Exact Incremental Projection Learning in the Presence of Noise

Masashi Sugiyama and 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@cs.titech.ac.jp, http://ogawa-www.cs.titech.ac.jp/~sugi.

Abstract

In many practical situations in neural network learning, training data is supplied one by one. Therefore, it is important to consider to add new training data to neural networks in order to further improve their generalization capability. In this paper, a method of incremental projection learning in the presence of noise is presented. The proposed method provides exactly the same learning result as that obtained by batch projection learning. By using the method, a criterion for redundancy of an additional datum is derived, and the relationship between a prior and a posterior learning results is studied. Moreover, a simple form of incremental projection learning under certain conditions is given. Finally, the effective of the proposed method is demonstrated through computer simulations.

Keywords

multilayer feedforward neural networks, generalization capability, incremental learning, projection learning, reproducing kernel Hilbert space (RKHS).

1 Introduction

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

In many practical situations, training data is supplied one by one. Therefore, it is important to consider to add new training data to NNs in order to further improve their generalization capability. Compared with the learning methods of human beings, it is natural to build a posterior learning result upon a prior result. This learning method is generally called *incremental learning*. Incremental learning also plays an important role when we work on *active learning*, which is extensively studied recently (MacKay [5], Fukumizu [3], Sugiyama and Ogawa [13]). In these methods, training data which should be learned next is determined by analyzing the intermediate learning result. Therefore, incremental learning is indispensable for performing active learning.

Many incremental learning methods have been devised so far. Many of them are based on the idea of generating a novel hidden unit when new training data is added, and adjusting weights on the connections to the novel unit (Platt [12], Kadirkamanathan and Niranjan [4], Vyšniauskas *et al.* [17], Molina and Niranjan [6], Yingwei *et al.* [18], Vijayakumar and Schaal [16]). Yamauchi and Ishii [20] took an interesting approach. First, the region which will be interfered with by incremental learning is inferred, and artificial training data which will prevent the interference is created. Then incremental learning takes place by using both newly added and created training data. Although computation becomes efficient by these methods, the optimal generalization may not be guaranteed. Recently, another incremental learning method has been proposed, which provides asymptotically the same generalization capability as that obtained by batch learning (Amari [2]). However, the optimal generalization in the non-asymptotic case may not be guaranteed.

Ogawa [9] formulated the NN learning problem as an *inverse problem* from the functional analytic point of view. It has been shown that the optimal image restoration filters such as *projection filter* (Ogawa [8]), *Wiener filter* (Ogawa and Oja [10]) *etc.* can be applied to the NN learning problem. These filters are called *projection learning*, *Wiener learning etc.* in the learning problem. Within the framework, incremental Wiener learning in the absence of noise has been devised (Vijayakumar and Ogawa [15]), in which generalization capability is proved to be exactly the same as that obtained by batch Wiener learning. In this paper, we present a method of incremental projection learning in the presence of noise, which 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. In Section 3, a method of incremental projection learning is proposed. Section 4 points out that some of the training data which is rejected in usual incremental learning methods have potential effectiveness, and an improved criterion for redundancy of an additional datum is derived. Section 5 studies the relationship between a prior and a posterior learning results where effective training data is classified into two categories as regards improving generalization capability. In Section 6, a simple form of the proposed incremental learning method under certain conditions is given. Finally, Section 7 is devoted to computer simulations, which demonstrates the effectiveness of the proposed incremental learning method.

2 Formulation of NN learning problem

In this section, the NN learning problem is formulated (See Ogawa [9]).

Let us consider a learning problem of three-layer feedforward NNs whose number of input and output units are L and 1, respectively. The relationship between input $x = (\eta_1, \cdots, \eta_L)$ and output y of the network is represented by using a function f_0 of L variables as

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

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

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

where y_i is degraded by additive noise n_i.

In many NN learning methods devised so far, learning algorithms are built upon certain architecture of NNs, i.e., a fixed number of hidden units, each with a prespecified sigmoidal or Gaussian functions. 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 given training data, and a NN which represents the approximated function is constructed in the second step.

