LETTER **Communicated by David Saad**

Improving Generalization Performance of Natural Gradient Learning Using Optimized Regularization by NIC

Hyeyoung Park
hypark@brain.riken.go.jp
Brain Science Institute, RIKEN, Saitama, Japan

Noboru Murata
noburu.murata@elec.waseda.ac.jp
Waseda University, Tokyo, Japan

Shun-ichi Amari
amari@brain.riken.go.jp
Brain Science Institute, RIKEN, Saitama, Japan

Natural gradient learning is known to be efficient in escaping plateau, which is a main cause of the slow learning speed of neural networks. The adaptive natural gradient learning method for practical implementation also has been developed, and its advantage in real-world problems has been confirmed. In this letter, we deal with the generalization performances of the natural gradient method. Since natural gradient learning makes parameters fit to training data quickly, the overfitting phenomenon may easily occur, which results in poor generalization performance. To solve the problem, we introduce the regularization term in natural gradient learning and propose an efficient optimizing method for the scale of regularization by using a generalized Akaike information criterion (network information criterion). We discuss the properties of the optimized regularization strength by NIC through theoretical analysis as well as computer simulations. We confirm the computational efficiency and generalization performance of the proposed method in real-world applications through computational experiments on benchmark problems.

1 Introduction

The natural gradient method is a stochastic gradient method originating with information geometry. Amari (1998) proposed the concept of natural gradient learning and proved that it is Fisher efficient in the case that the negative log likelihood is used as a loss function. Its dynamical property has been also studied by statistical-mechanical analysis in the large system size limit, and it has been shown that the natural gradient can escape from plateaus more efficiently than the standard gradient (Rattray, Saad, &

Amari, 1998). The convergence properties of on-line learning methods including natural gradient have also been analyzed using stochastic approximation theory (Bottou, 1998). In order to implement the concept of natural gradient for learning of feedforward neural networks, Amari, Park, and Fukumizu (2000) proposed an adaptive method of obtaining an estimate of the natural gradient. It is called the adaptive natural gradient method, and they extended it to various stochastic neural network models. A number of computational experiments on well-known benchmark problems have confirmed that adaptive natural gradient learning can be applied to various practical applications successfully, and the method can alleviate or even avoid plateaus.

Besides the problem of learning speed, generalization performance is another important problem in neural networks. When a network structure, an error function, and training data are fixed, all learning algorithms based on the gradient-descent method have the same equilibrium points in the error surface. Thus, in a theoretical sense, it is hard to say that the generalization performance of the natural gradient method is much different from that of the standard gradient method. In a practical sense, however, the results may be different.

Let us assume the following typical practical situation. First, we do not know the optimal complexity of network models for a given problem, and thus we use a large network with sufficient complexity. Second, since we do not know the minimum error that can be achieved by the network, we stop learning when the decrement of training error has been very small for a while and no improvement by learning is expected. In this situation, the solution obtained by the natural gradient method can be frequently different from that by the standard method, because standard gradient learning is subject to being trapped in a long plateau and can easily be misinterpreted as the end of the process. In this case, it is obvious that the generalization performances of the methods are different. Therefore, more careful consideration of the generalization performance is necessary when natural gradient learning is applied to practical problems.

Park (2001) has investigated this situation and has suggested solving it by using a regularization term. When applying the regularization method, it is very important to optimize the scale of regularization in order to have a good generalization performance. Various methods have been developed for obtaining the optimal scale of regularization (Sigurdsson, Larsen, & Hansen, 2000). A simple method is to use a validation set for optimizing the scale. Composing a validation set, however, requires using part of the learning data set. Thus, this method is not desirable when the learning data are insufficient. To solve this problem, the cross-validation method (Stone, 1974) has been widely used. To get an accurate estimate of the regularization parameter, however, the cross-validation method is computationally expensive. There also have been theoretical approaches to estimating the generalization error (Bishop, 1995; Moody, 1992; Rissanen, 1978; Vapnik,

1995) and applying it to optimization of the regularization strength. However, these methods have shortcomings in terms of efficient practical implementation, and some of them need additional assumptions. In addition, there have been few theoretical analyses on the properties of the obtained optimal values using those estimators.

In this letter, we used the concept of network information criterion (NIC) in order to optimize the regularization strength. The NIC (Murata, Yoshizawa, & Amari, 1994), which is one of the estimators for generalization errors, has been developed for general error functions and network models. Since natural gradient learning (Amari, 1998) and its adaptive version for stochastic neural networks (Park, Amari, & Fukumizu, 2000) have been derived from the same theoretical background as NIC, the natural gradient and NIC can be combined and applied to various stochastic learning models. In addition, since the theoretical meaning of applying NIC to optimize the regularization strength has been discussed by Murata (2001), the proposed method has strong theoretical justification as well. In this letter, we confirm the theoretical results by computer simulations. Furthermore, since the NIC and natural gradient learning use the common Fisher information matrix, we can also achieve computational efficiency by sharing the matrix information, as we discuss in section 3.

2 Theory of Learning with Regularization

2.1 Natural Gradient Learning. We begin with a brief introduction of the natural gradient method. Since natural gradient learning is a kind of stochastic gradient descent learning method, we consider a stochastic neural network model defined by

$$y = S\{f(\boldsymbol{x}, \boldsymbol{\theta})\}, \tag{2.1}$$

where $\boldsymbol{x}$ is an n-dimensional input subject to a probability distribution $q(\boldsymbol{x})$; $\boldsymbol{\theta} = (\theta^1, \ldots, \theta^m)^\tau$ is an m-dimensional column vector that plays the role of a coordinate system in the space of the networks; τ denotes the transposition; and the deterministic function $f(\boldsymbol{x}, \boldsymbol{\theta})$ specifies a network structure. Although the most popular type of f is given by $f(\boldsymbol{x}, \boldsymbol{\theta}) = \sum v_i \varphi\,(\boldsymbol{w}_i \cdot \boldsymbol{x})$, which is called a three-layer perceptron, the discussions in this letter have no restriction on the shape of function f except some regularity conditions such as differentiability.

The output y is emitted through a stochastic process denoted by $S\{\cdot\}$. The stochastic process can be defined by a conditional probability density function $p(y \mid \boldsymbol{x}; \boldsymbol{\theta})$ conditioned on input $\boldsymbol{x}$ and the weight parameter $\boldsymbol{\theta}$ that can be regarded as the parameter of the density function. Although the most typical example of $p(y \mid \boldsymbol{x}; \boldsymbol{\theta})$ is the gaussian noise model of the form

$$p(y \mid \boldsymbol{x}; \boldsymbol{\theta}) = \frac{1}{\sqrt{2\pi}} \exp\left[-\frac{1}{2}\{y - f(\boldsymbol{x}, \boldsymbol{\theta})\}^2\right], \tag{2.2}$$

natural gradient learning and NIC can be applied to various types of stochastic models (see Park et al., 2000, and Murata, et al., 1994, for details). Moreover, we proceed with the scalar output for simplicity, but its generalization to multivariate outputs can also be easily done.

From this stochastic viewpoint, we can consider a space of probability density functions $\{p(\boldsymbol{x}, y; \boldsymbol{\theta}) \mid \boldsymbol{\theta} \in \Re^m\}$ and define an appropriate pointwise loss function $d(\boldsymbol{z}, \boldsymbol{\theta}) = d(\boldsymbol{x}, y, \boldsymbol{\theta})$ for each piece of data $\boldsymbol{z} = (\boldsymbol{x}, y)$ and parameter $\boldsymbol{\theta}$ on the space. The ultimate goal of learning is to obtain the optimal parameter defined by

$$\boldsymbol{\theta}^* = \operatorname*{argmin}_{\boldsymbol{\theta}} E_Z[d(Z, \boldsymbol{\theta})], \tag{2.3}$$

where E_Z denotes the expectation with respect to the true distribution of random variable Z. Here, the optimality is discussed in terms of minimizing the expected loss under a given environment represented by a distribution of input and output, $p(\boldsymbol{z}; \boldsymbol{\theta}) = p(\boldsymbol{x}, y; \boldsymbol{\theta}) = p(y \mid \boldsymbol{x}; \boldsymbol{\theta})q(\boldsymbol{x})$. A natural way of estimating the optimal parameter is to adopt the empirical minimum loss estimator defined by

$$\hat{\boldsymbol{\theta}} = \operatorname*{argmin}_{\boldsymbol{\theta}} L(D, \boldsymbol{\theta}) = \operatorname*{argmin}_{\boldsymbol{\theta}} \frac{1}{n} \sum_{p=1}^{n} d(\boldsymbol{z}_p, \boldsymbol{\theta}), \tag{2.4}$$

when a sample set $D = \{(\boldsymbol{x}_p, y_p)\}_{p=1,\ldots,n}$ is observed.

To obtain $\hat{\boldsymbol{\theta}}$, gradient descent learning is widely used. The natural gradient learning method is based on the fact that the space of $p(\boldsymbol{x}, y; \boldsymbol{\theta})$ is a Riemannian space in which the metric tensor is given by the Fisher information matrix $F(\boldsymbol{\theta})$ defined by

$$F(\boldsymbol{\theta}) = E_{\boldsymbol{x}}[E_{y \mid \boldsymbol{x}; \boldsymbol{\theta}}[\nabla \log p(y \mid \boldsymbol{x}, \boldsymbol{\theta}) \nabla \log p(y \mid \boldsymbol{x}, \boldsymbol{\theta})^{\tau}]]. \tag{2.5}$$

Here, we use ∇ to denote a differential operator with respect to the parameter $\boldsymbol{\theta}$, and we will also use ∂_i in some cases to denote a partial differential operator with respect to the ith element of $\boldsymbol{\theta}$. The $E_{\boldsymbol{x}}[\cdot]$ and $E_{y \mid \boldsymbol{x}; \boldsymbol{\theta}}[\cdot]$ denote the expectation with respect to $q(\boldsymbol{x})$ and $p(y \mid \boldsymbol{x}; \boldsymbol{\theta})$, respectively. Using the inverse of the Fisher information matrix of equation 2.5, we can obtain the natural gradient $\tilde{\nabla} L$ and its learning algorithm for the stochastic systems,

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta_t \tilde{\nabla} L(\boldsymbol{\theta}_t) = \boldsymbol{\theta}_t - \eta_t (F(\boldsymbol{\theta}_t))^{-1} \nabla L(\boldsymbol{\theta}_t), \tag{2.6}$$

where $\nabla L(\boldsymbol{\theta})$ is the derivative of a loss function L with respect to $\boldsymbol{\theta}$.

