Incremental Projection Learning for Optimal Generalization

Masashi Sugiyama Hidemitsu Ogawa

Department of Computer Science,
Graduate School of Information Science and Engineering,
Tokyo Institute of Technology,

2-12-1, O-okayama, Meguro-ku, Tokyo, 152-8552, Japan.

sugi@og.cs.titech.ac.jp
http://ogawa-www.cs.titech.ac.jp/~sugi/

Abstract

In many practical situations in supervised learning, it is often expected to further improve the generalization capability after the learning process has been completed. One of the common approaches to improving the generalization capability is to add training examples. In view of the learning methods of human beings, it seems natural to build posterior learning results upon prior results, which is generally referred to as incremental learning. In this paper, a method of incremental projection learning (IPL) is presented. IPL provides exactly the same learning result as that obtained by batch projection learning. The effectiveness of the presented method is demonstrated through computer simulations.

Keywords

Multilayer feedforward neural networks; Generalization capability; Incremental learning; Projection learning; Reproducing kernel Hilbert space (RKHS)

Nomenclature

$\mathbf{C}^m$	m-dimensional unitary space
H	Hilbert space
$\langle \cdot, \cdot \rangle$	inner product in H or $\mathbf{C}^m$
$\| \cdot \|$	norm in H or $\mathbf{C}^m$
$\cdot \otimes \bar{\cdot}$	Neumann-Schatten product
$K(\cdot, \cdot)$	reproducing kernel of H
x_i	input of neural networks
y_i	output of neural networks
(x_i, y_i)	training examples
$y^{(m)}$	m-dimensional vector consisting of $\{y_i\}_{i=1}^m$
n_i	additive noise
$n^{(m)}$	m-dimensional vector consisting of $\{n_i\}_{i=1}^m$
w_i	weight of neural networks
u_i	basis function in neural networks
f	function of learning target
f_m	learning result from a set of m training examples
A_m	sampling operator for a set of m training examples
X_m	learning operator for a set of m training examples
ψ_i	sampling function of the i-th training example
Q_m	noise correlation matrix of a set of m training examples
q_{m+1}	noise covariance of the $(m+1)$-st training example
σ_{m+1}	noise variance of the $(m+1)$-st training example
I_m	identity matrix on $\mathbf{C}^m$
$e_i^{(m)}$	i-th vector of the standard basis in $\mathbf{C}^m$
E_n	ensemble average over noise
A^*	adjoint operator of A
$A^\dagger$	Moore-Penrose generalized inverse of A
$\mathcal{R}(A)$	range of A
$\mathcal{N}(A)$	null space of A
P_S	orthogonal projection operator onto a subspace S
α_{m+1}, β_{m+1}	scalars
$\tilde{\psi}_{m+1}$, ξ_{m+1}, $\tilde{\xi}_{m+1}$, ζ_{m+1}	functions in H
s_{m+1}, t_{m+1}	elements in $\mathbf{C}^m$
U_m	operator from $\mathbf{C}^m$ to $\mathbf{C}^m$
V_m	operator from H to H
Y_m	operator from $\mathbf{C}^m$ to H

1 Introduction

Supervised learning is obtaining an underlying rule by using training examples sampled from the environment. Neural networks (NNs) are expected not only to memorize the training examples, but also to acquire the generalization capability.

In many practical situations in neural network learning, it is often expected to further improve the generalization capability after the learning process has been completed. One of the common approaches to improving the generalization capability is to add training examples to the neural network. In view of the learning methods of human beings, it seems natural to build posterior learning results upon prior results. This learning method is generally called *incremental learning*. Incremental learning also plays an important role when we work on *active learning*, which has been extensively studied recently (MacKay, 1992b; Cohn, 1996; Fukumizu, 1996; Vijayakumar *et al.*, 1998; Sugiyama & Ogawa, 2000). In these methods, the choice of training examples to be learned next is determined by analyzing the intermediate learning result. Incremental learning is, therefore, indispensable for performing active learning.

Many incremental learning methods have been devised so far. Many of them are based on the idea of allocating novel hidden units when new training examples are added, and adjusting weights on the connections to the novel units. In this approach, the number of hidden units tends to increase with a gain in the total number of training examples. In order to prevent too many hidden units being allocated, Platt (1991) introduced the *novelty criteria* and proposed an incremental learning algorithm. Briefly, the algorithm can be described as follows: First, learning starts with no hidden units. The NN grows by allocating a new hidden unit whose input-output relation is a radial basis function based on the novelty of the training example given sequentially. If the training example has no novelty, then the existing parameters of the NN are adjusted by the least mean squares algorithm to fit the additional training example without adding novel units. If the training example satisfies the novelty criteria, then a new hidden unit is added and the weights on the connections to the unit are adjusted. A neural network trained by the algorithm is called the *resource allocating network (RAN)*. Kadirkamanathan and Niranjan (1993) gave an interpretation of RAN from the functional analytic point of view and showed that better approximations can be obtained by using the extended Kalman filter instead of the least mean squares algorithm. Moreover, Molina and Niranjan (1996) improved the algorithm by making it applicable to neural networks whose number of hidden units is limited. Yingwei *et al.* (1997, 1998) combined RAN with a procedure for pruning redundant hidden units.

In RAN and its derivatives, however, there is a crucial problem: Old training examples tend to be forgotten as the sequential process progress. In order to avoid the forgetfulness, the idea of storing some of the learned training examples in temporary memory has arisen (Yoneda *et al.*, 1992; Yamakawa *et al.*, 1993). But, it is difficult to store training examples sampled from a wide region into a limited memory space with these methods. Yamauchi and Ishii (1995) took an interesting approach to deal with the problem: First, a region which will be interfered with by incremental learning is inferred, and artificial training

examples which will prevent the interference is created. Then, incremental learning takes place by using both newly added and created training examples.

Although RAN, its derivatives, and other incremental learning methods (Zhang, 1994; Vyšniauskas *et al.*, 1995; Jutten & Chentouf, 1995; Vijayakumar & Schaal, 1998) improve the efficiency in computation, the optimal generalization capability is not theoretically guaranteed. Recently, Amari (1998) proposed the *natural gradient* method for training stochastic neural networks. He proved that the generalization capability obtained by the on-line version of the natural gradient method is asymptotically the same as that obtained by the batch version. A similar method is also given in Murata (1999). Even in these methods, however, the optimal generalization capability is not guaranteed in the non-asymptotic case. In practice, the number of training examples is always finite.

Ogawa (1989, 1992) formulated the NN learning problem as an *inverse problem* from the functional analytic point of view. In his papers, it has been shown that optimal image restoration filters such as the *projection filter* (Ogawa, 1987) and the *Wiener filter* (Ogawa & Oja, 1986) can be applied to the NN learning problem. These filters are called *projection learning*, *Wiener learning etc.* in the learning case. Within his framework, incremental Wiener learning in the absence of noise has been devised (Vijayakumar & Ogawa, 1998), in which the generalization capability is proved to be exactly the same as that obtained by batch Wiener learning even in the non-asymptotic case.

In this paper, we present a method of incremental projection learning in the presence of noise. This method provides exactly the same generalization capability as that obtained by batch projection learning. This paper is organized as follows: Section 2 formulates the NN learning problem. In Section 3 and Section 4, a method of incremental projection learning is proposed. Finally, Section 5 is devoted to computer simulations, demonstrating the effectiveness of the proposed incremental learning method. Properties of the proposed method will be studied in a separate paper (Sugiyama & Ogawa, 2000a).

2 Formulation of NN learning problem

In this section, the NN learning problem is formulated (see Ogawa, 1992).

Let us consider a learning problem of a three-layer feedforward NN whose numbers of input and output units are L and 1, respectively. The relationship between input $x = (\eta_1, \cdots, \eta_L)^\top$ and output y of the network can be expressed by using a function $f_0(x)$ of L variables as

$$y = f_0(x). \tag{1}$$

The NN learning problem is to obtain the optimal approximation to a target function f from a set of m training examples made up of inputs $x_i \in \mathbf{R}^L$ and corresponding outputs $y_i \in \mathbf{C}$:

$$\{(x_i, y_i) | y_i = f(x_i) + n_i\}_{i=1}^m, \tag{2}$$

where y_i is degraded by additive noise n_i.

In many NN learning methods devised so far, learning algorithms are built upon a certain architecture of NNs, i.e., a fixed number of hidden units, each with a prespecified

sigmoidal or Gaussian function. However, the restrictions sometimes prevent us from obtaining the optimal approximation. Therefore, we may divide our NN learning problem into two steps: The first step performs a function approximation from training examples, and a NN representing the approximated function is constructed in the second step.

To begin with, we explain the function approximation problem corresponding to the first step. Let $n^{(m)}$ and $y^{(m)}$ denote m-dimensional vectors whose i-th elements are n_i and y_i, respectively. $y^{(m)}$ is called a *sample value vector*, and a space to which $y^{(m)}$ belongs is called a *sample value space*. In this paper, the underlying function $f(x)$ is assumed to belong to a reproducing kernel Hilbert space H (Aronszajn, 1950; Bergman, 1970; Saitoh, 1988, 1997). If H is unknown, then it can be estimated by model selection methods (Akaike, 1974; MacKay, 1992a; Murata *et al.*, 1994; Sugiyama & Ogawa, 2001b, 2001c). Let $\mathcal{D}$ be the domain of f. The reproducing kernel $K(x, x')$ is a bivariate function defined on $\mathcal{D} \times \mathcal{D}$ which satisfies the following conditions:

- For any fixed x' in $\mathcal{D}$, $K(x, x')$ is a function of x in H.
- For any function f in H and for any x' in $\mathcal{D}$, it holds that

$$\langle f(\cdot), K(\cdot, x') \rangle = f(x'), \tag{3}$$

  where $\langle \cdot, \cdot \rangle$ denotes the inner product in H.