To begin with, we explain a function approximation problem which corresponds 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 which $y^{(m)}$ belongs to is called a *sample value space*. In this paper, the underlying function f is assumed to belong to a reproducing kernel Hilbert space H. Let $\mathcal{D}$ be the domain of f. The reproducing kernel 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 a reproducing kernel, then it is possible to deal with the value of a function at a point. 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 is expressed as

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

For this reason, ψ_i is called a *sampling function*. Once a training set $\{x_i\}_{i=1}^m$ is fixed, the relationship between f and $y^{(m)}$ can be represented as

$$y^{(m)} = A_m f + n^{(m)}, \tag{6}$$

where A_m is called a *sampling operator*. Note that A_m is always a linear operator. A_m is expressed by using the *Neumann-Schatten product*[1] as

$$A_m = \sum_{i=1}^{m} \left(e_i^{(m)} \otimes \overline{\psi_i} \right), \tag{7}$$

where $e_i^{(m)}$ is an m-dimensional vector where all elements are zero except the i-th element which is equal to one. Let us denote a learning result obtained from m training data 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 NNs 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. Since image and signal restoration problems discussed in Ogawa [8] and Ogawa *et al.* [11] are also formulated as the same form of inverse problems, the

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

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

optimal image restoration filters devised in these papers can be applied to the function approximation problem.

Now we go on to the second step, i.e., the construction of a NN which represents f_m. In this step, the number N of hidden units, an input-output function $u_i(x)$ of each hidden unit, and weights w_i on hidden-output connections are determined. A NN which represents a function obtained in the first step is called an *Optimally Generalizing NN* (OGNN). A general construction method of OGNNs was given in Ogawa [9]. The method shows that there exist infinite degrees of freedom in OGNNs. Utilizing these degrees of freedom effectively, Nakazawa and Ogawa [7] 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.

3 Incremental projection learning

As mentioned in the previous section, the NNs learning problem is divided into two steps. In this paper, we focus on the function approximation problem corresponding to the first step.

We adopt the projection learning criterion. Let E_n, A_m^*, $\mathcal{R}(A_m^*)$, and $P_{\mathcal{R}(A_m^*)}$ be the ensemble average over noise, the adjoint operator of A_m, the range of A_m^*, and the orthogonal projection operator onto $\mathcal{R}(A_m^*)$, respectively. Then, projection learning is defined as follows:

Definition 1 (Projection learning) *(Ogawa [8]) An operator X_m is called the projection learning operator if X_m minimizes the functional*

$$J_P[X_m] = E_n \|X_m n^{(m)}\|^2 \tag{9}$$

under the constraint

$$X_m A_m = P_{\mathcal{R}(A_m^*)}. \tag{10}$$

From eqs.(8) and (6), a learning result f_m can be decomposed as

$$f_m = X_m A_m f + X_m n^{(m)}. \tag{11}$$

The first and second terms of eq.(11) are called the *signal* and *noise components* of f_m, respectively. The projection learning criterion requires the signal component to coincide with the orthogonal projection of f onto $\mathcal{R}(A_m^*)$ and the noise component to minimize its variance.

Under the projection learning criterion, we shall devise an incremental learning method in the presence of noise. We call the method *incremental projection learning* (IPL). It has been shown that a learning result obtained by projection learning does not depend on the inner product in a sample value space (Yamashita and Ogawa [19]). Hence, the Euclidean inner product is adopted without loss of generality.

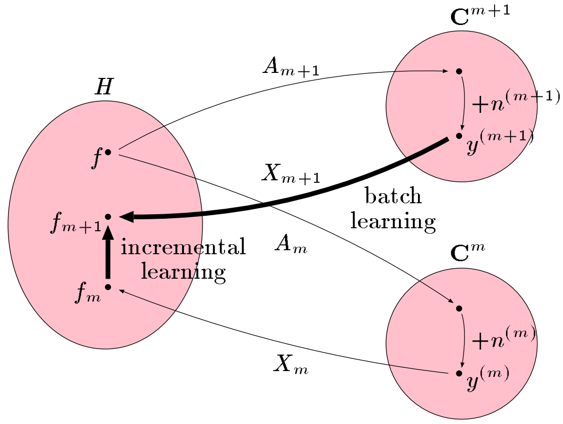


Figure 1: Exact incremental learning and batch learning.

