Channel Equalization using Fuzzy-ARTMAP Neural Network

Jung-Sik Lee* *Regular Member*, Jin-Hee Kim* *Associate Member*

ABSTRACT

This paper studies the application of a fuzzy-ARTMAP neural network to digital communications channel equalization. This approach provides new solutions for solving the problems, such as complexity and long training, which found when implementing the previously developed neural-basis equalizers. The proposed fuzzy-ARTMAP equalizer is fast and easy to train and includes capabilities not found in other neural network approaches; a small number of parameters, no requirements for the choice of initial weights, automatic increase of hidden units, no risk of getting trapped in local minima, and the capability of adding new data without retrainig previously trained data. In simulation studies, binary signals were generated at random in a linear channel with Gaussian noise. The performance of the proposed equalizer is compared with other neural net basis equalizers, specifically MLP and RBF equalizers.

I. Introduction

In digital communication systems, data symbols are transmitted at regular intervals. But time dispersion caused by non-ideal channel frequency response characteristics, or by multipath transmission, may create intersymbol interference (ISI). To deal with ISI, many researchers have been concerned with applying neural networks, such as multilayerperceptron (MLP) and radial basis functions (RBF), to equalizers[1-6]. The basic idea of applying neural network to equalization comes from the fact that channel equalizer problems can be regarded as pattern classification (detection). Previous studies have shown that neural network based equalizers are superior to linear equalizers in handling the situation where the channel suffers from high levels of additive noise and highly nonlinear distortion. However, each of these networks internally has significant shortcomings. MLP equalizers typically require long training and are sensitive to the initial choice of network parameters (especially initial weights). Furthermore, MLP equalizers need to decide by trial and error how many hidden units are needed. Also, RBF equalizers are simple and fast to train, but usually require a large number of centers, which increases the complexity of computation. In addition, it is not easy to determine both the number and the location of centers required to train.

In this paper, a fuzzy-ARTMAP neural network is applied to digital communication channel equalization. The main purpose of the proposed equalizer is to overcome the obstacles in implementing the previously developed neural-basis equalizers above. Since the advent of the fuzzy-ARTMAP neural network, the application of fuzzy-ARTMAP has been discussed mainly as it applies to pattern recognition fields. Through this paper, we describe how fuzzy-ARTMAP is combined with channel equalization techniques. The essential idea of this approach comes from the outstanding rec-

* 군산대학교 전자정보공학부(leejs@kunsan.ac.kr , jhk0221@kunsan.ac.kr)
논문번호 : 020048-0128 , 접수일자 : 2002년 1월 29일

※ 이 논문은 군산대학교 두뇌한국21 사업에 의하여 일부 지원되었음.

ognition capabilities and simple architecture of fuzzy-ARTMAP. The proposed fuzzy-ARTMAP equalizers provide attractive characteristics that are not found in previously developed neural-basis equalizers; a small number of parameters, fast and easy training, no requirement for the choice of initial weights, automatic increase of hidden units, and capability of adding new data without retraining previous patterns.

Section II presents a brief summary of the fuzzy-ARTMAP network and learning mechanism. Section III gives the structure and learning procedure for the fuzzy ARTMAP equalizer. Simulation results are provided in Section IV, and Section V gives the conclusions.

Ⅱ. Background of the Fuzzy-ARTMAP Neural Network

Since the advent of ART (adaptive resonance theory) as a cognitive and neural theory[7], a number of ART neural network architectures have been progressively developed. These models include ART2, ART3, fuzzy-ART, ARTMAP[9-12]. ART networks require that the input vectors be binary. ART2 networks are suitable for processing analog patterns. On the otherhand, ARTMAP is a class of neural network that performs incremental supervised learning of the categories for recognition.

Recently, a growing number of models computationally synthesize properties of neural networks, and fuzzy logic. Fuzzy-ARTMAP is one such model, combined with ARTMAP and fuzzy logic[8]. Fuzzy-ARTMAP is a neural network architecture that performs incremental supervised learning recognition categories and multi-dimensional maps in response to arbitrary sequences of analog or binary input vectors, and learns to classify inputs by a fuzzy set of features or a pattern of fuzzy membership values between 0 and 1 indicating the extent to which each feature is present. A schematic diagram of fuzzy-ARTMAP network is shown is Fig. 1.