Since it is difficult to obtain the Fisher information matrix $F(\boldsymbol{\theta})$ and its inverse in practical implementation, we need an approximation method. The

adaptive natural gradient (Amari et al., 2000) was originally developed for estimating the inverse of Fisher information matrix adaptively in on-line learning. In this letter, we present a similar estimation method for batch learning. Since we have a set of training data $\{(\boldsymbol{x}_p, y_p)\}_{p=1,\ldots,n}$ in batch learning, we can use the empirical distribution so as to derive an estimator $\tilde{F}(\boldsymbol{\theta})$ of Fisher information matrix,

$$\tilde{F}_n(\boldsymbol{\theta}) = \frac{1}{n}\sum_{p=1}^{n} \boldsymbol{h}_p(\boldsymbol{\theta})\boldsymbol{h}_p(\boldsymbol{\theta})^\tau, \tag{2.7}$$

where $\boldsymbol{h}_p(\boldsymbol{\theta})\boldsymbol{h}_p(\boldsymbol{\theta})^\tau = E_{y \mid \boldsymbol{x};\boldsymbol{\theta}}[\nabla \log p(y \mid \boldsymbol{x}_p;\boldsymbol{\theta}) \nabla \log p(y \mid \boldsymbol{x}_p;\boldsymbol{\theta})^\tau]$. The explicit form of $\boldsymbol{h}_p(\boldsymbol{\theta})$ can be analytically obtained by defining a specific form of $p(y \mid \boldsymbol{x}_p;\boldsymbol{\theta})$. In the case of the gaussian noise model in equation 2.2, $\boldsymbol{h}_p(\boldsymbol{\theta})$ is simply given by $\nabla f(\boldsymbol{x}_p;\boldsymbol{\theta})$ (see Park et al., 2000, for details). Then its inverse, $\tilde{F}_n^{-1} = (\tilde{F}_n(\boldsymbol{\theta}))^{-1}$, can be calculated successively by using the iterative equation,

$$\tilde{F}_n^{-1} = \frac{n}{n-1}\left(\tilde{F}_{n-1}^{-1} + \frac{\tilde{F}_{n-1}^{-1}\boldsymbol{h}_n\boldsymbol{h}_n^\tau\tilde{F}_{n-1}^{-1}}{n-1+\boldsymbol{h}_n^\tau F_{n-1}^{-1}\boldsymbol{h}_n}\right), \tag{2.8}$$

where the initial matrix F_0 is a positive definite symmetric matrix such as the identity matrix. This iterative technique can also be applied to calculating the NIC, as we show in section 3.

We should emphasize here that the definition of the Fisher information matrix depends not on the error function directly but on the stochastic neural network models that we use. Therefore, the natural gradient can be applied to arbitrary error functions satisfying some regularity conditions, such as differentiability. Although a typical error function can be defined as $L(\boldsymbol{\theta})$ of equation 2.4, we can also use other types of error functions, as we show in section 3.

2.2 Regularization Method. Due to noise in the observed samples and the limited number of samples, the estimate $\hat{\boldsymbol{\theta}}$ often has some divergence from $\boldsymbol{\theta}^*$ depending on the sample set, which results in large generalization errors. In order to decrease the generalization error $E_Z[d(Z, \hat{\boldsymbol{\theta}})]$, the regularization method is widely used. In the regularization method, we adopt an error function including a regularization term, which can be defined by

$$L_{\mathrm{reg}}(D, \alpha, \boldsymbol{\theta}) = \frac{1}{n}\sum_{p=1}^{n} d(\boldsymbol{z}_p, \boldsymbol{\theta}) + \frac{\alpha}{n} r(\boldsymbol{\theta}), \tag{2.9}$$

including a regularization parameter α. Note that in this definition, the contribution of the regularization is scaled as $1/n$. By minimizing this error

function, we obtain an estimator,

$$\bar{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \left\{ \frac{1}{n} \sum_{p=1}^{n} d(\boldsymbol{z}_p, \boldsymbol{\theta}) + \frac{\alpha}{n} r(\boldsymbol{\theta}) \right\}. \tag{2.10}$$

In order to get good generalization performances using the regularization method, it is very important to optimize the regularization strength α with respect to the expected generalization error

$$E_D[E_{\boldsymbol{Z}}[d(z, \bar{\boldsymbol{\theta}})]] = E_{\bar{\boldsymbol{\theta}}}[E_{\boldsymbol{Z}}[d(\boldsymbol{Z}, \bar{\boldsymbol{\theta}})]],$$

where both $E_D[\cdot]$ and $E_{\bar{\boldsymbol{\theta}}}[\cdot]$ denote the expectation with respect to $\bar{\boldsymbol{\theta}}$ that is dependent on the sample set D. However, there have been few theoretical works on the optimization of α. In this section, we review the relationship between the regularization strength and the generalization errors discussed in Amari and Murata (1997) and Murata (2001), and propose a modified NIC for optimizing the regularization strength α.

Before giving a formula for the optimal value of α, let us define two matrices that appear several times in the following discussions:

$$G^* = E_Z \left[\nabla d(Z, \boldsymbol{\theta}^*) \nabla d(Z, \boldsymbol{\theta}^*) \right], \quad Q^* = E_Z \left[\nabla \nabla d(Z, \boldsymbol{\theta}^*) \right]. \tag{2.11}$$

We also use the notation q^*_{ij}, g^*_{ij}, $q^*_{*}{}^{ij}$, and $g^*_{*}{}^{ij}$ for the elements at the ith row and the jth column of the matrix Q^*, G^*, Q^{*-1}, and G^{*-1}, respectively.

The asymptotic relationships among $\boldsymbol{\theta}^*$, $\hat{\boldsymbol{\theta}}$, and $\bar{\boldsymbol{\theta}}$ are given by the following lemmas.

Lemma 1. *Ensemble average and covariance matrix of the estimator $\hat{\boldsymbol{\theta}}$ derived by minimizing the empirical loss are given by*

$$E_D[\hat{\boldsymbol{\theta}}] = \boldsymbol{\theta}^* + \frac{1}{n}\boldsymbol{b} + o\left(\frac{1}{n}\right) \tag{2.12}$$

$$V_D[\hat{\boldsymbol{\theta}}] = \frac{1}{n} Q^{*-1} G^* Q^{*-1} + o\left(\frac{1}{n}\right), \tag{2.13}$$

where the ith element of the vector $\boldsymbol{b}$ is given by

$$b^i = \sum_{jkl} \left\{ q^*_{*}{}^{ij} q^*_{*}{}^{kl} \left(s^*_{jkl} - \frac{1}{2} \sum_{k'l'} q^*_{*}{}^{k'l'} t^*_{jkk'} g^*_{ll'} \right) \right\}, \tag{2.14}$$

and $t^*_{ijk} = E_Z[\partial_i \partial_j \partial_k d(Z, \boldsymbol{\theta}^*)]$, $s^*_{ijk} = E_Z[\partial_i \partial_j d(Z, \boldsymbol{\theta}^*) \partial_k d(Z, \boldsymbol{\theta}^*)]$.

Lemma 2. *The estimator is modified by the regularization term as*

$$\bar{\boldsymbol{\theta}} - \hat{\boldsymbol{\theta}} = -\frac{\alpha}{n} Q^{*-1} \nabla r(\bar{\boldsymbol{\theta}}) + O_p\left(\frac{1}{n^2}\right). \tag{2.15}$$

Note that the difference between $\bar{\boldsymbol{\theta}}$ and $\hat{\boldsymbol{\theta}}$ given in this lemma is a bias caused by the regularization. The proofs of these two lemmas are given in appendixes A and B. Note the difference between $o(\cdot)$ in equations 2.12 and 2.13 and $O_p(\cdot)$ in equation 2.15. Since $\boldsymbol{\theta}^*$ and $E_D[\bar{\boldsymbol{\theta}}]$ do not depend on D, the higher-order term, $o(1/n)$, is a deterministic quantity. On the contrary, $\hat{\boldsymbol{\theta}}$ and $\bar{\boldsymbol{\theta}}$ depend on D, and thus the higher-order term $O_p(1/n)$ is a stochastic quantity. Using the relationships among the estimators, we can obtain the expected generalization error at $\bar{\boldsymbol{\theta}}$.

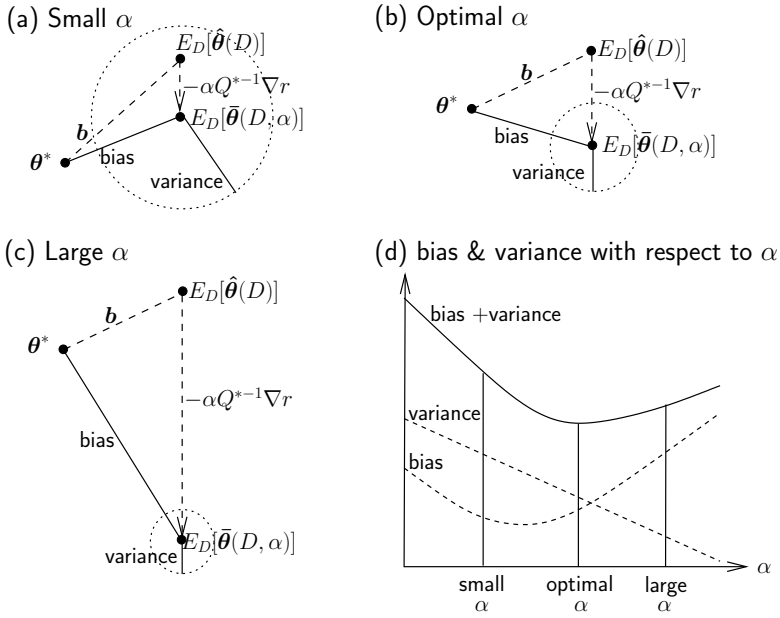
Lemma 3. *The expected generalization error at $\bar{\boldsymbol{\theta}}$ is estimated by*

$$\begin{aligned} E_D E_Z[d(Z, \bar{\boldsymbol{\theta}})] = E_Z[d(Z, \boldsymbol{\theta}^*)] &+ \frac{1}{2n} \operatorname{tr}\{Q^{*-1} G^*\} \\ &+ \frac{1}{2n^2}\{(\boldsymbol{b} - \alpha Q^{*-1} \nabla r(\boldsymbol{\theta}^*))^{\mathrm{T}} Q^* (\boldsymbol{b} - \alpha Q^{*-1} \nabla r(\boldsymbol{\theta}^*)) \\ &\qquad - 2\alpha \operatorname{tr}\{Q^{*-1} G^* Q^{*-1} \nabla \nabla r(\boldsymbol{\theta}^*)\}\} \\ &+ (\text{higher-order terms}). \end{aligned} \tag{2.16}$$

Here, the higher-order terms include nondominant terms with respect to n and α. The proof of this lemma is also given in appendix C. From the results, we can find the value of α, which minimizes the term of order $O(1/n^2)$ in equation 2.16,

$$\begin{aligned} &(\boldsymbol{b} - \alpha Q^{*-1} \nabla r(\boldsymbol{\theta}^*))^{\mathrm{T}} Q^* (\boldsymbol{b} - \alpha Q^{*-1} \nabla r(\boldsymbol{\theta}^*)) \\ &\quad -2\alpha \operatorname{tr}\{Q^{*-1} G^* Q^{*-1} \nabla \nabla r(\boldsymbol{\theta}^*)\}. \end{aligned} \tag{2.17}$$

It would be interesting to have some intuitive meaning of these two terms in equation 2.17. The first term is the sum of two biases: the original bias ($\boldsymbol{b}$) from $\hat{\boldsymbol{\theta}}$ and the additional bias ($-\alpha Q^{*-1} \nabla r(\boldsymbol{\theta}^*)$) from $\bar{\boldsymbol{\theta}}$. The second term is related to the mutual interaction of the variance ($Q^{*-1} G^* Q^{*-1}$) of $\hat{\boldsymbol{\theta}}$ and the Hessian of regularization function ($-2\alpha \nabla \nabla r(\boldsymbol{\theta}^*)$), which is the α-dependent part of total variance. In this sense, expression 2.17 shows a kind of bias-variance dilemma. When α is too small, both the bias and variance are large (see Figure 1a). As α increases, the bias and variance decrease, and the sum of bias and variance also decreases. However, when α is too large, the bias start to increase again (see Figure 1c). Therefore, we need to find an optimal value of α that minimizes the sum of the bias and variance, and the optimization of α using the expected generalization error 2.16 can



bias $= \boldsymbol{b} - \alpha Q^{*-1}\nabla r(\boldsymbol{\theta}^*)$, variance $=$ const.$-2\alpha \mathrm{tr}\{Q^{*-1}G^*Q^{*-1}\nabla\nabla r(\boldsymbol{\theta}^*)\}$

Figure 1: Bias-variance dilemma related to α.

be regarded as a solution of this bias-variance dilemma in the higher order (see Figures 1b and 1d). The optimal α is obtained as theorem 1.

