1953 Rev.

1999-3-11

Adaptive Method of Realizing Natural Gradient Learning for Multilayer Perceptrons

Shun-ichi Amari, Hyeyoung Park and Kenji Fukumizu
RIKEN Brain Science Institute
Wako-shi, Hirosawa, Saitama 351-0198, Japan
{amari, hypark, fuku}@brain.riken.go.jp

Abstract

The natural gradient learning method is known to have ideal performances for on-line training of multilayer perceptrons. It avoids plateaus which give rise to slow convergence of the backpropagation method. It is Fisher efficient whereas the conventional method is not. However, for implementing the method, it is necessary to calculate the Fisher information matrix and its inverse, which is practically very difficult. The present letter proposes an adaptive method of directly obtaining the inverse of the Fisher information matrix. It generalizes the adaptive Gauss-Newton algorithms and provides a solid theoretical justification to them. Simulations show that the proposed adaptive method works very well for realizing natural gradient learning.

1 Introduction

Natural gradient (Amari, 1998; Yang and Amari, 1998) gives an on-line learning algorithm which takes the Riemannian metric of the parameter space into account. The natural gradient method is based on information geometry (Amari, 1985; Amari and Nagaoka, 1999) and uses the Riemannian metric of the parameter space to define the steepest direction of a cost function. It may be regarded as a version of the stochastic descent method. It uses a matrix learning rate equal to the inverse of the Fisher information matrix, which plays the role of the Riemannian metric in the space of perceptrons. It overcomes two types of shortcomings involved in backpropagation learning, namely efficiency and slow convergence.

On-line learning may use a training example once when it is observed, while batch learning stores all the examples and any examples can be reused later. Therefore, it is in general true that the performance of on-line learning is worse than batch learning. This is true for the conventional backpropagation method. The paper (Amari, 1998) shows that natural gradient achieves Fisher efficiency. According to the Cramer-Rao bounds, this is the best asymptotic performance that any unbiased learning algorithm can achieve.

The backpropagation method is known to be very slow in convergence. This is because of plateaus in which the parameter is trapped in the process of learning, taking long time to get rid of them. Statistical-mechanical analysis (Saad and Solla, 1995) has made it clear that plateaus are ubiquitous in backpropagation learning, and various acceleration methods so far proposed cannot avoid or escape from plateaus. Amari (1998) and Yang and Amari (1998) suggested that the natural gradient algorithm has possibility of avoiding plateaus or quickly escaping from them. This has been theoretically confirmed (Rattray, Saad and Amari, 1998), where an ideal performance of the natural gradient method was theoretically demonstrated in the thermodynamical limit.

However, it is difficult to calculate the Fisher information matrix of multilayer perceptrons. Even when it is obtained, its inversion is computationally expensive. Yang and Amari (1998) gave an explicit form of the Fisher information matrix and the computational complexity of its inversion, by assuming that the distribution of input signals is Gaussian. Rattray, Saad and Amari (1998) gave it in terms of statistical-mechanical order parameters. These results show that it is difficult to implement natural gradient learning for practical large-scale problems, however excellent its performance is.

The present letter proposes an adaptive method of obtaining the inverse of the Fisher information matrix directly without any matrix inversion, by applying the Kalman filter technique. The proposed adaptive method generalizes the adaptive Gauss-Newton method (LeCun et al., 1998) and provides a solid theoretical justification based on different philosophical ideas. Computer experiments demonstrate that the proposed method has almost the same performance as the original natural gradient method and that its convergence speed is surprisingly faster than the conventional backpropagation method. This is a preliminary study, and more general performances of the proposed method will be reported in future by Park, Amari and Fukumizu (1999).

2 Natural Gradient for MLP

Let us consider a multilayer perceptron (MLP) which receives an n-dimensional input signal $\boldsymbol{x}$ and emits a scalar output signal y. When it has m hidden units and one linear output unit, its input-output behavior is written as

$$y = \sum_{\alpha=1}^{m} v_\alpha \varphi\left(\boldsymbol{w}_\alpha \cdot \boldsymbol{x} + b_\alpha\right) + b_0 + \xi \tag{1}$$