Note that the reproducing kernel is unique if it exists. In the theory of the Hilbert space, arguments are developed by regarding a function as a point in that space. Thus, the value of a function at a point can not be discussed within the general framework of the Hilbert space. However, if the Hilbert space has the reproducing kernel, then it is possible to deal with the value of a function at a point. Indeed, if a function $\psi_i(x)$ is defined as

$$\psi_i(x) = K(x, x_i), \tag{4}$$

then the value of f at a sample point x_i can be expressed as

$$f(x_i) = \langle f, \psi_i \rangle. \tag{5}$$

For this reason, ψ_i is called a *sampling function*. Let A_m be an operator mapping f to an m-dimensional vector whose i-th element is $f(x_i)$. A_m is called a *sampling operator*. Note that A_m is always a linear operator even when we are concerned with a non-linear function f. Indeed, A_m can be expressed by using the *Neumann-Schatten product*[1] as

$$A_m = \sum_{i=1}^{m} \left(e_i^{(m)} \otimes \overline{\psi_i} \right), \tag{6}$$

[1]For any fixed g in a Hilbert space H_1 and any fixed f in a Hilbert space H_2, the Neumann-Schatten product $(f \otimes \overline{g})$ is an operator from H_1 to H_2 defined by using any $h \in H_1$ as (Schatten, 1970)

$$(f \otimes \overline{g})h = \langle h, g \rangle f.$$

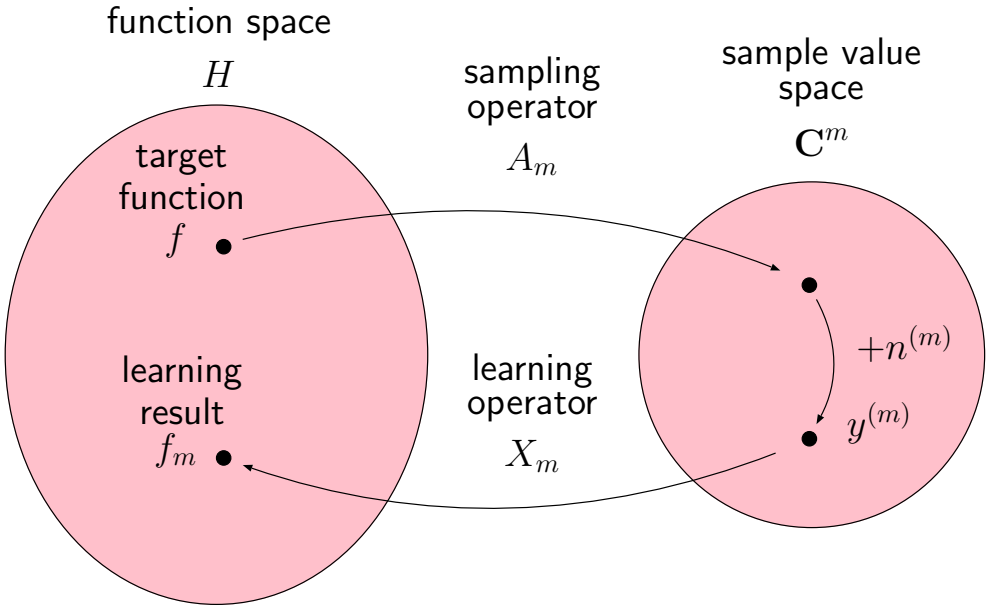


Figure 1: NN learning as an inverse problem.

where $e_i^{(m)}$ is the i-th vector of the so-called standard basis in $\mathbf{C}^m$, i.e., $e_i^{(m)}$ is the m-dimensional vector where all elements are zero except the i-th element which is equal to one. Then, the relationship between f and $y^{(m)}$ can be expressed as

$$y^{(m)} = A_m f + n^{(m)}. \tag{7}$$

Let us denote a learning result obtained from m training examples by f_m, and the relationship between $y^{(m)}$ and f_m as

$$f_m = X_m y^{(m)}, \tag{8}$$

where X_m is called a *learning operator*. Consequently, the first step of the NN learning problem can be reformulated as an inverse problem of obtaining X_m which provides the best approximation f_m to f under a certain criterion (Fig.1). Since image and signal restoration problems discussed in Ogawa (1987) and Ogawa *et al.* (1989) are also formulated as the same form of inverse problems, the criteria for optimal restoration discussed in these papers, e.g. the Wiener filter criterion and the projection filter criterion, can be applied to our function approximation problem.

Now we go on to the second step, i.e., the construction of a NN representing f_m. In this step, the number N of hidden units, basis functions $\{u_i(x)\}_{i=1}^N$, and weights $\{w_i\}_{i=1}^N$ on hidden-output connections are determined (Fig.2). In conventional neural networks, the following basis function is commonly used:

$$u_i(x) = \sigma(\sum_{j=1}^{L} w_{ij}\eta_j), \tag{9}$$

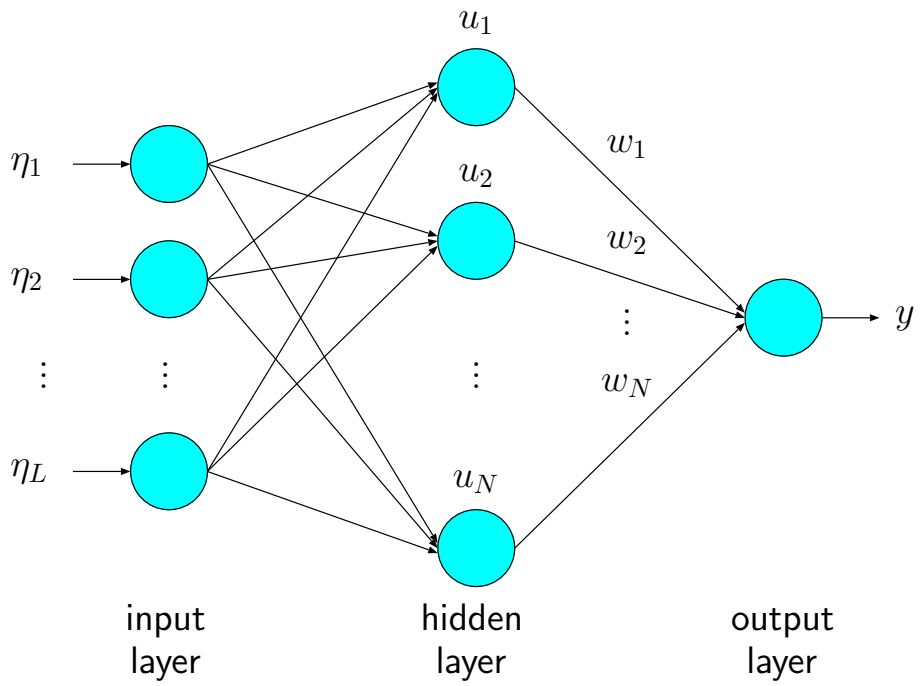


Figure 2: Optimally generalizing neural network (OGNN).

where $\sigma(\cdot)$ is a sigmoidal activation function and w_{ij} is a weight on the connection between the j-th input unit and the i-th hidden unit. A NN representing a function obtained in the first step is called an *optimally generalizing NN* (OGNN). A general construction method of OGNNs was given in Ogawa (1992, 1995). The method shows that there exist infinite degrees of freedom in OGNNs. Utilizing these degrees of freedom effectively, Nakazawa and Ogawa (1996) gave a robust construction method of OGNNs. NNs constructed by the method are specifically resistant to noise on the output of hidden units and connection faults.

In this paper, we focus on the function approximation problem corresponding to the first step and devise an incremental learning method.

3 Exact incremental learning

In this section, a method of incremental projection learning (IPL) is derived.

3.1 Learning criterion

As mentioned in the previous section, function approximation is performed on the basis of a learning criterion. In this paper, we adopt the projection learning criterion. Here, the definition of projection learning is reviewed.

Let us restrict our discussion within the case where the learning operator X_m is linear. In this case, it follows from eqs.(8) and (7) that the learning result f_m can be decomposed as

$$f_m = X_m A_m f + X_m n^{(m)}. \tag{10}$$

The first and second terms of eq.(10) are called the *signal* and *noise components* of f_m, respectively. Let E_n be the ensemble average over noise. Then, it follows from eq.(10) that

$$E_n f_m = X_m A_m f, \tag{11}$$

and hence the average of f_m belongs to $\mathcal{R}(X_m A_m)$, where $\mathcal{R}(\cdot)$ denotes the range of an operator. Let P_S be the orthogonal projection operator onto a subspace S. In order to minimize the bias of f_m, $X_m A_m f$ should agree with the orthogonal projection of f onto $\mathcal{R}(X_m A_m)$:

$$X_m A_m f = P_{\mathcal{R}(X_m A_m)} f. \tag{12}$$

From Albert (1972), the operator equation

$$X_m A_m = P_S \tag{13}$$

has a solution if and only if $S \subset \mathcal{R}(A_m^*)$, where A_m^* denotes the adjoint operator of A_m. Since bigger $\mathcal{R}(X_m A_m)$ provides better approximation, we adopt the largest one:

$$\mathcal{R}(X_m A_m) = \mathcal{R}(A_m^*). \tag{14}$$

For this reason, $\mathcal{R}(A_m^*)$ is called the *approximation space*. In order to reduce the generalization error, the variance of f_m should be minimized under the constraint of eq.(13) with $S = \mathcal{R}(A_m^*)$. This learning method is called projection learning:

Definition 1 (Projection learning) *(Ogawa, 1987) An operator X_m is called the projection learning operator if X_m minimizes the functional*

$$J_P[X_m] = E_n \| X_m n^{(m)} \|^2 \tag{15}$$

under the constraint

$$X_m A_m = P_{\mathcal{R}(A_m^*)}. \tag{16}$$

Let I_m and Y_m be the identity matrix on $\mathbf{C}^m$ and an arbitrary operator from $\mathbf{C}^m$ to H, respectively, and

$$Q_m = E_n \left(n^{(m)} \otimes \overline{n^{(m)}} \right), \tag{17}$$

$$U_m = A_m A_m^* + Q_m, \tag{18}$$

$$V_m = A_m^* U_m^\dagger A_m, \tag{19}$$

where † stands for the *Moore-Penrose generalized inverse*[2]. Then, the following proposition holds.

