ITERATIVE WEIGHTED LEAST SQUARES ALGORITHMS FOR NEURAL NETWORKS CLASSIFIERS

Takio Kurita
Electrotechnical Laboratory
1-1-4 Umezono, Tsukuba, 305 Japan
E-mail: kurita@etl.go.jp

Abstract

This paper discusses learning algorithms of layered neural networks from the standpoint of maximum likelihood estimation. Fisher information is explicitly calculated for the network with only one neuron. It can be interpreted as a weighted covariance matrix of input vectors. A learning algorithm is presented on the basis of Fisher's scoring method. It is shown that the algorithm can be interpreted as iterations of weighted least square method. Then those results are extended to the layered network with one hidden layer. It is also shown that Fisher information is given as a weighted covariance matrix of inputs and outputs of hidden units for this network. Tow new algorithms are proposed by utilizing this information. It is experimentally shown that the algorithms converge with fewer iterations than usual BP algorithm. Especially UFS (unitwise Fisher's scoring) method reduces to the algorithm in which each unit estimates its own weights by a weighted least squares method.

1. Introduction

Feed-forward neural networks with error back-propagation learning algorithm (BP) [1, 2] have been successfully applied to many problems including pattern recognition, robotics and control, vision, image analysis, etc. Squared-error criterion is usually used as cost function and is minimized to determine the weights of the network by the steepest descent method. For classification of K classes, desired outputs are usually set to K-dimensional binary vectors in which one element is unity corresponding to the correct class and all others are zero. In this case, outputs of this network are interpreted as estimates of Bayesian *a posteriori* probabilities [3].

The most popular alternative cost function for pattern classifiers is the cross-entropy between actual outputs and desired outputs [3, 4, 5]. This is derived by the assumption that desired outputs are independent, binary, random variables, and that the actual network outputs represent the conditional probabilities that these binary, random variables are one. It can also be interpreted as minimizing the Kullback-Liebler probability distance measure, maximizing mutual information, or as maximum likelihood parameter estimation [4, 5, 6, 7]. This cost function has yielded similar error rates with squared-error in a phoneme classification experiments [8]. It is, however, demonstrated that number of iterations required to converge is

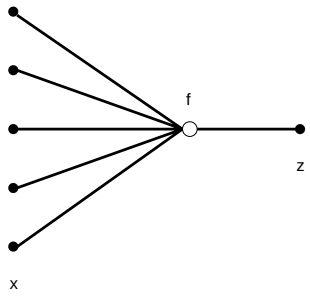


Figure 1. A neural network with only one unit.

fewer than squared-error criterion [9]. It is also shown that a cross-entropy cost function is minimized when outputs of the networks estimate Bayesian *a posteriori* probabilities [3]. This cost function was also used to determine the number of hidden units by information criteria [10].

This paper shows that iterative weighted least squares algorithms can be derived from the cross-entropy cost function by applying maximum likelihood estimation procedure. In section 2 we will study the network which consists of only one neuron from the view of maximum likelihood estimation in detail and show that Fisher information of the network is given as a weighted covariance matrix of inputs vectors [11]. A learning algorithm is presented on the basis of Fisher's scoring method, which is a kind of Newton-Raphson method and Fisher information is used instead of Hessian matrix. The algorithm can be interpreted as iterations of weighted least squares method [12]. Then we extend these results to layered neural networks with one hidden layer in section 3. Fisher information is also given as a weighted covariance matrix of inputs and outputs of hidden units. Tow new algorithms which utilize Fisher information are presented and are experimentally compared with BP algorithm.

2. The network with only one neuron

Here the networks which consists of only one neuron are studied from the view of maximum likelihood estimation. An example of the network with only one neuron is shown in Fig. 1.

2.1. Likelihood

Suppose that the input-output function of the neuron is logistic. Then the output z_p of the network is computed for an input $\boldsymbol{x}_p = (x_{p1}, \ldots, x_{pI})^T$ as

$$z_p = \frac{exp(\eta_p)}{1 + exp(\eta_p)}, \tag{1}$$

where $\eta_p = \sum_{i=1}^{I} a_i x_{pi}$ and a_i is the weight from the i-th input.