Theorem 1. *The optimal regularization strength α in equation 2.9, which asymptotically minimizes the expected generalization error, is given by*

$$\alpha_{\mathrm{opt}} = \frac{\boldsymbol{b}^{\mathrm{T}}\nabla r(\boldsymbol{\theta}^*) + \mathrm{tr}\{Q^{*-1}G^*Q^{*-1}\nabla\nabla r(\boldsymbol{\theta}^*)\}}{\nabla r(\boldsymbol{\theta}^*)^{\mathrm{T}}Q^{*-1}\nabla r(\boldsymbol{\theta}^*)}. \tag{2.18}$$

In practice, the above optimal α_{opt} is not accessible because we know neither the optimal parameter $\boldsymbol{\theta}^*$ nor the true distribution $p(\boldsymbol{z})$. Obviously, it is possible to use the empirical distribution and $\bar{\boldsymbol{\theta}}$ instead of $p(\boldsymbol{z})$ and $\boldsymbol{\theta}^*$ so as to get an approximation of α_{opt}. However, it is still computationally expensive for learning systems with a large number of modifiable parameters because the second and third differentials of the loss function need to be calculated.

Using the concept of NIC, we propose a method for obtaining an estimate of the optimal scale α in practice without directly calculating the third differentials of the loss function. The NIC, which is a generalized AIC (Akaike

information criterion; Akaike, 1974), is an estimator of the expected generalization error for arbitrary error functions. From the precise derivation of NIC (Murata et al., 1994), we can have an estimator of the generalization error for the error function L_{gen}, which is written as

$$E_Z\left[d(Z,\bar{\boldsymbol{\theta}})+\frac{\alpha}{n}r(\bar{\boldsymbol{\theta}})\right] \cong \frac{1}{n}\sum_{p=1}^{n} d(\boldsymbol{z}_p,\bar{\boldsymbol{\theta}})+\frac{\alpha}{n}r(\bar{\boldsymbol{\theta}})+\frac{1}{2n}\text{tr}\{\bar{Q}^{-1}\bar{G}\} + \frac{1}{2n}\text{tr}\{\bar{Q}^{*-1}\bar{G}^{*}\}, \tag{2.19}$$

where

$$\bar{G}=\frac{1}{n}\sum_{p=1}^{n}\left(\nabla d(\boldsymbol{z}_p,\bar{\boldsymbol{\theta}})+\frac{\alpha}{n}\nabla r(\bar{\boldsymbol{\theta}})\right)\left(\nabla d(\boldsymbol{z}_p,\bar{\boldsymbol{\theta}})+\frac{\alpha}{n}\nabla r(\bar{\boldsymbol{\theta}})\right)^{\tau}$$
$$\bar{Q}=\frac{1}{n}\sum_{p=1}^{n}\left(\nabla\nabla d(\boldsymbol{z}_p,\bar{\boldsymbol{\theta}})+\frac{\alpha}{n}\nabla\nabla r(\bar{\boldsymbol{\theta}})\right)$$
$$\bar{G}^{*}=E_Z\left[\left(\nabla d(Z,\boldsymbol{\theta}^{*})+\frac{\alpha}{n}\nabla r(\boldsymbol{\theta}^{*})\right)\left(\nabla d(Z,\boldsymbol{\theta}^{*})+\frac{\alpha}{n}\nabla r(\boldsymbol{\theta}^{*})\right)^{\tau}\right]$$
$$\bar{Q}^{*}=E_Z\left[\left(\nabla\nabla d(Z,\boldsymbol{\theta}^{*})+\frac{\alpha}{n}\nabla\nabla r(\boldsymbol{\theta}^{*})\right)\right].$$

Roughly speaking, the third term in equation 2.19 is given from an asymptotic expansion around $\bar{\boldsymbol{\theta}}$, and the fourth term is given from an asymptotic expansion around $\boldsymbol{\theta}^{*}$. The fourth term is not accessible because we do not know the true parameter $\boldsymbol{\theta}^{*}$. Therefore, in practice, the matrices $\bar{G}^{*}$ and $\bar{Q}^{*}$ are replaced with $\bar{G}$ and $\bar{Q}$, respectively, leading to the original definition of NIC given by Murata et al. (1994):

$$NIC(\alpha;D)=\frac{1}{n}\sum_{p=1}^{n} d(\boldsymbol{z}_p,\bar{\boldsymbol{\theta}})+\frac{\alpha}{n}r(\bar{\boldsymbol{\theta}})+\frac{1}{n}\text{tr}\left\{\bar{Q}^{-1}\bar{G}\right\}. \tag{2.20}$$

What we would like to get is an optimized α with respect to the expected generalization error for the original loss $d(Z,\bar{\boldsymbol{\theta}})$ without regularization penalty $\alpha r(\bar{\boldsymbol{\theta}})/n$. From the fact that

$$E_D E_Z[d(Z,\bar{\boldsymbol{\theta}})]=NIC(\alpha;D)-\frac{\alpha}{n}r(\bar{\boldsymbol{\theta}}), \tag{2.21}$$

we need to neglect the regularization term $\alpha r(\bar{\boldsymbol{\theta}})/n$ in equation 2.19. Moreover, noting that the term $\text{tr}\{\bar{Q}^{*-1}\bar{G}^{*}\}$ depends not on D but on α, it is desirable to extract the α-dependent part from the term. From this concern, we define a modified NIC for regularization as

$$NIC_{\text{reg}}(\alpha;D)=\frac{1}{n}\sum_{p=1}^{n} d(\boldsymbol{z}_p,\bar{\boldsymbol{\theta}})+\frac{1}{2n}\text{tr}\left\{\tilde{Q}^{-1}\bar{G}\right\}, \tag{2.22}$$

where

$$\tilde{Q} = \frac{1}{n}\sum_{p=1}^{n}\left(\nabla\nabla d(\boldsymbol{z}_p, \bar{\boldsymbol{\theta}}) + 2\frac{\alpha}{n}\nabla\nabla r(\bar{\boldsymbol{\theta}})\right).$$

Here, $\bar{\boldsymbol{\theta}}$, $\bar{G}$, and $\tilde{Q}$ are functions of α and data set D. By expanding equation 2.22 around $\alpha/n = 0$ and minimizing it with respect to α in leading order, we obtain the relationship of theorem 2. The assumption $\alpha/n \cong 0$ is valid under the condition that n is sufficiently larger than α, which is usually satisfied in real applications. We should also remark that the matrix $\tilde{Q}$ is different from the Hessian $\bar{Q}$. The factor 2 in the regularization term in $\tilde{Q}$ is necessary to reflect the influence of the α-dependent part in $\mathrm{tr}\{\bar{Q}^{*-1}\bar{G}^{*}\}$. This modified definition of NIC is justified in the following theorem and its proof.

Theorem 2. *By optimizing α with respect to NIC_{reg}, equation 2.22, we can obtain the asymptotic equality,*

$$\hat{\alpha}_{\mathrm{opt}} = \hat{\alpha}_{\mathrm{opt}}(D) = \frac{\hat{\boldsymbol{b}}^{\tau}\nabla r(\hat{\boldsymbol{\theta}}) + \mathrm{tr}\{\hat{Q}^{-1}\hat{G}\hat{Q}^{-1}\nabla\nabla r(\hat{\boldsymbol{\theta}})\}}{\nabla r(\hat{\boldsymbol{\theta}})^{\tau}\hat{Q}^{-1}\nabla r(\hat{\boldsymbol{\theta}})}, \tag{2.23}$$

where

$$\hat{G} = \frac{1}{n}\sum_{p=1}^{n}\nabla d(\boldsymbol{z}_p, \hat{\boldsymbol{\theta}})\nabla d(\boldsymbol{z}_p, \hat{\boldsymbol{\theta}})^{\tau}, \qquad \hat{Q} = \frac{1}{n}\sum_{p=1}^{n}\nabla\nabla d(\boldsymbol{z}_p, \hat{\boldsymbol{\theta}}),$$

and $\hat{\boldsymbol{b}}$ is defined in the same manner with $\boldsymbol{b}$ by using $\hat{\boldsymbol{\theta}}$ instead of $\boldsymbol{\theta}^{*}$.

The proof is given in appendix D.

We should discuss the relationship between $\hat{\alpha}_{\mathrm{opt}}$ and α_{opt}. The true optimum α_{opt} is obtained by minimizing the asymptotic expansion of the expected generalization error $E_D E_Z[d(Z, \bar{\boldsymbol{\theta}})]$ around $\boldsymbol{\theta}^{*}$, which is given in lemma 3. This quantity is deterministically obtained if we know $\boldsymbol{\theta}^{*}$. In real situations, we do not know $\boldsymbol{\theta}^{*}$, and thus we exploit the concept of NIC as an estimator of $E_D E_Z[d(Z, \bar{\boldsymbol{\theta}})]$. Since NIC uses the practically obtained estimator $\hat{\boldsymbol{\theta}}$ for a given data set, our estimator $\hat{\alpha}_{\mathrm{opt}}$ is also a data-dependent quantity. The relationship between $\hat{\alpha}_{\mathrm{opt}}$ and α_{opt} can thus be given from the relationship between $\hat{\boldsymbol{\theta}}$ and $\boldsymbol{\theta}^{*}$. From the relationship $\boldsymbol{\theta}^{*} = \hat{\boldsymbol{\theta}} + O_p(1/\sqrt{n})$ and equation 2.23, we can obtain the relationship

$$\hat{\alpha}_{\mathrm{opt}} = \alpha_{\mathrm{opt}} + O_p\left(\frac{1}{\sqrt{n}}\right). \tag{2.24}$$

From this, we can see that it is possible to substitute the minimization of the modified NIC for the minimization of the expected generalization errors, when the number of data is sufficiently large.