First, we show a general form of the projection learning operator. 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

$$\begin{aligned} Q_m &= E_n\left(n^{(m)} \otimes \overline{n^{(m)}}\right), &(12)\\ U_m &= A_m A_m^* + Q_m, &(13)\\ V_m &= A_m^* U_m^\dagger A_m, &(14)\end{aligned}$$

where $\dagger$ indicates the *Moore-Penrose generalized inverse*[2].

Proposition 1 *(Ogawa [8]) A general form of the projection learning operator is represented as*

$$X_m = V_m^\dagger A_m^* U_m^\dagger + Y_m(I_m - U_m U_m^\dagger). \tag{15}$$

Let us consider the case where the $(m+1)$-st training datum (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 data can be represented in a batch manner as

$$f_{m+1} = X_{m+1} y^{(m+1)}. \tag{16}$$

The suffix $m+1$ indicates the number of total training data. 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.1.

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

$$\begin{aligned} q_{m+1} &= E_n(n_{m+1} n^{(m)}), &(17)\\ \sigma_{m+1} &= E_n(n_{m+1}^2). &(18)\end{aligned}$$

[2] An operator X which satisfies the following four conditions is called the Moore-Penrose generalized inverse of an operator A (Albert [1]):

$$AXA = A,\ \ XAX = X,\ \ (AX)^* = AX,\ \ (XA)^* = XA.$$

Note that the Moore-Penrose generalized inverse is unique.

Note that q_{m+1} is an m-dimensional vector while σ_{m+1} is a scalar. Let m-dimensional vectors s_{m+1}, t_{m+1}, and a scalar α_{m+1} be

$$\begin{aligned} s_{m+1} &= A_m\psi_{m+1} + q_{m+1}, & (19)\\ t_{m+1} &= U_m^\dagger s_{m+1}, & (20)\\ \alpha_{m+1} &= \psi_{m+1}(x_{m+1}) + \sigma_{m+1} - \langle t_{m+1}, s_{m+1}\rangle. & (21)\end{aligned}$$

In this case, we have

Lemma 1 *U_{m+1} is non-negative if and only if*

$$\begin{aligned} s_{m+1} &\in \mathcal{R}(U_m), & (22)\\ \alpha_{m+1} &\geq 0. & (23)\end{aligned}$$

It follows from eqs.(13) and (12) that U_{m+1} is always non-negative[3]. Hence, eqs.(22) and (23) hold.

Whether α_{m+1} is zero or not is crucial in the derivation of IPL. First, we discuss the case $\alpha_{m+1} = 0$.

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

$$f_{m+1} = f_m. \tag{24}$$

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 data which causes $f_{m+1} = f_m$ is regarded as redundant. However, as shown in Section 4, the redundancy of training data can not be judged by simply comparing f_{m+1} with f_m.

Next, we focus on the case $\alpha_{m+1} > 0$. Let $\mathcal{N}(A_m)$ be the null space of A_m. In order to introduce the main theorem, we define the following notation.

$$\begin{aligned} \text{Matrix:} \quad \Gamma_{m+1} &= \sum_{i=1}^{m}\left(e_i^{(m+1)} \otimes \overline{e_i^{(m)}}\right). & (25)\\ \text{Functions:} \quad \tilde{\psi}_{m+1} &= P_{\mathcal{N}(A_m)}\psi_{m+1}, & (26)\\ \xi_{m+1} &= \psi_{m+1} - A_m^* t_{m+1}, & (27)\\ \tilde{\xi}_{m+1} &= V_m^\dagger \xi_{m+1}. & (28)\\ \text{Scalar:} \quad \beta_{m+1} &= y_{m+1} - f_m(x_{m+1}) \\ &\quad -\langle y^{(m)} - A_m f_m, t_{m+1}\rangle. & (29)\end{aligned}$$

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

$$f_{m+1} = f_m + \beta_{m+1}\zeta_{m+1}, \tag{30}$$

where ζ_{m+1} is given as follows:

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

$$\zeta_{m+1} = \frac{\tilde{\psi}_{m+1}}{\tilde{\psi}_{m+1}(x_{m+1})}. \tag{31}$$

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

$$\zeta_{m+1} = \frac{\tilde{\xi}_{m+1}}{\alpha_{m+1} + \langle \tilde{\xi}_{m+1}, \xi_{m+1}\rangle}. \tag{32}$$

Note that β_{m+1} depends on the value of y_{m+1} while ζ_{m+1} does not. The difference in the conditions (a) and (b) is studied in Section 4 and Section 5.