Let the set of learning samples be $\{< \boldsymbol{x}_p, t_p > | p = 1, \ldots, P\}$, where we assume that the teacher's signal t_p is given as binary (0 or 1). If we interpret the output z_p as an estimate of conditional probability given an input vector $\boldsymbol{x}_p$, then the log-likelihood of the network for the set of learning samples is given by

$$l = \sum_{p=1}^{P} \{t_p \ln z_p + (1 - t_p) \ln(1 - z_p)\}. \tag{2}$$

Thus the maximum likelihood estimate of the weights for the set of learning samples is computed as the one that maximizes this log-likelihood.

This can also be interpreted as minimizing the cross-entropy between actual outputs and desired outputs, minimizing Kullback-Liebler probability distance measure, or as maximizing mutual information [4, 5, 6, 7].

2.2. Fisher information

In the maximum likelihood estimation, Fisher information plays an important rule. Here we calculate Fisher information of the network with only one neuron explicitly.

The first and the second order derivatives of the log-likelihood (2) are given as

$$\begin{aligned} \frac{\partial l}{\partial a_i} &= \sum_{p=1}^{P} \delta_p x_{pi}, \\ \frac{\partial^2 l}{\partial a_l \partial a_i} &= -\sum_{p=1}^{P} \omega_p x_{pl} x_{pi}, \end{aligned} \tag{3}$$

where $\delta_p = t_p - z_p$ and $\omega_p = z_p(1 - z_p)$. In matrix form, we have

$$\begin{aligned} \nabla l &= \sum_{p=1}^{P} \delta_p \boldsymbol{x}_p = X^T \boldsymbol{\delta}, \\ \nabla^2 l &= -\sum_{p=1}^{P} \omega_p \boldsymbol{x}_p \boldsymbol{x}_p^T = -X^T W X \end{aligned} \tag{4}$$

where $X^T = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_P]$, $W = \mathrm{diag}(\omega_1, \ldots, \omega_P)$ and $\boldsymbol{\delta} = (\delta_1, \ldots, \delta_P)^T$.

Fisher information for the weights $\boldsymbol{a} = (a_1, \ldots, a_I)^T$, namely minus the expected value of the Hessian matrix $\nabla^2 l$, is given by

$$F = -E(\nabla^2 l) = X^T W X. \tag{5}$$

This means that Fisher information for the weights of the network is a weighted covariance matrix of input vectors $\{\boldsymbol{x}_p\}$. The weight of each sample is given by $\omega_p = z_p(1 - z_p)$ that has the maximum $\frac{1}{4}$ at $z_p = \frac{1}{2}$ and the minimum 0 at $z_p = 0$ or $z_p = 1$. This means that inputs whose output of the network is uncertain (z_p is near $\frac{1}{2}$) contribute more to Fisher information than those whose output is certain (z_p is near 0 or 1).

2.3. Iterative weighted least square algorithm

To obtain the weights which maximize the log-likelihood, it is necessary to have an optimization algorithm. Fisher's scoring method is a kind of Newton-Raphson method and Fisher information is used instead of Hessian matrix. For the network with only one neuron Fisher information is the same as Hessian except the sign. Therefor Fisher's scoring method and Newton-Raphson method reduce to the same algorithm [12].

Let the current estimate of the weights be $\boldsymbol{a}$. Then the estimate is repeatedly updated by $\boldsymbol{a}^* = \boldsymbol{a} + \delta \boldsymbol{a}$, where the increment $\delta \boldsymbol{a}$ is determined by solving the linear equation

$$F \delta \boldsymbol{a} = \nabla l. \tag{6}$$

Since $F\boldsymbol{a}$ is given by $F\boldsymbol{a} = X^T W \boldsymbol{\eta}$, the new estimate $\boldsymbol{a}^*$ can be obtained by solving the linear equation

$$X^T W X \boldsymbol{a}^* = X^T W \boldsymbol{\eta} + X^T \boldsymbol{\delta} = X^T W (\boldsymbol{\eta} + W^{-1} \boldsymbol{\delta}), \tag{7}$$

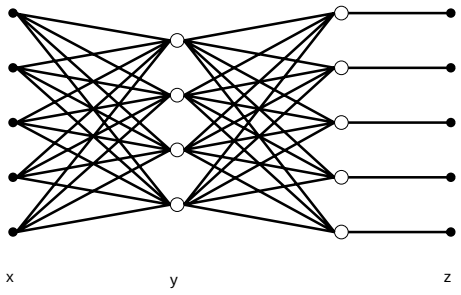


Figure 2. A layered neural network woth one hidden layer.

where $\boldsymbol{\eta} = (\eta_1, \ldots, \eta_P)^T$. This equation can be interpreted as the normal equation of a weighted least squares method to estimate $\boldsymbol{\eta} + W^{-1}\boldsymbol{\delta}$ from the input vectors X.

