Statistical Active Learning in Multilayer Perceptrons

Kenji Fukumizu
Brain Science Institute, RIKEN
Hirosawa 2-1, Wako
Saitama 351-0198 Japan
Tel: +81-48-467-9664
Fax: +81-48-467-9693
E-mail: fuku@brain.riken.go.jp

Abstract— **This paper proposes new methods of generating input locations actively in gathering training data, aiming at solving problems special to multilayer perceptrons. One of the problems is that the optimum input locations which are calculated deterministically sometimes result in badly-distributed data and cause local minima in back-propagation training. Two probabilistic active learning methods, which utilize the statistical variance of locations, are proposed to solve this problem. One is parametric active learning and the other is multi-point-search active learning. Another serious problem in applying active learning to multilayer perceptrons is the singularity of a Fisher information matrix, whose regularity is assumed in many methods including the proposed ones. A technique of pruning redundant hidden units is proposed to keep the regularity of a Fisher information matrix, which makes active learning applicable to multilayer perceptrons. The effectiveness of the proposed methods is demonstrated through computer simulations on simple artificial problems and a real-world problem in color conversion.**

Keywords— **Active learning, Multilayer perceptron, Fisher information matrix, Pruning.**

I. Introduction

When we train a learning machine like a feedforward neural network to estimate the true input-output relation of the target system, we must prepare input vectors, observe the corresponding output vectors, and pair them to make training data. It is well known that we can improve the ability of a learning machine by designing the input of training data. Such methods of selecting the location of input vectors have been long studied in the name of *experimental design* ([1]), *response surface methodology* ([2]), *active learning* ([3],[4]), and *query construction* ([5]). They are especially important when collecting data is very expensive.

This paper discusses statistical active learning methods for the multilayer perceptron (MLP) model. We consider learning of a network as statistical estimation of a regression function. The accuracy of the estimation is often evaluated using the generalization error, which is the mean square error between the true function and its estimate. In this paper, the objective of active learning is to reduce the generalization error. Using the statistical asymptotic theory, we can derive a criterion on where are effective input locations to minimize the generalization error ([6]).

K. Fukumizu is with the Brain Science Institute, RIKEN, Saitama, Japan. E-mail: fuku@brain.riken.go.jp

The main purpose of this paper is to solve problems related to special properties of multilayer networks. One problem is that a learning rule like the error back-propagation cannot always achieve the global minimum of the training error, while many statistical active learning or experimental design methods assume its availability. We see that learning with the optimal data which are calculated deterministically is trapped by local minima more often than passive learning. To overcome this problem, we propose probabilistic methods, which generate an input data with deviation from the optimal location.

Another problem is caused by the singularity of a Fisher information matrix. Many statistical active learning methods assume the regularity of a Fisher information matrix ([1],[4],[6]), which plays an important role in the asymptotic behavior of the least square error estimator ([7],[8],[9],[10],[11]). The Fisher information matrix of a MLP, however, can be singular if the network has redundant hidden units. Since active learning methods usually require that the prepared model includes the true function, the number of hidden units must be large enough to realize it with high accuracy. Thus, the model tends to be redundant especially in active learning. To solve this problem, we propose active learning with hidden unit pruning based on the regularity condition of a Fisher information matrix of a MLP ([12]). The method removes redundant hidden units to keep the regularity of a Fisher information matrix, and makes active learning methods applicable to the MLP model.

This paper is organized as follows. In Section II, we give basic definitions and terminology, and describe an active learning criterion. In Section III, we propose two novel active learning methods based on the probabilistic optimality of training data. In Section IV, we explain a problem concerning the singularity of a Fisher information matrix, and propose a pruning technique. Section V demonstrates the effectiveness of the proposed methods through an application to a real-world problem, and Section VI concludes this paper.

II. Active learning in statistical learning

A. Basic definitions and terminology

First, we give basic definitions and terminology, which our active learning methods are based on.

We discuss the three-layer perceptron model defined by

$$f^i(\boldsymbol{x};\boldsymbol{\theta}) = \sum_{j=1}^{H} w_{ij}\; s\left(\sum_{k=1}^{L} u_{jk}x_k + \zeta_j\right) + \eta_i, \quad (1 \le i \le M) \tag{1}$$

where $\boldsymbol{\theta} = (w_{11},\ldots,w_{MH},\eta_1,\ldots,\eta_M,u_{11},\ldots,u_{HL},\zeta_1,\ldots,\zeta_H)$ represents weights and biases, and $s(t) = \frac{1}{1+e^{-t}}$ is the sigmoidal function.

We assume that the target system which is estimated by a network is a function $\boldsymbol{f}(\boldsymbol{x})$, and the output of the system is observed with an additive Gaussian noise. Then, an output data $\boldsymbol{y}$ follows

$$\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{Z}, \tag{2}$$

where $\boldsymbol{Z}$ is a random vector with a zero mean and a scalar covariance $\sigma^2 I_M$. To obtain a set of training data $D = \{(\boldsymbol{x}^{(\nu)},\boldsymbol{y}^{(\nu)}) \mid \nu = 1,\ldots,N\}$, we prepare input vectors $X_N = \{\boldsymbol{x}^{(\nu)}\}$, feed them to the target system, and observe output vectors $\{\boldsymbol{y}^{(\nu)}\}$ subject to eq.(2). The problem of active learning is how to prepare X_N.

When a set of training data D is given, we employ the least square error (LSE) estimator $\hat{\boldsymbol{\theta}}$, that is,

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \sum_{\nu=1}^{N} \|\boldsymbol{y}^{(\nu)} - \boldsymbol{f}(\boldsymbol{x}^{(\nu)};\boldsymbol{\theta})\|^2. \tag{3}$$

Unlike linear models whose experimental design has been extensively studied in the field of statistics ([1]), the solution of eq.(3) cannot be rigorously calculated in the case of neural networks. An iterative learning rule like the error back-propagation is needed to obtain an approximation of $\hat{\boldsymbol{\theta}}$. To derive an active learning criterion, however, we assume the availability of $\hat{\boldsymbol{\theta}}$. A problem related to this assumption is discussed later.

We use the *generalization error* to evaluate the ability of a trained network. For the definition, we introduce *the environmental probability* Q, which gives independent input vectors in the actual environment where a trained network should be located. In system identification, for example, Q represents the distribution of input vectors which are given to the system. The generalization error is defined by

$$\mathcal{E}_{gen} = \int \|\boldsymbol{f}(\boldsymbol{x};\hat{\boldsymbol{\theta}}) - \boldsymbol{f}(\boldsymbol{x})\|^2 dQ(\boldsymbol{x}), \tag{4}$$

which is the mean square error between the true function and its estimate. The purpose of our active learning methods is to reduce the expectation of the generalization error $\mathrm{E}[\mathcal{E}_{gen}]$. The expectation $\mathrm{E}[\cdot]$ is taken with respect to training data, as $\hat{\boldsymbol{\theta}}$ is a random vector depending on the statistical training data D_N.