3 Proposed Method

From the result of theorem 2, we can say that the optimum α_{opt}, which minimizes the expected generalization error, is asymptotically equivalent to the optimum $\hat{\alpha}_{\text{opt}}$, which minimizes the NIC_{reg} equation 2.22. By exploiting the NIC_{reg} for optimizing α instead of using equation 2.23, for direct calculation of $\hat{\alpha}_{\text{opt}}$, we can avoid heavy computations for obtaining $\hat{\boldsymbol{b}}$, which requires the values of (number of parameters)^3 entries, that is, s_{ijk} and t_{ijk}. Based on this fact, we propose a computationally efficient algorithm for obtaining $\bar{\boldsymbol{\theta}}$ with the minimum generalization error as follows.

Corollary 3. *By minimizing the two loss functions simultaneously as*

$$\bar{\boldsymbol{\theta}}(\alpha) = \underset{\boldsymbol{\theta}}{\text{argmin}} \left\{ \sum_{p=1}^{n} d(\boldsymbol{z}_p, \boldsymbol{\theta}) + \alpha r(\boldsymbol{\theta}) \right\} \tag{3.1}$$

$$\hat{\alpha}_{\text{opt}} = \underset{\alpha}{\text{argmin}} \left\{ \sum_{p=1}^{n} d(\boldsymbol{z}_p, \bar{\boldsymbol{\theta}}(\alpha)) + \frac{1}{2} \text{tr}\{\tilde{Q}^{-1}(\alpha)\bar{G}(\alpha)\} \right\}, \tag{3.2}$$

we can obtain an estimator $\bar{\boldsymbol{\theta}}$ with optimally scaled $\hat{\alpha}_{\text{opt}}$ that gives the minimum expected generalization error.

In order to obtain $\bar{\boldsymbol{\theta}}(\alpha)$ for each α, we employ natural gradient learning. As mentioned in the previous section, natural gradient learning can be easily extended to arbitrary error functions. In the regularization methods, we consider an error function with a regularization term, $L_{\text{reg}}(D, \alpha, \boldsymbol{\theta})$, which is written by equation 2.9. Since the definition of the Fisher information matrix does not depend on the error function, the adaptive natural gradient learning algorithm with the regularization term can be written by

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta_t (\tilde{F}(\boldsymbol{\theta}_t))^{-1} \nabla \left\{ \sum_{p=1}^{n} d(\boldsymbol{z}_p, \boldsymbol{\theta}_t) + \alpha r(\boldsymbol{\theta}_t) \right\}, \tag{3.3}$$

where the Fisher information matrix and its inversion $(\tilde{F}(\boldsymbol{\theta}_t))^{-1}$ can be obtained by using the same method addressed in section 2.1.

One can see that the shape of this updating rule for natural gradient learning with regularization is different from that for the second-order methods. In the case of the second-order methods, the Hessian matrix depends on

the cost function, and the explicit form of the matrix includes additional terms given from $r(\boldsymbol{\theta})$ (see Bishop, 1995, for details). On the contrary, since the Fisher information matrix in the natural gradient depends on only the probability density function $p(y \mid \boldsymbol{x}, \boldsymbol{\theta})$, the forms of $L(\boldsymbol{\theta})$ and $r(\boldsymbol{\theta})$ have no influence on the form of $(\tilde{F}(\boldsymbol{\theta}))^{-1}$. However, we should note that the equilibria in the two gradient methods are same, because both use the same ∇L.

On the other hand, the problem of calculating the optimal α is rewritten as the problem of searching the minimum of a one-parameter function, equation 3.2. Compared to the cross-validation method that is widely used, the proposed method can save computational cost. In the case of K-fold cross validation, it is required to train networks K times in order to obtain an estimate of the generalization error for each α. In the case of the proposed NIC-based method, however, we can obtain an estimate of the generalization error by calculating the NIC defined in equation 2.22. Therefore, we can say that the proposed method can find an optimal value of α about K times faster than the K-fold cross-validation method, especially when the learning cost is high. In section 4, we compare the real processing times of the proposed method with other validation methods. In searching for the optimal α, we can use the line search method, which is used in the cross-validation method as well.

We also need to discuss the computational complexity of calculating the NIC. From equation 3.2, we can see that it is required to calculate $\bar{G}(\alpha)$ and $\tilde{Q}^{-1}(\alpha)$ at each α. However, noting that the two matrices are closely related to the Fisher information matrix $F(\boldsymbol{\theta})$ used in the natural gradient learning, we can obtain them with low computational cost. In this article, we show the adaptive method for calculating $\tilde{Q}^{-1}$ and $\bar{G}$ for the gaussian noise model, equation 2.2, which is a typical model. The method can be extended to various stochastic models, as discussed in Park et al. (2000).

By taking the negative log likelihood as the pointwise loss function and the regularization term as the weight decay term, the cost function is given by

$$\sum_{p=1}^{n} d(\boldsymbol{z}_p, \boldsymbol{\theta}) + \alpha r(\boldsymbol{\theta}) = \frac{1}{2} \sum_{p=1}^{n} \{y_p - f(\boldsymbol{x}_p, \boldsymbol{\theta})\}^2 + \frac{\alpha}{2} \|\boldsymbol{\theta}\|^2. \tag{3.4}$$

Then $\bar{G}$ can be estimated by

$$\bar{G} = \sum_{p=1}^{n} \{y_p - f(\boldsymbol{x}_p, \bar{\boldsymbol{\theta}})\}^2 \tilde{F}(\bar{\boldsymbol{\theta}}) + \left(\frac{\alpha}{n}\right)^2 \bar{\boldsymbol{\theta}} \bar{\boldsymbol{\theta}}^{\tau}. \tag{3.5}$$

The term $\tilde{F}(\bar{\boldsymbol{\theta}})$ can be obtained during the natural gradient learning process, and thus we do not need to recalculate it for obtaining NIC. For the inverse

of $\tilde{Q}^{-1}$ ($= \tilde{Q}_n^{-1}$), we can apply the iterative method of equation 2.8,

$$\tilde{Q}_n^{-1} = \frac{n}{n-1}\left(\tilde{Q}_{n-1}^{-1} + \frac{\tilde{Q}_{n-1}^{-1}\nabla f_n \nabla f_n^{\tau}\tilde{Q}_{n-1}^{-1}}{n-1+\nabla f_n^{\tau}\tilde{Q}_{n-1}^{-1}\nabla f_n}\right), \tag{3.6}$$

where $\nabla f_n = \nabla f(\boldsymbol{x}_n, \bar{\boldsymbol{\theta}})$, $\tilde{Q}_0^{-1} = \boldsymbol{I}/2\alpha$, and $\boldsymbol{I}$ is the identity matrix. Here, we used the Gauss-Newton approximation for the Hessian, which is written as

$$\tilde{Q} \cong \frac{1}{n}\sum_{p=1}^{n} \nabla f_p(\boldsymbol{x}_p, \bar{\boldsymbol{\theta}}) \nabla f_p(\boldsymbol{x}_p, \bar{\boldsymbol{\theta}})^{\tau} + 2\frac{\alpha}{n}\boldsymbol{I}. \tag{3.7}$$

From this, we can say that calculating NIC takes the same order of cost as that for a one-parameter update in natural gradient learning.

4 Computational Experiments

We investigate the properties of the proposed method compared to other methods through computer simulations. For a given training data set D, we employ natural gradient learning to obtain an estimate of the weight parameter,

$$\bar{\boldsymbol{\theta}}(D, \alpha) = \operatorname*{argmin}_{\boldsymbol{\theta}} \left\{ \sum_{\boldsymbol{z}_p \in D} d(\boldsymbol{z}_p, \boldsymbol{\theta}) + \alpha r(\boldsymbol{\theta}) \right\}.$$

In order to optimize α, we employ four criteria: the generalization error (L_{gen}), the modified NIC (NIC_{reg}), the validation error (L_{val}), and the cross-validation error (L_{cv}). In practical implementation, the generalization error is approximated by a sample mean for a test data set D_{tst}, which can be written by

$$L_{\text{gen}}(D, \alpha) = E_Z[d(Z, \bar{\boldsymbol{\theta}}(D, \alpha)] \cong \frac{1}{|D_{\text{tst}}|}\sum_{\boldsymbol{z}_p \in D_{\text{tst}}} d(\boldsymbol{z}_p, \bar{\boldsymbol{\theta}}(D, \alpha)), \tag{4.1}$$

where $|D_{\text{tst}}|$ is the number of examples in the set D_{tst}. The $NIC_{\text{reg}}(D, \alpha)$ can be calculated equation 2.22 for a given training set D. For validation error, we divided the training set D into two subsets: D_{trn} for obtaining $\bar{\boldsymbol{\theta}}(D_{\text{trn}}, \alpha)$ and D_{val} for obtaining the validation error; and the criterion is calculated by

$$L_{\text{val}}(D, \alpha) = \frac{1}{|D_{\text{val}}|}\sum_{\boldsymbol{z}_p \in D_{\text{val}}} d(\boldsymbol{z}_p, \bar{\boldsymbol{\theta}}(D_{\text{trn}}, \alpha)). \tag{4.2}$$

In the case of K-fold cross validation, we divided the data set D into K subsets: $D_1, D_2, \ldots, D_K$. For the kth training, we leave a subset D_k for validation and use the rest $D_{k^-} = D_1 \cup \cdots D_{k-1} \cup D_{k+1} \cup \cdots D_K$ to obtain $\bar{\boldsymbol{\theta}}(D_{k^-}, \alpha)$. Through K times of training, the cross-validation error is calculated by

$$L_{\mathrm{cv}}(D, \alpha) = \frac{1}{K} \sum_{k=1}^{K} \left\{ \frac{1}{|D_k|} \sum_{\boldsymbol{z}_p \in D_k} d(\boldsymbol{z}_p, \bar{\boldsymbol{\theta}}(D_{k^-}, \alpha)) \right\}. \tag{4.3}$$

By minimizing these criteria with respect to α, we can obtain the four optimal values: α_D^{gen}, α_D^{NIC}, α_D^{val}, and α_D^{cv}. These α_D are stochastic values depending on data set D and are distributed around the optimum α_{opt} that minimizes the expected generalization error, $E_D[L_{\mathrm{gen}}(D, \alpha)]$. The optimum can be estimated by

$$\alpha_{\mathrm{opt}} = \alpha_{\mathrm{opt}}^{\mathrm{gen}} = \operatorname*{argmin}_{\alpha} E_D[L_{\mathrm{gen}}(D, \alpha)]. \tag{4.4}$$

Therefore, the goodness of a criterion can be evaluated by the distributional properties of the corresponding α_D and $L_{\mathrm{gen}}(\alpha_D)$.