Thus the learning algorithm based on Fisher's scoring method for the network with only one neuron can be interpreted as iterations of a weighted least squares method.

From the equation (7) we can derive the following simple method to determine initial weights $\boldsymbol{a}_0$ of the iterative weighted least squares algorithm. Suppose that all weights are initially zero, namely $\boldsymbol{a} = \mathbf{o}$. Then $W = \frac{1}{4}I$, $\boldsymbol{\eta} = \mathbf{o}$, and $\boldsymbol{\delta} = \boldsymbol{t} - \frac{1}{2}\mathbf{1}$. Inserting these values into equation (7), we have the following equation

$$\boldsymbol{a}_0 = 4(X^T X)^{-1} X^T (\boldsymbol{t} - \frac{1}{2}\mathbf{1}). \tag{8}$$

This can be interpreted as linear regression that estimates the desired outputs $\boldsymbol{t} - \frac{1}{2}\mathbf{1}$ from the input vectors.

3. The network with one hidden layer

Next we extend the results for the network with only one neuron to layered networks with one hidden layer. An example of the network with one hidden layer is shown in Fig. 2.

3.1. Likelihood

For this network output vector $\boldsymbol{z}_p = (z_{p1}, \ldots, z_{pK})^T$ is computed for an input vector $\boldsymbol{x}_p = (x_{p1}, \ldots, x_{pI})^T$ as

$$y_{pj} = f(\zeta_{pj}) = \frac{exp(\zeta_{pj})}{1 + exp(\zeta_{pj})}, \quad \zeta_{pj} = \sum_{i=1}^{I} a_{ji} x_{pi}, \tag{9}$$

$$z_{pk} = f(\eta_{pk}) = \frac{exp(\eta_{pk})}{1 + exp(\eta_{pk})}, \quad \eta_{pk} = \sum_{j=1}^{J} b_{kj} y_{pj}, \tag{10}$$

where a_{ji} is the weight from the i-th input unit to the j-th hidden unit and b_{kj} is the weight from the j-th hidden unit to the k-th output unit.

Let the set of learning samples be $\{< \boldsymbol{x}_p, \boldsymbol{t}_p > | p = 1, \ldots, P\}$. We also assume that the teacher's vector $\boldsymbol{t}_p$ is given as K-dimensional binary vector in which one element is unity corresponding to the correct class and all others are zero. If we assume that the each element

of output vector of the network is conditionally independent, then the log-likelihood of the network for the set of learning samples is given by

$$l = \sum_{p=1}^{P} \sum_{k=1}^{K} \{t_{pk} \ln z_{pk} + (1 - t_{pk}) \ln(1 - z_{pk})\}. \tag{11}$$

Usually minus of this measure is called the cross-entropy cost function [3].

3.2. Fisher information

Here we calculate Fisher information for the network with one hidden layer explicitly.

The first and the second order derivatives of the log-likelihood (11) are given as

- The first order derivatives

$$\frac{\partial l}{\partial a_{ji}} = \sum_{p=1}^{P} \sigma_{pj} \nu_{pj} x_{pi}, \qquad \frac{\partial l}{\partial b_{kj}} = \sum_{p=1}^{P} \delta_{pk} y_{pj}, \tag{12}$$

  where $\sigma_{pj} = \sum_{k=1}^{K} \delta_{pk} b_{kj}$, $\nu_{pj} = y_{pj}(1 - y_{pj})$ and $\delta_{pk} = t_{pk} - z_{pk}$.