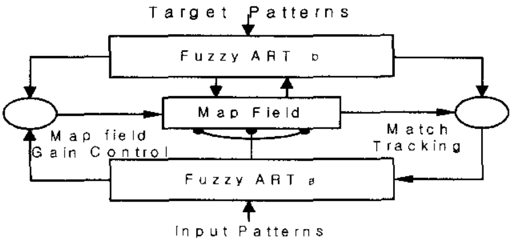

Fig.1 Fuzzy-ARTMAP structure block diagram

Fuzzy-ARTMAP utilizes a minimax learning rule that conjointly minimizes prediction error and maximizes generalization. As learning proceeds, the input and stored prototype of a category are said to resonate when they are sufficiently similar. When an input pattern is not sufficiently similar to any existing prototype, a new node (or hidden unit) is created to represent a new category with the input patterns as the prototype. The meaning of similarity depends on a vigilance parameter ρ, with $0 < \rho \leq 1$. If ρ is small, the similarity condition is easier to meet, resulting in a coarse categorization. On the other hand, if ρ is set close to 1, many finely divided categories are formed. As a result, a fuzzy-ARTMAP system increases the network architectures (number of clusters)to the minimum level necessary for perfect performance on the training data. By selecting the desired level for the vigilance parameter, the user has control over the performance of the work. Details of the fuzzy-ARTMAP network are given in [8].

Ⅲ. Implementation of the Fuzzy-ARTMAP Equalizer

In order to train a neural network to serve as a channel equalizer, it is necessary to generate appropriate training data. In this study, the network is trained to reconstruct the original signal(1 or -1) based on the signal received after transmission over a dispersive channel, as shown in Fig. 2.

Therefore, input patterns for the network consist of received signals and the corresponding target is the original transmitted signal. The channel is characterized by its transfer function, which in general has the form

$$H(Z) = \sum_{n=0}^{p} h_n Z^{-n} , \qquad (1)$$

where p is the channel order. If q denotes the equalizer order (number of tap delay elements in the equalizer), then there are $M = 2^{p+q+1}$ different sequences

$$A_k = [a_k, a_{k-1}, \ldots, a_{k-p-q}]^T \qquad (2)$$

that may be received (where each component is either 1 or -1). For a specific channel order and equalizer order, the required number of training patterns is M .

If pure training patterns were available , *they could be used directly, but if* fuzzy-ARTMAP is to be trained with noisy signals, preprocessing is necessary to prevent the network from learning the noise. In this study, the situation of a noisy transmission path is simulated by adding Gaussian noise to the received signal after each possible transmission sequence is passed through the transfer function. The training patterns are generated by applying the supervised K-means clustering algorithm [2] to remove the Gaussian noise:

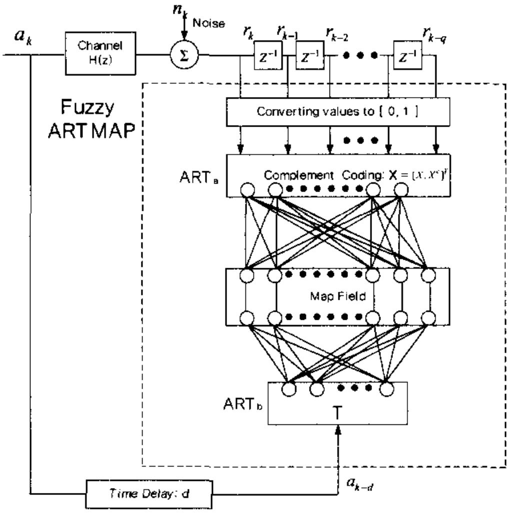

Fig.2 The structure of fuzzy-ARTMAP equalizer system

Algorithm:
{ Supervised K-means clustering}

```
if($\mathbf{A}_k = \mathbf{A}_i$){
    $counter_i = counter_i + 1;$
    $\mathbf{D}_i = \frac{(counter_i - 1)\cdot \mathbf{D}_i + \mathbf{R}_k}{counter_i};$
}
```

where

$$\left.\begin{aligned} &\mathbf{D}_i = [d_{i0}, d_{i1}, \ldots, d_{iq}]^T, \\ &\mathbf{R}_k = [r_k, r_{k-1}, \ldots, r_{k-q}]^T, \\ &i = 1, 2, \ldots, M, \end{aligned}\right\} \qquad (3)$$

and $\mathbf{A}_i$, $\mathbf{R}_k$, and $\mathbf{D}_i$ are the combination of $\mathbf{A}_k$, the received signal vectors, and a training pattern, respectively. As shown in (3), the number of components in the $\mathbf{R}_k$ is $q+1$. To facilitate the graphical *representation of the network input* vectors, the example given below are limited to equalizer order $q+1$, so that the input vectors have two components.

