Electronics and Communications in Japan, Part 3, Vol. 77, No. 1, 1994
Translated from Denshi Joho Tsushin Gakkai Ronbunshi, Vol. 76-A, No. 8, August 1993, pp. 1136-1143

A Fast Learning Algorithm for Neural Networks and Its Applications to Adaptive Equalizers

Teruyuki Miyajima, Takaaki Hasegawa, and Misao Haneishi, *Members*

Faculty of Engineering, Saitama University, Urawa, Japan 338

SUMMARY

This paper proposes a fast learning algorithm of neural networks and evaluates the performances of adaptive equalizers using neural networks trained by the proposed algorithm in a frequency-selective fading channel. The backpropagation (BP) algorithm which is used widely to train neural networks has a slow convergence rate because it is based on the gradient descent method.

This paper presents a fast learning algorithm using the recursive least squares (RLS) algorithm which has a fast convergence rate as an adaptive algorithm for adaptive linear filters. In the proposed algorithm, the sum of the squared error between the actual total input and the desired total input is used as the cost function to apply the RLS algorithm. A simulation result on the exclusive-OR problem indicates that the proposed algorithm is about 8.8 times faster than the BP for the number of iterations required to converge.

Recently, there has been interest in adaptive equalizers as an application field of neural networks. However, the performance of an adaptive equalizer using a neural network in a frequency-selective fading channel which is observed in land mobile communications has never been evaluated. Therefore, in this paper, the performances of adaptive equalizers using neural networks trained by the proposed algorithm in a frequency-selective fading channel are evaluated. Especially, an adaptive equalizer using the selectively unsupervised learning neural network proposed by the authors is considered. The adaptive equalizer can reject the false learning by carrying out learning selectively. It is shown that the adaptive equalizer is superior to the conventional one and the one using the conventional neural network.

Key words: Neural network; learning algorithm; objective function; adaptive equalizer; frequency selective fading channel.

1. Introduction

Neural networks which consist of a large number of nonlinear processing units acquire capabilities of high information processing by learning. Many good results have been obtained in many fields such as pattern classification [1]. The backpropagation (BP) algorithm [2] is used widely as a learning algorithm. However, the BP has a drawback in its slow convergence rate because it is based on the gradient descent method.

Many learning algorithms have been proposed to improve the convergence rate. These include: a dynamical learning rate adaptation [3]; the use of nonlinear optimization methods; a modification of the cost function [5]; an initialization method of the connection weights [6]; and reduction of neurons in the hidden layer [7].

ISSN1042-0967/94/0001-0075

Recently, learning algorithms using the recursive least squares (RLS) algorithm which has fast convergence rate as an adaptive algorithm for adaptive linear filters have been proposed independently and it has been reported that they have a fast convergence rate [8-10]. To apply the RLS algorithm, a piecewise-linear function instead of a sigmoid function is employed for the nonlinear output function in [8] or the desired total input is estimated and the sum of the squared error of the total input of each neuron is then minimized in [9] and [10].

In this paper, we propose a novel learning algorithm of neural networks using the RLS algorithm. In the proposed algorithm, the weighted sum of the squared error between the actual total input and the desired total input (it can be obtained by using the inverse function of the output function) is used to apply the RLS algorithm. It is shown that the proposed algorithm can considerably reduce the number of iterations required to converge for the exclusive-OR problem.

Next, an adaptive equalization is considered as an application which needs a small number of iterations to converge. It was reported that, if an adaptive channel equalization problem in digital communications is regarded as a pattern classification problem, the adaptive equalization of the channel with the intersymbol interference is not a linear separable problem [11]. The adaptive equalizer using a neural network is superior to the conventional adaptive equalizer employing a linear transversal filter because the neural network can separate patterns which are not linearly separable [11].

In land mobile communications where adaptive equalizers have been developed, compensation for variations of a channel is studied [12]. Thus, the performance of the adaptive equalizer using a neural network must be evaluated in a time-varying channel.

So far, some reports have been made concerning the performance evaluations of the adaptive equalizers using neural networks in time-varying channels [13, 14]. However, the channel characteristics considered in these reports are not realistic and the frequency-selective fading channel observed in land mobile communications has never been considered.

In this paper, a neural network trained by the learning algorithm proposed in the first half of this paper is applied to an adaptive decision feedback equalizer (DFE) and its performance in a frequency-selective fading channel is evaluated.