- The second order derivatives

$$\frac{\partial^2 l}{\partial a_{ml} \partial a_{ji}} = \begin{cases} \sum_{p=1}^{P} x_{pl} \nu_{pj} (1 - 2y_{pj}) \delta_p x_{pi} - \sum_{p=1}^{P} x_{pl} \nu_{pj} \chi_{pjj} \nu_{pj} x_{pi} & \text{if } m = j \\ -\sum_{p=1}^{P} x_{pl} \nu_{pm} \chi_{pmj} \nu_{pj} x_{pi} & \text{otherwise} \end{cases} \tag{13}$$

$$\frac{\partial^2 l}{\partial a_{ml} \partial b_{kj}} = \begin{cases} \sum_{p=1}^{P} x_{pl} \nu_{pj} \delta_{pk} - \sum_{p=1}^{P} x_{pl} \nu_{pj} b_{kj} \omega_{pk} y_{pj} & \text{if } m = j \\ -\sum_{p=1}^{P} x_{pl} \nu_{pm} b_{km} \omega_{pk} y_{pj} & \text{otherwise} \end{cases} \tag{14}$$

$$\frac{\partial^2 l}{\partial b_{nm} \partial a_{ji}} = \begin{cases} \sum_{p=1}^{P} x_{pi} \nu_{pj} \delta_{pn} - \sum_{p=1}^{P} y_{pj} \omega_{pn} b_{nj} \nu_{pj} x_{pi} & \text{if } m = j \\ -\sum_{p=1}^{P} y_{pm} \omega_{pn} b_{nj} \nu_{pj} x_{pi} & \text{otherwise} \end{cases} \tag{15}$$

$$\frac{\partial^2 l}{\partial b_{nm} \partial b_{kj}} = \begin{cases} -\sum_{p=1}^{P} y_{pm} \omega_{pk} y_{pj} & \text{if } n = k \\ 0 & \text{otherwise} \end{cases} \tag{16}$$

  where $\omega_{pk} = z_{pk}(1 - z_{pk})$, $\delta_p = \sum_{k=1}^{K} \delta_{pk}$ and $\chi_{pmj} = \sum_{k=1}^{K} b_{km} \omega_{pk} b_{kj}$.

From the relation on log-likelihood

$$0 = \mathrm{E}(\frac{\partial l_p}{\partial z_{pk}}) = \frac{\mathrm{E}(t_{pk}) - z_{pk}}{z_{pk}(1 - z_{pk})}, \tag{17}$$

Fisher information for the weights $A = [a_{ji}]$ and $B = [b_{kj}]$ is given by

$$F_{a_{ml} a_{ji}} = \sum_{p=1}^{P} x_{pl} \nu_{pm} \chi_{pmj} \nu_{pj} x_{pi}, \qquad F_{a_{ml} b_{kj}} = \sum_{p=1}^{P} x_{pl} \nu_{pm} b_{km} \omega_{pk} y_{pj} \tag{18}$$

$$F_{b_{nm} a_{ji}} = \sum_{p=1}^{P} y_{pm} \omega_{pn} b_{nj} \nu_{pj} x_{pi}, \qquad F_{b_{nm} b_{kj}} = \begin{cases} \sum_{p=1}^{P} y_{pm} \omega_{pk} y_{pj} & \text{if } n = k \\ 0 & \text{otherwise} \end{cases} \tag{19}$$

Thus Fisher information for the weights of the network with one hidden layer is given as a weighted covariance matrix of inputs $\{\boldsymbol{x}_p\}$ and outputs of hidden units $\{\boldsymbol{y}_p\}$.

3.3. Iterative weighted least squares algorithm

Next we consider learning algorithms of the weights which maximizes the log-likelihood (11).

One of the simplest optimization method is steepest decent method which use information on the first order derivatives of objective function. Let the current estimates of the weights be $\boldsymbol{\theta} = (a_{11}, \ldots, a_{JI}, b_{11}, \ldots, b_{KJ})^T$. Then the estimate is repeatedly updated by

$$\boldsymbol{\theta}^* = \boldsymbol{\theta} - \alpha \nabla l. \tag{20}$$

If we want to update the weights to each learning sample instead of the set of learning samples, this can be modified to

$$\boldsymbol{\theta}^* = \boldsymbol{\theta} - \alpha \nabla l_p, \tag{21}$$

where ∇l_p is the first order derivative to the sample $< \boldsymbol{x}_p, \boldsymbol{t}_p >$ and the relation $\nabla l = \sum_{p=1}^{P} \nabla l_p$ holds. In the following we call this SD (steepest decent) method.

