RKHS based Functional Analysis for Exact Incremental Learning

Sethu Vijayakumar*
RIKEN Brain Science Institute,
The Institute of Physical & Chemical Research,
Wako, Saitama 351-0198, Japan. Email: sethu@brain.riken.go.jp

Hidemitsu Ogawa
Department of Computer Science,Tokyo Institute of Technology
Meguro-ku, Tokyo-152, Japan. Email: ogawa@cs.titech.ac.jp

1 Abstract

We investigate the problem of incremental learning in artificial neural networks by viewing it as a sequential function approximation problem. A framework for discussing the generalization ability of a trained network in the original function space using tools of functional analysis based on Reproducing Kernel Hilbert Spaces(RKHS) is introduced. Using this framework, we devise a method of carrying out optimal incremental learning with respect to the entire set of training data by employing the results derived at the previous stage of learning and incorporating the newly available training data effectively. Most importantly, the incrementally learned function has the same (optimal) generalization ability as would have been achieved by using batch learning on the entire set of training data, hence, referred to as *exact* learning. This ensures that both the learning operator and the learned function can be computed using an online incremental scheme. Finally, we also provide a simplified closed form relationship between the learned functions before and after the incorporation of new data for various optimization criteria, opening avenues for work into selection of optimal training set. We also show that learning under this kind of framework is inherently well-suited for applying novel model selection strategies and introducing bias and apriori knowledge in a more systematic way. Moreover, it provides a useful hint in performing kernel based approximations, of which the regularization and SVM networks are special cases, in an online setting.

Keywords: Reproducing Kernel Hilbert Space(RKHS), functional analysis, incremental learning, generalization, model selection.

*Corresponding author. Tel +81-48-467-9664. Email: sethu@brain.riken.go.jp

2 Introduction

Multi-layer feedforward neural networks are essentially approximators, mapping from a finite dimensional input space to an output space. This paper discusses the learning problem in these kinds of networks as an inverse problem from the functional analytic point of view. The functional analytic approach to learning in Neural Networks(NNs) has been proved to improve generalizing ability because of their focus on the original function space rather than the sampled space. In earlier research work on this approach [13, 14], learning has been discussed under the framework of obtaining an optimal learning operator from a given set of training data. However, there is a need to look at cases where we have an optimally trained NN and there are additions to the training data to be incorporated at a later stage. Re-learning using the entire training set is a possible but not very efficient solution. These kind of situations also arise when we are dealing with 'active learning' in which the set of training data can be provided only one after the another and also uses the intermediate learned results for deciding on the next best sample. Recently, a lot of work on these lines have being carried out, some of them being the papers by Zhang[32], Jutten[5] et al. However, most of them, though being *incremental* in the nature of addition of training data or increase in the number of hidden units etc., almost always involves re-learning or re-training with the addition of information or units. Other incremental approaches like Kadirkamanathan and Niranjan[6] and Yingwei[31] have worked on specific radial basis function neural networks with sequential learning capability. However, in their work, the optimization criterion used reduces to an arbitrarily smoothed version of the memorization learning criterion, which may not be appropriate from a generalization point of view or the learning task at hand, as explained later. Dunkin, Shawe-Taylor and Koiran [3] have looked at an approximate incremental learning scheme based on training the network nodes one at a time and freezing the remaining network to realize a speed-up. This method relies on an importance factoring down scheme to correct mistakes learned by the intermediate stages.

In this paper, we use the inverse problem approach to formalize the concept of learning in neural networks without imposing any restriction on the type of basis functions used. The batch solutions for various optimization criteria are analytically derived, which in the present context turns out to be a kernel based approximation in Reproducing Kernel Hilbert Spaces. RKHS theory has been a well studied topic, stemming from the original works of Aronszajn[2], Kimeldorf and Wahba[7] and Parzen[17] to more recent studies on their application by Wahba[28], Saitoh[19] and many others. This kind of kernel based approximation scheme is also closely linked to regularization theory[4] and Support Vector Machines [23] based approximation schemes, a topic that we will briefly touch upon later. Based on this framework, an efficient method to incorporate new data incrementally, without re-learning from scratch, is formulated. This incremental method is shown to reduce the computational overhead while computing the new learning operator. Also, we analytically derive a relationship between the learned function before and after the addition of the new training datum, a result that can be utilized in the problems of active training data selection.