4 Effectiveness of additional training data

In this section, we point out that some of the training data which is rejected in usual incremental learning methods have potential effectiveness as a matter of fact. Based on this, an improved criterion for redundancy of additional training data is derived.

In many incremental learning methods devised so far, an additional training datum (x_{m+1}, y_{m+1}) is rejected if the posterior result f_{m+1} is exactly the same as the prior result f_m (Platt [12], Kadirkamanathan and Niranjan [4], Molina and Niranjan [6], Yingwei *et al.* [18]). However, this sometimes leads us to waste valuable information. So as to make the claim sure, we show a simple example:

Let a function space H be spanned by

$$\{\sin 6x, \sin 10x, \sin 15x\}, \tag{33}$$

and the inner product in H be defined as

$$\langle f, g\rangle = \frac{2}{\pi}\int_0^{\frac{\pi}{2}} f(x)g(x)\,dx. \tag{34}$$

Let an original function be $f = 9\sin 6x + 5\sin 15x$. For the sake of simplicity, the learning takes place in the absence of noise in this example. The original function f and a learning result f_1 obtained by using $(x_1, y_1) = (\frac{\pi}{5}, f(\frac{\pi}{5}))$ are shown as solid and dotted lines, respectively, in Fig.2 (a). Adding $(x_2, y_2) = (\frac{\pi}{3}, f(\frac{\pi}{3}))$ to f_1, we obtain a learning result f_2, which agrees with f_1. Now we comply with the usual criterion for redundancy, i.e., we reject (x_2, y_2) since it causes $f_2 = f_1$. A learning result f_2' obtained by adding $(x_3, y_3) = (\frac{\pi}{9}, f(\frac{\pi}{9}))$ to f_1 is shown as a dashed line in Fig.2 (b). On the other hand, if we use (x_2, y_2) without rejection and add (x_3, y_3) to f_2, we obtain a learning result f_3 shown as a solid line in the same figure. f_3 agrees with the original function f. The example says that f_3 acquires higher generalization capability compared with f_2'. This implies (x_2, y_2) is essentially useful.

The reason why (x_2, y_2) has potential effectiveness can be understood from the functional analytic point of view. The geometrical relationships between the original function f, learning results f_1, f_2, f_2', and f_3 in the

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

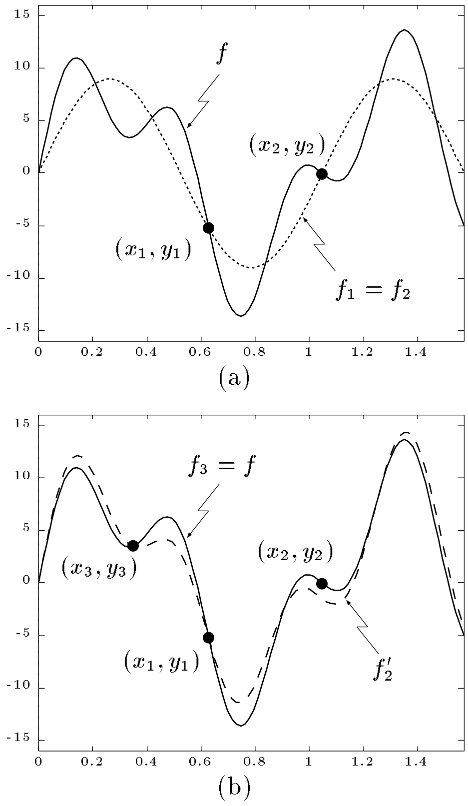


Figure 2: Example of the training data which is regarded as redundant in traditional incremental learning methods but it is effective.