Similar to the case of the neural networks with only one unit, we can develop Fisher's scoring algorithm which uses Fisher information. The estimate is repeatedly changed by

$$\boldsymbol{\theta}^* = \boldsymbol{\theta} + \delta\boldsymbol{\theta}, \tag{22}$$

where the increment $\delta\boldsymbol{\theta}$ is determined by solving the linear equation

$$F\delta\boldsymbol{\theta} = \nabla l. \tag{23}$$

In the following we call this FS (Fisher's scoring) method.

Since there is possibility that Fisher information matrix F becomes singular, in the subsequent experiments we used $F + \beta I$ instead of F itself, where β is a constant and I is the unit matrix.

3.4. Unitwise iterative weighted least squares algorithm

In FS method it is necessary to compute Fisher information F ($(IJ + JK) \times (IJ + JK)$) and solve the linear equation (23) with $IJ + JK$ unknown parameters at each step of learning. This is not easy for large networks.

Here we propose an algorithm which uses only block diagonal elements of Fisher information and neglects the other elements.

Fisher information related with the weights from input units to the unit j in the hidden layer is given by

$$F_{Aj} = \left[\sum_{p=1}^{P} x_{pl} \nu_{pj} \chi_{pjj} \nu_{pj} x_{pi} \right] = X^T W_{Aj} X, \tag{24}$$

where

$$X = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_P]^T \tag{25}$$

$$W_{Aj} = \mathrm{diag}(\nu_{pj} \chi_{pjj} \nu_{pj}). \tag{26}$$

This is a weighted covariance matrix of input vectors. From the equations (22) and (23), the normal equation of weighted least squares estimation to obtain the current estimates of the weights $\boldsymbol{\theta}^*_{Aj} = (a_{j1}, \ldots, a_{jI})^T$ is given by

$$X^T W_{AJ} X \boldsymbol{\theta}^*_{Aj} = X^T W_{AJ} (\boldsymbol{\zeta}_j + W_{Aj}^{-1} \boldsymbol{\delta}_{Aj}) \tag{27}$$

where

$$\boldsymbol{\zeta}_j = (\zeta_{1j}, \ldots, \zeta_{Pj})^T \tag{28}$$

$$\boldsymbol{\delta}_{Aj} = (\sigma_{1j}\nu_{1j}, \ldots, \sigma_{Pj}\nu_{Pj})^T. \tag{29}$$

Similarly, Fisher information related with the weights from hidden units to the unit k in the output layer and the corresponding normal equation are given by

$$F_{Bk} = \left[\sum_{p=1}^{P} y_{pm}\omega_{pk}y_{pj}\right] = Y^T W_{Bk} Y \tag{30}$$

and

$$Y^T W_{Bk} Y \boldsymbol{\theta}^*_{Bk} = Y^T W_{Bk}(\boldsymbol{\eta}_k + W_{Bk}^{-1}\boldsymbol{\delta}_{Bk}) \tag{31}$$

where

$$Y^T = [\boldsymbol{y}_1, \ldots, \boldsymbol{y}_P] \tag{32}$$

$$W_{Bk} = \mathrm{diag}(\omega_{pk}) \tag{33}$$

$$\boldsymbol{\eta}_k = (\eta_{1k}, \ldots, \eta_{Pk})^T \tag{34}$$

$$\boldsymbol{\delta}_{Bk} = (\omega_{1k}, \ldots, \omega_{Pj})^T. \tag{35}$$

The estimates $\{\boldsymbol{\theta}_{Aj} | j = 1, \ldots, J\}$ and $\{\boldsymbol{\theta}_{Bk} | k = 1, \ldots, K\}$ are repeatedly updated by solving these normal equations. We call this UFS (Unitwise Fisher's scoring) method. In this algorithm each unit of the network estimates its own weights by the iterative weighted least squares algorithm which is similar with the one derived for the network with only one neuron in section 2.3.

Next we will consider the recursive formulas to solve these normal equations for the weighted least squares. This reveals the close relation between the proposed UFS algorithm and the back-propagation learning algorithm.

