptive complex radial basis function network," *IEEE Journal on SAC*, vol.13, No. 1, pp. 121-131, Jan. 1995.

[3] M. J. Bradley and P. Mars, "Application of recurrent neural networks to communication channel equalization," *IEEE Trans. on Neural Networks*, vol. 5, No. 2, March 1994.

[4] S. Chen, B. Mulgrew, and S. McLaughlin, "Adaptive Bayesian equalizer with decision feedback," *IEEE Trans. Signal Processing*, vol. 41, No. 9, pp. 2918-2927, September 1993.

[5] S. Sju, G. J. Gibson, and C. F. N. Cowan, "Multilayer perception structures and performance comparison with the standard architecture," *IEEE Proceedings Part I*, vol. 137, No. 4, pp. 221-225, August 1990.

[6] He Peiyu and Wang Yongde, "A neural network equalizer with self-adjusting structure for non-linear channels," *Journal of Sichuan University*, vol. 33, No. 4, pp. 405-409, Aug. 1996.

Hongrui Jiang

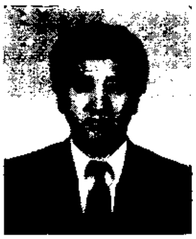
1993~1997 : B.S., Dept. of Microwave Telecommunication Engineering, Xidian University, Xian City, China

1997~2001 : M.S., majoring in Processing of Signal & Information, Dept. of Electronic information, Guilin Institute of Electronic Technology, Guilin City, China

2001~present : Ph.D. student, majoring in Electronic Telecommunication Engineering, Dept. of Electronic Engineering, Inha University, Korea

<Main Interesting fields> Neural Networks, Telecommunication and so on

곽 경 섭(Kyung-Sup Kwak)



1977년 2월 : 인하대학교 전기공학과 졸업(공학사)

1981년 12월: 미국 University of Southern California 공학석사

1988년 2월 : 미국 University of Southern California 공학박사 (통신이론 및 시스템)

1988년 2월~1989년 2월 : 미국 Hughes Network Systems 연구원

1989년 2월~1990년 3월 : 미국 IBM Network Analysis Center 연구원

1996년 3월~현재 : 인하대학교 정보통신공학부 교수

1999년 1월~현재 : 통신학회 상임이사

2000년 3월~현재 : 정보통신전문대학원 원장

<주관심 분야> 위성 및 이동 통신, 멀티미디어 통신