4.1 Toy Problem. To investigate the distributional properties of α_D and $L_{\mathrm{gen}}(\alpha_D)$, we first conducted computer simulations using a simple toy problem. We used the gaussian noise model with simple weight decay regularization, as discussed in section 3. The training data are generated by a teacher network with a two input–three hidden–one output structure (see Figure 2). The output y includes the Gaussian noise with zero mean and 10^{-2} variance. To get average results, we generated 100 different training sets with 250 examples in each set. We trained three student network models with

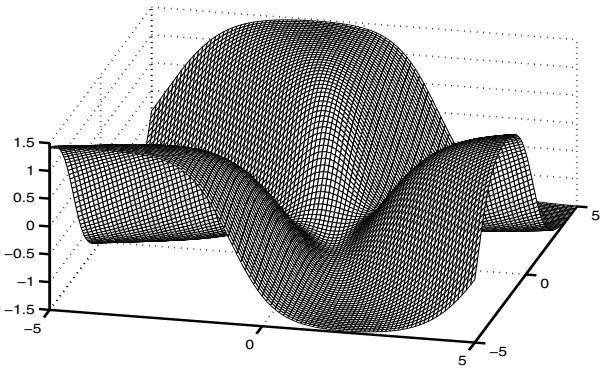


Figure 2: Input-output mapping of teacher network.

Table 1: Average Results on Toy Problem for Different Sizes of Network Models.

		Three Hiddens	Four Hiddens	Five Hiddens
Average generalization error	$E_D[L_{\text{gen}}(0)]$	0.00533	0.00560	0.00562
	$E_D[L_{\text{gen}}(\alpha_D^{\text{gen}})]$	0.00527	0.00534	0.00540
	$E_D[L_{\text{gen}}(\alpha_D^{\text{NIC}})]$	0.00531	0.00543	0.00551
	$E_D[L_{\text{gen}}(\alpha_D^{\text{val}})]$	0.00540	0.00579	0.00608
	$E_D[L_{\text{gen}}(\alpha_D^{\text{cv}})]$	0.00533	0.00540	0.00557
Relative processing time	NIC	1.0	1.0	1.0
	Validation	0.64	0.71	0.75
	Cross valid	7.76	8.40	8.51

different numbers of hidden units: three, four, and five hidden units. For approximating L_{gen}, we used a test set with 10^5 examples generated from the teacher network. For obtaining L_{val}, we divided each training set into two subsets: D_{trn} with 200 data and D_{val} with 50 data. For obtaining L_{cv}, we employed 10-fold cross validation.

The average results are given in Table 1. Table 1 shows the average of $L_{\text{gen}}(\alpha_D)$ for four different criteria and $L_{\text{gen}}(0)$ at $\alpha = 0$. The $L_{\text{gen}}(\alpha_D^{\text{gen}})$ is ideal, giving the best generalization performance for a given data set, but it cannot be used in practical applications. The simple validation method is even worse than the naive learning without regularization. The proposed method can give a competent performance compared to the cross-validation method, which has a high computational cost (see the relative processing time in Table 1). We can also see the effect of regularization on the different sizes of network models. As the number of hidden units increases, the difference between L_{gen} with regularization and L_{gen} without regularization also increases. Nevertheless, for all methods, the smaller model gives a better generalization performance than the larger one.

We conducted further investigation on detail properties of the criteria for the smallest (optimal) network model. Figure 3 shows the average curves of the criteria with respect to α—over 100 learning tasks with different data sets. From the figure, we can see that the curve of $E_D[NIC_{\text{reg}}(\alpha)]$ has a slightly steeper slope than that of $E_D[L_{\text{gen}}(\alpha)]$. This tendency also corresponds to the property of the AIC, that is, selecting a larger model in model selection tasks. Nevertheless, we can still see that the curves of $E_D[NIC_{\text{reg}}(\alpha)]$ and $E_D[L_{\text{cv}}(\alpha)]$ have similar slopes, and they are closer to that of $E_D[L_{\text{gen}}(\alpha)]$, compared to that of $E_D[L_{\text{val}}(\alpha)]$.

The optimum α_{opt} can be estimated by the minimum of the average curve of $E_D[L_{\text{gen}}(\alpha)]$, (i.e., $\alpha_{\text{opt}}^{\text{gen}}$), and it gives 0.02684. In addition to the average curves, we also plotted the median and quartiles of $L_{\text{gen}}(\alpha)$ in the inset box of Figure 3. From the graph, we can see that the random fluctuation of $L_{\text{gen}}(\alpha)$ depending on data set D is quite large, which causes a large

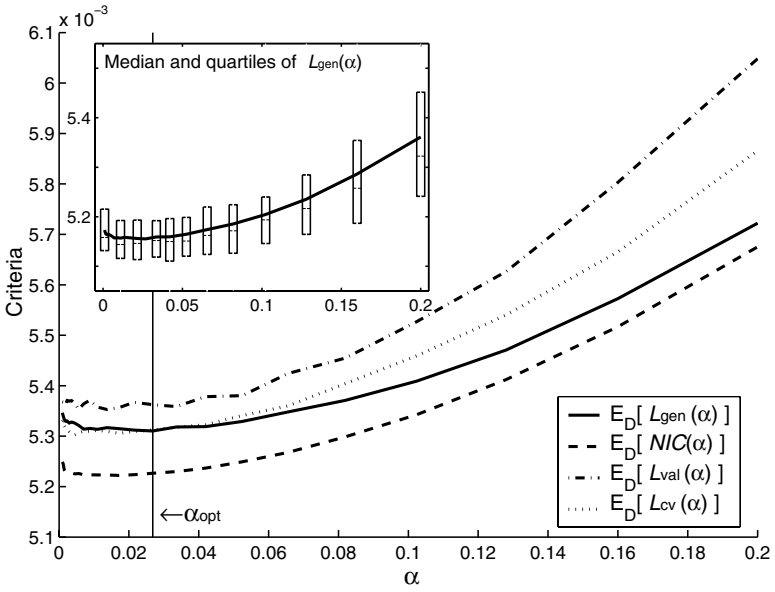


Figure 3: Average curves of the four criteria—the generalization error, the modified NIC, the validation error, and the cross-validation error (large graph)—and median and quartiles of the generalization errors (small graph).

fluctuation of optimal α_D^{gen} as well. When each data set D is not so large, the random fluctuation of data in D causes the large variance of $L_{\text{gen}}(\alpha, D)$ and α_D^{gen}. Concerning this phenomenon, we can say that it is more important to investigate the distribution properties of the optimal values $\alpha_D^{\text{gen}}, \alpha_D^{\text{NIC}}, \alpha_D^{\text{val}}$, and α_D^{cv} depending on each training set D.

Figure 4 shows the distributions of the obtained optimal α_D. In the figure, one can see that the α_D^{NIC} and α_D^{cv} concentrate in the range of smaller values than $\alpha_{\text{opt}}^{\text{gen}}(0.02684)$, and they have a smaller mean and variance than α_D^{gen}. The small value of mean $E_D[\alpha_D^{\text{NIC}}]$ is supposed to correspond to the property of the AIC, which chooses a more complex model than an optimal one, because a small regularization strength implies a small penalty of complicating function shapes made by the learning network. Nevertheless, the small variances of α_D^{NIC} and α_D^{cv} are desirable properties, giving robustness against the stochastic noise in given data sets. For the same reason, the large variance of α_D^{val} is the main cause of the poor performance of the simple validation method. For some data sets, the generalization performance of the simple validation method is very poor and results in poor average performance.

In addition to the properties of α_D itself, the data-dependent property of $L_{\text{gen}}(\alpha_D)$ is also important. In Figure 5, we show the correlation between the

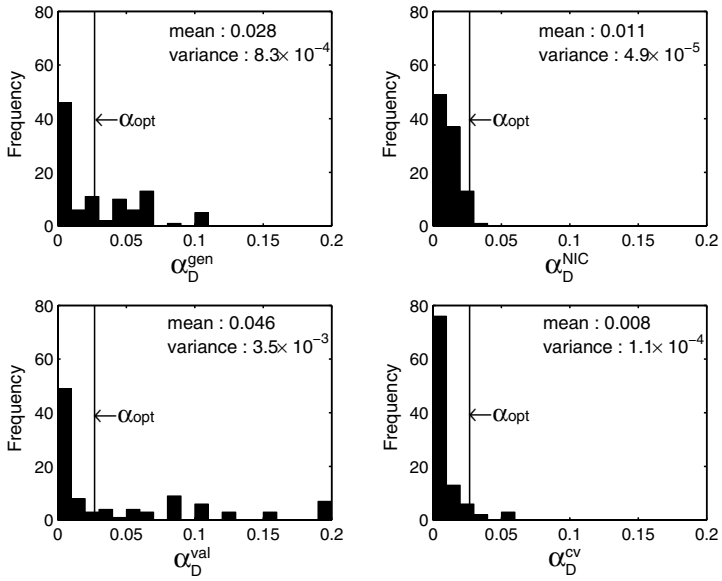


Figure 4: Distributions of optimal values of α depending on the training data set for the network model with three hidden units.

ideal generalization performance ($L_{\text{gen}}(\alpha_D^{\text{gen}})$) and the generalization performance of other methods ($L_{\text{gen}}(\alpha_D^{\text{NIC}})$, $L_{\text{gen}}(\alpha_D^{\text{val}})$, $L_{\text{gen}}(\alpha_D^{\text{cv}})$, and $L_{\text{gen}}(0)$). As shown in the figure, the generalization error obtained using NIC gives a slightly higher correlation with $L_{\text{gen}}(\alpha_D^{\text{gen}})$ than those obtained using other criteria (see the correlation coefficients in the upper-right corner of each box). The high correlation also implies the stability of the proposed method.

It is necessary to determine whether the high correlation is preserved in larger models. Figure 6 shows the correlation between $L_{\text{gen}}(\alpha_D^{\text{gen}})$ and $L_{\text{gen}}(\alpha_D^{\text{NIC}})$ (upper row) and the correlation between $L_{\text{gen}}(\alpha_D^{\text{gen}})$ and $L_{\text{gen}}(\alpha_D^{\text{cv}})$ (lower row), for the model with three, four, and five hidden units (each column). The histograms to the left of the data plots show the distribution of the generalization errors ($L_{\text{gen}}(\alpha_D^{\text{NIC}})$ in the upper row and $L_{\text{gen}}(\alpha_D^{\text{cv}})$ in the lower row), and the ones at the bottom show the distribution of $L_{\text{gen}}(\alpha_D^{\text{NIC}})$. For both NIC and cross validation, we can see that the correlation decreased as the network size increased. At the same time, the mean and variance of the generalization errors also increased as the network size increased. This deterioration is expected to be improved by using a more sophisticated function for the regularization term. Nevertheless, we can still see good correlations, and the proposed method is as good as the cross-validation method.