If the input vectors $\{\boldsymbol{x}^{(\nu)}\}$ are independent samples from the environmental distribution Q, such learning is called *passive*. Active learning is, of course, expected to be superior to passive learning with respect to the generalization error. When the number of training data is sufficiently large, and if the true function is included in the model, the statistical asymptotic theory tells that $\mathrm{E}[\mathcal{E}_{gen}]$ of passive learning is approximately $\frac{\sigma^2}{N} \times S$, where S is the dimension of $\boldsymbol{\theta}$ ([8],[10]).

B. Criterion of statistical active learning

Because our principle of learning is to minimize the expectation of the generalization error, in order to construct an active learning method we must evaluate how $\mathrm{E}[\mathcal{E}_{gen}]$ depends on X_N. There are several kinds of methods, in general, to estimate the generalization error. One is to use the statistical asymptotic theory ([7]), and another one is to use resampling techniques like bootstrap ([13]) or cross-validation ([14]). The concept of structural risk minimization (SRM, [15]) developed by Vapnik also gives a solid basis to discuss generalization problems. In this paper, we employ a method based on the asymptotic theory. The resampling techniques, which estimate the generalization error using given training data, is not suitable for active learning in which we have to know how the generalization error depends on an input point before the data is actually generated. We do not adopt the SRM principle either, because it is based on the bound of the worst case unlike our objective to minimize the expectation of the generalization error.

For the approximation of eq.(4), we assume that the true function $\boldsymbol{f}(\boldsymbol{x})$ is completely included in the model and $\boldsymbol{f}(\boldsymbol{x};\boldsymbol{\theta}_o) = \boldsymbol{f}(\boldsymbol{x})$. This assumption is not rigorously satisfied in practical problems. In general, the expectation of the generalization error can be decomposed as

$$\begin{aligned}\mathrm{E}[\mathcal{E}_{gen}] &= \mathrm{E}\left[\int \|\boldsymbol{f}(\boldsymbol{x};\hat{\boldsymbol{\theta}}) - \boldsymbol{f}(\boldsymbol{x};\boldsymbol{\theta}_o)\|^2 dQ(\boldsymbol{x})\right] \\ &\quad + \int \|\boldsymbol{f}(\boldsymbol{x};\boldsymbol{\theta}_o) - \boldsymbol{f}(\boldsymbol{x})\|^2 dQ(\boldsymbol{x}),\end{aligned} \tag{5}$$

where $\boldsymbol{\theta}_o$ is the parameter that gives $\min_{\boldsymbol{\theta}} \int \|\boldsymbol{f}(\boldsymbol{x};\boldsymbol{\theta}_o) - \boldsymbol{f}(\boldsymbol{x})\|^2 dQ(\boldsymbol{x})$. The first and the second terms in eq.(5) are called the variance and the bias of the model respectively. Moody ([16]), for example, discusses the generalization error in the framework of *nonparametric regression* which allows the model bias. However, it is very difficult to describe explicitly the dependence of $\mathrm{E}[\mathcal{E}_{gen}]$ on X_N if the model bias exists. Therefore, we assume that the bias of the model is small enough to be neglected, and that active learning is supposed to reduce the variance term. In Section IV, we discuss how to solve the problem of the model bias.

Similar to Cohn's discussion ([4]), application of the asymptotic theory ([9],[10]) or local linearization under the bias-free assumption shows

$$\mathrm{E}\left[\mathcal{E}_{gen}\right] \approx \sigma^2 \mathrm{Tr}\left[I(\boldsymbol{\theta}_o)J^{-1}(\boldsymbol{\theta}_o;X_N)\right], \tag{6}$$

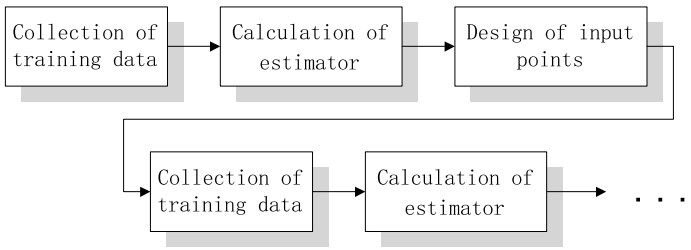

Fig. 1. Sequential active learning

where the matrix $I(\boldsymbol{\theta})$ and $J(\boldsymbol{\theta}; X_N)$ are defined by

$$I(\boldsymbol{\theta}) = \int I(\boldsymbol{x};\boldsymbol{\theta}) dQ(\boldsymbol{x}), \tag{7}$$

$$J(\boldsymbol{\theta}; X_N) = \sum_{\nu=1}^{N} I(\boldsymbol{x}^{(\nu)};\boldsymbol{\theta}), \tag{8}$$

$$I_{ab}(\boldsymbol{x};\boldsymbol{\theta}) = \frac{\partial \boldsymbol{f}(\boldsymbol{x};\boldsymbol{\theta})^T}{\partial \theta_a} \frac{\partial \boldsymbol{f}(\boldsymbol{x};\boldsymbol{\theta})}{\partial \theta_b}. \tag{9}$$

The matrix $I(\boldsymbol{\theta})$ and $J(\boldsymbol{\theta}; X_N)$ are called Fisher information matrixes or asymptotic covariance matrixes. Note that the matrix $I(\boldsymbol{\theta})$ is averaged with the environmental probability Q, while $J(\boldsymbol{\theta}; X_N)$ is calculated using empirical data X_N. Replacing the unknown parameter $\boldsymbol{\theta}_o$ with its current estimate $\hat{\boldsymbol{\theta}}$, we adopt the following as the criterion of active learning;

$$\mathrm{Tr}\left[I(\hat{\boldsymbol{\theta}}) J^{-1}(\hat{\boldsymbol{\theta}}; X_N) \right]. \tag{10}$$

This criterion is equivalent to Q-optimality ([1]) if the model is linear. Thus, our active learning criterion for neural networks is a nonlinear extension of Q-optimality. In the rest of this paper, we discuss special problems caused by nonlinearity of neural networks. Similar criterions are derived by MacKay ([3]) and Cohn ([4]). However, their criterions are based on the error of one point to avoid the integral calculation. We perform the numerical integral calculation to keep the principle of minimizing the generalization error.

C. Problem of deterministic active learning

In this subsection, we explain that a simple implementation of the above active learning criterion has a problem. We employ sequential active learning which is commonly used in experimental design ([1]), because we should update $\hat{\boldsymbol{\theta}}$ in eq.(10) to obtain a more accurate estimate each time a new training data is added. Design of the next input point, observation of the response, and estimation of $\boldsymbol{\theta}$ are iteratively performed in sequential learning (Fig.1).