Consider a set of earlier learning samples $\{< \boldsymbol{x}_p, \boldsymbol{t}_p > | p = 1, \ldots, N-1\}$ and a new learning sample $< \boldsymbol{x}_N, \boldsymbol{t}_N >$. Then one can estimate the parameters $\boldsymbol{\theta}_{Aj}^{(N)}$, which is optimal with respect to the set of earlier learning samples and the new learning samples, using the estimates $\boldsymbol{\theta}_{Aj}^{(N-1)}$ for the set of earlier learning samples. The recursive formula is given by

$$\boldsymbol{\theta}_{Aj}^{(N)} = \boldsymbol{\theta}_{AJ}^{(N-1)} + Q_{Aj}^{(N)} \boldsymbol{x}_N \sigma_{Nj}\nu_{Nj}, \tag{36}$$

where the matrix $Q_{Aj}^{(N)}$ gives the estimates of the inverse of the weighted covariance matrix ${}^{(N)}X^T\,{}^{(N)}W_{Aj}^{(N)}X$ and its recursive formula is given by

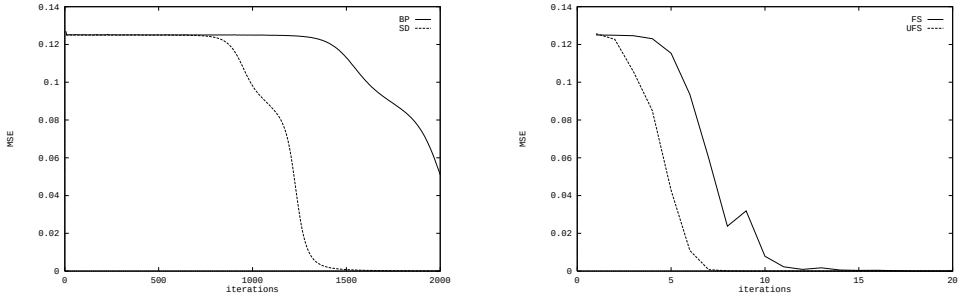
$$Q_{Aj}^{(N)} = Q_{Aj}^{(N-1)} - \frac{\nu_{Nj}\chi_{Njj}\nu_{Nj} Q_{Aj}^{(N-1)} \boldsymbol{x}_N \boldsymbol{x}_N^T Q_{Aj}^{(N-1)}}{1 + \nu_{Nj}\chi_{Njj}\nu_{Nj} \boldsymbol{x}_N^T Q_{Aj}^{(N-1)} \boldsymbol{x}_N}. \tag{37}$$

Similarly the recursive formula for the estimates $\boldsymbol{\theta}_{Bk}$ is given by

$$\boldsymbol{\theta}_{Bk}^{(N)} = \boldsymbol{\theta}_{Bk}^{(N-1)} + Q_{Bk}^{(N)} \boldsymbol{y}_N \delta_{Nj} \tag{38}$$

$$Q_{Bk}^{(N)} = Q_{Bk}^{(N-1)} - \frac{\omega_{Nj} Q_{Bk}^{(N-1)} \boldsymbol{y}_N \boldsymbol{y}_N^T Q_{Bk}^{(N-1)}}{1 + \omega_{Nj} \boldsymbol{y}_N^T Q_{Bk}^{(N-1)} \boldsymbol{y}_N}. \tag{39}$$

By comparing the equations (36) and (38) with the back-propagation algorithm in which the cross-entropy cost function is used as the cost function, one can notice that the estimates Q_{Aj} or Q_{Bk} of the inverse of the covariance matrices are used instead of the constant learning rate of the back-propagation algorithm.



(a) (b)

Figure 3. Results of learning of XOR problem.

4. Experiment

The learning algorithms described in this paper are compared with usual BP algorithm which uses square-error criterion.

4.1. XOR problem

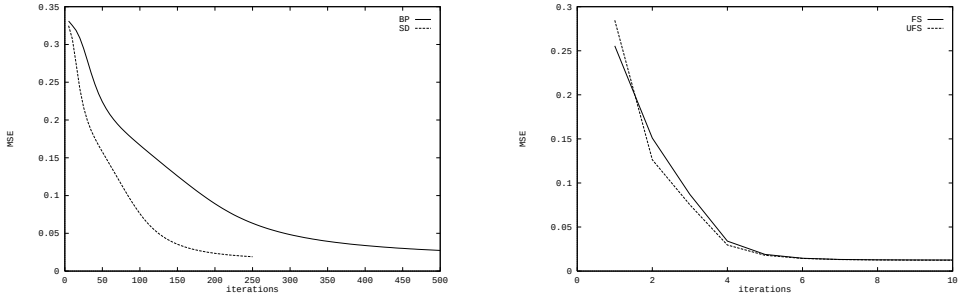
The problem used in this experiment is the XOR problem. The network has 2 inputs, 2 hidden units and 1 output unit. Initial weights of the network were randomly generated within the interval $[-0.5, 0.5]$. For all algorithms the same initial weights were used. In BP and SD methods, the learning rate α was set to 0.25. The results of learning are shown in Fig. 3 (a) and (b). It is noticed that SD is faster than BP for this problem. This result agrees with the result of Holt [9]. FS and UFS methods converged with less than 30 iteration for this problem.