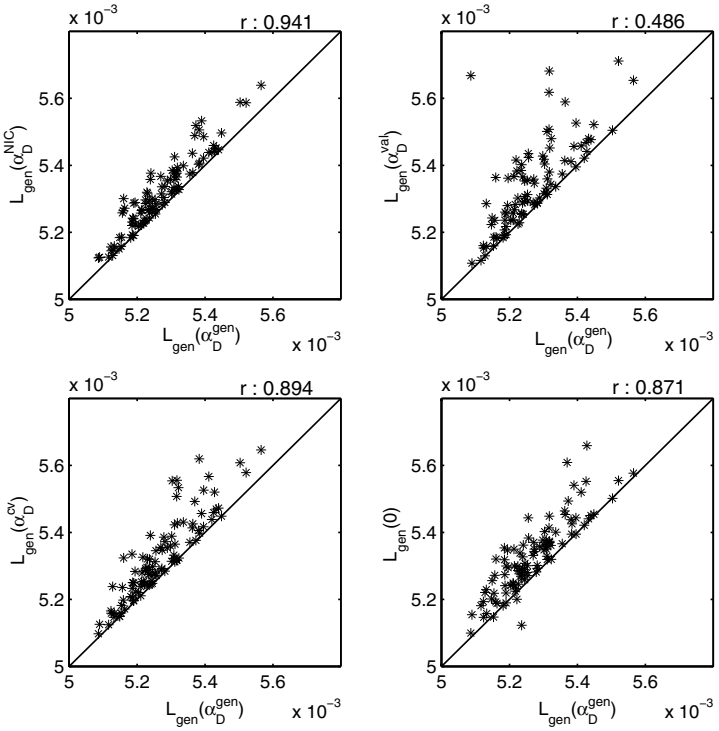


Figure 5: Correlations among generalization errors at optimized values α using different criteria for the network model with three hidden units. r denotes the correlation coefficient.

4.2 Real-World Problems. In order to check the practical applicability of the proposed method, we conducted experiments on three real-world problem: the sonar data classification problem (Gorman & Sejnowski, 1988), the pen-based digits recognition problem, and the computer system activity estimation problem. The sonar data and the pen-digit data are well-known benchmark data for classification task and are available from the UCI machine learning repository (http://www.ics.uci.edu/∼mlearn/MLSummary.html). The computer system activity estimation problem is also public benchmark data for regression tasks and is available from the DELVE database (http://www.cs.toronto. edu/∼delve/).

The task of the sonar data problem is to train a network to discriminate between sonar signals bounced off a metal cylinder and those bounced off a roughly cylindrical rock. The data are composed of two sets: a training set with 104 data and a test set with 104 data. We used a multilayer perceptron with 60 inputs–6 hiddens–1 output, and trained the network 10 times with

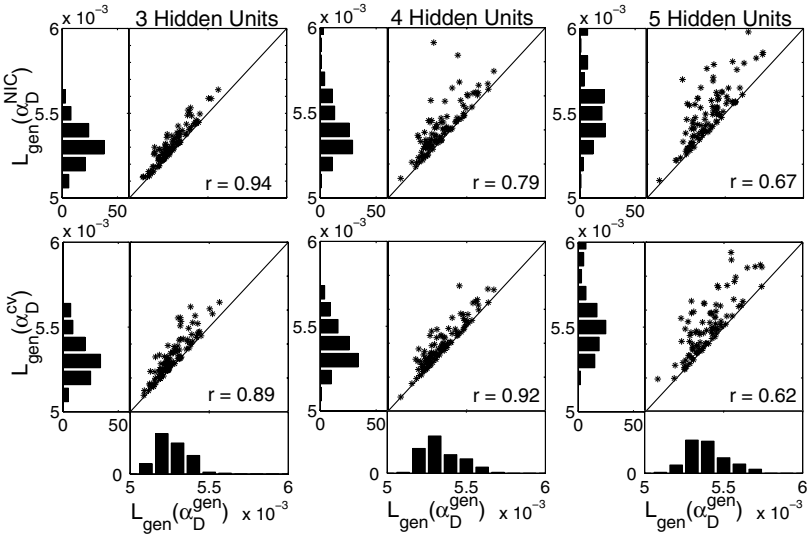


Figure 6: Correlations and distributions of generalization errors for different sizes of network model.

different initializations so as to get average results. The learning stopped when the classification rate for training data became 100% or the decreased error became smaller than 10^{-4}. In order to find the optimal α, we started from $\alpha = 0.5$ and conducted a line search by decreasing the value using $\alpha_{new} = 0.7 \times \alpha_{old}$. We applied the four criteria: L_{gen}, NIC_{reg}, L_{val}, and L_{cv}. For obtaining L_{gen}, we used the test data set with 104 data. For obtaining L_{val}, we divided the training set into two subsets: one with 78 data for training and the other with 26 data for validation. For obtaining L_{cv}, we employed the eight-fold cross-validation method.

The task of the pen-based digit recognition problem is to recognize digit numbers captured from a pressure-sensitive tablet. The raw data are preprocessed so as to be represented by 16-dimensional feature vectors. For the recognition task, we used a multilayer perceptron with 16 inputs–15 hiddens–10 outputs and trained the network 10 times with different initializations so as to get average results. The stopping condition and the searching strategy for α are the same as those for the sonar data problem. We used 700 data for training and 1000 data for testing. For obtaining L_{gen}, we used the test data set. For obtaining L_{val}, we divided the training set into two subsets: one with 500 data for training and the other with 200 data for validation. For obtaining L_{cv}, we employed the five-fold cross-validation method. For these two classification problems, the number of hidden units

was chosen so as to obtain 100% classification rates for training data sets when $\alpha = 0$.

The task of the computer system activity estimation problem is to estimate the number of users in a multiuser computer system from 12 types of system information such as the number of system calls in a second. We used a multilayer perceptron with 12 inputs–15 hiddens–1 output and trained the network 10 times with different initializations to get average results. The learning stopped when the decreased training error became smaller than 10^{-6}. The line search condition for optimizing α is the same as that for the two previous problems. We used 512 data for training, and 4096 data for testing and obtaining L_{gen}. For obtaining L_{val}, we divided the training set: one with 384 data for training and the other with 128 data for validation. For obtaining L_{cv}, we employed the four-fold cross-validation method.

The average results are given in Table 2. For the classification problems we also calculated the classification error rates (CE) to evaluate the generalization performance. The $E[\cdot]$ in the table denotes the average over a number of trainings with different initialization. From the results, we confirmed that the proposed method can achieve good generalization performance in practical applications. Furthermore, we showed the relative processing time of the methods using NIC_{reg}, L_{val}, and L_{cv} in Table 2. From the result, we can say that the proposed method is competent for practical applications in the sense of computational efficiency as well as generalization performance.

Table 2: Average Values of Test Error and/or the Classification Rate on Test Data for Different Criteria.

		SONAR	PEN-DIGITS	COMP-ACT
Average generalization error	$E[L_{\text{gen}}(0)]$	0.053	0.056	0.1702
	$E[L_{\text{gen}}(\alpha_D^{\text{gen}})]$	0.041	0.042	0.0279
	$E[L_{\text{gen}}(\alpha_D^{\text{NIC}})]$	0.045	0.044	0.0293
	$E[L_{\text{gen}}(\alpha_D^{\text{val}})]$	0.051	0.046	0.0420
	$E[L_{\text{gen}}(\alpha_D^{\text{cv}})]$	0.046	0.044	0.0408
Average classification error	$E[CE(0)]$	13.2%	6.64%	-
	$E[CE(\alpha_D^{\text{gen}})]$	10.8%	4.98%	-
	$E[CE(\alpha_D^{\text{NIC}})]$	11.8%	5.35%	-
	$E[CE(\alpha_D^{\text{val}})]$	14.5%	5.43%	-
	$E[CE(\alpha_D^{\text{cv}})]$	11.9%	5.30%	-
Relative processing time	NIC	1.0	1.0	1.0
	Validation	0.65	0.97	0.88
	Cross valid	6.44	3.42	3.04

5 Conclusion and Discussion

In this article, we deal with the generalization performance of natural gradient learning. Due to the ability of avoiding plateaus and the fast convergence of the natural gradient method, the generalization problem of the natural gradient learning could be more serious than that of the standard gradient method. To solve the problem, we presented a natural gradient learning method with a regularization term and proposed an efficient and theoretically justified method of estimating the optimal scale of regularization using a modified NIC. By combining the natural gradient and the modified NIC, we can build up a learning strategy that gives good generalization performance and can be applied to various stochastic neural network models. Since it was proved that the optimization with respect to the NIC is asymptotically equivalent to the optimization with respect to the ensemble average of the generalization error, we can expect to maximize the generalization performance of the trained networks by the proposed method. Compared to the cross-validation method, the optimization process for the regularization strength is computationally efficient.

We also investigated the properties of the optimal regularization strength obtained by the proposed method through computer simulations. Its distribution properties of concentrating on small values correspond to the properties of AIC, which is likely to choose a more complex model than the optimal one in model selection. Hagiwara (2002) noted that the tendency is due to the singular structure of the space of neural networks. As a future study, it would be important to investigate the relationship between singularities and the regularization strength more deeply. Nevertheless, the proposed method showed superiority in generalization performance compared to the simple validation method and the cross-validation method.

Appendix A. Proof of Lemma 1

Hereafter, we use the tensor-like expressions and Einstein's notation without notice, such as $a^i b_i = \sum_i^n a^i b_i$. We also use the differential operator ∂_i: $f \mapsto \frac{\partial f}{\partial \theta^i}$ for derivatives with respect to parameter $\boldsymbol{\theta} = (\theta^1, \ldots, \theta^m)^\tau$.

Let us define $\boldsymbol{\omega} = (\omega^1, \ldots, \omega^m)^\tau$ as

$$\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^* = \frac{\boldsymbol{\omega}}{\sqrt{n}}, \tag{A.1}$$

and expand the derivative of the empirical loss around the optimal parameters $\boldsymbol{\theta}^*$. By using $\sum_{p=1}^n \partial_i d(\boldsymbol{z}_p, \hat{\boldsymbol{\theta}}) = 0$, we obtain

$$\frac{1}{\sqrt{n}} \sum_{p=1}^n \partial_i d(\boldsymbol{z}_p, \hat{\boldsymbol{\theta}}) = \frac{1}{\sqrt{n}} \sum_{p=1}^n \partial_i d(\boldsymbol{z}_p, \boldsymbol{\theta}^*)$$

$$+\left(\frac{1}{n}\sum_{p=1}^{n}\partial_i\partial_j d(\boldsymbol{z}_p,\boldsymbol{\theta}^*)-q^*_{ij}+q^*_{ij}\right)\omega^j$$
$$+\frac{1}{2\sqrt{n}}\left(\frac{1}{n}\sum_{p=1}^{n}\partial_i\partial_j\partial_k d(\boldsymbol{z}_p,\boldsymbol{\theta}^*)-t^*_{ijk}+t^*_{ijk}\right)\omega^j\omega^k$$
$$+\text{(higher-order term)}=0. \tag{A.2}$$

By solving equation A.2 for $\boldsymbol{\omega}$, we obtain the following relation:

$$\omega^i=-q^{ij}_*\left(\frac{1}{\sqrt{n}}\sum_{p=1}^{n}\partial_j d(\boldsymbol{z}_p,\boldsymbol{\theta}^*)\right)$$
$$-q^{ij}_*\left(\frac{1}{n}\sum_{p=1}^{n}\partial_j\partial_k d(\boldsymbol{z}_p,\boldsymbol{\theta}^*)-q^*_{jk}\right)\omega^k$$
$$-q^{ij}_*\frac{1}{2\sqrt{n}}\left(\frac{1}{n}\sum_{p=1}^{n}\partial_j\partial_k\partial_l d(\boldsymbol{z}_p,\boldsymbol{\theta}^*)-t^*_{jkl}+t^*_{jkl}\right)\omega^k\omega^l$$
$$+\text{(higher-order term)}. \tag{A.3}$$