The simplest sequential active learning method is described as follows ([1]). When we have $n-1$ training data D_{n-1} and the corresponding LSE estimator $\hat{\boldsymbol{\theta}}_{n-1}$, we select the next input $\boldsymbol{x}^{(n)}$ according to

$$\boldsymbol{x}^{(n)} = \arg\min_{\boldsymbol{x}} \mathrm{Tr}\left[I(\hat{\boldsymbol{\theta}}_{n-1}) J^{-1}(\hat{\boldsymbol{\theta}}_{n-1}; X_{n-1} \cup \{\boldsymbol{x}\}) \right]. \tag{11}$$

We call this *deterministic active learning*, because the location of the next input is selected deterministically.

In the case of neural networks, this method does not necessarily work well. Training of a neural network does not always give the correct LSE estimator because of local minima and plateaus. The above method tend to generate training data that are trapped by local minima more easily. We explain the reason briefly. It is known that the optimal data that minimize the left hand side of eq.(6) can be approximated by a data set on a fixed number of input locations, because any Fisher information matrix at $\boldsymbol{\theta}_o$ can be approximately realized using a data set on $\frac{S(S+1)}{2}+1$ points ([1], Theorem 2.1.2). Therefore, it is very likely that the same input positions are repeatedly selected in deterministic active learning. Obviously such a training data set makes the convergence of the back propagation much more difficult.

We illustrate this influence with a simple experiment using a MLP network with 2 input, 2 hidden, and 1 output unit. The target function is also defined by a parameter in this model (Fig.2). The normal distribution $N(0, 16I_2)$ is used for Q, where $N(\boldsymbol{m}, \Sigma)$ means the normal distribution with $\boldsymbol{m}$ as its mean and Σ as its variance-covariance matrix. Fig.3 shows the average of generalization errors for 50 trials changing initial training data set. The result of deterministic active learning is inferior to that of passive one after 60 data. We find that the parameter sometimes does not approach to $\hat{\boldsymbol{\theta}}$ because of the excessive localization of training data, which is shown clearly in Fig.4.

III. Probabilistic active learning

A. Probabilistic active learning methods

We propose two probabilistic active learning methods. One is the parametric active learning, which utilizes a parametric probability family to generate a new input point.

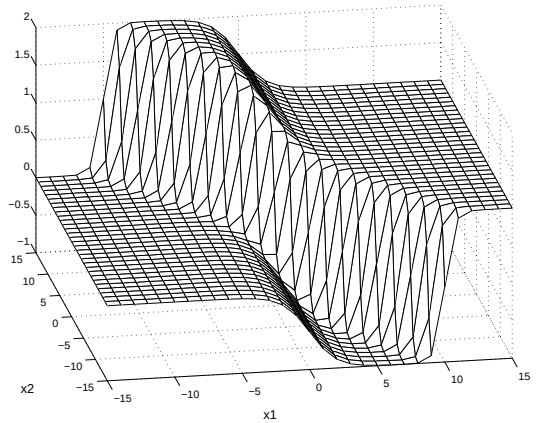

Fig. 2. The true function of 2-2-1 MLP model

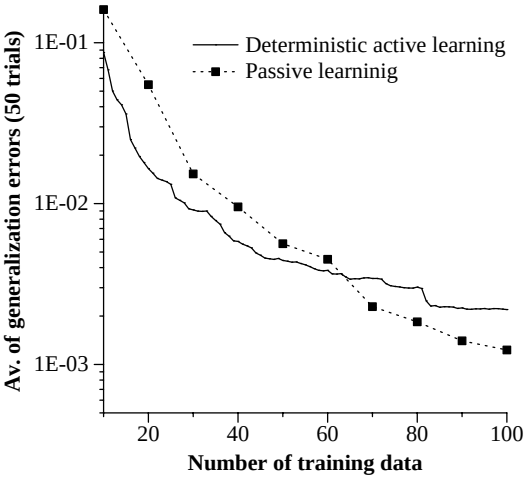

Fig. 3. Deterministic active learning

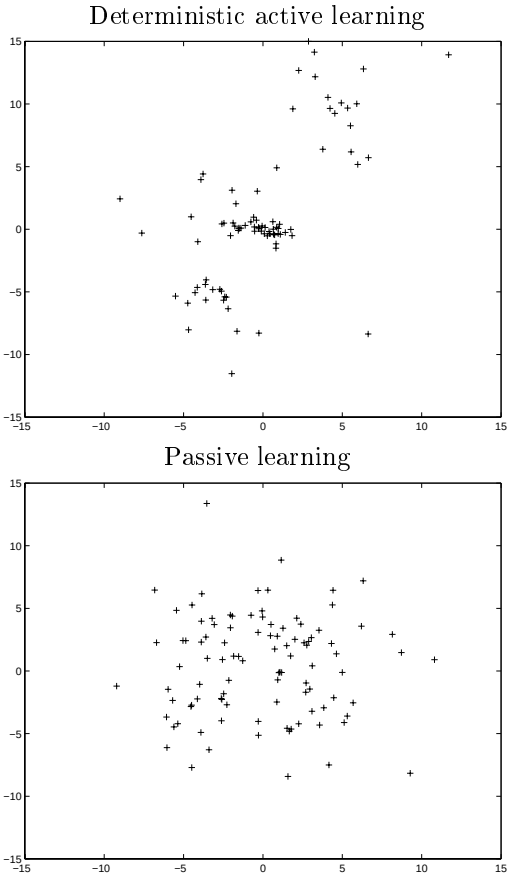

Fig. 4. Distributions of input data

This is a slight refinement of the method proposed by Fukumizu ([17]). The other is the multi-point-search active learning, which generates a finite number of input points as candidates and selects the best one. In both methods, we introduce randomness which is expected to solve the problem of excessive localization.

A.1 Parametric active learning

Instead of optimizing a point $\boldsymbol{x}$ in eq.(10), we introduce a parametric family of the density functions $\{r(\boldsymbol{x};\boldsymbol{v})\}$ for generating $\boldsymbol{x}$, and try to optimize the density. A possible choice of $\{r(\boldsymbol{x};\boldsymbol{v})\}$ is a normal mixture model defined by

$$r(\boldsymbol{x};\boldsymbol{v}) = \sum_{k=1}^{K} \frac{c_k}{(2\pi\tau_k^2)^{L/2}} \exp\left(-\frac{1}{2\tau_k^2}\|\boldsymbol{x}-\boldsymbol{m}_k\|^2\right), \quad (12)$$

where $c_k \geq 0$ $(k = 1,\ldots,K)$, $\sum_{k=1}^{K} c_k = 1$, and $\boldsymbol{v} = (c_1, \boldsymbol{m}_1, \tau_1, \ldots, c_K, \boldsymbol{m}_K, \tau_K)$ is a variable parameter vector. Since a normal mixture converges to a point distribution if τ_k goes to zero, we should restrict the value of τ_k in $[A,\infty)$ for a positive A.