[2]An operator X is called the Moore-Penrose generalized inverse of an operator A if X satisfies the following four conditions (Albert, 1972).

$$AXA = A, \;\; XAX = X, \;\; (AX)^* = AX, \;\; \text{and} \;\; (XA)^* = XA.$$

Note that the Moore-Penrose generalized inverse is unique and denoted as $A^\dagger$.

Proposition 1 *(Ogawa, 1987) A general form of the projection learning operator is given as*

$$X_m = V_m^\dagger A_m^* U_m^\dagger + Y_m(I_m - U_m U_m^\dagger). \tag{20}$$

There are various methods of calculating the projection learning operator X_m and the projection learning result f_m by matrix operation. Here, we show one of the simplest methods valid for a finite dimensional Hilbert space H.

When the dimension of H, denoted by μ, is finite, functions in H can be expressed in the form of

$$f(x) = \sum_{j=1}^{\mu} a_j \varphi_j(x), \tag{21}$$

where $\{\varphi_j\}_{j=1}^{\mu}$ is an orthonormal basis in H and $\{a_j\}_{j=1}^{\mu}$ is its coefficients. Let us consider a μ-dimensional parameter space in which functions in H are expressed as

$$f = (a_1, a_2, \cdots, a_\mu)^\top. \tag{22}$$

If we regard this parameter space as H, then the sampling function ψ_i is expressed as

$$\psi_i = (\varphi_1(x_i), \varphi_2(x_i), \cdots, \varphi_\mu(x_i))^*, \tag{23}$$

where $(a_1, a_2, \cdots, a_\mu)^*$ denotes the complex conjugate of the transpose of $(a_1, a_2, \cdots, a_\mu)$. Hence, the sampling operator A_m becomes an $m \times \mu$ matrix whose (i, j)-element is

$$[A_m]_{ij} = \varphi_j(x_i). \tag{24}$$

In practice, the calculation of the Moore-Penrose generalized inverse is sometimes unstable. To overcome the unstableness, we recommend to use *Tikhonov's regularization* (Tikhonov & Arsenin, 1997):

$$A_m^\dagger \longleftarrow A_m^*(A_m A_m^* + \epsilon I)^{-1}, \tag{25}$$

where ϵ is a small constant, say $\epsilon = 10^{-4}$.

Under the projection learning criterion, we devise an incremental learning method in the presence of noise. We call the method *incremental projection learning* (IPL). It has been shown that learning results obtained by projection learning are invariant under the inner product used in the sample value space (Yamashita & Ogawa, 1992). Hence, the Euclidean inner product is adopted without loss of generality.

3.2 Incremental projection learning

Let us consider the case where the $(m+1)$-st training example (x_{m+1}, y_{m+1}) is added to f_m. It follows from eq.(8) that a learning result f_{m+1} obtained from $(m+1)$ training examples $\{(x_i, y_i)\}_{i=1}^{m+1}$ in a batch manner can be expressed as

$$f_{m+1} = X_{m+1} y^{(m+1)}. \tag{26}$$

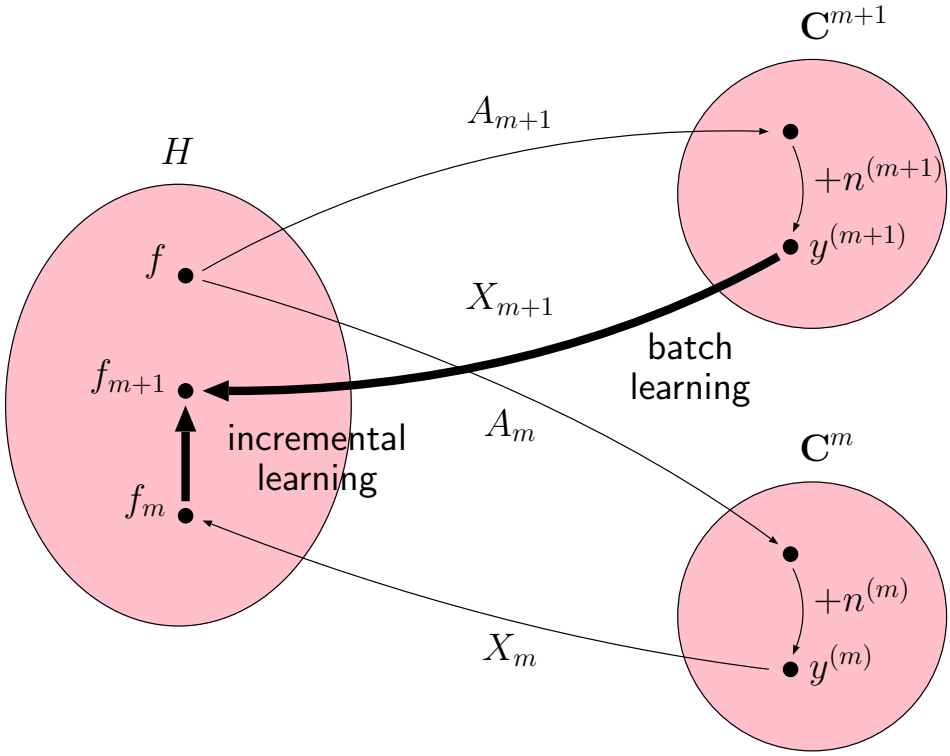


Figure 3: Exact incremental learning and batch learning.

The suffix $m+1$ indicates the total number of training examples. In order to devise an exact incremental learning method, let us calculate f_{m+1} by using f_m and (x_{m+1}, y_{m+1}), as illustrated in Fig.3.

Let the noise characteristics of an additional training example (x_{m+1}, y_{m+1}) be

$$q_{m+1} = E_n(\overline{n_{m+1}} n^{(m)}), \tag{27}$$
$$\sigma_{m+1} = E_n|n_{m+1}|^2, \tag{28}$$

where $\overline{n_{m+1}}$ denotes the complex conjugate of n_{m+1}. Note that q_{m+1} is an m-dimensional vector while σ_{m+1} is a scalar. Let an $(m+1) \times m$ matrix Γ_{m+1}, m-dimensional vectors s_{m+1}, t_{m+1}, and a scalar α_{m+1} be

$$\Gamma_{m+1} = \sum_{i=1}^{m} \left(e_i^{(m+1)} \otimes \overline{e_i^{(m)}} \right), \tag{29}$$
$$s_{m+1} = A_m \psi_{m+1} + q_{m+1}, \tag{30}$$
$$t_{m+1} = U_m^{\dagger} s_{m+1}, \tag{31}$$
$$\alpha_{m+1} = \psi_{m+1}(x_{m+1}) + \sigma_{m+1} - \langle t_{m+1}, s_{m+1} \rangle. \tag{32}$$

Γ_{m+1} expands an m-dimensional vector h into an $(m+1)$-dimensional vector, while Γ_{m+1}^*

removes the $(m+1)$-th element as follows:

$$\begin{pmatrix} h \\ 0 \end{pmatrix} = \Gamma_{m+1} h, \quad h = \Gamma_{m+1}^{*} \begin{pmatrix} h \\ c \end{pmatrix}, \tag{33}$$

where c is a scalar. From eq.(5), $\psi_{m+1}(x_{m+1})$ in eq.(32) agrees with $\|\psi_{m+1}\|^2$. Then, we have the following lemmas.

Lemma 1 *A_{m+1} can be expressed by using A_m as*

$$A_{m+1} = \Gamma_{m+1} A_m + e_{m+1}^{(m+1)} \otimes \overline{\psi_{m+1}}. \tag{34}$$

Lemma 2 *U_{m+1} can be expressed by using U_m as*

$$U_{m+1} = \begin{pmatrix} U_m & s_{m+1} \\ s_{m+1}^{*} & \psi_{m+1}(x_{m+1}) + \sigma_{m+1} \end{pmatrix}, \tag{35}$$

where s_{m+1}^{*} is the complex conjugate of the transpose of s_{m+1}.

Lemma 3 *It holds that*

$$s_{m+1} \in \mathcal{R}(U_m), \tag{36}$$

$$\alpha_{m+1} \geq 0. \tag{37}$$

Proofs of all lemmas are given in Appendix. In these proofs, the following proposition plays an important role.

Proposition 2 *(Albert, 1972) Let S_m, s, and ν be a non-negative[3] m-dimensional matrix, an m-dimensional vector, and a scalar, respectively. Let S_{m+1} be an $(m+1)$-dimensional matrix defined as*

$$S_{m+1} = \begin{pmatrix} S_m & s \\ s^{*} & \nu \end{pmatrix}. \tag{38}$$

Then, it holds that

$$s \in \mathcal{R}(S_m), \tag{39}$$

$$\nu - \langle S_m^{\dagger} s, s \rangle \geq 0, \tag{40}$$

if and only if S_{m+1} is non-negative.

From Albert (1972), it is easily confirmed that $\mathrm{rank}(U_{m+1}) = \mathrm{rank}(U_m)$ if and only if $\alpha_{m+1} = 0$. Whether α_{m+1} is zero or not is crucial in the derivation of IPL. First, we shall discuss the case where $\alpha_{m+1} = 0$.