An important point to be noted is that the results are extendable to any general kind of optimization criterion; although from a generalization point of view, an optimization criterion which focuses on the original function space rather than the sampled space is recommended [14]. Moreover, the incremental learning worked on here is *exact* in the sense that batch learning on the complete data set would give exactly the same results.

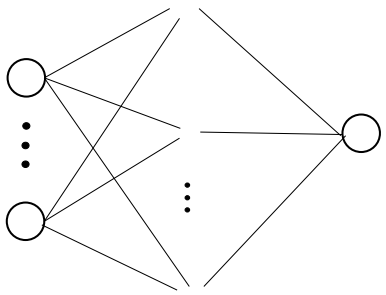


Figure 1: NN as a real valued function

3 Neural Network learning as an inverse problem

In this section, we provide the basic functional analytic formulation of the learning problem and define the notations and operators used for the analysis.

Let us begin by considering a three-layer feedforward neural network whose number of input, hidden, and output units are L, N, and 1, respectively as shown in Fig.1. Let $\xi_1, \xi_2, ..., \xi_L$ be the inputs to the input layer neurons. Let these inputs constitute the elements of an L-dimensional vector x, referred to as the 'input vector'. An input-output relation of the n-th hidden unit is denoted by $u_n(x)$ and called the basis function. In conventional NNs, the following basis function is used:

$$u_n(x) = \sigma(\sum_{l=1}^{L} w_{nl}\xi_l), \tag{1}$$

where $\sigma(.)$ is a sigmoidal function. However, here we generalize to any basis functions of L-variables. Let w_n be the corresponding weight connecting the n-th hidden unit and the output unit. In this case, the network can be considered as a real valued function $f_0(x)$ of L variables and expressed as follows:

$$f_0(x) = \sum_{n=1}^{N} w_n u_n(x). \tag{2}$$

Let the set of M input vectors $\{x_m\}_{m=1}^{M}$ be the *training set*, $\{y_m\}_{m=1}^{M}$ represent the corresponding desired output values, where $y_m = f(x_m)$ and $y \equiv (y_1\, y_2 \,...\, y_m)^T$ denote the vector whose elements contain the output values. The *learning problem* is to construct a neural network by using a training set so that the NN expresses the best approximation $f_0(x)$ to a desired function $f(x)$ under some given learning criterion.

The learning problem of neural networks can, hence, be discussed from the point of view of function approximation. In other words, the learning problem can be regarded as the same problem as that of the general sampling theorem [12] and image restoration [15]. In this context, x_m and y_m can be called the m-th *sample point* and the corresponding *sample value* of the desired function $f(x)$, respectively.

Let H be the set of functions which includes $f(x)$, the function to be approximated by the neural network. Assume that H is a Reproducing Kernel Hilbert Space(RKHS) with a *reproducing kernel* $K(x, x')$. Let D be the domain of the basis functions, which is a subset of the L-dimensional Euclidean space $\mathbb{R}^L$. The reproducing kernel $K(x, x')$ is a bivariate function defined on $D \times D$ which satisfies the following two conditions:

1. For any fixed x' in D , $K(x, x')$ is a function in H.
2. For any function f in H and for any x' in D, it holds that
$$(f(x), K(x, x')) = f(x'), \tag{3}$$
where the left hand side of eq.(3) denotes the inner product in H.

In the theory of Hilbert space, arguments are developed by regarding a function as a point in that space. Thus, things such as 'value of a function at a point' cannot be discussed under the general framework of Hilbert space. However, if the Hilbert space has a reproducing kernel[1], then it is possible to deal with the value of a function at a point. Indeed, if we define functions $\psi_m(x)$ as
$$\psi_m(x) = K(x, x_m) : 1 \leq m \leq M, \tag{4}$$
then, the value of f at a sample point x_m is expressed in Hilbert space language as the inner product of f and ψ_m as
$$f(x_m) = (f, \psi_m). \tag{5}$$
Once the training set $\{x_m\}_{m=1}^M$ is fixed, the vector $y \equiv (y_1\, y_2 \ldots y_m)^T$ is uniquely determined from f. So, we can introduce an operator A which transforms f to y:
$$y = Af. \tag{6}$$
The operator A, called the *sampling operator*, becomes a linear operator even when we are concerned with nonlinear neural networks. It is expressed by using the Schatten product as
$$A = \sum_{m=1}^{M} e_m \otimes \overline{\psi_m}, \tag{7}$$
where $\{e_m\}_{m=1}^M$ is the so-called natural basis in $\mathbb{R}^M$, i.e., the vector e_m is the M-dimensional vector consisting of zero elements except the element m equal to 1. The Schatten product denoted by $(. \otimes \bar{.})$ is defined by
$$(e_m \otimes \overline{\psi_m})f = (f, \psi_m)e_m. \tag{8}$$
Hence, the learning problem can be reformulated as the problem of obtaining an estimate , say f_0, to f from y in the model (See Fig.2). This can be considered as an *inverse problem* equivalent to obtaining an operator X which provides f_0 from y:
$$f_0 = Xy. \tag{9}$$
The operator X is called the *learning operator*. It is also referred to elsewhere in other fields by different names like *restoration operator, reconstruction operator, approximation operator*, etc. This operator X can be optimized based on different criteria and using various techniques [21, 10]. Examples of the *learning criteria* we will work with includes the memorization criterion, the Wiener criterion, the projection criterion, etc.