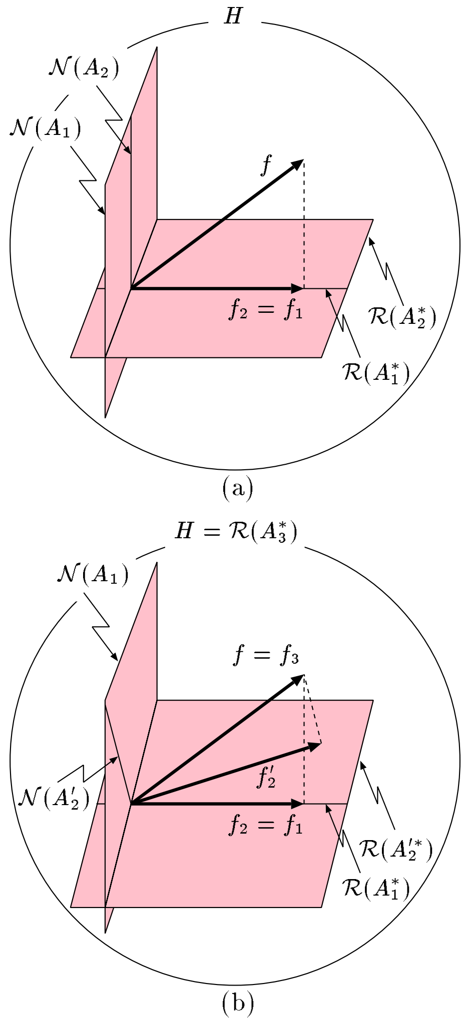


Figure 3: Geometrical interpretation of the training data which is regarded as redundant in traditional methods but it is effective.

function space H are shown in Fig.3. In the absence of noise, a projection learning result f_m is coincident with the orthogonal projection of f onto $\mathcal{R}(A_m^*)$. $\mathcal{R}(A_m^*)$ is called the *approximation space for* f_m. Since f belongs to $\mathcal{R}(A_1^*) + \mathcal{N}(A_2)$ in this example, we have

$$f_2 = P_{\mathcal{R}(A_2^*)} f = P_{\mathcal{R}(A_1^*)} f = f_1, \tag{35}$$

as shown in Fig.3 (a). Rejecting (x_2, y_2) and adding (x_3, y_3) to f_1, we obtain f_2' (See Fig.3 (b)). In this case, the approximation space for f_2', denoted by $\mathcal{R}(A_2'^*)$, becomes a two-dimensional subspace. Since f does not belong to $\mathcal{R}(A_2'^*)$, f_2' does not agree with f. On the other hand, if we use (x_2, y_2) without rejection and add (x_3, y_3) to f_2, we obtain f_3. In this case, $\mathcal{R}(A_3^*)$ becomes a three-dimensional subspace which coincides with H. Since f belongs to $\mathcal{R}(A_3^*)$, f_3 agrees with f. After all, the difference between f_3 and f_2' is caused by the difference in approximation spaces, i.e., $\mathcal{R}(A_1^*)$ is a proper subspace of $\mathcal{R}(A_2^*)$.

So far, additinal training data was said to be redundant if it causes $f_{m+1} = f_m$. However, the redundancy of additional training data can not be judged by simply comparing f_{m+1} with f_m. Now we define *real* effectiveness and redundancy of an additional training datum. Let f_m be a learning result obtained by using $\{(x_i, y_i)\}_{i=1}^m$, and $\hat{f}_{m+1}$ be a learning result obtained by adding $(\hat{x}, \hat{y})$ to f_m. Let f_{m+i} and $\hat{f}_{m+i+1}$ be learning results obtained by adding $\{(x_{m+j}, y_{m+j})\}_{j=1}^i$ to f_m and $\hat{f}_{m+1}$, respectively.