[3] An operator T is said to be *non-negative* if $\langle Tf, f \rangle \geq 0$ for any f. If $\langle Tf, f \rangle > 0$ for any $f \neq 0$, T is said to be *positive definite*.

Theorem 1 *If* $\alpha_{m+1} = 0$*, then*

$$f_{m+1} = f_m. \tag{41}$$

A proof of Theorem 1 is given in Section 4. Theorem 1 says that the learning result does not change at all by adding (x_{m+1}, y_{m+1}) if $\alpha_{m+1} = 0$. Generally, the training examples which cause $f_{m+1} = f_m$ are regarded as redundant. However, as shown in Sugiyama and Ogawa (2001a), the redundancy of training examples can not be judged by simply comparing f_{m+1} with f_m.

Now, we shall discuss the case where $\alpha_{m+1} > 0$. Let $\mathcal{N}(A_m)$ and $P_{\mathcal{N}(A_m)}$ be the null space of A_m and the orthogonal projection operator onto $\mathcal{N}(A_m)$, respectively. Let functions $\tilde{\psi}_{m+1}$, ξ_{m+1}, $\tilde{\xi}_{m+1}$, and a scalar β_{m+1} be defined as

$$\begin{aligned}
\tilde{\psi}_{m+1} &= P_{\mathcal{N}(A_m)}\psi_{m+1} \quad (= \psi_{m+1} - A_m^{\dagger}A_m\psi_{m+1}), && (42)\\
\xi_{m+1} &= \psi_{m+1} - A_m^{*}t_{m+1}, && (43)\\
\tilde{\xi}_{m+1} &= V_m^{\dagger}\xi_{m+1}, && (44)\\
\beta_{m+1} &= y_{m+1} - f_m(x_{m+1}) - \langle y^{(m)} - A_m f_m, t_{m+1}\rangle. && (45)
\end{aligned}$$

Then, we have the following theorem.

Theorem 2 (Incremental projection learning) *When* $\alpha_{m+1} > 0$*, a posterior projection learning result* f_{m+1} *can be obtained by using prior results* f_m*,* A_m*,* $U_m^{\dagger}$*,* $V_m^{\dagger}$*, and* $y^{(m)}$ *as*

$$f_{m+1} = f_m + \beta_{m+1}\zeta_{m+1}, \tag{46}$$

where ζ_{m+1} *is given as follows:*

(a) When $\psi_{m+1} \notin \mathcal{R}(A_m^{*})$*,*

$$\zeta_{m+1} = \frac{\tilde{\psi}_{m+1}}{\tilde{\psi}_{m+1}(x_{m+1})}. \tag{47}$$

(b) When $\psi_{m+1} \in \mathcal{R}(A_m^{*})$*,*

$$\zeta_{m+1} = \frac{\tilde{\xi}_{m+1}}{\alpha_{m+1} + \langle \tilde{\xi}_{m+1}, \xi_{m+1}\rangle}. \tag{48}$$

A proof of Theorem 2 is also given in Section 4. Note that learning results obtained by IPL in Theorem 2 are exactly the same as those obtained by batch projection learning. In the second term of the right-hand side of eq.(46), β_{m+1} depends on the value of y_{m+1} while ζ_{m+1} does not. $\tilde{\psi}_{m+1}(x_{m+1})$ in eq.(47) is equivalent to $\|\tilde{\psi}_{m+1}\|^2$ (see eqs.(5) and (42)). The condition $\psi_{m+1} \notin \mathcal{R}(A_m^{*})$ means that ψ_{m+1} is linearly independent of $\{\psi_i\}_{i=1}^{m}$, i.e., the approximation space $\mathcal{R}(A_{m+1}^{*})$ becomes wider than $\mathcal{R}(A_m^{*})$. In contrast, $\psi_{m+1} \in \mathcal{R}(A_m^{*})$ means that ψ_{m+1} is linearly dependent of $\{\psi_i\}_{i=1}^{m}$, and hence the approximation space $\mathcal{R}(A_{m+1}^{*})$ is equal to $\mathcal{R}(A_m^{*})$. Whether $\psi_{m+1} \in \mathcal{R}(A_m^{*})$ or not can be easily checked since $\psi_{m+1} \in \mathcal{R}(A_m^{*})$ if and only if

$$P_{\mathcal{N}(A_m)}\psi_{m+1} = \tilde{\psi}_{m+1} = 0. \tag{49}$$

In practice, we recommend to use the following criterion.

$$\text{if } \|\tilde{\psi}_{m+1}\|^2 < \epsilon \text{ then } \psi_{m+1} \in \mathcal{R}(A_m^*),$$

where ϵ is a small constant, say $\epsilon = 10^{-4}$. Properties of IPL are studied in detail in a separated paper (Sugiyama & Ogawa, 2001a).

4 Proofs of Theorem 1 and Theorem 2

This section is devoted to proving Theorem 1 and Theorem 2.

Let an $(m+1)$-dimensional vector h_{m+1}, a scalar γ_{m+1}, and an m-dimensional matrix T_{m+1} be defined as

$$\begin{aligned} h_{m+1} &= e_{m+1}^{(m+1)} - \Gamma_{m+1}(t_{m+1} + X_m^* \xi_{m+1}), & (50)\\ \gamma_{m+1} &= 1 + \|t_{m+1}\|^2, & (51)\\ T_{m+1} &= I_m - \frac{t_{m+1} \otimes \overline{t_{m+1}}}{\gamma_{m+1}}. & (52) \end{aligned}$$

The following lemmas are prepared for proving the theorems.

Lemma 4 *$U_{m+1}^\dagger$ can be expressed by using $U_m^\dagger$ as follows:*

(i) When $\alpha_{m+1} > 0$,

$$U_{m+1}^\dagger = \begin{pmatrix} U_m^\dagger + \frac{t_{m+1} \otimes \overline{t_{m+1}}}{\alpha_{m+1}} & -\frac{t_{m+1}}{\alpha_{m+1}} \\ -\frac{t_{m+1}^*}{\alpha_{m+1}} & \frac{1}{\alpha_{m+1}} \end{pmatrix}. \quad (53)$$

(ii) When $\alpha_{m+1} = 0$,

$$U_{m+1}^\dagger = \begin{pmatrix} T_{m+1} U_m^\dagger T_{m+1} & \frac{T_{m+1} U_m^\dagger t_{m+1}}{\gamma_{m+1}} \\ \frac{(T_{m+1} U_m^\dagger t_{m+1})^*}{\gamma_{m+1}} & \frac{\langle U_m^\dagger t_{m+1}, t_{m+1} \rangle}{\gamma_{m+1}^2} \end{pmatrix}. \quad (54)$$

Lemma 5 *$\alpha_{m+1} = 0$ if and only if the following three conditions hold.*

$$\begin{aligned} \xi_{m+1} &= 0, & (55)\\ q_{m+1} &= Q_m t_{m+1}, & (56)\\ \sigma_{m+1} &= \langle Q_m^\dagger q_{m+1}, q_{m+1} \rangle. & (57) \end{aligned}$$

Lemma 6 *V_{m+1} can be expressed by using V_m as follows:*

(i) When $\alpha_{m+1} > 0$,

$$V_{m+1} = V_m + \frac{\xi_{m+1} \otimes \overline{\xi_{m+1}}}{\alpha_{m+1}}. \quad (58)$$

(ii) When $\alpha_{m+1} = 0$,

$$V_{m+1} = V_m. \tag{59}$$

Lemma 7 *$V_{m+1}^\dagger$ can be expressed by using $V_m^\dagger$ as follows:*

(i) When $\alpha_{m+1} > 0$ *and* $\psi_{m+1} \notin \mathcal{R}(A_m^*)$,

$$\begin{aligned} V_{m+1}^\dagger &= V_m^\dagger + \frac{\alpha_{m+1} + \langle \tilde{\xi}_{m+1}, \xi_{m+1} \rangle}{\tilde{\psi}_{m+1}(x_{m+1})^2} \tilde{\psi}_{m+1} \otimes \overline{\tilde{\psi}_{m+1}} \\ &\quad - \frac{\tilde{\xi}_{m+1} \otimes \overline{\tilde{\psi}_{m+1}} + \tilde{\psi}_{m+1} \otimes \overline{\tilde{\xi}_{m+1}}}{\tilde{\psi}_{m+1}(x_{m+1})}. \end{aligned} \tag{60}$$

(ii) When $\alpha_{m+1} > 0$ *and* $\psi_{m+1} \in \mathcal{R}(A_m^*)$,

$$V_{m+1}^\dagger = V_m^\dagger - \frac{\tilde{\xi}_{m+1} \otimes \overline{\tilde{\xi}_{m+1}}}{\alpha_{m+1} + \langle \tilde{\xi}_{m+1}, \xi_{m+1} \rangle}. \tag{61}$$

(iii) When $\alpha_{m+1} = 0$,

$$V_{m+1}^\dagger = V_m^\dagger. \tag{62}$$

Lemma 8 *X_{m+1} can be expressed by using X_m as*

$$X_{m+1} = X_m \Gamma_{m+1}^* + \left(\zeta_{m+1} \otimes \overline{h_{m+1}} \right) + Y_{m+1} \Gamma_{m+1} (I_m - U_m U_m^\dagger) \Gamma_{m+1}^*, \tag{63}$$

where Y_{m+1} is an arbitrary operator from $\mathbf{C}^{m+1}$ to H and ζ_{m+1} is given as follows:

(i) When $\alpha_{m+1} > 0$ *and* $\psi_{m+1} \notin \mathcal{R}(A_m^*)$, ζ_{m+1} *is given as eq.(47).*

(ii) When $\alpha_{m+1} > 0$ *and* $\psi_{m+1} \in \mathcal{R}(A_m^*)$, ζ_{m+1} *is given as eq.(48).*

(iii) When $\alpha_{m+1} = 0$, ζ_{m+1} *is given as*

$$\zeta_{m+1} = \frac{X_m t_{m+1}}{\gamma_{m+1}} + Y_{m+1} h_{m+1}. \tag{64}$$

Lemma 9 *It holds that*

$$\langle y^{(m+1)}, h_{m+1} \rangle = \beta_{m+1}. \tag{65}$$

Lemma 10 *If* $\alpha_{m+1} = 0$, *then*

$$\beta_{m+1} = 0. \tag{66}$$

Based on the above arrangements, we shall prove Theorem 1 and Theorem 2.