We optimize the density by finding the best $\boldsymbol{v}$ to minimize

$$\mathrm{Tr}\left[I(\hat{\boldsymbol{\theta}}_{n-1})\left(J(\hat{\boldsymbol{\theta}}_{n-1};X_{n-1}) + J(\hat{\boldsymbol{\theta}}_{n-1};r_{\boldsymbol{v}})\right)^{-1}\right], \quad (13)$$

where

$$J(\boldsymbol{\theta};r_{\boldsymbol{v}}) = \int J(\boldsymbol{\theta};\boldsymbol{x})r(\boldsymbol{x};\boldsymbol{v})d\boldsymbol{x}. \quad (14)$$

The algorithm is described as follows.

[PARAMETRIC ACTIVE LEARNING]

1. Prepare an initial set of training data D_{N_0}.
2. Calculate the initial estimator $\hat{\boldsymbol{\theta}}_{N_0}$ with respect to D_{N_0}.
3. Prepare an initial parameter $\boldsymbol{v}_{N_0}$.
4. $n := N_0 + 1$.
5. Find $\boldsymbol{v}_n$ that minimizes

$$\mathrm{Tr}\left[I(\hat{\boldsymbol{\theta}}_{n-1})\left(J(\hat{\boldsymbol{\theta}}_{n-1};X_{n-1}) + J(\hat{\boldsymbol{\theta}}_{n-1};r_{\boldsymbol{v}})\right)^{-1}\right],$$

using a numerical optimization method.
6. Generate an input data $\boldsymbol{x}^{(n)}$ from $r(\boldsymbol{x};\boldsymbol{v}_n)$.
7. Feed $\boldsymbol{x}^{(n)}$ to the target system. Observe a response $\boldsymbol{y}^{(n)}$.
8. Set $D_n := D_{n-1} \cup (\boldsymbol{x}^{(n)}, \boldsymbol{y}^{(n)})$.
9. Calculate the LSE estimator $\hat{\boldsymbol{\theta}}_n$ with respect to D_n.
10. $n := n + 1$.
11. if $n > N$, then END, otherwise go to **5**.

Although the selected data are optimal only probabilistically at best, they distribute over the input space more widely than those of deterministic active learning. we can expect this prevents the excessive localization of training data. However, this method needs the integral calculation of $J(\hat{\boldsymbol{\theta}}_{n-1};r_{\boldsymbol{v}})$ in each iteration of numerical optimization. The calculation cost is very expensive.

A.2 Multi-point-search active learning

We consider a method in which multiple candidates of the next point are generated and the best one that minimizes

$$\mathrm{Tr}\left[I(\hat{\boldsymbol{\theta}}_{n-1})J^{-1}(\hat{\boldsymbol{\theta}}_{n-1}; X_{n-1}\cup\{\boldsymbol{x}\})\right]$$

is selected. If the number of candidates increases according to the number of training data, the best one converges to the true optimal location. This method is more random in early stage of the training, and it comes to generate the optimal data gradually. It aims at avoiding local minima for a small number of training data. Learning in early stage is especially important, because it is very difficult to converge to $\boldsymbol{\theta}_o$ if data are generated based on a wrong estimate of $\boldsymbol{\theta}_0$. If we generate random candidates subject to Q, the learning moves from passive to active.

The algorithm is described as follows. The number of candidates for the nth training data, K_n, is an increasing function of n.

[MULTI-POINT-SEARCH ACTIVE LEARNING]

1. Prepare an initial set of training data D_{N_0}.

2. Calculate the initial estimator $\hat{\boldsymbol{\theta}}_{N_0}$ with respect to D_{N_0}.

3. $n := N_0 + 1$.

4. Generate K_n input data $\boldsymbol{x}_{<1>}, \ldots, \boldsymbol{x}_{<K_n>}$. Choose $\boldsymbol{x}^{(n)}$ according to

$$\boldsymbol{x}^{(n)} = \arg\min_{\boldsymbol{x}_{<j>}} \mathrm{Tr}\left[I(\hat{\boldsymbol{\theta}}_{n-1})J^{-1}(\hat{\boldsymbol{\theta}}_{n-1}; X_{n-1}\cup\{\boldsymbol{x}_{<j>}\})\right].$$

5. Feed $\boldsymbol{x}^{(n)}$ to the target system. Observe a response $\boldsymbol{y}^{(n)}$.

6. Set $D_n := D_{n-1}\cup\{(\boldsymbol{x}^{(n)}, \boldsymbol{y}^{(n)})\}$.

7. Calculate the LSE estimator $\hat{\boldsymbol{\theta}}_n$ with respect to D_n.

8. $n := n + 1$.

9. if $n > N$, then END, otherwise go to **4**.

This method does not require numerical minimization. This remarkably saves the computational cost, which is often a problem of active learning methods.

B. Comparison with other active learning methods

There have been proposed other active learning methods which use Fisher information matrixes. We briefly review them and compare them with ours.

The most famous criterion of experimental design is *D-optimality* ([1]), which selects input data that maximize $\det J(\boldsymbol{\theta}_0, X_N)$. It is known ([18]) that under some conditions D-optimality is equivalent to the minimax criterion that selects the input data X_N according to

$$\min_{X_N}\max_{\boldsymbol{x}} \mathrm{E}\left[\|\boldsymbol{f}(\boldsymbol{x};\hat{\boldsymbol{\theta}}) - \boldsymbol{f}(\boldsymbol{x};\boldsymbol{\theta}_o)\|^2\right]. \tag{15}$$

In a sequential implementation of D-optimality ([1]), the selected point attains the maximum of the expected variance of the estimation defined by

$$V(\boldsymbol{x}) = \mathrm{E}\left[\|\boldsymbol{y} - \boldsymbol{f}(\boldsymbol{x};\hat{\boldsymbol{\theta}})\|^2\right]. \tag{16}$$

Kindermann et al. ([19]) proposes an active learning method for neural networks based on this criterion. They use the bootstrap to estimate $V(\boldsymbol{x})$. The computational cost of this method is very expensive, since we have to perform both the bootstrap and the numerical optimization of the input point.

These criterions are clearly different from ours in that they do not minimize the generalization error. It depends on the purpose of learning which criterion should be applied.

Cohn ([4]) proposes a method that uses reference points to avoid the integral calculation of $I(\boldsymbol{\theta})$. He uses a random reference point $\boldsymbol{x}_r$, and selects the next point that minimizes

$$\mathrm{Tr}\left[I(\boldsymbol{x}_r;\hat{\boldsymbol{\theta}}_{n-1})J^{-1}(\hat{\boldsymbol{\theta}}_{n-1}; X_{n-1}\cup\{\boldsymbol{x}\})\right]. \tag{17}$$

Although our methods look similar to this, they are essentially different in that the objective of our method is to minimize the generalization error. Note that the above criterion is different from eq.(10) even if $\boldsymbol{x}_r$ is taken from Q, because the operations of min and integral are not changeable. However, we can expect that this method also has an effect of avoiding localization of input points by the variation of reference points.

C. Experimental results on active learning methods

We show simple experimental results to compare the performance and property of active learning methods.