[1] A Hilbert space always possesses a reproducing kernel if it is separable[2]

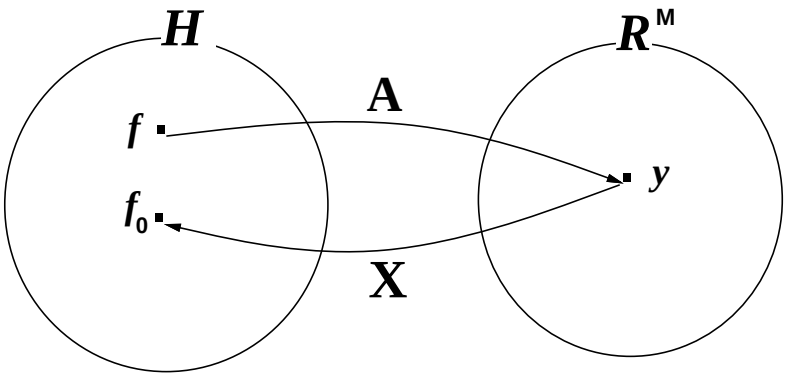


Figure 2: NN learning as an inverse problem

4 Learning with optimal generalization ability

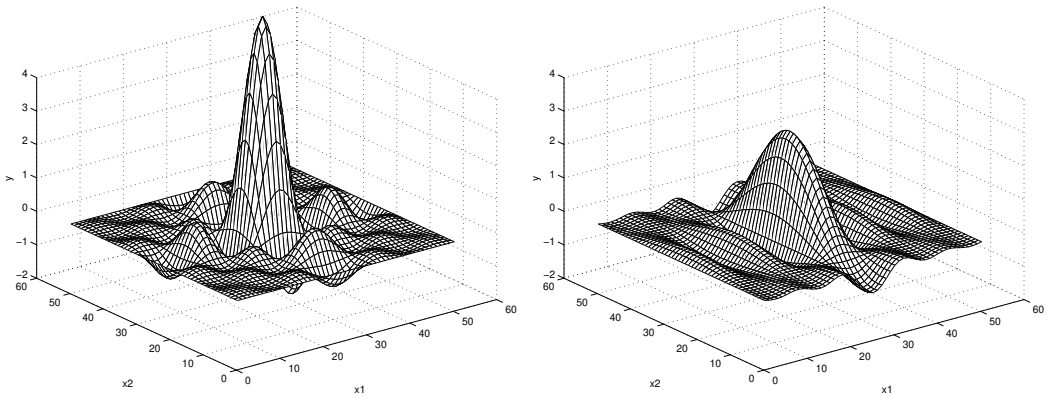
In approximation theory, we clearly see three stages in the development of the learning problem (Poggio and Girosi[18]). First, there is the *model selection* problem, which addresses which space to search for the optimal approximation. This is equivalent to the problem of selecting the architecture, the number of hidden units and basis functions in conventional NN literature. In our case, this is analogous to the question of which Hilbert space is relevant for a particular class of problems. The other two stages involves finding the *optimal approximation* within this model space and devising fast and cheap *implementations* of the algorithm to obtain this approximation. In this section, we will deal, in turn, with each of the above issues for the batch learning case; results which will be extended to the incremental learning scenario in the following section.

4.1 Model selection and optimal search space

In Section 3, we talked about the Hilbert space H that contains the target function we are trying to approximate or learn. The selection of this search space has a huge impact on the approximation results and is studied under model selection. Although the emphasis of this paper is not on the model selection problem, in this section, we show that this framework is already well adapted for carrying out model selection and introducing bias in a *systematic* way. We show that the optimal basis functions for the approximation are automatically determined based on the search space used.