Definition 2 $(\hat{x}, \hat{y})$ *is said to be effective if there exists at least one set of training data which causes* $f_{m+i} \neq \hat{f}_{m+i+1}$. *Conversely, training data which is not effective is said to be redundant.*

Note that the above concepts depends on f, f_m, A_m, and $U_m^\dagger$. Based on the definition of redundancy, a criterion for redundancy of an additional datum is given as follows:

Theorem 3 (Redundancy criterion) (x_{m+1}, y_{m+1}) *is redundant if* $\xi_{m+1} = 0$, *where* ξ_{m+1} *is the function defined by eq.(27)*

It is shown that $f_{m+1} = f_m$ if and only if one of the following four conditions holds:

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

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

(c) $\alpha_{m+1} > 0$, $\psi_{m+1} \in \mathcal{R}(A_m^*)$, $\zeta_{m+1} \neq 0$, and $\beta_{m+1} = 0$,

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

where α_{m+1}, β_{m+1}, ψ_{m+1}, and ζ_{m+1} are given by eqs.(21), (29), (4), and (32), respectively. Among these conditions, $\xi_{m+1} = 0$ if and only if (a) or (d) holds. Note that the condition (a) and (d) do not depend on the value of y_{m+1} while (b) and (c) do, which implies that an additional datum is judged to be redundant if it causes $f_{m+1} = f_m$ independently of y_{m+1}.

5 Improving generalization capability through IPL

The previous section clarified the redundancy of additional training data. In this section, the characteristics of effective additional training data are studied from the viewpoint of improving generalization capability. The mean of noise is assumed to be zero through this section.

In this section, we measure the generalization error of a learning result f_m by

$$J_G = E_n\|f - f_m\|^2. \tag{36}$$

Eq.(36) can be decomposed as follows:

Proposition 2 *(Takemura [14]) It holds that*

$$J_G = \|f - E_n f_m\|^2 + E_n\|E_n f_m - f_m\|^2. \tag{37}$$

The first and second terms of eq.(37) are called the *bias* and *variance* of the generalization error, respectively. Let J_b and J_v be the changes in the bias and variance of the generalization error by adding a training datum, respectively, i.e.,

$$J_b = \|f - E_n f_m\|^2 - \|f - E_n f_{m+1}\|^2, \tag{38}$$
$$J_v = E_n\|E_n f_m - f_m\|^2 - E_n\|E_n f_{m+1} - f_{m+1}\|^2. \tag{39}$$

Then, we have

Theorem 4 *For any additional datum (x_{m+1}, y_{m+1}) satisfying $\xi_{m+1} \neq 0$, the following relations hold:*

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

$$J_b \geq 0, \quad J_v \leq 0. \tag{40}$$

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

$$J_b = 0, \quad J_v > 0. \tag{41}$$

Theorem 4 says that additional training data satisfying $\psi_{m+1} \notin \mathcal{R}(A_m^*)$ reduces or maintains the bias of the generalization error while it increases or maintains the variance. On the other hand, additional training data satisfying $\psi_{m+1} \in \mathcal{R}(A_m^*)$ maintains the bias while it reduces the variance. Note that additional training data satisfying $\psi_{m+1} \notin \mathcal{R}(A_m^*)$ possibly causes $J_b = 0$ and $J_v = 0$, which yields $f_{m+1} = f_m$. However, it is not redundant since $\xi_{m+1} \neq 0$ as shown in the previous section.

6 Simple representation of IPL

In this section, a simple form of IPL under certain conditions is given.

Suppose the noise correlation matrix is positive and diagonal, i.e.,

$$Q_{m+1} = \mathrm{diag}(\sigma_1, \sigma_2, \cdots, \sigma_{m+1}), \tag{42}$$

where $\sigma_i > 0$ for all i. Let an operator V'_m from H to H be

$$V'_m = A_m^* Q_m^{-1} A_m. \tag{43}$$

In this case, we have

Theorem 5 *If Q_m is given by eq.(42) with $\sigma_i > 0$ for all i, a posterior projection learning result f_{m+1} is obtained by using prior results f_m and $V'^{\dagger}_m$ as*

$$f_{m+1} = f_m + \beta'_{m+1}\zeta'_{m+1}, \tag{44}$$

where

$$\beta'_{m+1} = y_{m+1} - f_m(x_{m+1}), \tag{45}$$

and ζ'_{m+1} are given as follows:

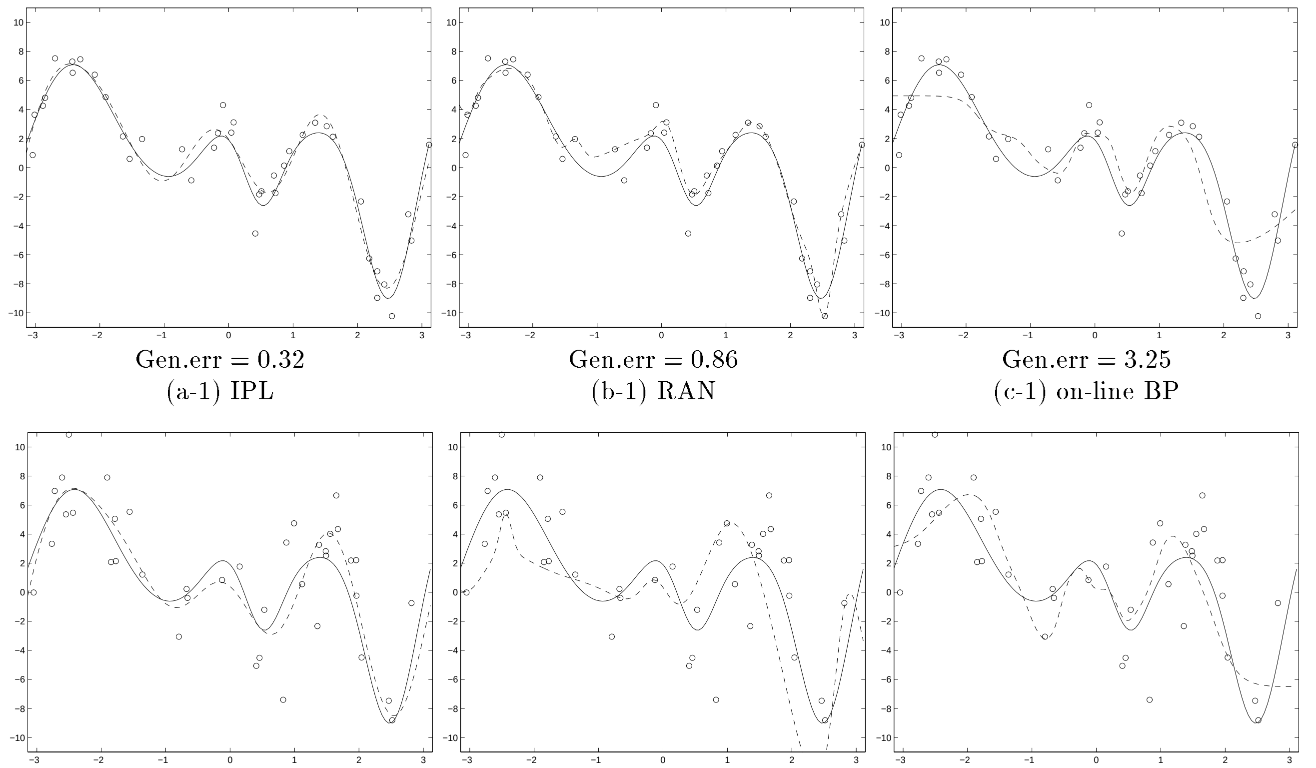
(a) When $\psi_{m+1} \notin \mathcal{R}(A_m^*)$,

$$\zeta'_{m+1} = \frac{\tilde{\psi}_{m+1}}{\tilde{\psi}_{m+1}(x_{m+1})}. \tag{46}$$

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

$$\zeta'_{m+1} = \frac{V'^{\dagger}_m \psi_{m+1}}{\sigma_{m+1} + \langle V'^{\dagger}_m \psi_{m+1}, \psi_{m+1}\rangle}. \tag{47}$$

Compared with Theorem 2, eq.(29) is replaced with eq.(45) in Theorem 5. This implies that Theorem 5 does not require $\{y_i\}_{i=1}^m$ for calculating f_{m+1}. In the case $\psi_{m+1} \notin \mathcal{R}(A_m^*)$, eq.(31) is the same as eq.(46). On the other hand, in the case $\psi_{m+1} \in \mathcal{R}(A_m^*)$, eq.(32) is replaced by eq.(47) where α_{m+1} does not appear. Although α_{m+1} played an important role in the derivation of Theorem 2, it is not required for Theorem 5 since it is always positive if the noise correlation matrix is positive.