On the other hand, we have already proposed the selectively unsupervised learning neural network (SULNN) [15]. After the learning by the known learning data, the SULNN can track changes of data in a nonstationary environment by creating the teacher signals based on the outputs of the network and carrying out learning. Although false teacher signals are created when the decisions are incorrect due to noise, and so on, the SULNN can reject the false learning by carrying out learning selectively. Therefore, the SULNN is suitable for a pattern classification in a nonstationary environment with noise. In practice, the detection system using the SULNN is superior to the conventional detection system in an electrical power line spread spectrum communication in the case of appliances changing [16].

In this paper, the DFE using the SULNN is especially considered and its effectiveness is shown.

2. Proposal of Fast Learning Algorithm

2.1. Neural networks

Consider a neural network which consists of L layers. The total input $y_i^{(l)}(t)$ and output $x_i^{(l)}(t)$ of the i-th neuron in the l-th layer at time t are given as follows:

$$y_i^{(l)}(t)=\sum_{j=0}^{N_{l-1}} w_{ij}^{(l)}(t-1)x_j^{(l-1)}(t) \tag{1}$$

$$x_i^{(l)}(t)=f(y_i^{(l)}(t)) \tag{2}$$

where N_{l-1} is the number of neurons in the l - 1st layer, $w_{ij}^{(l)}(t)$ is a connection weight from the j-th neuron in the l - 1st layer to the i-th neuron in the l-th layer at time t and $f(\cdot)$ is the nonlinear output function. The constant for the threshold is always 1 and is expressed as $x_0^{(l)}(t)$. The output of a neuron in the input layer ($l = 1$) is the input of the neuron.

2.2. Derivation of learning algorithm

The cost function is defined as the following equation and it is minimized:

$$E(t)=\frac{1}{2}\sum_{n=0}^{t}\lambda^{t-n}\sum_{i=1}^{N_L}e_i^{(L)}(n,t)^2 \tag{3}$$

where

$$e_i^{(L)}(n,t)=d_i^{(L)}(n)-y_i^{(L)}(n,t) \tag{4}$$

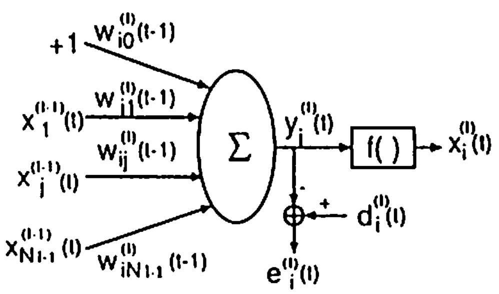


Fig. 1. Neuron model.

The forgetting factor is expressed as λ $(0 < \lambda \leq 1)$ and is used to track variations of data in a nonstationary environment. The actual total input and the error at the total input of the i-th neuron in the output layer of the network with connection weights at time t in case of the input signal at time n are expressed as $y_i^{(L)}(n, t)$ and $e_i^{(L)}(n, t)$, respectively. The desired total input of the i-th neuron in the output layer at time n is expressed as $d_i^{(L)}(n)$, and it can be obtained by using the inverse function of the output function. Taking the partial derivative of Eq. (3) with respect to $w_{ij}^{(l)}(t)$ and setting it equal to zero

$$\sum_{n=0}^{t} \lambda^{t-n} e_i^{(l)}(n, t) x_j^{(l-1)}(n, t) = 0 \tag{5}$$

where $x_j^{(l-1)}(n, t)$ is the output of the j-th neuron in the l - 1st layer of the network with connection weights at time t for the input signal at time n. The error signal $e_i^{(l)}(n, t)$ is given by Eq. (4) for the output layer and is given by the following equation for the l-th hidden layer:

$$e_i^{(l)}(n, t) = f'(y_i^{(l)}(n, t)) \sum_{k=1}^{N_{l+1}} w_{ki}^{(l+1)}(t) e_k^{(l+1)}(n, t) \tag{6}$$

As shown in Fig. 1, $e_i^{(l)}(n, t)$ in a hidden layer is regarded as the error between the desired and actual total input similarly to that in the output layer:

$$e_i^{(l)}(n, t) = d_i^{(l)}(n) - \sum_{k=0}^{N_{l-1}} w_{ik}^{(l)}(t) x_k^{(l-1)}(n, t) \tag{7}$$

Substituting Eq. (7) in Eq. (5) yields a normal equation [17]. A recursive learning algorithm can be obtained from the normal equation by using the RLS algorithm [17].

The resulting learning algorithm is summarized as follows.

Step 1. Initialize.

The connection weights $w_{ij}^{(l)}(0)$ are initialized to small random values. The $((N_{l-1} + 1) \times (N_{l-1} + 1))$ correlation matrices $P^{(l)}(0)$ are initialized to identity matrices.