The training patterns which come from the transfer function(either directly or after the noise removal) have components that are not in the correct range for fuzzy-ARTMAP. The actual range depends on the transfer function; however, the binary sigmoid

$$\frac{1}{1+e^{-\alpha x}} \qquad (4)$$

converts the interval [-n, n] to [0,1], and thus is suitable for making the required conversion for any transfer function. The final input vectors after converting and complement coding procedures are

$$\mathbf{X}_i = [\mathbf{x}_i,\ \mathbf{x}_i^c]^T,\ i = 1,2,\ldots,M \qquad (5)$$

where

$$\mathbf{x}_i = [x_{i0}, x_{i1}, \ldots, x_{iq}], \qquad (6)$$

$$\mathbf{x}_i^c = [1 - x_{i0}, 1 - x_{i1}, \ldots, 1 - x_{iq}], \qquad (7)$$

$$x_{ij} = \frac{1}{1+\exp(-\alpha \cdot d_{ij})},\ j = 0,1,\ldots,q \qquad (8)$$

The target value for each generated training pattern is the correct value for a_{k-d} for the desired delay, d. The appropriate value of d is determined by the dominant term in the transfer function. A target value of 1 is represented by the vector (1,0); the target value of 1 is given by the vector(0,1). To generate training patterns for a given channel input matrix and noisy channel output vector, it is necessary to estimate the channel order. This is done using regression analysis [3]. For details of training algorithms, readers are referred to see the paper [8].

Ⅳ. Simulation Results

The fuzzy-ARTMAP equalizer was applied to several linear channels with different transfer functions. Among the favorable characteristics of this network is the fact that there are relatively few network parameters to be determined. The steepness of the sigmoid function(α in (4)) used to convert the input patterns into the required interval(0, 1) and the vigilance parameter for the networks must be set by the user. The network is not particularly sensitive to the values of either of these parameters. Sigmoid steepness parameter values in the range (0.7, 1.0) were used. The value of the vigilance influences the number of clusters formed (as is well known), but fuzzy-ARTMAP networks increase the vigilance, if required, to ensure that the training data are learned perfectly. The value of the vigilance has a more pronounced effect on the performance of the network after training, since the network will reject as unknown any input that is not sufficiently similar (based on the vigilance value) to the training patterns.

In Fig. 3, there are 16 patterns that are estimated from the noisy received signals by using the supervised K-means clustering. As shown in Fig. 3, high vigilance results in more clusters, and the error rate performance with high vigilance was better than with low vigilance. The vigilance value also affects the region which each cluster will accept when the net is applied (after training). The cluster will accept all points that fall within the boundaries shown, and also points somewhat

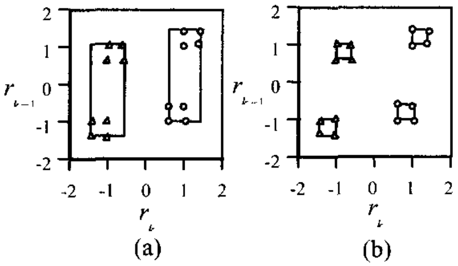

Fig. 3 Comparison of the number of clusters $H(Z) = 0.2 + 1.0Z^{-1} + 0.2Z^{-2}$, $q = 1$, $d = 1$
circles : patterns with $a_{k-1} = 1$,
triangles : patterns with $a_{k-1} = -1$
(a) vigilance parameter = 0.7, 2 clusters
(b) vigilance parameter = 0.85, 4 clusters

outside the dotted rectangle region.