The first experiment is a very simple one to see the basic properties of various active learning methods. We use the MLP model with 1 input, 1 hidden, and 1 output unit. The target function is given by

$$f(\boldsymbol{x}) = s(\boldsymbol{x}), \tag{18}$$

which is realized by the model. The total number of training data is 100. The initial 10 data are given passively subject to $Q = N(0,1)$. The deviation of the noise added to the output is 0.1. We compare the following 5 methods;

A. parametric active learning
B. multi-point-search active learning
C. maximum variance point ([19])
D. usage of reference points ([4])
E. passive learning

In method A, we use a mixture model of four normal distributions. The candidates in method B are generated by Q, and $K_n = \left\lceil\sqrt{10(n-10)}\right\rceil$. In Method C, we use 20 bootstrap samples in estimating $V(\boldsymbol{x})$. In Method D, the probability to generate reference points is Q.

Fig.5 shows the average of generalization errors for 50 sets of training data. Active learning with method A, B, and D outperform passive learning. The proposed methods, A and B, show good performance in generalization error. Interestingly, method D is as good as A and B, though the criterion does not precisely minimize the generalization error. Method C also shows effectiveness in small number

of training data, while the final result is worse than the result of passive learning. This is reasonable because the criterion of method C is different from minimization of the generalization error. In fact, there are many training data selected very far from the high-density region of Q.

Next, we apply active learning methods to see the performances in a little complicated problem, which is the same as the one in Section II.C (Fig.2). We omit the method C in this simulation, since it is computationally very expensive and we know from the previous experiment that the performance in the generalization error is not so high. Fig.6 shows the average of generalization errors for 50 data sets. In this case, the multi-point-search method shows the best performance. Although parametric active learning still shows much better performance than passive learning, its effect is not so remarkable as the effect of the multi-point-search method. One reason is that the density model $r(\boldsymbol{x};\boldsymbol{v})$, which is the mixture of 4 normal distributions, is not sufficient to express the optimal density. In fact, as we can see in Fig.7, the distribution of the input data in the parametric method is more concentrated around the center than the multi-point-search method. Although Method D shows effectiveness in early stage of learning, it is worse than passive learning after the number of data becomes large. This seems natural because the criterion is not equivalent to the generalization error.

In both simulations, our probabilistic active learning methods show significant reduction of the generalization error. The multi-point-search method shows almost the best in both simulations. In the parametric method, we have to choose carefully a density family $r(\boldsymbol{x};\boldsymbol{v})$, which has an essential influence on the performance. It is also a disadvantage of the parametric method that the deviation of data from the optimal position remains even after the training has converged successfully. On the other hand, in the multi-point-search method, we have only to choose the number of candidates at each sampling. It automatically increases the optimality of selected locations. Cohn's method also shows effectiveness in the generalization error in spite of the difference of the criterion. However, in the latter simulation, the effect becomes comparatively small as the increase of the number of data.

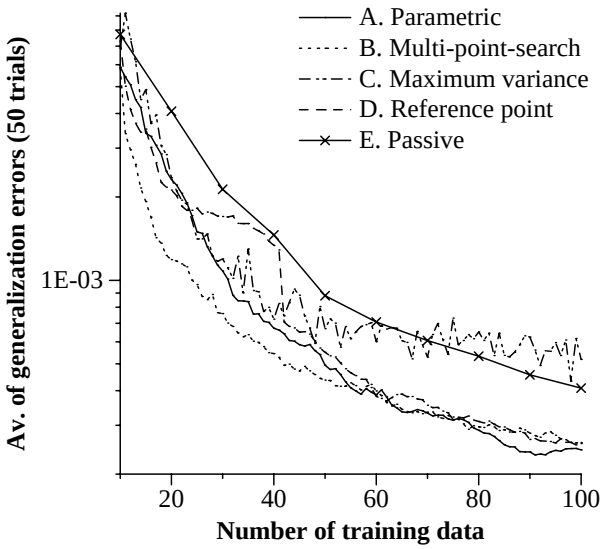

Fig. 5. Comparison of active learning methods (1-1-1)

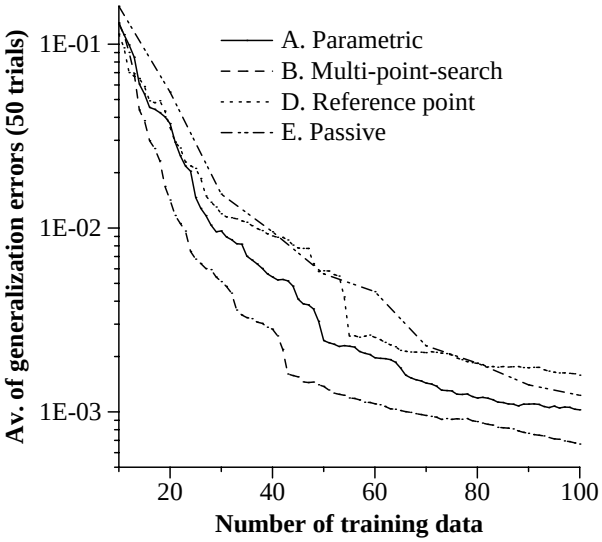

Fig. 6. Comparison of active learning methods (2-2-1)

IV. Model selection in active learning

A. Model mismatch problem in active learning

In the previous sections, we assume that the true function is completely included in the model or can be approximated by the model with high accuracy. This assumption is too strong in actual machine learning problems. On the other hand, it is easy to see that active learning does not work if the model has a large bias. The data set given by active learning is far from optimal, for example, if we estimate a quadratic function by the linear regression model believing that the true function is linear.

Model selection is, then, especially important in active learning. It is known that a MLP can approximate any continuous function on a compact set with arbitrary accuracy ([20],[21],[22]). Therefore, a network with a sufficiently large number of hidden units can almost realize the true function. A network with many hidden units, however, causes a critical problem in active learning, in addition to the increase of the generalization error caused by surplus parameters. It is proved that the Fisher information matrix at the true parameter is singular if and only if the model has surplus hidden units to realize the true function ([12]). Even if the true function cannot be realized perfectly, the Fisher information of a network with almost redundant hidden units is very close to a singular one, which makes an algorithm using the inverse matrix numerically unstable. We should establish a method of keeping the Fisher information matrix non-singular during learning. We will describe a solution to this problem in this section. This method is first introduced in Fukumizu

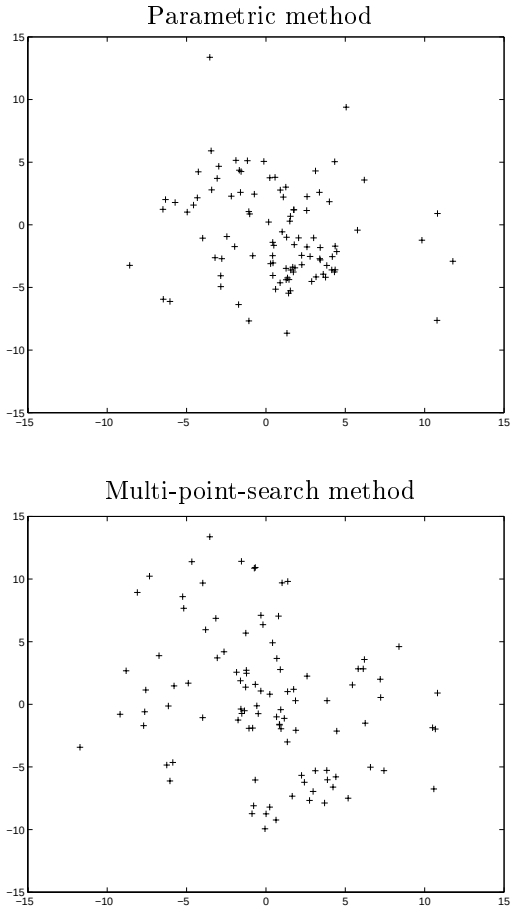

Fig. 7. Distributions of input data

([17]), and we give its full description here.