Step 2. Calculate output signals.

Calculate the total input $y_i^{(l)}(t)$ and output $x_i^{(l)}(t)$ using Steps (1) and (2).

Step 3. Update the Kalman gain and the correlation matrices.

Update the Kalman gain $K^{(l)}(t)$ and the correlation matrices $P^{(l)}(t)$ using the following equations:

$$K^{(l)}(t) = \frac{P^{(l)}(t-1) X^{(l-1)}(t)}{\lambda + X^{(l-1)T}(t) P^{(l)}(t-1) X^{(l-1)}(t)} \tag{8}$$

$$P^{(l)}(t) = \frac{1}{\lambda}\left(P^{(l)}(t-1) - K^{(l)}(t) X^{(l-1)T}(t) P^{(l)}(t-1)\right) \tag{9}$$

where $X^{(l-1)}(t)$ is a vector whose component is the output of the neuron in the l - 1st layer $x_i^{(l-1)}(t)$ $(i \in [0, N_{l-1}])$, and the superscript T denotes transposition.

Step 4. Calculate error signals.

For the output layer, calculate the error signal by

$$e_i^{(L)}(t) = d_i^{(L)}(t) - y_i^{(L)}(t) \tag{10}$$

where

$$d_i^{(L)}(t) = f^{-1}(t_i(t)) \tag{11}$$

where $t_i(t)$ is the desired output of the i-th neuron in the output layer. For the $l(l \in [2, L - 1])$-th hidden layer, calculate the error signal by

$$e_i^{(l)}(t) = f'(y_i^{(l)}(t)) \sum_{j=1}^{N_{l+1}} w_{ji}^{(l+1)}(t-1) e_j^{(l+1)}(t) \tag{12}$$

Step 5. Update the connection weights.

Update the connection weight vectors from l - 1st layer to the i-th neuron in the l-th layer by

$$W_i^{(l)}(t) = W_i^{(l)}(t-1) + K^{(l)}(t) e_i^{(l)}(t) \tag{13}$$

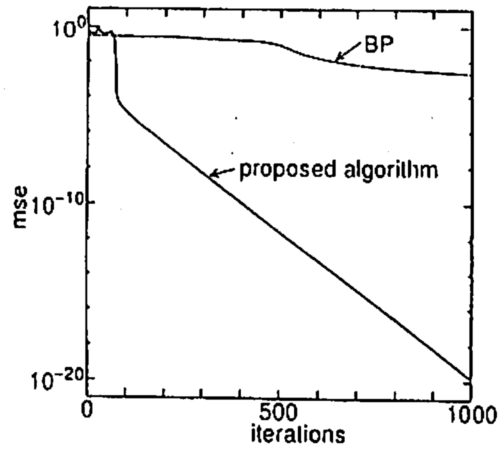


Fig. 2. Learning curves of XOR.

Repeat Step 2 to Step 5 until the error converges to within an acceptable range.

3. Behavior of Fast Learning Algorithm

Now the performance of the learning algorithm proposed in section 2 is evaluated via computer simulations. The performance of the proposed algorithm is compared with that of the BP. The exclusive-OR problem [2] was considered. The network has two neurons in the input layer, two neurons in the hidden layer, and a single neuron in the output layer. The weights were initialized to random values distributed uniformly between −0.5 and 0.5. The initial weight values of the proposed algorithm and BP were the same. A thousand trials with different initial values were carried out. The four learning patterns were represented in the same order.

The forgetting factor of the proposed algorithm was tested every 0.05 from 0.5 to 0.95 and the learning rate and inertia rate of the BP were tested every 0.2 from 0.2 to 1.0. The parameters whose convergence probability was more than 80 percent and which provided the fastest convergence were employed. As a result, the forgetting factor, learning rate, and momentum rate were set to be 0.8, 1.0, and 0.8, respectively. The sigmoid function which is used widely for the exclusive-OR problem was used as the output function:

$$f(x)=\frac{1}{1+e^{-x}} \tag{14}$$

Table 1. XOR simulation results

	Proposed algorithm	BP
Number of average learning cycle	234	2057
Convergence rate [%]	98.9	81.0
CPU time per cycle (ms)	0.202	0.125

The inverse function of Eq. (14) is given by

$$f^{-1}(x)=\ln\frac{x}{1-x} \tag{15}$$

Since, if the desired output signal is set to be 0 or 1, Eq. (15) becomes infinity, the desired output signal was thus set to be 0.01 or 0.99.