where $\boldsymbol{w}_\alpha$ is an n-dimensional connection weight vector from the input to the α-th hidden unit ($\alpha = 1, \cdots, m$); v_α is the connection weight from the α-th hidden unit to the output unit; b_α and b_0 are biases to the α-th hidden node and the output node, respectively; and ξ is a random noise subject to $N(0, \sigma^2)$. The function φ is a sigmoidal type activation function. The parameters $\{\boldsymbol{w}_1, \cdots, \boldsymbol{w}_m; \boldsymbol{b}; \boldsymbol{v}\}$ can be summarized into a single $\{m(n+2)+1\}$-dimensional vector $\boldsymbol{\theta}$. It is a stochastic network because of noise ξ. The set S of all such stochastic multilayer perceptrons forms a manifold where $\boldsymbol{\theta}$ plays the role of a coordinate system in the space S.

A multilayer perceptron having parameter $\boldsymbol{\theta}$ is connected with the conditional probability distribution of output y conditioned on input $\boldsymbol{x}$,

$$p(y|\boldsymbol{x}, \boldsymbol{\theta}) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{1}{2\sigma^2}\left\{y - f(\boldsymbol{x}, \boldsymbol{\theta})\right\}^2\right], \tag{2}$$

where

$$f(\boldsymbol{x}, \boldsymbol{\theta}) = \sum_{\alpha=1}^{m} v_\alpha \varphi\left(\boldsymbol{w}_\alpha \cdot \boldsymbol{x} + b_\alpha\right) + b_0 \tag{3}$$

is the mean value of y given input $\boldsymbol{x}$. The space S of multilayer perceptrons is identified with the set of all the conditional probability distributions of Eq. (2). Its logarithm is

$$l(y|\boldsymbol{x}, \boldsymbol{\theta}) = -\frac{1}{2\sigma^2}\left\{y - f(\boldsymbol{x}, \boldsymbol{\theta})\right\}^2 - \log(\sqrt{2\pi}\sigma) \tag{4}$$

and is called the log likelihood. This can be regarded as the negative of the square of an error when y is a target value given $\boldsymbol{x}$, except for a scale and a constant term. Hence, the maximization of the likelihood is equivalent to the minimization of the square error

$$l^*\left(y, \boldsymbol{x}, \boldsymbol{\theta}\right) = \frac{1}{2}\left\{y - f(\boldsymbol{x}, \boldsymbol{\theta})\right\}^2. \tag{5}$$

Let us consider training of a perceptron. Given a sequence of training examples $\{(\boldsymbol{x}_1, y_1^*),$

$(\boldsymbol{x}_2, y_2^*)\,, \cdots\}$, the conventional on-line backpropagation learning algorithm is written as

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta_t \nabla l^* \left(\boldsymbol{x}_t, y_t^*; \boldsymbol{\theta}_t\right), \tag{6}$$

where $\boldsymbol{\theta}_t$ is the network parameter at time t, η_t is a learning rate which may depend on t,

$$\nabla l^*(\boldsymbol{x}, y^*, \boldsymbol{\theta}) = \left\{ \frac{\partial}{\partial \theta_i} l^*(\boldsymbol{x}, y^*; \boldsymbol{\theta}) \right\} \tag{7}$$

is the gradient of the loss function l^* and y_t^* is the desired output signal given from the teacher.

The gradient ∇l^* is in general believed to be the steepest direction of scalar function of l^*. This is true only when the space S is Euclidean and $\boldsymbol{\theta}$ is an orthonormal coordinate system. In our case of stochastic perceptrons, the parameter space S has a Riemannian structure (Amari, 1998; Yang and Amari, 1998), so that the ordinary gradient does not represent the steepest direction of the loss function. The steepest descent direction of the loss function $l^*(\boldsymbol{\theta})$ in a Riemannian space is given (Amari, 1998) by

$$-\tilde{\nabla} l^*(\boldsymbol{\theta}) = -G^{-1}(\boldsymbol{\theta}) \nabla l^*(\boldsymbol{\theta}), \tag{8}$$

where G^{-1} is the inverse of a matrix $G = (g_{ij})$ called the Riemannian metric tensor. This gradient is called natural gradient of the loss function $l^*(\boldsymbol{\theta})$ in the Riemannian space, and it suggests the natural gradient descent algorithm of the form

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta_t \tilde{\nabla} l^*(\boldsymbol{\theta}) \tag{9}$$