A comparison of the performance of the fuzzy - ARTMAP equalizer to that of a linear equalizer, and two other neural network equalizers is illustrated in Fig. 4 for the transfer function

$$H(Z)=0.5+1.0Z^{-1} \qquad (9)$$

with $q=1$ and $d=0$. As shown in Fig. 4(a), the nonlinear decision boundaries cannot be achieved by the linear equalizer. The response regions for the radial basis function network and the fuzzy-ARTMAP equalizer are similar. In the case of the MLP equalizer, the number of units used in input, hidden, and output layers were two, eight, and one, respectively. The RBF equalizer uses eight centers which are estimated from supervised K-means clustering. For the fuzzy-ARTMAP equalizer, the vigilance, and sigmoid steepness parameters were 0.95, and 0.6 respectively.

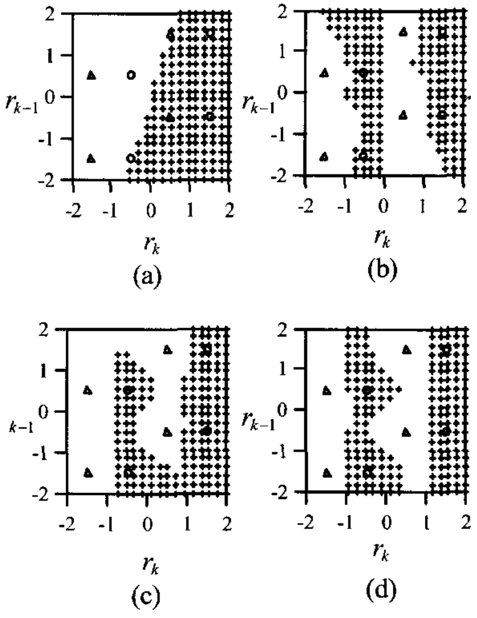

Fig. 4 Comparison of nonlinear decision boundary :
$H(Z)=0.5+1.0Z^{-1}$, $q=1$, $d=0$
circles : patterns with $a_k=1$,
triangles : patterns with $a_k=-1$
near equalizer (b) MLP equalizer
F equalizer(d)Fuzzy ARTMAP equalizerer

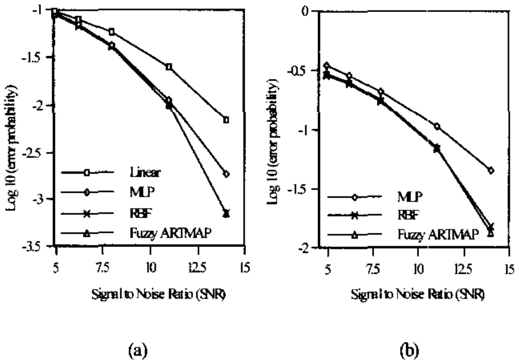

Fig. 5 Error rate comparison
$H(Z)=0.5+1.0Z^{-1}$, $q=1$
(a) $d=1$ (desired symbol = a_{k-1})
(b) $d=0$ (desired symbol = a_k)

Fig. 5 shows the error rate comparison of one linear and three kinds of neural network equalizers over two different channel delays that were introduced in training. Fig. 5(b) shows worse performance than in Fig. 5(a), because, for

non-minimum phase channel in (9), training without delay element (i.e., $d=0$) results in non-linear decision boundary. As shown in the graph, the performance of the fuzzy-ARTMAP equalizer is superior to that of both the linear and MLP equalizers, while producing results as favorable as those in RBF equalizer. Although the performance of the RBF equalizer is almost the same as that of the fuzzy-ARTMAP equalizer, the fuzzy-ARTMAP equalizer is a more attractive candidate than the RBF equalizer considering the cost and efforts required in neural network implementation. Here, MLP and RBF equalizers use eight hidden units, or centers, while the fuzzy-ARTMAP equalizer requires four clusters for a linear decision boundary case. In Table 1, the training efforts of the fuzzy-ARTMAP equalizer are compared with that of both MLP and RBF equalizers.

The training speed of the fuzzy-ARTMAP equalizer over the MLP or RBF equalizer can be much faster as the order of channel impulse response increases.