An example of the learning curve is shown in Fig. 2. From this figure, one can observe that the mean-squared error of the proposed algorithm becomes small in fewer iterations in comparison with that of the BP.

Next, the number of iterations and CPU time required to converge are considered. The learning was terminated when the squared error fell below 0.001 four successive iterations. If the learning did not converge after 10,000 iterations, it was terminated.

The results are shown in Table 1. The proposed algorithm is about 8.8 times faster than the BP for the number of iterations required to converge. The CPU time required for an iteration of the proposed algorithm is about 1.6 times as much as that of the BP. Hence, the proposed algorithm is about 5.5 times faster than the BP for the CPU time required to converge.

It is noted that a convergence of the proposed algorithm depends on the initial weight values as the BP and thus it is not guaranteed to converge to the global minimum of the cost function and there is a possibility to converge to a local minimum.

It is clear from Eqs. (8) to (13) that the proposed algorithm joins the error propagation of the BP with the RLS algorithm. Thus, if the number of inputs to a neuron is N, the computational complexity of the BP is proportional to N while that of the proposed algorithm is proportional to N^2 since the computational complexity which is required for the RLS algorithm [17] and is proportional to N^2 is dominant.

Therefore, if N is very large, the CPU time required to converge for the proposed algorithm will be very long. However, the small number of iterations is essentially needed in a system, such as an adaptive equalizer, where the learning data are presented at stated periods and it is necessary to track changes of the environment for a short time. In particular, the proposed algorithm will be useful in such a system. Thus, in section 4, an application of a neural network trained by the proposed algorithm to an adaptive equalizer is discussed.

4. Application to Adaptive Equalizer

4.1. Adaptive decision feedback equalizers

Recently, an adaptive decision feedback equalizer (DFE) which is a nonlinear equalizer is considered as a countermeasure of frequency-selective fading in land mobile communications [12]. The DFE consists of the fractionally tap-spaced (two taps/symbol) feedforward filter and the feedback filter whose inputs are previous decisions. Its tap coefficients are adjusted to minimize the mean-squared error between the equalizer output and the teacher (reference) signal by using the RLS algorithm, and so on. It is known that the DFE is simple and effective.

It is reported that a nonlinear decision boundary is needed to achieve the optimum classification of input patterns to a DFE when the feedback signals to the DFE are correct [18]. However, the conventional DFE creates only a linear decision boundary because the DFE uses a linear transversal filter for the feedforward filter. On the other hand, neural networks can create any nonlinear decision boundaries. It is known that the DFE using a neural network is superior to the conventional DFE in static frequency-selective channels [18, 19]. In this section, the performance of the DFE using a neural network trained by the algorithm proposed in section 2 in a frequency-selective fading channel is evaluated. In particular, the DFE using the SULNN [15] which is suitable for dynamical pattern classifications is considered.

4.2. DFE using SULNN

The structure of the DFE using the SULNN is shown in Fig. 3. Since the processing signals are complex-valued signals, there are neurons corresponding to

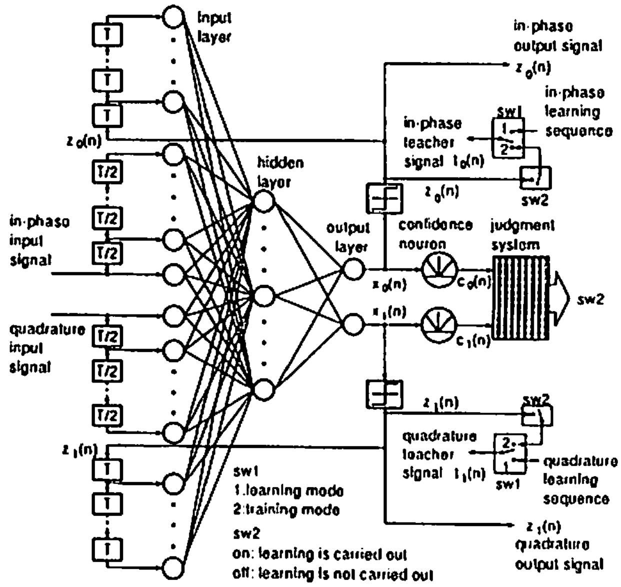


Fig. 3. Structure of the DFE using SULNN.

real and imaginary components in the input and output layer. The differences between the conventional neural network and the SULNN are the confidence neurons and the judgment system. These operations are described as follows.

Usually, a DFE operates in two modes, i.e., the learning mode and the tracking mode. In the learning mode, learning is carried out for the known learning data. In the tracking mode, for the unknown information data, the evaluation and the learning are carried out simultaneously to track variations of a channel by using the decision results as the teacher signals. This is referred to as the decision direction mode in which, if a decision is incorrect due to noise, etc., false learning is carried out by a false teacher signal and the performance will degrade.