(Proof of Theorem 1) A learning result f_{m+1} obtained from $(m+1)$ training examples in a batch manner is given as eq.(26). Since $y^{(m)}$ belongs to $\mathcal{R}(U_m)$, it follows from eqs.(26), and (63) that f_{m+1} is expressed as

$$f_{m+1} = f_m + \langle y^{(m+1)}, h_{m+1} \rangle \zeta_{m+1}. \tag{67}$$

When $\alpha_{m+1} = 0$, eqs.(65)–(67) yield eq.(41). ■

(Proof of Theorem 2) Theorem 2 is clear from eq.(26), Lemma 8, and Lemma 9. ■

5 Computer simulations

In this section, two kinds of computer simulations are performed to demonstrate the effectiveness of the proposed incremental learning method. In Section 5.1, IPL is compared with usual incremental learning methods. In Section 5.2, IPL is applied to a real world problem such as learning of the sensorimotor map of two-joint robot arms.

5.1 Performance comparison: IPL vs. RAN and on-line BP

We shall compare IPL with the resource allocating network (RAN) proposed by Platt (1991) and so-called on-line back propagation (on-line BP). In RAN, radial basis functions (RBFs) are adopted as basis functions. Briefly, the algorithm of RAN can be described as follows: Learning starts with no hidden units and the NN grows by allocating a new hidden unit based on the novelty in the additional training example. If the training example has no novelty, then the existing parameters of the NN are adjusted by the least mean squares algorithm to fit the additional example without adding novel units. Otherwise, a new hidden unit is added and the weights on the connections to the unit are adjusted. On the other hand, sigmoidal functions are adopted as hidden activation functions in on-line BP.

Let us consider the problem of approximating the following function.

$$f(x) = 2x - 14e^{-3(x-2.5)^2} - 5e^{-6(x-0.5)^2} + 3e^{-3x^2} + 12e^{-(x+2.5)^2}, \tag{68}$$

whose domain is $[-\pi, \pi]$. Learning simulations are carried out in the following conditions:

(a) IPL: H is spanned by $\{1, \sqrt{2}\cos kx, \sqrt{2}\sin kx\}_{k=1}^{4}$ and the inner product is defined as

$$\langle f, g\rangle = \frac{1}{2\pi}\int_{-\pi}^{\pi} f(x)\overline{g(x)}dx. \tag{69}$$

(b) RAN: Parameters are assigned as $\delta_{max} = 1$, $\kappa = 0.87$, $\delta_{min} = 0.05$, and $\epsilon = 0.01$.

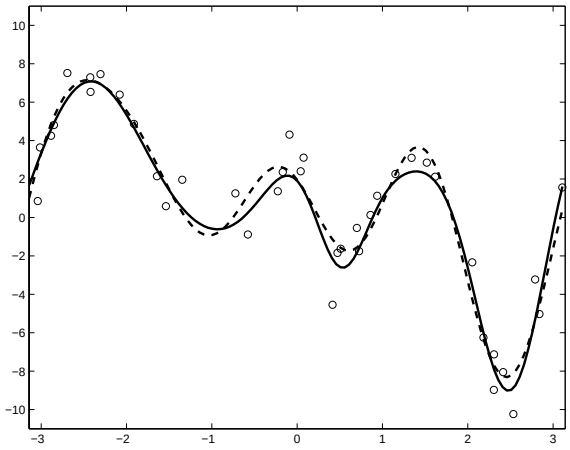
(c) on-line BP: The number of hidden units is fixed to 30 throughout the learning process.

Note that the target function f does not belong to H in (a), and it is not realizable in (b) and (c). In this simulation, we measure the generalization error of a learning result $f_0(x)$ by

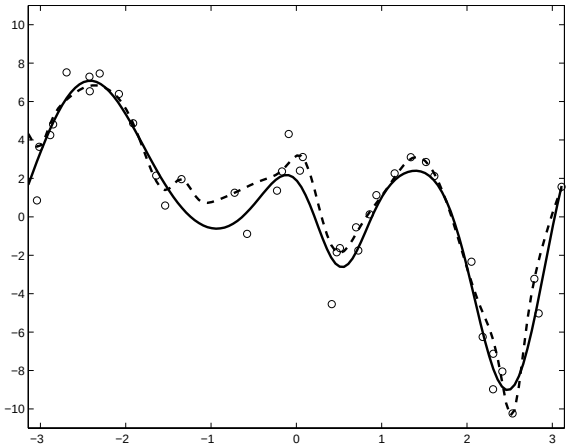
$$\text{Gen.err} = \frac{1}{126}\sum_{k=0}^{125}\left[f(-\pi + 0.05k) - f_0(-\pi + 0.05k)\right]^2. \tag{70}$$

Forty training examples $\{(x_i, y_i)\}_{i=1}^{40}$ is randomly sampled from the domain.

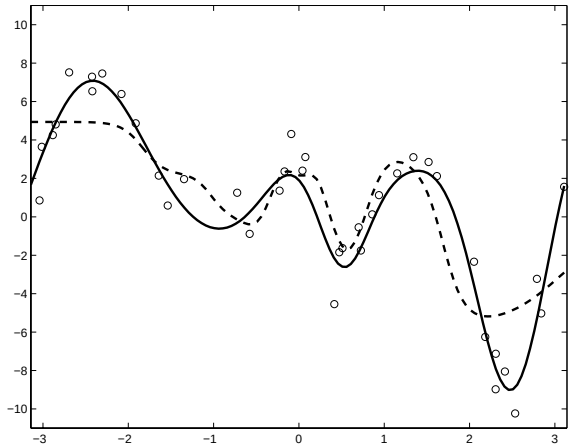
Learning results in the case $Q_m = I_m$ are shown in Fig.4. The solid and dashed lines denote the target function f and a learning result of each method, respectively. $\circ$ indicates a training example. The generalization errors of IPL, RAN, and on-line BP



(a) IPL: Gen.err = 0.32



(b) RAN: Gen.err = 0.86

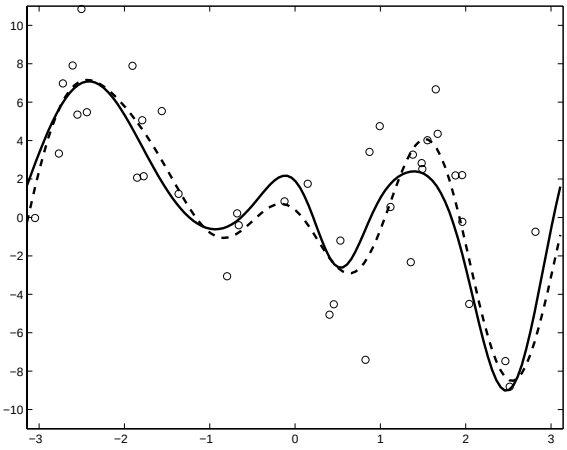


(c) on-line BP: Gen.err = 3.25

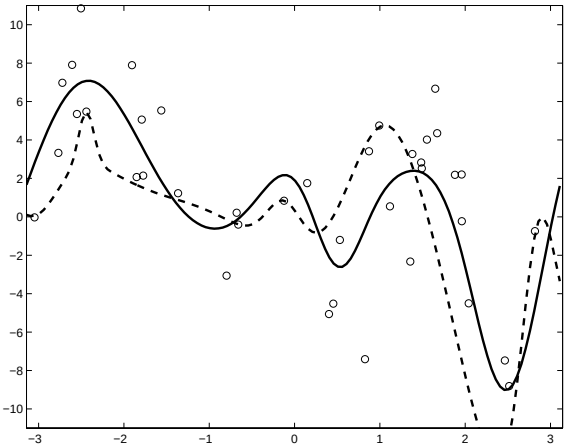
Figure 4: Learning simulation 1 ($Q_m = I_m$). The solid and dotted lines denote the target function $f(x)$ and a learning result, respectively. $\circ$ indicates a training example.

measured by eq.(70) are 0.32, 0.86, and 3.25, respectively. The results show that IPL provides better generalization capability than RAN and on-line BP. Note that the result of RAN is also good enough. Learning results in the case $Q_m = 3I_m$ are shown in Fig.5. The generalization errors of IPL, RAN, and on-line BP are 1.23, 8.61, and 3.89, respectively. In the second simulation, IPL also provides better generalization capability than RAN and on-line BP. The generalization errors of RAN and on-line BP are very large, which implies that RAN and on-line BP may not sufficiently suppress the effect of noise.