A function space that has extensive practical significance in one or multi dimensional signal processing is the space of *band-limited* functions, defined as a function whose Fourier transform is zero outside a hyper-rectangle[2] in the frequency domain. Such a function space forms a Hilbert space (also referred to as the Paley-Wiener space) and it's reproducing kernel can be determined as follows. Consider for simplicity, the function of two variables; if the

[2]It has been pointed out by one of the reviewers that band-limitation using hyper-rectangle cutoffs, which result in a tensor-product kernel in multidimensional case, may introduce preferred directions in the frequency distribution. An alternate radially symmetric frequency distribution can be considered to overcome this flaw.



(a) (b)

Figure 3: Reproducing kernels for 2-dimensional band-limited function with cutoff frequency parameters (a) $\sigma_1 = 8$, $\sigma_2 = 8$ and (b) $\sigma_1 = 10$, $\sigma_2 = 3$. Note the radial asymmetry in (b).

Fourier transform $F(\omega_1, \omega_2)$ of the function $f(x) = f(\xi_1, \xi_2)$ is zero for $|\omega_1| > \sigma_1, |\omega_2| > \sigma_2$, then the sampling series is based on the functions

$$K(x) = \frac{\sigma_1 \sigma_2}{\pi^2} sinc[\frac{\sigma_1}{\pi}\xi_1] sinc[\frac{\sigma_2}{\pi}\xi_2], \tag{10}$$

where *sinc* is the function

$$sinc(u) = sin(\pi u)/\pi u. \tag{11}$$

The function $K(x, x') = K(x - x')$ forms the reproducing kernel for the Hilbert space of band-limited functions. Fig.3 shows examples of such two-dimensional reproducing kernels for band-limited function spaces defined by parameters (a) σ_1=8, σ_2=8 and (b) σ_1=10, σ_2=3.

The approximation using these kinds of Reproducing Kernel Hilbert Spaces (RKHS) is close to the Radial Basis Function (RBF) networks but the kernel of eq.(10) or it's generalization to L dimensions,

$$K(x, x') = \prod_{l=1}^{L} \frac{\sigma_l}{\pi} sinc[\frac{\sigma_l}{\pi}(\xi_l - \xi'_l)], \tag{12}$$

with $x = (\xi_1, ..., \xi_l)$, $x' = (\xi'_1, ..., \xi'_l)$, is not radially symmetrical. However, it has a central 'lump' whose width and shape is controlled automatically by the L parameters σ_l specifying the band-limitation in the frequency domain. Note that the sampling theorem provides a proof that asymptotically, approximations using these kinds of basis functions can approximate any band-limited function to arbitrary accuracy. Adjustment of the parameters σ_l in eq.(12), which can potentially make this frequency band arbitrarily large, can be carried out through cross validation or other model selection techniques like complexity penalization in addition to systematically introducing prior biases on the smoothness by controlling the frequency content of the search space. An example of learning using the Paley-Wiener space is shown in Section 6.2, clearly illustrating the bias-variance tradeoff that results. A standard practice in NN literature of using regularizers e.g. in splines, may be looked upon

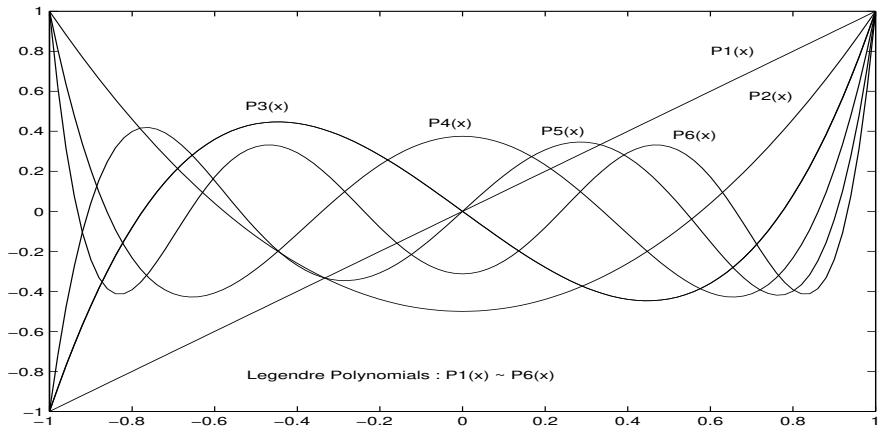


Figure 4: Legendre Polynomials : An example of orthogonal polynomials of increasing degree

as methods of controlling the frequency content of the search space, although their exact relationship is still a topic of active research. More about their relationship will be handled under the discussion at the end of the paper.