The DFE using the SULNN rejects the false learning by selective learning in the tracking mode. A confidence neuron corresponds to a neuron in the output layer and its output is the absolute value of the output of the corresponding neuron in the output layer:

$$c_i = |x_i^{(L)}| \quad i = 0, 1 \tag{16}$$

where c_i is the output of a confidence neuron and is referred to as the confidence. The real and imaginary components of the output signal are represented by the subscripts $i = 0$ and $i = 1$, respectively. The judgment system determines whether learning is carried out based

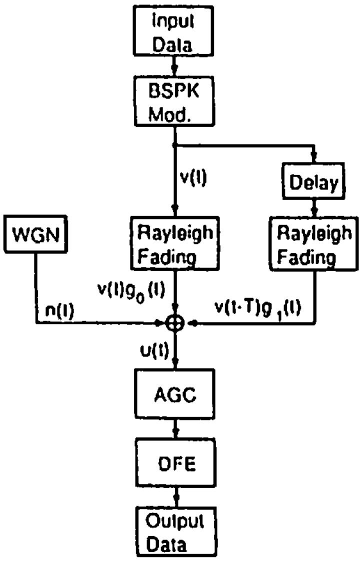


Fig. 4. Simulation model.

on the confidence values for the present input data. If the learning is carried out, the decision results are used as the teacher signals as in the decision direction mode. The judgment criterion depends on the modulation scheme employed. The judgment criterion for BPSK is described as follows:

$$\begin{cases} t_0 = z_0,\ t_1 = 0 & (c_0 > T_s) \\ \text{learning is not carried out} & (c_0 \leqq T_s) \end{cases} \tag{17}$$

where c_0 is the confidence for the real component of the equalizer output and T_s is a predetermined threshold and is referred to as the self-judgment threshold. The decision z_i is given by

$$z_i = \begin{cases} 1 & x_i^{(L)} > 0 \\ -1 & x_i^{(L)} \leqq 0 \end{cases} \tag{18}$$

The false learning is rejected by setting the self-judgment threshold appropriately.

Since BPSK is considered as the modulation scheme in this paper, the transmitted symbol takes a value {-1, 1}. Therefore, the sigmoid function which takes value (-1, 1) as the output function is employed:

$$f(x) = \frac{1-e^{-x}}{1+e^{-x}} \tag{19}$$

The inverse function of Eq. (19) is given by

Table 2. Components of simulation

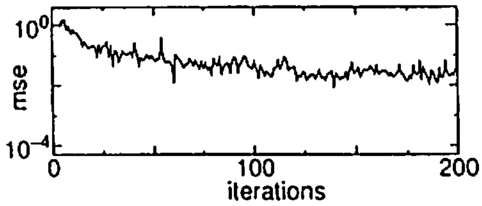
Length of learning data series	50 bit
Length of actual data series	500 bit
Modulation method	BPSK
Demodulation method	synchronous detection
Roll-off rate	1.0
Number of taps	FF 4
	FB 1
Tap space	FF $T/2$
	FB T
Number of neurons in input level	10
Number of neurons in hidden level	8
Number of neurons in output level	2
Forgetting factor (RLS)	0.98
Forgetting factor (proposed method)	0.99
BP learning rate	0.4
BP inertia rate	0.2
$f_D T$	1/12800
Delay time of delayed wave	T

FF: Feedforward
FB: Feedback

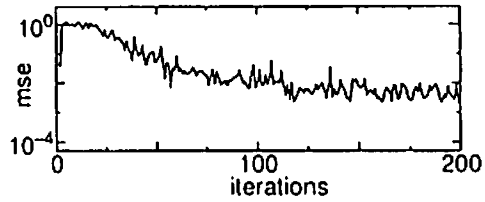
$$f^{-1}(x) = \ln \frac{1+x}{1-x} \tag{20}$$

Since the DFE considered uses the learning algorithm proposed in section 2, the teacher signal t_0 is ± 0.99 instead of ± 1.

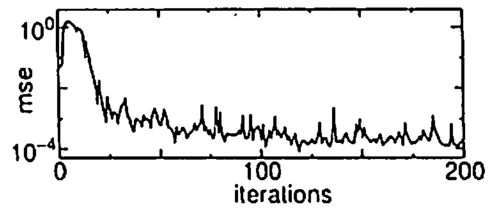
Since the sigmoid function (19) which takes value (-1, 1) is used as the output function, the confidence takes value (0, 1). Thus, when the self-judgment threshold is set to be 0, the learning is carried out for all the data and the DFE using the SULNN is equal to the DFE using the conventional neural network.