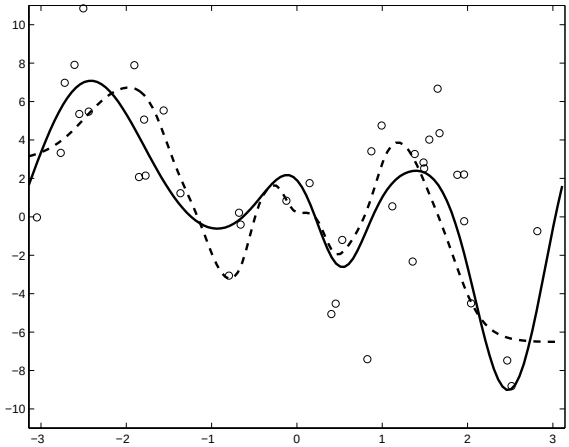
From the point of view of learning criteria, the reason why IPL works well can be explained as follows: For the signal component of the learning result, the projection learning



(a) IPL: Gen.err = 1.23



(b) RAN: Gen.err = 8.61



(c) on-line BP: Gen.err = 3.89

Figure 5: Learning simulation 2 ($Q_m = 3I_m$). The solid and dotted lines denote the target function $f(x)$ and a learning result, respectively. $\circ$ indicates a training example.

criterion is aimed at minimizing the generalization error (Ogawa, 1987) while the criteria of RAN and on-line BP are aimed at fitting an additional example. For the noise component of the learning result, the projection learning criterion requires the effect of noise to be systematically suppressed. On the other hand, RAN and on-line BP avoid over-fitting the noisy data by smoothing the learning results, which is attained by appropriately determining the width of RBFs, the number of hidden units, *etc.* Furthermore, since learning results obtained by IPL are exactly the same as those obtained by batch projection learning, IPL provides better generalization capability than RAN and on-line BP.

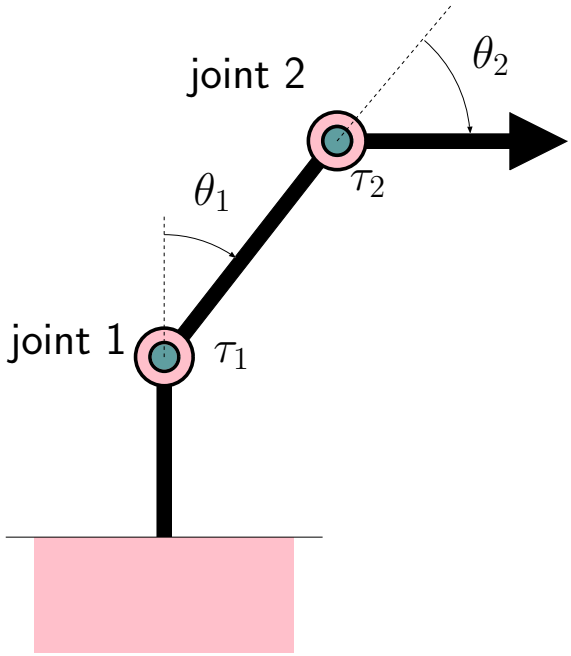


Figure 6: A two-joint robot arm.

5.2 Sensorimotor map learning

Let us consider learning of sensorimotor maps of a two-joint robot arm shown in Fig.6. A sensorimotor map is a mapping from joint angle θ_i, angular velocity $\dot{\theta}_i$, and angular acceleration $\ddot{\theta}_i$ to torque τ_i which should be applied to each joint:

$$\tau_i = f^{(i)}(\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2, \ddot{\theta}_1, \ddot{\theta}_2) \quad \text{for } i = 1, 2, \tag{71}$$

where $-\pi \leq \theta_1, \theta_2 \leq \pi$, $-a \leq \dot{\theta}_1 \leq a$, $-b \leq \dot{\theta}_2 \leq b$, $-c \leq \ddot{\theta}_1 \leq c$, and $-d \leq \ddot{\theta}_2 \leq d$.

Function spaces H_i to which $f^{(i)}$ belong are given as follows (Vijayakumar, 1998):

$$\begin{aligned} H_1 &= \mathcal{L}\{\ddot{\theta}_1, \ddot{\theta}_2, \ddot{\theta}_1 \cos\theta_2, \ddot{\theta}_2 \cos\theta_2, \\ &\qquad \dot{\theta}_2^{\,2} \sin\theta_2, \dot{\theta}_1\dot{\theta}_2 \sin\theta_2, \sin\theta_1, \sin\theta_1 \cos\theta_2, \sin\theta_2 \cos\theta_1\}, \qquad (72) \\ H_2 &= \mathcal{L}\{\ddot{\theta}_1, \ddot{\theta}_2, \ddot{\theta}_1 \cos\theta_2, \dot{\theta}_2 \sin\theta_2, \sin\theta_1 \cos\theta_2, \sin\theta_2 \cos\theta_1\}, \qquad (73) \end{aligned}$$

where $H = \mathcal{L}(\varphi_1, \varphi_2, \cdots, \varphi_k)$ means that H is spanned by $\varphi_1, \varphi_2, \cdots, \varphi_k$. The inner product in H_i is defined as

$$\langle f, g \rangle = \frac{1}{64\pi^2 abcd} \int_{-d}^{d} \int_{-c}^{c} \int_{-b}^{b} \int_{-a}^{a} \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} f(x)\overline{g(x)}\, d\theta_1 d\theta_2 d\dot{\theta}_1 d\dot{\theta}_2 d\ddot{\theta}_1 d\ddot{\theta}_2. \tag{74}$$

We shall perform a learning simulation of the sensorimotor map $f^{(1)}$. From eqs.(72) and (74), an orthonormal basis $\{\varphi_i^{(1)}\}_{i=1}^{9}$ in H_1 is given as follows:

$$\varphi_1^{(1)} = \frac{\sqrt{3}}{c}\ddot{\theta}_1, \tag{75}$$

$$\varphi_2^{(1)} = \frac{\sqrt{3}}{d}\ddot{\theta}_2, \tag{76}$$

$$\varphi_3^{(1)} = \frac{\sqrt{6}}{c}\ddot{\theta}_1 \cos\theta_2, \tag{77}$$

$$\varphi_4^{(1)} = \frac{\sqrt{6}}{d}\ddot{\theta}_2 \cos\theta_2, \tag{78}$$

$$\varphi_5^{(1)} = \frac{\sqrt{10}}{b^2}\dot{\theta}_2^{\,2} \sin\theta_2, \tag{79}$$

$$\varphi_6^{(1)} = \frac{\sqrt{18}}{ab}\dot{\theta}_1\dot{\theta}_2 \sin\theta_2, \tag{80}$$

$$\varphi_7^{(1)} = \sqrt{2}\sin\theta_1, \tag{81}$$

$$\varphi_8^{(1)} = 2\sin\theta_1\cos\theta_2, \tag{82}$$

$$\varphi_9^{(1)} = 2\sin\theta_2\cos\theta_1. \tag{83}$$

Hence, the reproducing kernel of H_1 is given as

$$K_1(x, x') = \sum_{i=1}^{9} \varphi_i^{(1)}(x)\overline{\varphi_i^{(1)}(x')}. \tag{84}$$

Suppose sample values are degraded by additive noise.

In Fig.7, the change in the generalization error is shown by the solid line. In this simulation, the generalization error of a learning result f_0 is measured by $\|f - f_0\|^2$. The generalization error tends to decrease as the number of training examples increases, and it becomes sufficiently small with 20–30 training examples. In contrast, if we use LASS (Vijayakumar & Schaal, 1998), which is a non-parametric learning method, around 15,000 training examples in total is required for obtaining a sufficiently good result. Consequently, IPL provides much faster convergence to the target function than LASS.

6 Conclusion

A method of incremental projection learning (IPL) was presented. IPL provides exactly the same learning result as that obtained by batch projection learning even in the non-asymptotic case. It has been demonstrated through computer simulations that IPL provides better generalization capability than RAN and on-line BP, and IPL shows considerably faster convergence to the target function. Properties of IPL will be studied in detail in a separate paper (Sugiyama & Ogawa, 2001a).

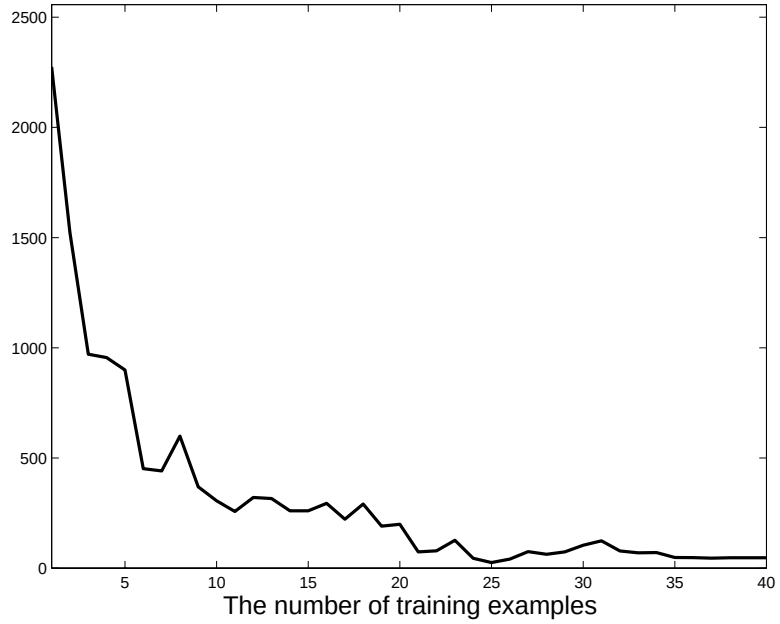


Figure 7: Learning simulation 3. Sensorimotor map learning. The change in the generalization error with respect to the number of training examples is shown by the solid line.