(Amari, 1998; Edelman et al. 1998). In the case of stochastic multilayer perceptron, the Riemannian metric tensor $G(\boldsymbol{\theta}) = (g_{ij}(\boldsymbol{\theta}))$ is given by the Fisher information matrix (Amari, 1998),

$$g_{ij}(\boldsymbol{\theta}) = E\left[\frac{\partial l(y|\boldsymbol{x}; \boldsymbol{\theta})}{\partial \theta_i} \frac{\partial l(y|\boldsymbol{x}; \boldsymbol{\theta})}{\partial \theta_j} \right], \tag{10}$$

where E denotes expectation with respect to the input-output pair $(\boldsymbol{x}, y)$ given by Eq. (2). Note that we do not use the teacher signal y^* to define it. It was proved that the on-line learning method based on the natural gradient is asymptotically efficient as the optimal batch algorithm is. It was also suggested that natural gradient learning is much easier to get rid of plateaus than ordinary gradient learning (Amari, 1998; Yang and Amari, 1998). The statistical physical approach elucidated an ideal behavior of natural gradient learning (Rattray, Saad and Amari, 1998) avoiding plateaus.

2.1 Adaptive estimation of natural gradient

The Fisher information matrix of Eq. (10) at $\boldsymbol{\theta}_t$ can be rewritten, by using Eq. (4), as

$$\begin{aligned} G_t &= E\left[\frac{\partial l(y|\boldsymbol{x};\boldsymbol{\theta}_t)}{\partial \boldsymbol{\theta}_t}\frac{\partial l(y|\boldsymbol{x};\boldsymbol{\theta}_t)'}{\partial \boldsymbol{\theta}_t}\right] && (11)\\ &= \frac{1}{4\sigma^4}E\left[\{y - f(\boldsymbol{x};\boldsymbol{\theta}_t)\}^2\right]E\left[\frac{\partial f(\boldsymbol{x};\boldsymbol{\theta}_t)}{\partial \boldsymbol{\theta}_t}\frac{\partial f(\boldsymbol{x};\boldsymbol{\theta}_t)'}{\partial \boldsymbol{\theta}_t}\right] && (12)\\ &= \frac{1}{4\sigma^2}E\left[\frac{\partial f(\boldsymbol{x};\boldsymbol{\theta}_t)}{\partial \boldsymbol{\theta}_t}\frac{\partial f(\boldsymbol{x};\boldsymbol{\theta}_t)'}{\partial \boldsymbol{\theta}_t}\right] && (13)\end{aligned}$$

where $'$ denotes transposition of a vector or matrix. For the calculation of expectation, we have to know the probability distribution $q(\boldsymbol{x})$ of input $\boldsymbol{x}$. Such knowledge about the input distribution, however, is hardly given in practical problems. In addition, it is difficult to obtain an explicit form of G even if we know it. Moreover, inversion of G is computationally costful when the number m of the hidden units is large. All of these suggest that natural gradient learning is not practical.

In order to overcome these difficulties, we propose an adaptive method of directly obtaining an estimate of $G^{-1}(\boldsymbol{\theta}_t)$ which does not require costful inversion of G.

Since the Fisher information at $\boldsymbol{\theta}_t$ is obtained by the expectation over inputs $\boldsymbol{x}$ of $\nabla f(\nabla f)'$, we make use of the adaptive method of obtaining $\hat{G}_t$, which is an estimate of $G(\boldsymbol{\theta}_t)$, given by

$$\hat{G}_t = (1-\varepsilon_t)\,\hat{G}_{t-1} + \varepsilon_t \nabla f\left(\boldsymbol{x}_{t-1},\boldsymbol{\theta}_{t-1}\right) \nabla f\left(\boldsymbol{x}_{t-1},\boldsymbol{\theta}_{t-1}\right)', \tag{14}$$

where ε_t is a time dependent learning rate. Typical examples are $\varepsilon_t = c/t$ and $\varepsilon_t = \varepsilon$. When we put $\varepsilon_t = 1/t$, $\hat{G}_t$ is equal to the arithmetic mean of $\nabla f_i \nabla f_i'$ except for a scalar factor $(1/4\sigma^2)$, where $\nabla f_i = \nabla f\left(\boldsymbol{x}_i,\boldsymbol{\theta}_i\right)$, $i = 1,\cdots,t$. When $\boldsymbol{\theta}_i$ converges to $\boldsymbol{\theta}^*$, $\hat{G}_t$ converges to $G\left(\boldsymbol{\theta}^*\right)$.