(a) conventional DFE



(b) BP



(c) proposed algorithm

Fig. 5. Learning curves.

4.3. Simulation model

An equivalent lowpass system is used in the simulation. The simulation model is shown in Fig. 4. Table 2 shows the simulation specification. The channel characteristic is a two-ray model which consists of the direct wave and the delayed one. Each wave has the same average power and is independently Rayleigh faded. The delay time of the delayed wave is set to be one symbol duration T. It is assumed that the estimation of carrier phase is obtained. The demodulated equivalent baseband signal at t is expressed as [20]

$$u(t)=v(t)g_0(t)+v(t-T)g_1(t)+n(t) \qquad (21)$$

where $v(t)$ is the value of the transmitted complex baseband signal, $n(t)$ is the additive white Gaussian noise expressed as an equivalent lowpass system and $g_i(t)$ is zero mean complex-valued Gaussian process and is band-limited to the maximum Doppler frequency $\pm f_D$.

The maximum Doppler frequency is expressed as f_D. The envelope of $g_i(t)$ is Rayleigh-distributed, and its phase is distributed uniformly.

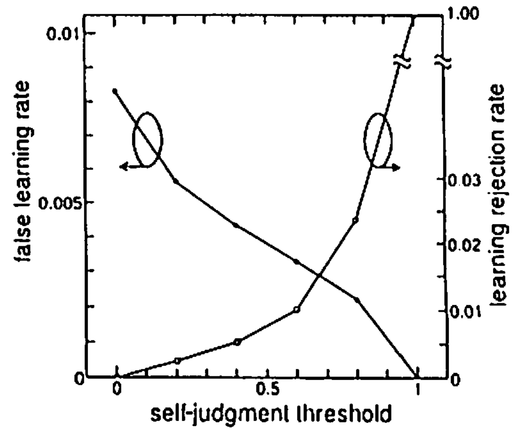


Fig. 6. The false learning rate and the learning rejection rate vs. self-judgment threshold.

In this paper, the condition where f_DT = 1/12,800 is considered. For example, this condition corresponds to f_D = 80 Hz when the transition rate is 1024 kbit/s and f_D = 80 Hz corresponds to a vehicle speed of 96 km/h when the carrier frequency is 900 MHz. The connection weights $w_{ij}^{(l)}$ and the correlation matrices $P^{(l)}$ are initialized in each burst to improve tracking ability.

As mentioned in section 3, there is a possibility to converge to a local minimum for the proposed algorithm. However, the performance degradation due to the convergence to a local minimum can be limited within a burst because the connection weights $w_{ij}^{(l)}$ are initialized in each burst.

4.4. Performance evaluation

First, the usefulness of a neural network and the convergence rate of the proposed algorithm are considered. Figure 5 depicts the learning curves where E_b/N_0 = 20 dB. The energy of the received signal including the delayed wave per bit is expressed as E_b, and N_0 describes the power spectral density of noise. It is assumed that the teacher signals are known and the feedback signals are correct. The conventional DFE is an equalizer whose tap coefficients are complex-valued and the RLS algorithm is employed for learning. The mean-squared error of the BP (b) and that of a neural network trained by the proposed algorithm are smaller than that of the conventional DFE (a). This is due to the nonlinearity of neural networks. Moreover, one can see that the proposed algorithm is faster than the BP and thus is suitable for a time-variant environment such as fading channels.

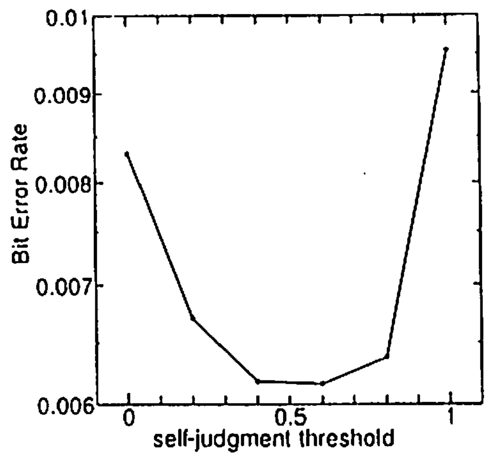


Fig. 7. Bit error rate vs. self-judgment threshold.