Knowing that

$$E_{Z_1,\ldots,Z_n}\left[\frac{1}{\sqrt{n}}\sum_{p=1}^{n}\partial_i d(Z_p,\boldsymbol{\theta}^*)\right]=0$$
$$E_{Z_1,\ldots,Z_n}\left[\frac{1}{\sqrt{n}}\sum_{p=1}^{n}\partial_i d(Z_p,\boldsymbol{\theta}^*)\frac{1}{\sqrt{n}}\sum_{p'=1}^{n}\partial_j d(Z_{p'},\boldsymbol{\theta}^*)\right]=g^*_{ij}$$
$$E_{Z_1,\ldots,Z_n}\left[\frac{1}{\sqrt{n}}\sum_{p=1}^{n}\partial_i d(Z_p,\boldsymbol{\theta}^*)\frac{1}{\sqrt{n}}\sum_{p'=1}^{n}\partial_j d(Z_{p'},\boldsymbol{\theta}^*)\right.$$
$$\left.\times\frac{1}{\sqrt{n}}\sum_{p''=1}^{n}\partial_k d(Z_{p''},\boldsymbol{\theta}^*)\right]=O\left(\frac{1}{\sqrt{n}}\right),$$

where $E_{Z_1,\ldots,Z_n}[\cdot]$ denotes an expectation with respect to independent and identically different random variables $Z_1,\ldots,Z_n$, and that

$$\left(\frac{1}{n}\sum_{p=1}^{n}\partial_j\partial_k d(\boldsymbol{z}_p,\boldsymbol{\theta}^*)-q^*_{jk}\right),\quad\left(\frac{1}{n}\sum_{p=1}^{n}\partial_j\partial_k\partial_l d(\boldsymbol{z}_p,\boldsymbol{\theta}^*)-t^*_{jkl}\right)$$

are of the order $O_p(1/\sqrt{n})$, and replacing ω^i by imposing equation A.3 repeatedly, the average of $\boldsymbol{\omega}$ can be calculated as

$$\begin{aligned}
E_{Z_1,\ldots,Z_n}[\omega^i] &= \frac{1}{\sqrt{n}} q_*^{ij} q_*^{kl} E_{Z_1,\ldots,Z_n}\left[\frac{1}{n}\sum_{p=1}^{n} \partial_j \partial_k d(Z_p, \boldsymbol{\theta}^*) \sum_{p=1}^{n} \partial_l d(Z_{p'}, \boldsymbol{\theta}^*)\right] \\
&\quad - \frac{1}{2\sqrt{n}} q_*^{ij} q_*^{kl} q_*^{k'l'} t_{jkk'}^* E_{Z_1,\ldots,Z_n}\left[\frac{1}{n}\sum_{p=1}^{n} \partial_l d(Z_p, \boldsymbol{\theta}^*) \sum_{p=1}^{n} \partial_{l'} d(Z_{p'}, \boldsymbol{\theta}^*)\right] \\
&\quad + o\left(\frac{1}{\sqrt{n}}\right) \\
&= \frac{1}{\sqrt{n}}\left(q_*^{ij} q_*^{kl} s_{jkl}^* - \frac{1}{2} q_*^{ij} q_*^{kl} q_*^{k'l'} t_{jkk'}^* g_{ll'}^*\right) + o\left(\frac{1}{\sqrt{n}}\right) \qquad \text{(A.4)} \\
&= \frac{1}{\sqrt{n}} b^i + o\left(\frac{1}{\sqrt{n}}\right). \qquad \text{(A.5)}
\end{aligned}$$

Consequently, we obtain

$$E_D[\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^*] = E_D\left[\frac{\boldsymbol{\omega}}{\sqrt{n}}\right] = \frac{1}{n}\boldsymbol{b} + o\left(\frac{1}{n}\right). \qquad \text{(A.6)}$$

The covariance of the estimator is given by

$$\begin{aligned}
V_D[\hat{\boldsymbol{\theta}}] &= E_D\left[\left\{\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^* - \frac{1}{n}\boldsymbol{b} + o\left(\frac{1}{n}\right)\right\}\left\{\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^* - \frac{1}{n}\boldsymbol{b} + o\left(\frac{1}{n}\right)\right\}^{\tau}\right] \\
&= \frac{1}{n} Q^{*-1} G^* Q^{*-1} + o\left(\frac{1}{n}\right) \qquad \text{(A.7)}
\end{aligned}$$

from ordinary asymptotic theory.

Appendix B. Proof of Lemma 2

Let

$$\boldsymbol{\zeta} = (\zeta^1, \ldots, \zeta^m) = \bar{\boldsymbol{\theta}} - \hat{\boldsymbol{\theta}} \qquad \text{(B.1)}$$

be the difference between two estimators. Taking account of

$$\frac{1}{n}\sum_{p=1}^{n} \partial_i d(\boldsymbol{z}_p, \bar{\boldsymbol{\theta}}) + \frac{\alpha}{n} \partial_i r(\bar{\boldsymbol{\theta}}) = 0 \qquad \text{(B.2)}$$

and

$$\frac{1}{n}\sum_{p=1}^{n} \partial_i d(\boldsymbol{z}_p, \hat{\boldsymbol{\theta}}) = 0, \qquad \text{(B.3)}$$

we can expand the empirical loss with the regularization term as follows:

$$\begin{aligned}
&\frac{1}{n}\sum_{p=1}^{n}\partial_i d(\boldsymbol{z}_p, \hat{\boldsymbol{\theta}}+\boldsymbol{\zeta}) + \frac{\alpha}{n}\partial_i r(\bar{\boldsymbol{\theta}}) \\
&\quad = \frac{1}{n}\sum_{p=1}^{n}\partial_i d(\boldsymbol{z}_p, \hat{\boldsymbol{\theta}}) + \frac{1}{n}\sum_{p=1}^{n}\partial_i\partial_j d(\boldsymbol{z}_p, \hat{\boldsymbol{\theta}})\zeta^j + \frac{\alpha}{n}\partial_i r(\bar{\boldsymbol{\theta}}) \\
&\qquad + (\text{higher-order term}) \\
&\quad = \frac{1}{n}\sum_{p=1}^{n}\partial_i\partial_j d(z_p, \hat{\boldsymbol{\theta}})\zeta_j + \frac{\alpha}{n}\partial_i r(\bar{\boldsymbol{\theta}}) + (\text{higher-order term}) \\
&\quad = 0. \qquad \text{(B.4)}
\end{aligned}$$

Here we define the following values,

$$\hat{g}_{ij} = \frac{1}{n}\sum_{p=1}^{n}\partial_i d(\boldsymbol{z}_p, \hat{\boldsymbol{\theta}})\partial_j d(\boldsymbol{z}_p, \hat{\boldsymbol{\theta}}) \qquad \text{(B.5)}$$

$$\hat{q}_{ij} = \frac{1}{n}\sum_{p=1}^{n}\partial_i\partial_j d(\boldsymbol{z}_p, \hat{\boldsymbol{\theta}}), \qquad \text{(B.6)}$$

which are similar to $g_{ij}(\boldsymbol{\theta}^*)$, $q_{ij}(\boldsymbol{\theta}^*)$ but are defined at parameter $\hat{\boldsymbol{\theta}}$ and averaged with respect to the empirical distribution based on the given examples. Then we obtain

$$\begin{aligned}
\zeta^i &= -\frac{\alpha}{n}\hat{q}^{ij}\partial_j r(\bar{\boldsymbol{\theta}}) + O\left(\frac{1}{n^2}\right) \\
&= -\frac{\alpha}{n}\hat{q}^{ij}\partial_j \hat{r} + O\left(\frac{1}{n^2}\right), \qquad \text{(B.7)}
\end{aligned}$$

where $\hat{r} = r(\hat{\boldsymbol{\theta}})$.

Appendix C. Proof of Lemma 3

Taking the regularization term and the bias into account, the estimator is decomposed as

$$\begin{aligned}
\bar{\boldsymbol{\theta}} = \boldsymbol{\theta}^* + \frac{\boldsymbol{\omega}}{\sqrt{n}} + \frac{\boldsymbol{b}}{n} - \frac{\alpha}{n}\left(Q^{*-1}\nabla r(\boldsymbol{\theta}^*) + Q^{*-1}\nabla\nabla r(\boldsymbol{\theta}^*)\frac{\boldsymbol{\omega}}{\sqrt{n}}\right) \\
+ (\text{higher-order term}), \qquad \text{(C.1)}
\end{aligned}$$

where $\boldsymbol{\omega} = (\omega^1, \ldots, \omega^m)^\tau$ is a random variable that is subject to a normal distribution $N(0, \boldsymbol{\Sigma})$, $\boldsymbol{\Sigma} = Q^{-1}(\boldsymbol{\theta}^*)G(\boldsymbol{\theta}^*)Q^{-1}(\boldsymbol{\theta}^*)$.

By letting

$$\dot{r}^i = q_*^{ij}\partial_j r(\boldsymbol{\theta}^*) \qquad \ddot{r}_j^i = q_*^{ik}\partial_j\partial_k r(\boldsymbol{\theta}^*),$$

the ensemble average of pointwise loss is given by

$$\begin{aligned} E_D E_Z[d(Z,\bar{\boldsymbol{\theta}})] \;=\; & E_Z[d(Z,\boldsymbol{\theta}^*)] \\ & + E_Z\left[\partial_i\partial_j d(Z,\boldsymbol{\theta}^*)\right] \\ & \times E_D\left[\frac{1}{2}\left\{\frac{\omega^k}{\sqrt{n}}\left(\delta_k^i - \frac{\alpha}{n}\ddot{r}_k^i\right) + \frac{1}{n}\left(b^i - \alpha\dot{r}^i\right)\right\}\right. \\ & \left. \times\left\{\frac{\omega^l}{\sqrt{n}}\left(\delta_l^j - \frac{\alpha}{n}\ddot{r}_l^j\right) + \frac{1}{n}\left(b^j - \alpha\dot{r}^j\right)\right\}\right] \\ & + (\text{higher-order term}) \\ =\; & E_Z\left[d(Z,\boldsymbol{\theta}^*)\right] \\ & + \frac{1}{2n}\left(\delta_k^i - \frac{\alpha}{n}\ddot{r}_k^i\right)\left(\delta_l^j - \frac{\alpha}{n}\ddot{r}_l^j\right) q_*^{kk'} q_*^{ll'} g_{k'l'}^* q_{ij}^* \\ & + \frac{1}{2n^2}\left(b^i - \alpha\dot{r}^i\right)\left(b^j - \alpha\dot{r}^j\right) q_{ij}^* \\ & + (\text{higher-order term}), \end{aligned} \tag{C.2}$$

where δ_k^i is the Kronecker's delta, which has a value of 1 when $i = k$ and otherwise is 0. Here, the equality $E_Z\left[\partial_i d(Z,\boldsymbol{\theta}^*)\right] = 0$ is used. By rewriting equation C.2 using the matrix form, we can easily obtain equation 2.16 in lemma 3.