Our purpose is to obtain $\hat{G}_t^{-1}$ directly. To this end, we use the well-known technique of the Kalman filter by applying the following identity which holds for a nonsingular symmetric $n \times n$ matrix A and an $n \times k$ matrix $B (k < n)$,

$$\left(\alpha A + \beta B B'\right)^{-1} = \frac{1}{\alpha}A^{-1} - \left(\frac{\sqrt{\beta}}{\alpha}A^{-1}B\right)\left(I + \frac{\beta}{\alpha}B'A^{-1}B\right)^{-1}\left(\frac{\sqrt{\beta}}{\alpha}A^{-1}B\right)', \tag{15}$$

where α and β are scalars (see Bottou, 1998). We apply the identity to the right-hand side of (14) for $A = \hat{G}_{t-1}$ and $B = \nabla f_t$. We then have the following adaptive algorithm to obtain an estimate of the inverse of Fisher information

$$\hat{G}_{t+1}^{-1} = \frac{1}{1-\varepsilon_t}\hat{G}_t^{-1} - \frac{\varepsilon_t}{(1-\varepsilon_t)} \frac{\hat{G}_t^{-1}\nabla f_t \left(\nabla f_t\right)' \hat{G}_t^{-1}}{(1-\varepsilon_t) + \varepsilon_t \nabla f_t' \hat{G}_t^{-1} \nabla f_t}. \tag{16}$$

When ε_t is small, we may approximate it by a simpler

$$\hat{G}_{t+1}^{-1} = (1+\varepsilon_t)\,\hat{G}_t^{-1} - \varepsilon_t \hat{G}_t^{-1} \nabla f_t \left(\nabla f_t\right)' \hat{G}_t^{-1}. \tag{17}$$

The related natural gradient learning algorithm is given by

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta_t \hat{G}_t^{-1} \nabla l\left(\boldsymbol{x}_t, y_t^*; \boldsymbol{\theta}_t\right). \tag{18}$$

Eqs. (17) and (18) are the new method which we propose to implement natural gradient learning.

Since we have introduced one more learning rate ε_t in addition to η_t, we need to study how to choose ε_t. We see that ε_t determines how quickly and how accurately, $\hat{G}_t^{-1}$ converges to $G^{-1}(\boldsymbol{\theta}_t)$. In the final stage where $\boldsymbol{\theta}_t$ is chose to $\boldsymbol{\theta}^*$, an error in G^{-1} is not serious since the estimator $\boldsymbol{\theta}_t$ is consistent even when G^{-1} is misspecified. Here, an adequate choice of η_t is much more important. There are lots of theories concerning determination of η_t. See, for example, Amari (1998) for its adaptive determination. Avoidance of plateaus is an important benefit of natural gradient. If ε_t is too small, there is a serious delay for adjusting $\hat{G}^{-1}$ so that the parameter might approach plateaus, even though it escapes eventually from them. On the other hand, there is numerical instability for large ε_t. All in all, we suggest to use a constant ε_t which is not so small. The performance is insensitive to ε_t in a reasonable constant range. These are confirmed by computer simulations.

It should be noted here that the natural gradient method given by Eqs. (8) and (9) is completely different from the Newton-Raphson method in which $G(\boldsymbol{\theta})$ is replaced by the Hessian

$$H(\boldsymbol{\theta}) = E\left[\nabla\nabla l^*(\boldsymbol{x}, y^*; \boldsymbol{\theta})\right]. \tag{19}$$

$H(\boldsymbol{\theta})$ depends on the specific cost function l^* which includes the teacher signal y^* explicitly, but $G(\boldsymbol{\theta})$ is the metric of the underlying space S independent of the target function to be approximated. However, when the log likelihood is used as the cost function and the teacher signal y is generated by a network having parameter $\boldsymbol{\theta}^*$,

$$H(\boldsymbol{\theta}^*) = G(\boldsymbol{\theta}^*) \tag{20}$$

holds at $\boldsymbol{\theta}^*$. Hence, the natural gradient is equivalent to the Newton method at around the optimal point.