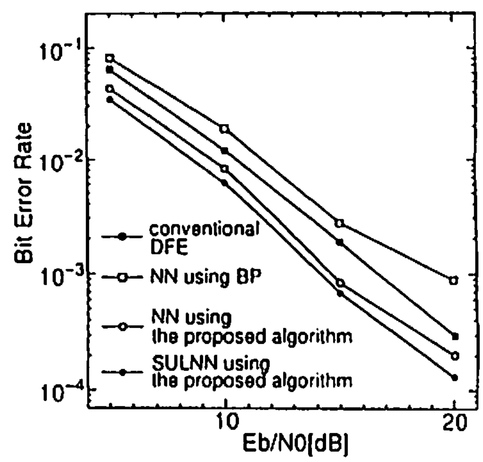


Fig. 8. Bit error rates.

Next, the DFE using the SULNN trained by the proposed algorithm is considered. As mentioned in section 4.2, the performances of the DFE depend on the self-judgment threshold. Here, we consider how the performances depend on the self-judgment threshold. The false learning rate and the learning rejection rate as a function of the self-judgment threshold are shown in Fig. 6, where E_b/N_0 = 10 dB. It is clear from this figure that the false learning rate decreases and the learning rejection rate increases as the self-judgment threshold becomes higher. If the self-judgment threshold is too low, the false learning rate becomes high and, as a result, the equalization performance degrades. On the other hand, if the self-judgment threshold is too high, the learning rejection rate becomes high and, as a result, it is difficult to track variations of the channel. Hence, the self-judgment threshold must be set appropriately between 0 and 1.

Figure 7 shows the bit error rate as a function of the self-judgment threshold. One can observe that the bit error rate of the DFE using the SULNN trained by the proposed algorithm is smaller than that of the DFE using the conventional neural network trained by the proposed algorithm (corresponding to T_s = 0) by setting the self-judgment threshold appropriately. The optimum value of the threshold is 0.6 in this figure. Although the optimum value depends on conditions such as E_b/N_0, sufficient results were obtained by the threshold of about 0.6. In the following simulation, the self-judgment threshold is set to be 0.6.

The bit error rate characteristics are shown in Fig. 8. Due to the fact that the DFE trained by the BP is not able to track variations of the channel because the BP has a slow convergence rate, the DFE provides the worst result. Due to the fact that the DFE using the conventional neural network trained by the proposed algorithm can track variations of the channel because of the fast convergence rate property of the proposed algorithm and the nonlinearity of the neural network is useful, the DFE is superior to the conventional DFE. The DFE using the SULNN trained by the proposed algorithm provides the best result because of the rejection of the false learning by selective learning in addition to the fast convergence rate property and the nonlinearity.

From the foregoing results, it was shown that the DFE using the SULNN trained by the proposed algorithm has the best performance and the DFE using the conventional neural network trained by the proposed algorithm has good performance. These techniques are useful in equalizing a frequency-selective fading channel.

5. Conclusions

In this paper, we propose a fast learning algorithm of neural networks using the RLS algorithm and consider an application to an adaptive decision feedback equalizer in a frequency-selective fading channel.

In the proposed algorithm, the sum of the squared error between the actual total input and the desired total input is employed for the cost function to apply the RLS algorithm which is useful as an adaptive algorithm for an adaptive linear filter. A simulation result on the exclusive-OR problem indicated that the proposed algorithm is about 8.8 times faster than the backpropagation algorithm for the number of iterations required to converge. The proposed algorithm will be useful in application fields where the small number of iterations is needed.

Moreover, the proposed algorithm was applied to the SULNN we proposed and the DFE using it was considered. The DFE can reject the false learning due to selective learning. The performances of various types of DFE using neural networks in a frequency-selective fading channel were evaluated. Consequently, it was shown that the DFE using the SULNN trained by the proposed algorithm is superior to the conventional DFE, the DFE using the conventional neural network trained by the BP and the DFE using the conventional neural network trained by the proposed algorithm.

Reduction of computational complexity of the proposed algorithm is a future problem. The performance evaluations of the DFE in various conditions where the delay time of the delay wave is longer or the maximum Doppler frequency is higher should be considered. Development of the optimization technique of the self-judgment threshold is the subject of current investigations.

Acknowledgement. The authors would like to thank to Prof. Y. Kobayashi of Saitama University for his helpful guidance.