B. Pruning for regularity of a Fisher information matrix

Our pruning technique is based on the following theorem.

Theorem 1 ([12]) The Fisher information matrix of a three-layer perceptron at a parameter $\boldsymbol{\theta} = (w_{11}, \ldots, w_{MH}, \eta_1, \ldots, \eta_H, \zeta_{11}, \ldots, \zeta_{1L}, \ldots, \zeta_{H1}, \ldots, \zeta_{HL})$ is singular if and only if one of the following three conditions holds;

(1) there exists j such that $\boldsymbol{u}_j := (u_{j1}, \ldots, u_{jL})^T = \mathbf{o}$.
(2) there exists j such that $\boldsymbol{w}_j := (w_{1j}, \ldots, w_{Mj})^T = \mathbf{o}$.
(3) there exist different j_1 and j_2 such that $(\boldsymbol{u}_{j_1}, \zeta_{j_1}) = \pm(\boldsymbol{u}_{j_2}, \zeta_{j_2})$.

According to this theorem, we can keep the Fisher information of a network non-singular by checking the above three conditions which indicate the existence of redundant hidden units, and by pruning them if any. The parameter should be modified in (1) and (3) to keep the function unchanged, when the redundant hidden unit is removed. The following is the pruning procedure. Note that we use the relation $s(-t) = 1 - s(t)$ in the derivation of (D). We write $\mathcal{H}_j$ for the jth hidden unit.

[Pruning procedure]
(A) If $\boldsymbol{u}_j = \mathbf{o}$, then
[P1] eliminate $\mathcal{H}_j$ and $\eta_i \mapsto \eta_i + w_{ij} s(\zeta_j)$ for all i.
(B) If $\boldsymbol{w}_j = \mathbf{o}$, then
[P2] eliminate $\mathcal{H}_j$.
(C) If $(\boldsymbol{u}_{j_1}, \zeta_{j_1}) = (\boldsymbol{u}_{j_2}, \zeta_{j_2})$, then
[P3] eliminate $\mathcal{H}_{j_2}$ and $w_{ij_1} \mapsto w_{ij_1} + w_{ij_2}$ for all i.
(D) If $(\boldsymbol{u}_{j_1}, \zeta_{j_1}) = -(\boldsymbol{u}_{j_2}, \zeta_{j_2})$, then
[P4] eliminate $\mathcal{H}_{j_2}$ and $w_{ij_1} \mapsto w_{ij_1} - w_{ij_2}$, $\eta_i \mapsto \eta_i + w_{ij_2}$ for all i.

In most problems, there is little possibility that a Fisher information is perfectly singular. However, we should remove almost redundant hidden units to ensure the stability of the inverse. At the same time, necessary hidden units should not be removed because it results in the increase of the model bias. We must establish a criterion to determine when hidden units should be removed.

We eliminate a hidden unit if the inequality

$$\int \|\boldsymbol{f}(\boldsymbol{x}; \tilde{\boldsymbol{\theta}}) - \boldsymbol{f}(\boldsymbol{x}; \hat{\boldsymbol{\theta}})\|^2 dQ < \frac{A}{N} \tag{19}$$

is satisfied for the LSE estimator $\hat{\boldsymbol{\theta}}$ and a pruned estimator $\tilde{\boldsymbol{\theta}}$ derived from [P1]-[P4]. The constant A is a positive number. If there is no redundant hidden unit, it is known that the LSE estimator approaches to $\boldsymbol{\theta}_o$ in the order of $N^{-1/2}$. The asymptotic behavior in the existence of redundant hidden units is very complicated and still an open problem. Therefore, we put an assumption

$$\mathrm{E}\left[\int \|\boldsymbol{f}(\boldsymbol{x}; \hat{\boldsymbol{\theta}}) - \boldsymbol{f}(\boldsymbol{x}; \boldsymbol{\theta}_{\mathrm{o}})\|^2 dQ(\boldsymbol{x})\right] = O(N^{-1}), \tag{20}$$

and use eq.(19) as heuristics.

We employ a pruning procedure during the batch back-propagation algorithm, in which one training example in a fixed data set is used for one update of the parameter. The condition of eq.(19) is checked for every candidate of a pruned estimator $\tilde{\boldsymbol{\theta}}$ once in T updates. Eq.(19) can be satisfied during the training if the optimal parameter $\boldsymbol{\theta}_0$ is located within the order of $N^{-1/2}$ distance of the parameter set that realizes networks with redundant hidden units, Therefore, the following pruning algorithm is expected to eliminate only almost redundant hidden units. The conditions in (a)-(d) are derived by calculating eq.(19). In the following, we write $\hat{s}_j$ for $s(\hat{\boldsymbol{u}}_j^T \boldsymbol{x} + \hat{\zeta}_j)$.

[BP with hidden unit pruning]
1. $t := 1$.
2. Update $\hat{\boldsymbol{\theta}}$ with respect to $(\boldsymbol{x}^{(t \bmod N)}, \boldsymbol{y}^{(t \bmod N)})$ using the back-propagation rule.
3. If $t \bmod T = 0$, then execute the following four sub-procedure:

(a) If $\|\hat{\boldsymbol{w}}_j\|^2 \int (\hat{s}_j - s(\hat{\zeta}_j))^2 dQ(\boldsymbol{x}) < A/N$, then execute [P1].

(b) If $\|\hat{\boldsymbol{w}}_j\|^2 \int (\hat{s}_j)^2 dQ(\boldsymbol{x}) < A/N$, then execute [P2].

(c) If $\|\hat{\boldsymbol{w}}_{j_2}\|^2 \int (\hat{s}_{j_2} - \hat{s}_{j_1})^2 dQ(\boldsymbol{x}) < A/N$ for $j_1 \neq j_2$, then execute [P3].

(d) If $\|\hat{\boldsymbol{w}}_{j_2}\|^2 \int (1-\hat{s}_{j_2}-\hat{s}_{j_1})^2 dQ(\boldsymbol{x}) < A/N$ for $j_1 \neq j_2$, then execute [P4].

4. $t := t+1$.