Another approach to selection of the relevant Hilbert space H involves parameterization of the space based on the highest degree of polynomial within the space. For example, if we consider the Gram-Schmidt orthogonalization of the independent polynomial functions $1, t, t^2, ...$ within the Hilbert space $L_2[-1, 1]$, we can generate a set of orthonormal sequences given by

$$e_n(t) = \sqrt{\frac{2n+1}{2}} P_n(t), \qquad n = 0, 1, 2, ... \tag{13}$$

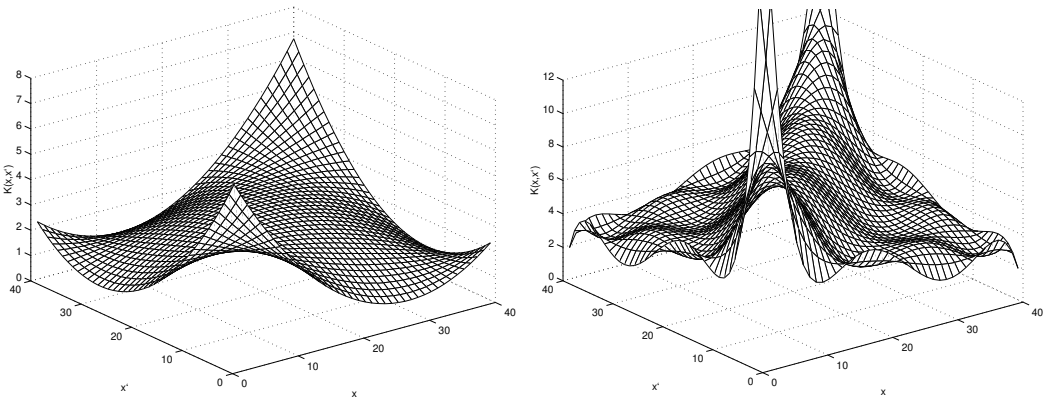
where $P_n(t)$ are the well known Legendre polynomials (See Fig.4)

$$P_n(t) = \frac{1}{2^n n!} \frac{d^n}{dt^n}(t^2 - 1)^n. \tag{14}$$

Based on this set of OrthoNormal Bases (ONB), we can generate the reproducing kernel for a function subspace with the highest degree of polynomial N as

$$\begin{aligned} K(x, x') &= \sum_{n=0}^{N} e_n(x) e_n(x') \\ &= \begin{cases} \frac{(N+1)^2}{2(1-x^2)}[P_N(x)^2 - 2xP_N(x)P_{N+1}(x) + P_{N+1}(x)^2] & \text{if } x = x' \\ \frac{N+1}{2(x-x')}[P_{N+1}(x)P_N(x') - P_N(x)P_{N+1}(x')] & \text{otherwise.} \end{cases} \end{aligned} \tag{15}$$

Fig.5 shows how the reproducing kernel looks graphically for spaces with highest polynomial degrees equals to two and five, respectively. The x and x' axis represent the variables of the bi-variate reproducing kernel function and the z axis shows the value taken by these functions.



(a) (b)

Figure 5: Reproducing kernels $K(x, x')$ for function spaces spanned by polynomial functions with (a) highest degree polynomial = 2 and (b) highest degree polynomial = 5

A family of such stratified function spaces can be used with novel model selection strategies as described in Schuurmans[20] etc. Of course, the choice of the appropriate kernels for a given problem should reflect the prior knowledge on data like, for example, the smoothness properties of the desired solution. Another important question is: which Hilbert space and kernels correspond to standard approximation schemes used conventionally. Table 1 gives examples of kernel functions corresponding to some RKHS and the type of decision surface they describe, recovering some well known approximation schemes like Gaussian RBF, MLP under constraint etc.