REFERENCES

1. R. P. Lippmann. Pattern classification using neural networks. IEEE Commun. Mag., 27, p. 47-64 (1989).
2. D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representation by backpropagating errors. Nature, 323, pp. 533-536 (1986).
3. R. A. Jacobs. Increased Rates of Convergence Through Learning Rate Adaptation. Neural Networks, 1, pp. 295-307 (1988).
4. H. Takagi, S. Sakaue, and H. Togawa. Evaluation of nonlinear optimization methods for the learning algorithm of artificial neural networks. Trans. I.E.I.C.E., J74-D-II, 4, pp. 528-535 (1991).
5. Y. Tan, Y. Kato, and T. Ejima. Error Functions to Improve Learning Speed in PDP Models. Trans. I.E.I.C.E., J73-D-II, 12, pp. 2022-2028 (1990).
6. Q. Jia, N. Toda, and S. Usui. A study on initial value setting of the backpropagation learning algorithm. Trans. I.E.I.C.E., J73-D-II, 8, pp. 1179-1185 (1990).
7. M. Hagiwara. Backpropagation with artificial selection—Reduction of the number of learning times and that of hidden units. Trans. I.E.I.C.E., J74-D-II, 6, pp. 812-818 (1991).
8. M. R. Azimi-Sadjadi and R. Liou. Fast learning process of multilayer neural networks using recursive least-squares method. IEEE Trans. Signal Process., 40, 2, pp. 446-450 (1992).
9. R. S. Scalero and N. Tepedelenlioglu. A fast new algorithm for training feedforward neural networks. IEEE Trans. Signal Process., 40, 1, pp. 202-210 (1992).
10. F. Palmieri and S. A. Shah. Fast Training of Multilayer Perceptrons Using Multilinear Parametrization. Proc. IJCNN 1990, 1, pp. 696-699 (1990).
11. G. J. Gibson, S. Siu, and C. F. N. Cowan. Multilayer Perceptron Structures Applied to Adaptive Equalizers for Data Communications. Proc. ICASSP 1989, pp. 1183-1186 (1989).
12. S. Sampei. Development of Japanese adaptive equalizing technology toward high bit rate data transmission in land mobile communications. Trans. I.E.I.C.E., E74, 6, pp. 1512-1521 (1991).
13. T. Miyajima and T. Hasegawa. An Adaptive Decision Feedback Equalizer Using Selectively Unsupervised Learning Neural Network in Time Variant Channel. 1992 I.E.I.C.E. Spring Nat'l Conf., A-151 (1992).
14. R. Kaneda, T. Manabe, and S. Fujii. An Evaluation of the Adaptive Equalizer Using Forward-Only Counterpropagation Networks. 1992 I.E.I.C.E. Spring Nat'l Conf., A-150 (1992).
15. T. Miyajima, T. Hasegawa, Y. Hakura, and M. Haneishi. Performance of a selectively unsupervised learning neural network. Trans. I.E.I.C.E., J75-D-II, 1, pp. 137-145 (1992).
16. T. Daihoji and T. Hasegawa. Neural Network Detection Systems for Electric Power Line Spread-Spectrum Communications in Case of Appliances Changing. 1991 I.E.I.C.E. Autumn Nat'l Conf., A-94 (1991).
17. S. Haykin. Introduction to Adaptive Filters. Macmillan Publishing Company (1987).
18. S. Chen, G. J. Gibson, C. F. N. Cowan, and P. M. Grant. Reconstruction of binary signals using an adaptive radial-basis function equalizer. Signal Processing, 22, pp. 77-93 (1992).
19. S. Siu, G. J. Gibson, and C. F. N. Cowan. Decision feedback equalization using neural network structures and performance comparison with standard architecture. IEE Proc. I, 137, 4, pp. 221-225 (1990).
20. S. Sampei. Antifading techniques. Proc. I.E.I.C.E., 73, 8, pp. 829-835 (1990).

AUTHORS (from left to right)

Teruyuki Miyajima received a B.E. and an M.E. in 1989 and 1991, respectively, in Electrical Engineering from Saitama University, where he is currently in the doctoral program. His research interests are in neural networks and communication systems.

Takaaki Hasegawa received a B.E., an M.E., and a Ph.D. in Electrical Engineering from Keio University in 1981, 1983, and 1986, respectively. He joined the Faculty of Engineering, Saitama University, in 1986. Since 1991, he has been an Associate Professor. His research interests are in spread spectrum communications, signal processing, and human communications. Dr. Hasegawa is a member of IEEE and SITA of Japan.

Misao Haneishi received a B.S. in Electrical Engineering from Saitama University in 1967, an M.S. in Electrical Engineering from Tokyo Metropolitan University in 1969, and a Dr. of Eng. degree from Tokyo Institute of Technology in 1981. In 1969, he joined the Faculty of Engineering, Saitama University, and since then has been engaged in research on microwave antenna engineering. He is now a Professor of Electrical Engineering at Saitama University.