5. If $t > t_{MAX}$, then END. Otherwise go to **2**.

A positive constant A and a natural number T control the possibility of pruning. The constant A should be sufficiently large so that the inverse of a Fisher information matrix can be stably calculated. On the other hand, A should be small enough for the pruning procedure to preserve eliminate necessary hidden units. The optimization of these values is also very difficult, because it requires to know the exact asymptotic behavior of the estimator in the existence of redundant hidden units. Therefore, we decide them heuristically in this paper.

C. Active learning with hidden unit pruning

The pruning procedure keeps the information matrix nonsingular and makes the active learning methods applicable to a MLP even if we first prepare a surplus number of hidden units. The modification of active learning methods is simple. We have only to use the BP with hidden unit pruning instead of the usual back-propagation.

We demonstrate the effect of the modified methods through experiments in which the true function is not included in the MLP model. We use the MLP model with 4 input units, 7 hidden units, and 1 output unit. The true function is given by

$$f(\boldsymbol{x}) = \mathrm{erf}(x_1),$$

where $\boldsymbol{x} = (x_1, x_2, x_3, x_4)$ and $\mathrm{erf}(t)$ is the *error function* defined by

$$\mathrm{erf}(t) = \sqrt{\frac{2}{\pi}} \int_0^t e^{-x^2} dx.$$

The graph of the error function resembles that of the sigmoidal function, while they never coincide by any affine transforms. The model has many almost redundant hidden units in this case. We set $Q = N(0, 9I_4)$. The final number of training data is 300. The deviation of the noise added to the output is 0.01. In the parametric active learning method, the mixture model of 5 normal distributions is used for $r(\boldsymbol{x}; \boldsymbol{v})$. To save the computational cost, we generate 5 data at one time. When new data are generated, all data are presented 50000 times cyclically in the BP training, and the pruning conditions are checked every 100 updates of parameters from 40000th through 50000th cycle. In the multi-point-search active learning, n candidates are generated for selecting nth training data. In this case, all the data are presented 5000 times cyclically each time a new data is added, and the pruning condition is checked every 100 updates from 4000th through 5000th cycle.

Fig.8 shows the average of generalization errors for 10 sets of training data. We find that the active learning methods reduce the error remarkably, though the bias-free assumption is not satisfied. Fig.9 shows a typical learning curve of the multi-point-search method, and the transition of the number of hidden units during the learning. We see the elimination of redundant hidden units. The generalization error is reduced by both the effect of pruning and active learning.

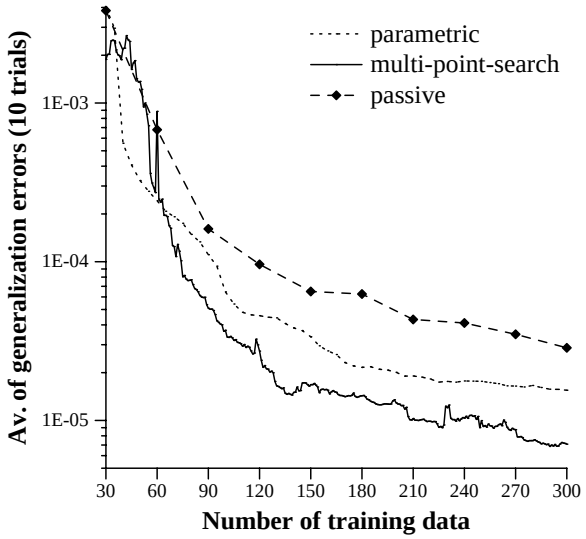

Fig. 8. Active/Passive learning : $f(\boldsymbol{x}) = \mathrm{erf}(x_1)$.

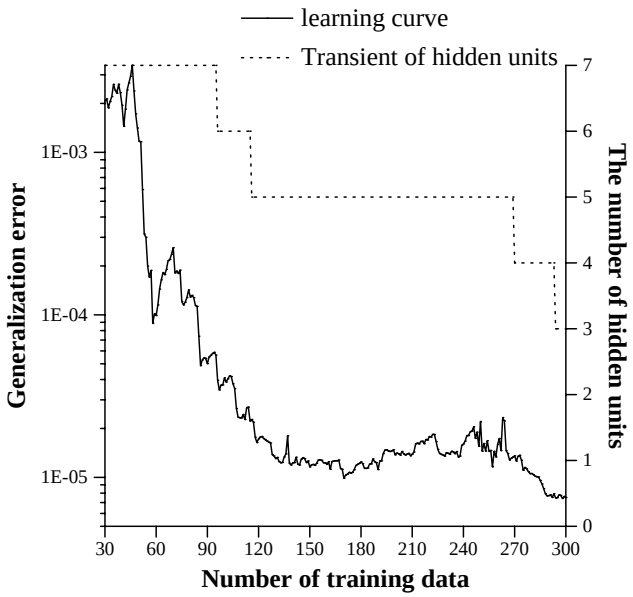

Fig. 9. A typical learning curve of multi-point-search active learning.

V. Application to a color conversion problem

We apply our active learning methods with the pruning technique for a color conversion problem, which is found in many color reproduction systems using CMY (cyan, magenta, and yellow) ink. The problem is to simulate a specific color reproduction system like a color printer, which produces a color print for a given CMY input signal. The print result can be physically measured and represented by

a color system like RGB. It is very important to know the function from CMY to RGB of a specific system to achieve accurate color reproduction. It is known that the function from CMY to RGB can be theoretically given by the Neugebauer equations ([23]). However, a practical system like a color xerography has a very complicated mechanism, and the theoretical equations cannot predict the actual result with high accuracy. The neural network approach is one of the methods to approximate the non-linear relation of the color conversion ([24]). Moreover, since the precise measurement of color is costly, active learning is a promising way to simulate the system.

In this paper, to demonstrate the effectiveness of our active learning methods, we estimate the relation of the Neugebauer equations instead of using a real color reproduction system. It is known that the Neugebauer equations approximate the real system well in the case of offset printing. Then, if we can verify effectiveness in estimating the theoretical equations, we can also expect the effectiveness in approximating a real system.

The model which we use has 3 input and 3 output units. The initial number of hidden units is 8. For the parameters of the Neugebauer equations, we use the relation in [25]. We add an independent Gaussian noise $N(0, 10^{-4}I_3)$ to the output of the Neugebauer equations to simulate a measurement noise. Since we have no meaningful reason to assume a special environmental distribution, we use the uniform distribution on $[0,1]^3$ for Q. After the initial training with 30 examples which are given passively, we train a network by the parametric and multi-point-search active learning method, collecting training samples one by one up to 150. In the parametric method, we use a mixture model of 8 normal distributions restricted on $[0,1]^3$. In the multi-point-search method, the number of candidates is $\left\lceil \sqrt{10(n-30)} \right\rceil$.