4.2. Pattern Recognition Problem

Experiment for classification of Fisher's Iris data was also performed. In this case the network has 4 inputs, 2 hidden units and 3 output units. Each of input features was normalized to zero mean and unit variance. Initial weights of the network were also randomly generated. In BP and SD methods, the learning rate α was set to $\frac{1}{150}$. The results of learning are shown in Fig. 4 (a) and (b). It is noticed that results are similar to the XOR problem.

5. Conclusion

This paper discussed learning algorithms of layered neural networks from the standpoint of maximum likelihood estimation. For the network with only one neuron Fisher information was explicitly calculated. It can be interpreted as a weighted covariance matrix of input vectors. A learning algorithm was presented on the basis of Fisher's scoring method. The algorithm can be interpreted as iterations of weighted least square method. Then those results were extended to the network with one hidden layer. For this network Fisher information was also a weighted covariance matrix of inputs and outputs of hidden units. Tow new algorithms were proposed by utilizing this information. The algorithms converged with fewer iterations than usual BP algorithm. Especially UFS (unitwise Fisher's scoring) method reduces to the algorithm in which each unit estimates its own weights by a weighted least squares method. In



(a) (b)

Figure 4. Results of learning of a pattern recognition problem.

this paper we considered learning algorithms of neural networks for pattern classification. The same approach can be applied for neural networks for function approximation.

References

[1] Rumelhart,D.E., Hinton,G.E., and Williams,R.J. : Learning representations by back-propagating errors, Nature, Vol.323-9, pp.533-536 (1986).

[2] Rumelhart,D.E., Hinton,G.E., and Williams,R.J. : Learning internal representations by error propagation, in *Parallel Distributed Processing* Volume 1, McCleland,J.L., Rumelhart,D.E., and The PDP Research group, Cambridge, MA: MIT Press, 1986.

[3] Richard,M.D. and Lippmann,R.P. : Neural network classifiers estimate Bayesian *a posteriori* probabilities, Neural Computation, Vol.3, No.4, pp.461-483 (1991).

[4] Baum,E.B. and Wilczek,F. : Supervised learning of probability distributions by neural networks, In *Neural Information Processing Systems*, D.Anderson, ed.,pp.52-61. American Institute of Physics, New York (1988).

[5] Hinton,G.E. : Connectionist learning procedures, Artificial Intelligence 40, 185-234 (1989).

[6] Bridle,J.S. :Training stochastic model recognition algorithms as networks can lead to maximum mutual information estimation of parameters. In *Neural Information Processing Systems 2*, David S.Touretzky, ed., pp.211-217, Morgan Kaufmann (1990).

[7] Gish,H. : A probabilistic approach to the understanding and training of neural network classifiers. In Proceedings of IEEE Conference on Acoustics Speech and Signal Processing, pp.1361-1364 (1990).

[8] Hampshire,J.B. and Waibel,A.H. : A novel objective function for improved phoneme recognition using time-delay neural networks, IEEE Trans. on Neural Networks, Vol.1, No.2, pp.216-228 (1990).

[9] Holt,M.J.J. and Semanani,S. : Convergence of back propagation in neural networks using a log-likelihood cost function, Electronics Letters, Vol.26, No.23 (1990).

[10] Kurita,T. : A Method to Determine the Number of Hidden Units of Three Layered Neural Networks by Information Criteria, Trans. of IEICE Japan, J73-D-II, 1872-1878, 1990 (in Japanese).

[11] Kurita,T. : On Maximum Likelihood Estimation of Feed-Forward Neural Net Parameters, IEICE Tech. Report, NC91-36, 1991 (in Japanese).

[12] McCullagh,P. and Nelder FRS,J.A. : *Generalized Linear Models*, Chapman and Hall, 1989.