The Hessian $H(\boldsymbol{\theta})$ is not necessarily positive-definite. Moreover, it includes the second derivatives. To avoid calculations of the second derivatives and to keep $H(\boldsymbol{\theta})$ positive-definite, the Gauss-Newton method approximates

$H(\boldsymbol{\theta})$, by neglecting the terms of the second derivatives by the sample average,

$$\bar{G}(\boldsymbol{\theta}) \quad = \quad \frac{1}{T}\sum_{t=1}^{T} \nabla \boldsymbol{f}(\boldsymbol{x}_t, \boldsymbol{\theta}) \nabla \boldsymbol{f}(\boldsymbol{x}_t, \boldsymbol{\theta})'. \tag{21}$$

LeCun et al. (1998) also describes stochastic or adaptive Gauss-Newton method, when l^* is a quadratic error function. The proposed method generalizes the adaptive Gauss-Newton algorithms and provides a solid theoretical justification even when l^* is not quadratic. The proposed adaptive method is computationally costless. It should be noted that, when $\boldsymbol{\theta}$ is close to a plateau, $\bar{G}(\boldsymbol{\theta})$ is almost singular and its inverse diverges.

3 Experimental Results

3.1 Simple toy model

We conducted an experiment for comparing the convergence speeds among the conventional gradient, the natural gradient, and the proposed adaptive methods. We used a simple MLP model

$$y = \sum_{\alpha=1}^{2} v_\alpha \varphi\left(\boldsymbol{w}_\alpha \cdot \boldsymbol{x}\right) + \xi, \tag{22}$$

having 2-dimensional input, and no bias terms. Assuming that the input is subject to the Gaussian distribution $N(0, I)$, where I is the identity matrix, we can get the Fisher information and its inverse explicitly so that we have the exact natural gradient learning algorithm. Training data $(\boldsymbol{x}_t, y_t^*)$, $t = 1, 2, \cdots$ at each time t is given by the noisy output y_t^* from a teacher network. The teacher network is an orthogonal committee machine having the same structure as the student network, with the true parameter defined as

$$|\boldsymbol{w}_1| = |\boldsymbol{w}_2| = 1, \quad \boldsymbol{w}_1^T \boldsymbol{w}_2 = 0, \quad v_\alpha = 1. \tag{23}$$

The variance of output noise is put equal to 0.1. We chose initial values of the parameters randomly subject to the uniform distribution on small intervals:

$$w_{\alpha i} \sim 0.3 + U\left[-10^{-8}, 10^{-8}\right], \quad v_\alpha \sim 0.2 + U\left[-10^{-4}, 10^{-4}\right], \quad \alpha, i = 1, 2. \quad (24)$$

We conducted the three learning algorithms simultaneously with the same initial value of parameters and the same training data sequence under various conditions. A typical result of convergence speed is shown in Fig. 1.

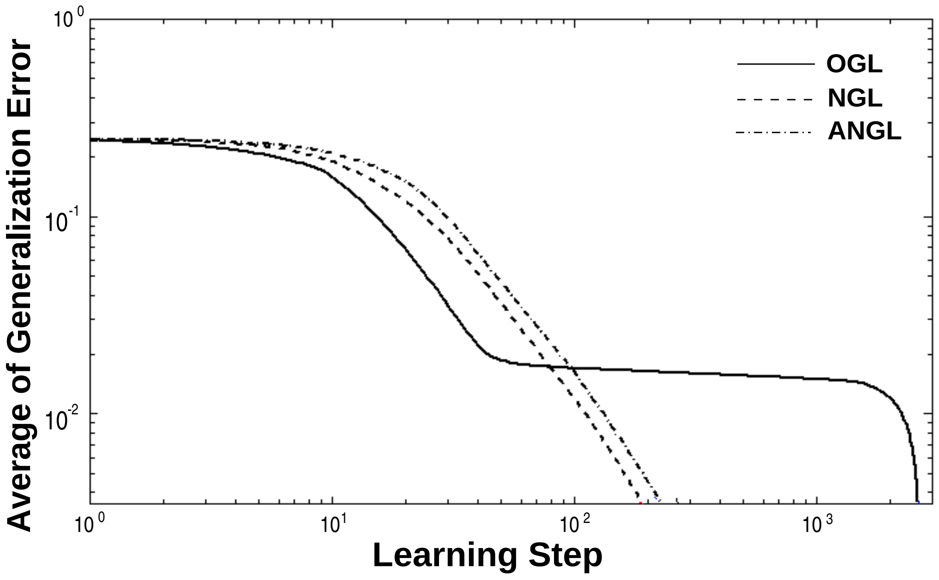

Figure 1: Average learning curves for the three algorithms.(OGL: the ordinary gradient learning, NGL: the natural gradient learning, ANGL: the adaptive natural gradient learning)