Appendix

A Proof of lemma 1

Since $e_i^{(m+1)} = \Gamma_{m+1} e_i^{(m)}$ for $1 \leq i \leq m$, it follows from eqs.(6) and (29) that

$$A_{m+1} = \sum_{i=1}^{m+1} \left(e_i^{(m+1)} \otimes \overline{\psi_i} \right) = \Gamma_{m+1} A_m + e_{m+1}^{(m+1)} \otimes \overline{\psi_{m+1}}, \tag{85}$$

which implies the lemma. ■

B Proof of lemma 2

Eqs.(17), (27), and (28) yield

$$Q_{m+1} = \begin{pmatrix} Q_m & q_{m+1} \\ q_{m+1}^* & \sigma_{m+1} \end{pmatrix}. \tag{86}$$

It follows from eq.(18) that

$$U_{m+1} = A_{m+1} A_{m+1}^* + Q_{m+1}. \tag{87}$$

Substituting eqs.(34) and (86) into eq.(87), we have eq.(35). ■

C Proof of lemma 3

It follows from eqs.(18) and (17) that U_{m+1} is always non-negative. Then, Lemma 3 is clear from Lemma 2 and Proposition 2. ■

D Proof of lemma 4

Lemma 4 is clear from eq.(35) and Theorem 3 in Albert (1969). ■

E Proof of lemma 5

Since Q_{m+1} is non-negative, it follows from eq.(86) and Proposition 2 that

$$\begin{aligned} q_{m+1} &= Q_m Q_m^\dagger q_{m+1}, && (88)\\ \sigma_{m+1} &\geq \langle Q_m^\dagger q_{m+1}, q_{m+1}\rangle. && (89) \end{aligned}$$

From eqs.(31) and (36), it holds that

$$U_m t_{m+1} = U_m U_m^\dagger s_{m+1} = P_{\mathcal{R}(U_m)} s_{m+1} = s_{m+1}. \quad (90)$$

It follows from eqs.(30) and (88) that

$$\begin{aligned} \langle t_{m+1}, s_{m+1}\rangle &= \langle t_{m+1}, A_m \psi_{m+1}\rangle + \langle t_{m+1}, q_{m+1}\rangle \\ &= \langle A_m^* t_{m+1}, \psi_{m+1}\rangle + \langle t_{m+1}, Q_m Q_m^\dagger q_{m+1}\rangle. && (91) \end{aligned}$$

Similarly, it follows from eqs.(90) and (18) that

$$\begin{aligned} \langle t_{m+1}, s_{m+1}\rangle &= \langle t_{m+1}, U_m t_{m+1}\rangle \\ &= \langle t_{m+1}, A_m A_m^* t_{m+1}\rangle + \langle t_{m+1}, Q_m t_{m+1}\rangle \\ &= \|A_m^* t_{m+1}\|^2 + \langle t_{m+1}, Q_m Q_m^\dagger Q_m t_{m+1}\rangle. && (92) \end{aligned}$$

Let an m-dimensional vector b_{m+1} be defined as

$$b_{m+1} = q_{m+1} - Q_m t_{m+1}. \quad (93)$$

Then, it follows from eqs.(32), (89), (91), (92), (43), and (93) that

$$\begin{aligned} \alpha_{m+1} &= \|\psi_{m+1}\|^2 + \sigma_{m+1} - \langle t_{m+1}, s_{m+1}\rangle \\ &\geq \|\psi_{m+1}\|^2 + \langle Q_m^\dagger q_{m+1}, q_{m+1}\rangle - \langle t_{m+1}, s_{m+1}\rangle && (94)\\ &= \|\psi_{m+1}\|^2 + \langle Q_m^\dagger q_{m+1}, q_{m+1}\rangle - \langle t_{m+1}, s_{m+1}\rangle - \langle s_{m+1}, t_{m+1}\rangle + \langle t_{m+1}, s_{m+1}\rangle \\ &= \|\psi_{m+1}\|^2 - \langle A_m^* t_{m+1}, \psi_{m+1}\rangle - \langle \psi_{m+1}, A_m^* t_{m+1}\rangle + \|A_m^* t_{m+1}\|^2 \\ &\quad + \langle Q_m^\dagger q_{m+1}, q_{m+1}\rangle - \langle Q_m^\dagger Q_m t_{m+1}, q_{m+1}\rangle \\ &\quad - \langle q_{m+1}, Q_m^\dagger Q_m t_{m+1}\rangle + \langle Q_m^\dagger Q_m t_{m+1}, Q_m t_{m+1}\rangle \\ &= \|\psi_{m+1} - A_m^* t_{m+1}\|^2 + \langle Q_m^\dagger (q_{m+1} - Q_m t_{m+1}), q_{m+1} - Q_m t_{m+1}\rangle \\ &= \|\xi_{m+1}\|^2 + \langle Q_m^\dagger b_{m+1}, b_{m+1}\rangle, && (95) \end{aligned}$$

and hence

$$\alpha_{m+1} \geq \|\xi_{m+1}\|^2 + \langle Q_m^\dagger b_{m+1}, b_{m+1}\rangle. \tag{96}$$

Since $Q_m^\dagger$ is non-negative, the second term in the right-hand side of eq.(96) is always non-negative. Hence, eqs.(96) and (89) yield that $\alpha_{m+1} = 0$ if and only if the following three conditions hold:

$$\begin{aligned} \xi_{m+1} &= 0, & (97)\\ b_{m+1} &\in \mathcal{N}(Q_m^\dagger), & (98)\\ \sigma_{m+1} &= \langle Q_m^\dagger q_{m+1}, q_{m+1}\rangle. & (99) \end{aligned}$$

Since it follows from eq.(88) that

$$b_{m+1} = (q_{m+1} - Q_m t_{m+1}) \in \mathcal{R}(Q_m) \perp \mathcal{N}(Q_m^\dagger), \tag{100}$$

eq.(98) is equivalent to

$$b_{m+1} = 0, \tag{101}$$

which concludes the proof. ■

F Proof of lemma 6

It follows from eq.(19) that

$$V_{m+1} = A_{m+1}^* U_{m+1}^\dagger A_{m+1}. \tag{102}$$

Eqs.(102), (34), (53), and (43) yield eq.(58). Similarly, eqs.(102), (34), (54), and (55) yield eq.(59). ■

G Proof of lemma 7

Eqs.(60) and (61) are clear from eq.(58) and Theorem 4.6 in Albert (1972). Eq.(62) is clear from eq.(59). ■

H Proof of lemma 8

It follows from eq.(20) that

$$X_{m+1} = V_{m+1}^\dagger A_{m+1}^* U_{m+1}^\dagger + Y_{m+1}(I_{m+1} - U_{m+1}U_{m+1}^\dagger). \tag{103}$$

When $\alpha_{m+1} > 0$, eqs.(103), (34), (53), (35), and (43) yield

$$\begin{aligned} X_{m+1} &= V_{m+1}^\dagger A_m^* U_m^\dagger \Gamma_{m+1}^* + \frac{V_{m+1}^\dagger \xi_{m+1} \otimes \overline{(e_{m+1}^{(m+1)} - \Gamma_{m+1} t_{m+1})}}{\alpha_{m+1}} \\ &\quad + Y_{m+1}\Gamma_{m+1}(I_m - U_m U_m^\dagger)\Gamma_{m+1}^*. \end{aligned} \tag{104}$$

If $\psi_{m+1} \notin \mathcal{R}(A_m^*)$, it follows from eqs.(60), (44), and (20) that

$$V_{m+1}^\dagger A_m^* U_m^\dagger = V_m^\dagger A_m^* U_m^\dagger - \frac{\tilde{\psi}_{m+1} \otimes \overline{X_m^* \xi_{m+1}}}{\tilde{\psi}_{m+1}(x_{m+1})}, \tag{105}$$

since $\mathcal{R}(V_m^\dagger) = \mathcal{R}(A_m^*)$ (see Ogawa, 1987). Eqs.(43), (42), and (5) yield

$$\langle \xi_{m+1}, \tilde{\psi}_{m+1} \rangle = \langle \psi_{m+1} - A_m^* t_{m+1}, \tilde{\psi}_{m+1} \rangle = \tilde{\psi}_{m+1}(x_{m+1}). \tag{106}$$

It follows from eqs.(60), (106), and (43) that

$$V_{m+1}^\dagger \xi_{m+1} = \frac{\alpha_{m+1} \tilde{\psi}_{m+1}}{\tilde{\psi}_{m+1}(x_{m+1})}. \tag{107}$$

Substituting eqs.(105) and (107) into eq.(104), we have eqs.(63) and (47). If $\psi_{m+1} \in \mathcal{R}(A_m^*)$, it follows from eq.(61), (44), and (20) that

$$V_{m+1}^\dagger A_m^* U_m^\dagger = V_m^\dagger A_m^* U_m^\dagger - \frac{\tilde{\xi}_{m+1} \otimes \overline{X_m^* \xi_{m+1}}}{\alpha_{m+1} + \langle \tilde{\xi}_{m+1}, \xi_{m+1} \rangle}. \tag{108}$$

Eqs.(61) and (44) yield

$$V_{m+1}^\dagger \xi_{m+1} = \tilde{\xi}_{m+1} - \frac{\tilde{\xi}_{m+1}}{\alpha_{m+1} + \langle \tilde{\xi}_{m+1}, \xi_{m+1} \rangle} = \frac{\alpha_{m+1} \tilde{\xi}_{m+1}}{\alpha_{m+1} + \langle \tilde{\xi}_{m+1}, \xi_{m+1} \rangle}. \tag{109}$$

Substituting eqs.(108) and (109) into eq.(104), we have eqs.(63) and (48). Finally when $\alpha_{m+1} = 0$, eqs.(103), (62), (34), (54), (35), and (55) yield eqs.(63) and (64). ■