V. Conclusion

In this paper, a new fuzzy-ARTMAP equalizer system is developed mainly for solving the problems of long time of training and complexity which are often encountered in previously developed neural-basis equalizers such as MLP and RBF equalizers. The fuzzy-ARTMAP equalizer is fast and easy to train and includes capabilities not found in other neural network approaches; a small number of parameters, no requirements for the choice of initial weights, automatic increase of hidden units, no risk of getting trapped in local minima, and capability of adding new data without retraining previously trained data. Throughoutthe simulation studies, it was found that the fuzzy-ARTMAP equalizer performed favorably better than the MLP equalizer, while requiring relatively smaller computation steps in training. The main advantage of the proposed fuzzy-ARTMAP is fast training due to the structural simplicity of fuzzy-ARTMAP. The superiority of fuzzy-ARTMAP to other neural networks makes the implementations of fuzzy-ARTMAP equalizer feasible.

Table 1:Training efforts comparison between fuzzy-ARTMAP, RBF, and MLP equalizers

	fuzzy -ARTMAP	RBF	MLP
Performance sensitivity for initial weights	not sensitive	not sensitive	sensitive
The way of determining number of hidden units (cluster units)	automatic increase by itself during the training period	trial and error	trial and error
Number of hidden units required in training	4	8	8
Training speed(training time for MLP is set to 1)	approx. 1/5	approx. 1/3	1

REFERENCES

[1] 이정식, "RBF 신경망 등화기 운용에 있어서의 센터수 축소화 알고리즘", *한국통신학회* 논문지, 제24권, 제4B호, pp. 771-776, Apr., 1999.

[2] J. Lee, C. D. Beach, and N. Teped-elenlioglu, "A Practical Radial Basis Function Equalizer," *IEEE Trans. Neural Networks*, vol. 10, pp. 450-455, Mar. 1999.

[3] 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.

[4] J. Lee, C. D. Beach, and N. Teped-elenlioglu, "Channel Equalization using Radial Basis Function Network," *ICASSP*, Atlanta, Georgia, vol. 3, pp. 1719-1722, 1996.

[5] 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, 1991.

[6] 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, pp. 107-119, 1990.

[7] S. Grossberg, "Adaptive Pattern Classification and Universal Recording: I. Parallel Development and Coding of Neural Feature Detectors," *Biological Cybernetics*, pp. 121-134, 1976.

[8] G. A. Carpenter, S. Grossberg, N. Markuzon, J.H. Reynolds, and D.B. Rosen, "Fuzzy ARTMAP: A Neural Network Architecture for Incremental Supervised Learning of Analog Multidimensional Maps," *IEEE Trans. Neural Networks*, vol. 3, pp. 698-713, Sep. 1992.

[9] G. A. Carpenter and S. Grossberg, "ART2: Stable Self-organization of Pattern Recognition Codes for Analog Input Patterns," *Applied Optics*, vol. 26, pp. 4919-4930,1987.

[10] G. A. Carpenter and S. Grossberg, "ART3: Hierarchial Search Using Chemical Transmitters in Self-Organizing Pattern Recognition Architectures," *Neural Networks*, vol. 3,pp. 129-152, 1990.

[11]G. A. Carpenter, S. Grossberga, and D. B. Rosen, "Fuzzy-ART: Fast Stable Learning and Categorization of Analog Patterns by an Adaptive Resonance System," *Neural Networks*, vol. 4, pp. 759-771, 1991.

[12]G. A. Carpenter, S. Grossberg, J. H. Reynolds, "ARTMAP: Supervised Real-Time Learning and Classification of Non-stationary Data by a Self-Organizing Neural Network," *Neural Networks*, vol. 4,pp. 565-588, 1991.

이 정 식(Jung-Sik Lee) 정회원

1983년 : 한양대학교 전자통신과 공학사
1990년 : Florida Institute of Tech., Elec. Eng. M.S.
1996년 : Florida Institute of Tech., Elec. Eng. Ph.D.
1986년-1988년 : OPC 중앙연구소 연구원
1997년-현재:군산대학교 전자정보공학부조교수

<주관심분야>적응신호처리,디지털 통신, 신경망 응용

김 진 희(Jin-Hee Kim) 준회원

2001년 2월 : 군산대학교 정보통신공학과 공학사
2001년 3월～현재 : 군산대학교 전자정보공학부 석사과정

<주관심분야> 디지털통신, 신호처리