As shown in Fig. 1, the proposed learning algorithm can give the convergence speed as fast as that of the exact natural gradient learning algorithm. They can escape from plateaus much faster than the conventional one in this example. The learning rate was selected empirically to get an accurate estimator and a good convergence speed for each algorithm. Fig. 1 is a typical case with a small constant learning rate η. We also tried the case with

$\eta_t = c/t$ and others. Various choices of ε_t are also checked. When G is close to a singular matrix, G^{-1} and hence $\tilde{\nabla} l^*$ become too large. In such a case, we made η_t smaller to avoid too large adjustment of the parameter vector at one time.

It is interesting that the adaptive method works very well in this example, but has a tendency of being trapped in a plateau when ε_t is chosen to small, while the natural gradient method does not. This is due to a time delay for adjusting G^{-1}. We have checked the effect of different choices of ε_t. However, there are little changes in the performance within an adequate range of ε_t that is in range 0.05–0.5. When ε_t is too large, numerical instability emerges. When it is too small, for example, 0.01, the plateau phenomena become remarkable.

3.2 Extended XOR problem

To show that the proposed learning algorithm can be applied to practical problems, we treated a pattern classification problem. We used the extended exclusive OR problem which is a benchmark problem for pattern classifiers. Fig. 2 shows the pattern set to be classified into two categories. It consists of 9 clusters each of which is assigned to one of the two classes marked by different symbols $\circ$ and $+$. Each cluster is generated subject to a Gaussian distribution specified by a mean and a common covariance matrix. Each cluster has 200 elements, but there are some overlapped parts between the two different clusters.

We used the MLP model with 12 hidden units and one nonlinear output unit. Since it is difficult to obtain an explicit form of natural gradient, we conducted experiments for ordinary gradient learning and proposed learning. The desired outputs y are put equal to 1 and 0 for the two classes. Since the size of the training examples is fixed, we used them repeatedly and the

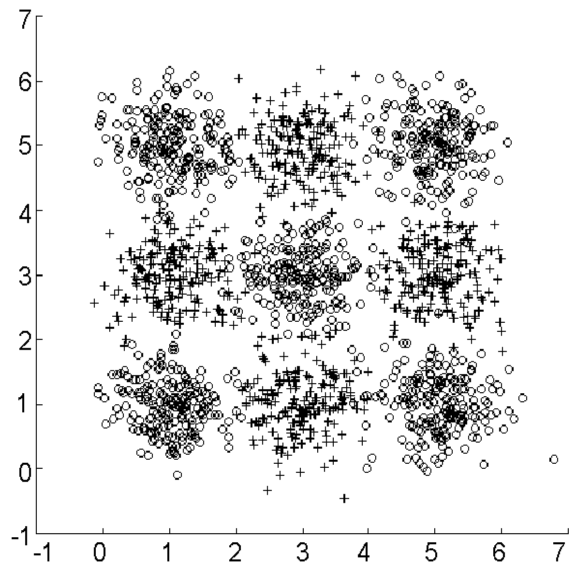

Figure 2: Extended XOR problem

mean square training error over the set was calculated at each cycle. The algorithms started with random initial values generated randomly subject to

$$w_{\alpha i}, v_\alpha, b_\alpha, b_o \sim U\left[-10^{-1}, 10^{-1}\right], \qquad \alpha, i = 1 \ldots m. \tag{25}$$

We conducted 10 independent runs. The generalization error sometimes does not decrease to a desired level in the case of ANGL, but it mostly fails in the case of the conventional method. We do not know if the state converged to a bad local minimum or is trapped to a plateau. Fig. 3 shows the best learning curve for each algorithm. Since we hardly get the zero training error, we stopped learning when there is no more increase in the classification rate for training data set. The resulting training errors were 0.0754 and 0.0675 for ordinary gradient learning and proposed learning, respectively. The necessary learning cycles to get the same classification rates(98.11%) are 154300 and 1500 for ordinary gradient learning and proposed learning. This means that the convergence speed of proposed learning was surprisingly quick in this example.