Appendix D. Proof of Theorem 2

The modified NIC_{reg}, equation 2.22, can be rewritten as

$$\frac{1}{n}\sum_{p=1}^{n} d(\boldsymbol{z}_p,\bar{\boldsymbol{\theta}}) + \frac{1}{2n}\tilde{q}^{jk}\bar{g}_{jk}, \tag{D.1}$$

where $\tilde{q}^{ik}$ and $\bar{g}_{jk}$ is the element at ith row and jth column of $\tilde{Q}^{-1}$ and $\bar{G}$, respectively. By using equation B.7, the first term becomes

$$\begin{aligned} & \frac{1}{n}\sum_{p=1}^{n} d(\boldsymbol{z}_p,\hat{\boldsymbol{\theta}}+\boldsymbol{\zeta}) \\ =\; & \frac{1}{n}\sum_{p=1}^{n} d(\boldsymbol{z}_p,\hat{\boldsymbol{\theta}}) + \frac{1}{2}\left(\frac{\alpha}{n}\right)^2 \hat{q}_{ij}\hat{q}^{ik}\hat{q}^{jl}\partial_k\hat{r}\partial_l\hat{r} + O\left(\frac{1}{n^3}\right) \\ =\; & \frac{1}{n}\sum_{p=1}^{n} d(\boldsymbol{z}_p,\hat{\boldsymbol{\theta}}) + \frac{1}{2}\left(\frac{\alpha}{n}\right)^2 \hat{q}^{ij}\partial_i\hat{r}\partial_j\hat{r} + O\left(\frac{1}{n^3}\right). \end{aligned} \tag{D.2}$$

Noting that

$$\begin{aligned}\tilde{q}_{ij} &= \frac{1}{n}\sum_{p=1}^{n}\partial_i\partial_j d(\boldsymbol{z}_p, \hat{\boldsymbol{\theta}}+\boldsymbol{\zeta}) + 2\frac{\alpha}{n}\partial_i\partial_j r(\hat{\boldsymbol{\theta}}+\boldsymbol{\zeta}) \\ &= \frac{1}{n}\sum_{p=1}^{n}\partial_i\partial_j d(\boldsymbol{z}_p, \hat{\boldsymbol{\theta}}) + \frac{1}{n}\sum_{p=1}^{n}\partial_i\partial_j\partial_k d(\boldsymbol{z}_p, \hat{\boldsymbol{\theta}})\zeta^k + 2\frac{\alpha}{n}\partial_i\partial_j r(\hat{\boldsymbol{\theta}}) \\ &\quad + 2\frac{\alpha}{n}\partial_i\partial_j\partial_k r(\hat{\boldsymbol{\theta}})\zeta^k \\ &\quad + (\text{higher-order term}) \\ &= \hat{q}_{ij} - \frac{\alpha}{n}(\hat{t}_{ijk}\hat{q}^{kl}\partial_l r(\hat{\boldsymbol{\theta}}) - 2\partial_i\partial_j r(\hat{\boldsymbol{\theta}})) + O\left(\frac{1}{n^2}\right) \end{aligned} \tag{D.3}$$

$$\begin{aligned}\tilde{g}_{ij} &= \frac{1}{n}\sum_{p=1}^{n}\left(\partial_i d(\boldsymbol{z}_p, \hat{\boldsymbol{\theta}}+\boldsymbol{\zeta}) + \frac{\alpha}{n}\partial_i r(\hat{\boldsymbol{\theta}}+\boldsymbol{\zeta})\right) \\ &\quad \times \left(\partial_j d(\boldsymbol{z}_p, \hat{\boldsymbol{\theta}}+\boldsymbol{\zeta}) + \frac{\alpha}{n}\partial_j r(\hat{\boldsymbol{\theta}}+\boldsymbol{\zeta})\right) \\ &= \frac{1}{n}\sum_{p=1}^{n}\partial_i d(\boldsymbol{z}_p, \hat{\boldsymbol{\theta}})\partial_j d(\boldsymbol{z}_p, \hat{\boldsymbol{\theta}}) \\ &\quad + \frac{1}{n}\sum_{p=1}^{n}\partial_i\partial_k d(\boldsymbol{z}_p, \hat{\boldsymbol{\theta}})\zeta^k\partial_j d(\boldsymbol{z}_p, \hat{\boldsymbol{\theta}}) + \frac{1}{n}\sum_{p=1}^{n}\partial_i d(\boldsymbol{z}_p, \hat{\boldsymbol{\theta}})\partial_j\partial_k d(\boldsymbol{z}_p, \hat{\boldsymbol{\theta}})\zeta^k \\ &\quad + \frac{1}{n}\sum_{p=1}^{n}\partial_i d(\boldsymbol{z}_p, \hat{\boldsymbol{\theta}})\frac{\alpha}{n}\partial_j r(\hat{\boldsymbol{\theta}}) + \frac{1}{n}\sum_{p=1}^{n}\partial_j d(\boldsymbol{z}_p, \hat{\boldsymbol{\theta}})\frac{\alpha}{n}\partial_i r(\hat{\boldsymbol{\theta}}) \\ &\quad + (\text{higher-order term}) \\ &= \hat{g}_{ij} - \frac{\alpha}{n}(\hat{s}_{kij} + \hat{s}_{kji})\hat{q}^{kl}\partial_l\hat{r} + O\left(\frac{1}{n^2}\right), \end{aligned} \tag{D.4}$$

and that the equality

$$\partial q^{ij} = -q^{ik}q^{lj}\partial q_{kl} \tag{D.5}$$

holds for the derivative of the inverse matrix, the second term becomes

$$\begin{aligned}&\frac{1}{2n}\hat{q}^{ij}\hat{g}_{ij} - \frac{1}{2n}\frac{\alpha}{n}\hat{q}^{ij}\partial_i\hat{r}((\hat{s}_{jkl} + \hat{s}_{jlk})\hat{q}^{kl} - \hat{t}_{jkl}\hat{q}^{kk'}\hat{q}^{ll'}\hat{g}_{k'l'}) \\ &\quad - \frac{1}{n}\frac{\alpha}{n}\hat{q}^{ik}\hat{q}^{jl}\hat{g}_{kl}\partial_i\partial_j\hat{r} + O\left(\frac{1}{n^3}\right) \\ &= \frac{1}{2n}\hat{q}^{ij}\hat{g}_{ij} - \frac{1}{n}\frac{\alpha}{n}(\hat{b}^i\partial_i\hat{r} + \hat{q}^{ik}\hat{q}^{jl}\hat{g}_{kl}\partial_i\partial_j\hat{r}) + O\left(\frac{1}{n^3}\right). \end{aligned} \tag{D.6}$$

Neglecting higher-order terms and finding the value of α that minimizes the quadratic form

$$\frac{1}{2}\left(\frac{\alpha}{n}\right)^2 \hat{q}^{ij}\partial_i\hat{r}\partial_j\hat{r} - \frac{1}{n}\frac{\alpha}{n}(\hat{b}^i\partial_i\hat{r} + \hat{q}^{ik}q^{jl}g_{kl}\partial_i\partial_j\hat{r}), \tag{D.7}$$

we obtain

$$\begin{aligned}\hat{\alpha}_{\text{opt}} &= \frac{\hat{b}^i\partial_i\hat{r} + \hat{q}^{ik}q^{jl}g_{kl}\partial_i\partial_j\hat{r}}{\partial_i\hat{r}\partial_j\hat{r}\hat{q}^{ij}} \\ &= \frac{\hat{\boldsymbol{b}}^\tau\nabla r(\hat{\boldsymbol{\theta}}) + \text{tr}\{\hat{Q}^{-1}\hat{G}\hat{Q}^{-1}\nabla\nabla r(\hat{\boldsymbol{\theta}})\}}{\nabla r(\hat{\boldsymbol{\theta}})^\tau\hat{Q}^{-1}\nabla r(\hat{\boldsymbol{\theta}})}.\end{aligned} \tag{D.8}$$

References

Akaike, H. (1974). A new look at the statistical model identification. *IEEE trans. on Automatic Control*, *16*, 716–724.

Amari, S. (1998). Natural gradient works efficiently. *Neural Computation*, *10*, 251–276.

Amari, S., & Murata, N. (1997). Statistical analysis of regularization constant—From Bayes, MDL, and NIC points of view. *Lecture Notes in Computer Science*, *1240*, 284–293.

Amari, S., Park, H., & Fukumizu, K. (2000). Adaptive method of realizing natural gradient learning for multilayer perceptrons. *Neural Computation*, *12*, 1399–1409.

Bishop, C. M. (1995). *Neural networks for pattern recognition.* New York: Oxford University Press.

Bottou, L. (1998). Online algorithms and stochastic approximations. In D. Saad (Ed.), *Online learning and neural networks*. Cambridge: Cambridge University Press.

Gorman, R. P., & Sejnowski, T. J. (1988). Analysis of hidden units in a layered network trained to classify sonar targets. *Neural Networks*, *1*, 75–89.

Hagiwara, K. (2002). On the problem in model selection of neural network regression in overrealizable scenario. *Neural Computation*, *14*, 1979–2002.

Moody, M. E. (1992). The effective number of parameters: An analysis of generalization and regularization in nonlinear learning systems. In J. E. Moody, S. J. Hanson, & R. P. Lippmann (Eds.), *Advances in Neural Information Processing Systems*, *4* (pp. 847–854). San Mateo, CA: Morgan Kaufmann.

Murata, N. (2001). *Bias of estimators and regularization terms*. Dagstuhl seminar 01301. Available on-line: http://www.dagstuhl.de/Seminars/.

Murata, N., Yoshizawa, S., & Amari, S. (1994). Network information criterion—Determining the number of hidden units for an artificial neural network model. *IEEE Trans. on Neural Networks*, *5*(6), 865–872.

Park, H. (2001). Practical consideration on generalization property of natural gradient learning. *Lecture Notes in Computer Science*, *2084*, 402–409.

Park, H., Amari, S., & Fukumizu, K. (2000). Adaptive natural gradient learning algorithms for various stochastic models. *Neural Networks*, *13*, 755–764.

Rattray, M., Saad, D., & Amari, S. (1998). Natural gradient descent for on-line learning. *Physical Review Letters*, *81*, 5461–5464.

Rissanen, J. (1978). Modelling by shortest data description. *Automatica*, *14*, 465–471.

Sigurdsson, S., Larsen, J., & Hansen, L. K. (2000). On comparison of adaptive regularization methods. *Proc. of Neural Network for Signal Processing* (X, pp. 221–230). Piscataway, NJ: IEEE Press.

Stone, M. (1974). Cross-validatory choice and assessment of statistical predictions. *Journal of Royal Statistical Society*, *36*, 111–133.

Vapnik, V. (1995). *The nature of statistical learning theory*. Berlin: Springer-Verlag.

Received December 6, 2002; accepted July 10, 2003.