I Proof of lemma 9

It follows from eqs.(50), (43), (8), (5), and (45) that

$$\begin{aligned}
\langle y^{(m+1)}, h_{m+1} \rangle &= \langle y^{(m+1)}, e_{m+1}^{(m+1)} - \Gamma_{m+1}(t_{m+1} + X_m^* \xi_{m+1}) \rangle \\
&= y_{m+1} - \langle y^{(m)}, t_{m+1} \rangle - \langle y^{(m)}, X_m^* \xi_{m+1} \rangle \\
&= y_{m+1} - \langle y^{(m)}, t_{m+1} \rangle - \langle y^{(m)}, X_m^* (\psi_{m+1} - A_m^* t_{m+1}) \rangle \\
&= y_{m+1} - \langle y^{(m)}, t_{m+1} \rangle - \langle X_m y^{(m)}, \psi_{m+1} \rangle + \langle A_m X_m y^{(m)}, t_{m+1} \rangle \rangle \\
&= y_{m+1} - \langle y^{(m)}, t_{m+1} \rangle - \langle f_m, \psi_{m+1} \rangle + \langle A_m f_m, t_{m+1} \rangle \rangle \\
&= y_{m+1} - f_m(x_{m+1}) - \langle y^{(m)} - A_m f_m, t_{m+1} \rangle \\
&= \beta_{m+1},
\end{aligned} \tag{110}$$

which implies eq.(65). ■

J Proof of lemma 10

Since $y^{(m+1)} \in \mathcal{R}(U_{m+1})$, it holds that

$$U_{m+1}U_{m+1}^{\dagger}y^{(m+1)} = P_{\mathcal{R}(U_{m+1})}y^{(m+1)} = y^{(m+1)}. \tag{111}$$

When $\alpha_{m+1} = 0$, it follows from eqs.(35), (54), and (111) that

$$y_{m+1} = \langle y^{(m)}, t_{m+1} \rangle. \tag{112}$$

Therefore, it follows from eqs.(45), (112), (5), (43), and (55) that

$$\begin{aligned}
\beta_{m+1} &= y_{m+1} - f_m(x_{m+1}) - \langle y^{(m)} - A_m f_m, t_{m+1} \rangle \\
&= -f_m(x_{m+1}) + \langle A_m f_m, t_{m+1} \rangle \\
&= -\langle f_m, \psi_{m+1} \rangle + \langle f_m, A_m^* t_{m+1} \rangle \\
&= -\langle f_m, \psi_{m+1} - A_m^* t_{m+1} \rangle \\
&= -\langle f_m, \xi_{m+1} \rangle \\
&= 0,
\end{aligned} \tag{113}$$

which implies eq.(66). ■

References

[1] Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control, AC-19(6)*, 716–723.

[2] Albert, A. (1969). Conditions for positive and nonnegative definiteness in terms of pseudoinverses. *SIAM Journal on Applied Mathematics, 17*, 434–440.

[3] Albert, A. (1972). *Regression and the Moore-Penrose pseudoinverse*. New York and London: Academic Press.

[4] Amari, S. (1998). Natural gradient works efficiently in learning. *Neural Computation, 10(2)*, 251–276.

[5] Aronszajn, N. (1950). Theory of reproducing kernels. *Transactions of the American Mathematical Society, 68*, 337–404.

[6] Bergman, S. (1970). *The kernel function and conformal mapping*. Providence, Rhode Island: The American Mathematical Society.

[7] Cohn, D. A. (1996). Neural network exploration using optimal experiment design. *Neural Networks, 9(6)*, 1071–1083.

[8] Fukumizu, K. (1996). Active learning in multilayer perceptrons. In D. Touretzky et al. (Eds.), *Advances in Neural Information Processing Systems 8* (pp. 295–301). Cambridge: MIT Press.

[9] Jutten, C., & Chentouf, R. (1995). A new scheme for incremental learning. *Neural Processing Letters, 2(1)*, 1–4.

[10] Kadirkamanathan, V., & Niranjan, M. (1993). A function estimation approach to sequential learning with neural networks. *Neural Computation, 5(6)*, 954–975.

[11] MacKay, D. J. C. (1992a). Bayesian interpolation. *Neural Computation, 4(3)*, 415–447.

[12] MacKay, D. J. C. (1992b). Information-based objective functions for active data selection. *Neural Computation, 4(4)*, 590–604.

[13] Molina, C., & Niranjan, M. (1996). Pruning with replacement on limited resource allocating networks by F-projections. *Neural Computation, 8(4)*, 855–868.

[14] Murata, N. (1999). Statistical study on on-line learning. In D. Saad (Ed.), *On-line Learning in Neural Networks* (pp. 63–92). Cambridge University Press.

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

[16] Nakazawa, S., & Ogawa, H. (1996). Optimal realization of optimally generalizing neural networks. *IEICE Technical Report, NC96-60*, 17–24. (in Japanese)

[17] Ogawa, H. (1987). Projection filter regularization of ill-conditioned problem. *Proceedings of SPIE, Inverse Problems in Optics, 808* (pp. 189–196).

[18] Ogawa, H. (1989). Inverse problem and neural networks. *Proceedings of IEICE 2nd Karuizawa Workshop on Circuits and Systems* (pp. 262–268). Karuizawa, Japan. (in Japanese)

[19] Ogawa, H. (1992). Neural network learning, generalization and over-learning. *Proceedings of the ICIIPS'92, International Conference on Intelligent Information Processing & System, 2* (pp. 1–6). Beijing, China.

[20] Ogawa, H. (1995). Neural networks and generalization ability, *IEICE Technical Report, NC95-8*, 57–64. (in Japanese)

[21] Ogawa, H., & Oja, E. (1986). Projection filter, Wiener filter, and Karhunen-Loève subspaces in digital image restoration. *IEEE Transactions on Acoustics, Speech & Signal Processing, ASSP-34(6)*, 1643–1653.

[22] Ogawa, H., Oja, E., & Lampinen, J. (1989). Projection filters for image and signal restoration. *Proceedings of the IEEE International Conference on Systems Engineering* (pp. 93–97). Dayton, USA.

[23] Platt, J. (1991). A resource-allocating network for function interpolation. *Neural Computation, 3(2)*, 213–225.

[24] Saitoh, S. (1988). *Theory of reproducing kernels and its applications*. Pitsman Research Notes in Mathematics Series, 189. UK: Longman Scientific & Technical.

[25] Saitoh, S. (1997). *Integral transform, reproducing kernels and their applications*. Pitsman Research Notes in Mathematics Series, 369. UK: Longman.

[26] Schatten, R. (1970). *Norm ideals of completely continuous operators*. Berlin: Springer-Verlag.

[27] Sugiyama, M., & Ogawa, H. (2000). Incremental active learning for optimal generalization. *Neural Computation, 12(12)*, 2909–2940.

[28] Sugiyama, M., & Ogawa, H. (2001a). Properties of incremental projection learning. *Neural Networks 14(1)*, 67–78. (Its latest version is available at '`http://ogawa-www.cs.titech.ac.jp/~sugi/publications/2001/ipl2.ps.gz`'.)

[29] Sugiyama, M., & Ogawa, H. (2001b). Subspace information criterion for model selection. *Neural Computation, 13(8)*.

[30] Sugiyama, M., & Ogawa, H. (2001c). Theoretical and experimental evaluation of subspace information criterion. *Machine Learning, Special Issue on New Methods for Model Selection and Model Combination*.

[31] Tikhonov, A. N., & Arsenin, V. Y. (1977). *Solutions of Ill-Posed Problems*. Washington DC: V. H. Winston.

[32] Vijayakumar, S. (1998). Computational theory of incremental and active learning for optimal generalization. *Ph.D Thesis*, Department of Computer Science, Tokyo Institute of Technology, Japan.

[33] Vijayakumar, S., & Ogawa, H. (1998). RKHS based functional analysis for exact incremental learning. *Neurocomputing, 29(1–3)*, 85–113.

[34] Vijayakumar, S., Sugiyama, M., & Ogawa, H. (1998). Training data selection for optimal generalization with noise variance reduction in neural networks. In Marinaro & Tagliaferri (Eds.), *Neural Nets WIRN Vietri-98* (pp. 153–166). Springer-Verlag.

[35] Vijayakumar, S., & Schaal, S. (1998). Local adaptive subspace regression. *Neural Processing Letters, 7(3)*, 139–149.

[36] Vyšniauskas, V., Groen, F. C. A., & Kröse, B. J. A. (1995). Orthogonal incremental learning of a feedforward network. *Proceedings of International Conference on Artificial Neural Networks* (pp. 311–316). Paris, France.

[37] Yamakawa, H., Masumoto, D., Kimoto, T., & Nagata, S. (1993). Active data selection and subsequent revision for sequential learning. *IEICE Technical Report, NC92-99*, 33–40. (in Japanese)

[38] Yamashita, Y., & Ogawa, H. (1992). Optimum image restoration and topological invariance. *The transactions of the IEICE D-II, J75-D-II(2)*, 306–313. (in Japanese)

[39] Yamauchi, K., & Ishii, N. (1995). An incremental learning method with recalling interfered patterns. *Proceedings of IEEE International Conference on Neural Networks ICNN'95, 6* (pp. 3159–3164).

[40] Yingwei, L., Sundararajan, N., & Saratchandran, P. (1997). A sequential learning scheme for function approximation using minimal radial basis function neural networks. *Neural Computation, 9(2)*, 461–478.

[41] Yingwei, L., Sundararajan, N., & Saratchandran, P. (1998). Performance evaluation of a sequential minimal radial basis function neural network learning algorithm. *IEEE Transactions on Neural Networks, 9(2)*, 308–318.

[42] Yoneda, T., Yamanaka, M., & Kakazu, Y. (1992). Study on optimization of grinding conditions using neural networks — a method of additional learning —. *Journal of the Japan Society of Precision Engineering, 58(10)*, 1707–1712. (in Japanese)

[43] Zhang, B. T. (1994). An incremental learning algorithm that optimizes network size and sample size in one trial. *Proceedings of International Conference on Neural Networks* (pp. 215-220). Orlando, USA.