Fig.10 shows the average of generalization errors for 30 trials with different initial training examples. We can see the results of the active learning methods remarkably outperform the result of passive learning. If we evaluate their effect by $l_{para}(i)/l_{passive}(i)$, where $l_{para}(i)$ and $l_{passive}(i)$ are the value of the graphs at ith data for parametric active learning and passive learning respectively, the average of $l_{para}(i)/l_{passive}(i)$ for $i = 50, \ldots, 100$ is 0.80, and the average for $i = 50, \ldots, 150$ is 0.85. The effect of the multi-point-search method in the same manner is 0.61 for $i = 50, \ldots, 100$, and 0.67 for $i = 50, \ldots, 150$. These results clearly show the effectiveness of our methods in this real-world application. The multi-point-search method shows a better result than the parametric method throughout the training, and the advantage of the latter method over passive learning is smaller at the final stage of learning. One of the reasons of this is the probabilistic aspect as described in Section IV. Especially, it is difficult to find a suitable density family on a compact input space. This problem always arises if the input space is bounded, and is a disadvantage of the parametric method.

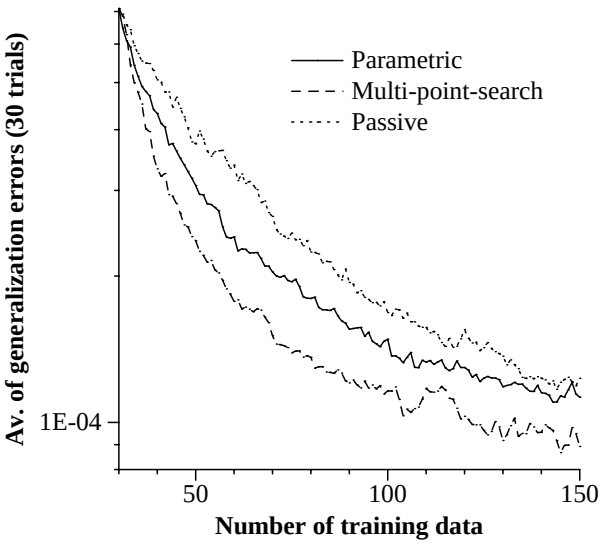

Fig. 10. Active learning of a color conversion problem

VI. Conclusion

We discussed statistical active learning methods for the purpose of applying them to the multilayer perceptron model. We explained the problem of local minima in active learning of neural networks, and proposed two probabilistic active learning methods to prevent local minima. This problem does not appear in linear models, in which the least square error estimator can be solved directly.

We explained the importance of model selection especially in active learning. The derivation of many active learning methods requires that the model includes the true function. This is essential to the effect of active learning, while we cannot assure it in many practical applications. On the other hand, too many hidden units make the active learning methods inapplicable because of the singularity of Fisher information matrixes. To solve this problem, we proposed the active learning method with pruning to keep the Fisher information nonsingular, based on the theorem that clarifies the singularity condition of a Fisher information matrix of a three-layer perceptron. Experimental results showed that active learning with pruning eliminated surplus hidden units and had a remarkable effect of reducing the generalization error.

Acknowledgments

The author would like to express gratitude to Dr. Sumio Watanabe for his encouragement and helpful comments.

References

[1] V.V. Fedorov. *Theory of Optimal Experiments*. Academic Press, New York, 1972.

[2] R.H. Myers, A.I. Khuri, and W.H. Carter, Jr. Response surface methodology: 1966-1988. *Technometrics*, 31(2):137–157, 1989.

[3] D. MacKay. Information-based objective functions for active data selection. *Neural Computation*, 4(4):305–318, 1992.

[4] D.A. Cohn. Neural network exploration using optimal experiment. In J. Cowan et al., editor, *Advances in Neural Information Processing Systems 6*, pages 679–686, San Meteo, 1994. Morgan Kaufmann.

[5] P. Sollich. Query construction, entropy and generalization in neural network models. *Physical Review E*, 49:4637–4651, 1994.

[6] K. Fukumizu and S. Watanabe. Error estimation and learning data arrangement for neural networks. In *Proceedings of IEEE International Conference on Neural Networks*, volume 2, pages 777–780, June 1994.

[7] H. Cramér. *Mathematical method of statistics*, pages 497–506. Princeton University Press, Princeton, NJ, 1946.

[8] H. Akaike. A new look at the statistical model identification. *IEEE Trans. Automatic Control*, 19(6):716–723, 1974.

[9] H. White. Learning in artificial neural networks: a statistical perspective. *Neural Computation*, 1:425–464, 1989.

[10] N. Murata, S. Yoshizawa, and S. Amari. Network information criterion – determining the number of hidden units for an artificial neural network model. *IEEE Trans. Neural Networks*, Vol.5, No.9:865–872, 1994.

[11] S. Watanabe and K. Fukumizu. Probabilistic design of layered neural networks based on their unified framework. *IEEE Transaction on Neural Networks*, 6(3), 1995.

[12] K. Fukumizu. A regularity condition of the information matrix of a multilayer perceptron network. *Neural Networks*, Vol.9, No.5:871–879, 1996.

[13] B. Efron and R. Tibshirani. *An introduction to the bootstrap*. Chapman and Hall, New York, 1993.

[14] M. Stone. Cross-validatory choice and assessment of statistical predictions. *J. Royal Statist. Soc.*, 36:111-133, 1974.

[15] V.N. Vapnik. *Estimation of dependences based on empirical data*. Springer-Verlag, New York, 1982.

[16] J. Moody. The effective number of parameters: An analysis of generalization and regularization in nonlinear learning systems. In *Advances in Neural Information Processing Systems* ,

[17] K. Fukumizu. Active learning in multilayer perceptrons. In D. S. Touretzky et al., editor, *Advances in Neural Information Processing Systems 8*, pages 295–301, Cambridge, 1996. MIT Press.

[18] J. Kiefer and J. Wolfowitz. The equivalence of two extremum problem. *Canadian Journal of Mathematics*, 12:363–366, 1960.

[19] J. Kindermann, G. Paass, and F. Weber. Query construction for neural networks using the bootstrap. In *Proceedings of International Conference on Artificial Neural Networks 95*, pages 135–140, 1995.

[20] G. Cybenco. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2(4):303–314, 1989.

[21] K. Funahashi. On the approximate realization of continuous mapping by neural networks. *Neural Network*, 2:183–192, 1989.

[22] K. Hornik, M. Stinchcombe, and H. White. Multi-layer feedforward networks are universal approximators. *Neural Networks*, 2:359–366, 1989.

[23] H.E.J. Neugebauer. Die theoretishen Grundlagen des Mehrfarben buchdrucks. *Zeitschrift für wissenschaftliche Photographik, Photophysik, und Photochemie*, 36(4):73–89, 1937.

[24] T. Iga, Y. Arai, and S. Usui. Trend of a present color management technology in the industry. In *Proceedings of 5th International Conference on Neural Information Processing (ICONIP'98)*, pages 40–43, 1998.

[25] J.A.C. Yule. *Principles of color reproduction*, Appendix E. John Willey & Sons, New York, 1967.