Gen.err = 0.32 (a-1) IPL | Gen.err = 0.86 (b-1) RAN | Gen.err = 3.25 (c-1) on-line BP

Gen.err = 1.23 (a-2) IPL | Gen.err = 8.61 (b-2) RAN | Gen.err = 3.89 (c-2) on-line BP

Figure 4: Learning simulation. Solid and dotted lines denote the original function f and a learning result, respectively. $\circ$ indicates training data. The upper three graphs show learning results in the case $Q_m = I_m$ while the bottom three graphs show learning results in the case $Q_m = 3I_m$.

7 Computer simulations

In this section, computer simulations are performed to show the effectiveness of the proposed incremental learning method.

First, IPL is compared with a resource allocating network (RAN) proposed by Platt [12], where radial basis functions (RBFs) are adopted as its hidden activation functions. In RAN, a novel hidden unit is added if an additional datum satisfies the *novelty criteria*. Next, IPL is compared with so-called on-line back propagation (on-line BP), where each training data is used once and never used again. Sigmoidal functions are adopted as hidden activation functions.

Let us consider the problem of approximating the following function:

$$\begin{aligned} f &= 2x - 14e^{-3(x-2.5)^2} - 5e^{-6(x-0.5)^2} \\ &\quad +3e^{-3x^2} + 12e^{-(x+2.5)^2}, \end{aligned} \tag{48}$$

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

(a) IPL: H is spanned by $\{1, \sin ix, \cos ix\}_{i=1}^{4}$, and the inner product in H is defined as

$$\langle f, g \rangle = \frac{1}{2\pi}\int_{-\pi}^{\pi} f(x)g(x)dx. \tag{49}$$

(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 through the learning process.

Note that the original 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 by

$$\text{Gen.err} = \frac{1}{126}\sum_{i=0}^{125}\left[f(-\pi + 0.05i) - f_0(-\pi + 0.05i)\right]^2. \tag{50}$$

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

Learning results in the case $Q_m = I_m$ are shown in the upper half of Fig.4. Solid and dashed lines denote the original function f and a learning result of each method, respectively. $\circ$ indicates training data. The generalization errors of IPL, RAN, and on-line BP measured by eq.(50) are 0.32, 0.86, and 3.25, respectively. The results say that IPL provides a better generalization capability than RAN and on-line BP do. Note that RAN also works well in this simulation. Learning results in the case $Q_m = 3I_m$ are shown in the bottom half of Fig.4. 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 a better generalization capability than RAN and on-line BP do. The generalization errors of the learning results 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 criterion aims for minimizing the generalization error while the criteria of RAN and on-line BP aims for fitting an additional datum. 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 to the noisy data by smoothing a learning result, which is achieved by appropriately determining the width of RBFs, the number of hidden units, *etc.* Since a learning result obtained by IPL is exactly the same as that obtained by batch projection learning, IPL provides a better generalization capability than RAN and on-line BP do.

8 Conclusion

A method of incremental projection learning in the presence of noise was presented. The proposed method provides exactly the same learning result as that obtained by batch projection learning even in the non-asymptotic case. It is demonstrated through computer simulations that the proposed method provides a better generalization capability than RAN and on-line BP do.

References

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

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

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

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

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

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

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

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

[9] Ogawa, H. (1992). Neural network learning, generalization and over-learning. *Proceedings of the ICIIPS'92, 1*, Beijing, China, 1–6.

[10] 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.

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

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

[13] Sugiyama, M., & Ogawa, H. (1998a). Active learning for noise suppression. *IEICE Technical Report, NC98-21*, 87–94. (in Japanese).

[14] Takemura, A. (1991). *Modern mathematical statistics.* Sobunsha (in Japanese).

[15] Vijayakumar, S., & Ogawa, H. (1998). RKHS based functional analysis for exact incremental learning. *Neurocomputing : Special Issue on Theoretical analysis of real valued function classes*, Elsevier Science (in press).

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

[17] Vyšniauskas, V., Groen, F. C. A., & Kröse, B. J. A. (1995). Orthogonal incremental learning of a feedforward network. *Proceedings of ICANN*, Paris, France, 311–316.

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

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

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