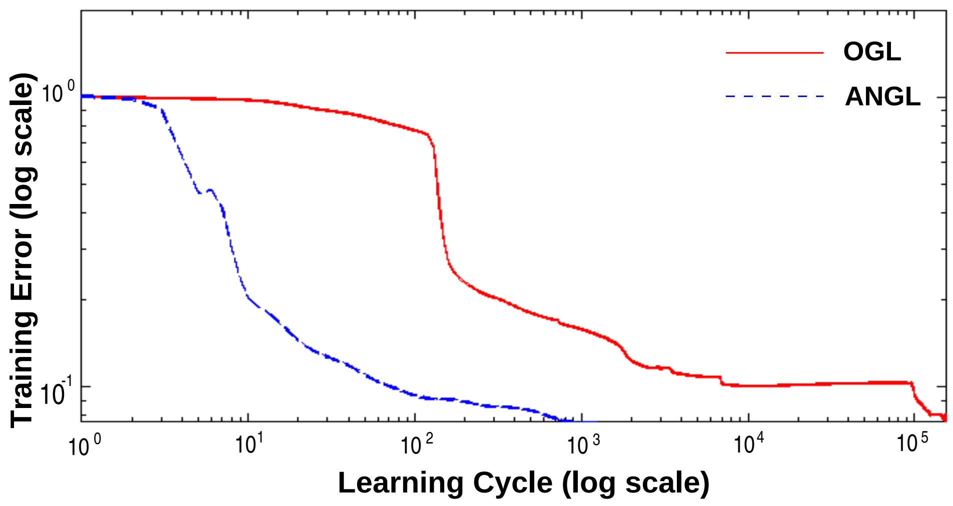

Figure 3: Learning curves for extended XOR problem(OGL: the ordinary gradient learning, ANGL: the adaptive natural gradient learning)

4 Conclusions and Discussions

In this paper, we proposed an adaptive calculation of the inverse of Fisher information matrix without matrix inversion and an adaptive natural gradient learning algorithm using it. With a simple toy problem, we showed that the proposed learning algorithm is as fast as the exact natural gradient learning algorithm, which is proved to be Fisher efficient. The proposed method is remarkably faster than the conventional learning method. The experiment with a benchmark pattern recognition problem showed that the proposed learning algorithm can be also applied to the practical problem successfully. It is demonstrated that the improvement of the convergence speed by the proposed learning method over the ordinary gradient learning method is remarkable.

This letter is a preliminary study of the topological and metrical structure of the parameter space of MLPs. More detailed theoretical and experimental results will be published in future.

Acknowledgments The authors express their gratitude to Dr. Leon Bottou and Dr. Noboru Murata for their discussions and suggestions.

References

S. Amari, “Differential-Geometrical Method in Statistics”, *Springer Lecture Note in Statistics*, vol.28, Springer, 1985.

S. Amari, “Natural Gradient Works Efficiently in Learning”, *Neural Computation,* Vol.10 pp.251-276, 1998.

S. Amari and H. Nagaoka, “Information Geometry”, AMS and Oxford University Press, 1999.

L. Bottou, “Online algorithms and stochastic approximations”, in Online Learning in Neural Networks, ed. D. Saad, Cambridge University Press, pp.9-42, 1998.

A. Edelman, T. Arias and S.T. Smith, “The geometry of algorithms with orthogonality constraints”, *SIAM Journal of Matrix Analysis and Applications*, to appear, 1998.

Y. LeCun, L. Bottou, G.B. Orr and K.-R. Müller, “Efficient backprop”, in Neural Networks—Tricks of the Trade, Springer Lecture Notes in Computer Sciences 1524, pp.5-50, 1998.

H. Park, S. Amari and K. Fukumizu, “Adaptive natural gradient learning algorithms for various stochastic models”, submitted.

M. Rattray, D. Saad and S. Amari, “Natural gradient descent for on-line learning”, *Physical Review Letters*, vol.81, pp.5461-5464, 1998.

D. Saad and S.A. Solla, “On-line learning in soft committee machines”, *Phys. Rev. E*, 52, pp.4225-4243, 1995.

H.H. Yang and S. Amari, “Complexity issues in natural gradient descent method for training multilayer perceptrons”, *Neural Computation*, 10, pp.2137-2157, 1998.