Kernel Functions	**Approximation Scheme**
$K(x,x') = \frac{\sigma}{\pi} sinc[\frac{\sigma}{\pi}(x-x')]$	Band-limited Paley Wiener space
$K(x,x') = exp(-\|x-x'\|^2)$	Gaussian RBF
$K(x,x') = (1 + x * x')^d$	Polynomial of degree d
$K(x,x') = tanh(x * x' - \theta)$	Multi layer perceptrons[3]
$K(x,x') = B_{2n+1}(x-x')$	B-Splines[4]
$K(x,x') = \frac{\sin(d+1/2)(x-x')}{\sin(x-x')/2}$	Trigonometric polynomials of degree d

Table 1: Examples of RKHS Kernels and the decision surfaces they define.

An extensive analysis of Reproducing Kernel Hilbert Spaces (RKHS) including Sobolev spaces for different kinds of problem domains can be found in Saitoh[19]. A separate paper will address approaches and results of model selection under this framework, especially for real world problems like the learning of the sensorimotor map in a complex anthropomorphic robot arm.

[3] Only for some values of θ

[4] B_n are piecewise polynomials of degree n, definitions of which can be found in Unser et al.[22].

4.2 Optimization Criteria

Once we decide on the search space, next, we need some kind of measure to evaluate our learning results. As described in Section 3, the learning operator X can be optimized based on different criteria. Here, we will discuss three such specific criteria and provide analytical solutions for optimization under the linear search constraint. In practice, the suitable criterion should be chosen based on the requirements of the problem.

4.2.1 Wiener Criterion

The mathematical representation of the *Wiener criterion* J_W for the *noiseless* case is given in the following equation:

$$\text{min. over } X : J_W[X] = E_f \|XAf - f\|^2, \tag{16}$$

where $\|.\|$ is the norm in H and E_f is the expectation taken over the ensemble $\{f\}$. An operator X satisfying the above criterion is called a Wiener learning operator. The criterion aims at reducing the difference between the original function f and the function f_0 reconstructed by using the learning operator X. The thing to be noted is that this minimization is done in the original function space H and in an *averaged sense* with respect to all the possible functions to be approximated.

The Wiener criterion can be transformed into a more useful form [16]. Let R be the *correlation operator* of the function ensemble and be defined as

$$R = E_f(f \otimes \overline{f}). \tag{17}$$

In order to use the Wiener criterion, it is essential that we know or estimate the correlation operator R, which acts as an apriori information on the distribution in the function space. This apriori is analogous to the Bayesian prior i.e., the covariance matrix in Gaussian processes [29].

The necessary and sufficient condition for the Wiener criterion to be satisfied by an operator X is given as

$$XARA^* = RA^*, \tag{18}$$

where A^* is the adjoint operator of A. A general form of the solution of eq.(18) is given as

$$X = RA^*U^\dagger + W_1(I - UU^\dagger), \tag{19}$$

where W_1 is any operator from $\mathbb{R}^M$ to H, $U^\dagger$ represents the Moore-Penrose generalized inverse of U [1] and U is defined as

$$U = ARA^*. \tag{20}$$

Once the training set is fixed, A can be calculated using eqs.(7) and (4). Hence, corresponding to this sampling operator, a learning operator X satisfying the Wiener criterion can be obtained using eqs.(19) and (20).

4.2.2 Memorization Criterion

Another optimization criterion is the Memorization criterion . It minimizes the following functional:

$$\text{min. over } X : J_M[X] = \|AXy - y\|^2 = \sum_{m=1}^{M} (f_0(x_m) - y_m)^2, \tag{21}$$

where $\|.\|$ is the norm in the sampled space $\mathbb{R}^M$. As is evident from the criterion, it reduces error over the sampled space. Minimization of the above functional yields the general form of the solution for X as

$$X = A^\dagger + W_2 - A^\dagger A W_2 A A^\dagger, \tag{22}$$

where W_2 is any operator from $\mathbb{R}^M$ to H. In standard backpropagation algorithms, the Memorization criterion is generally employed along with an additional *regularizer* term, which is decided based on some heuristics.

4.2.3 Projection Criterion

Finally, we discuss about the Projection optimization criterion. The criterion for X to satisfy Projection learning is that for all $f \in H$,

$$f_0 = XAf, \tag{23}$$

becomes the best approximation (orthogonal projection) of function f in a subspace $\mathcal{R}(A^*)$. The minimization of this criterion leads to the general form of the solution:

$$X = A^\dagger + W_3(I - AA^\dagger), \tag{24}$$

where W_3 is any operator from $\mathbb{R}^M$ to H.

In general, from an optimal generalization point of view, an optimization criterion which reduce errors over the original function space (eg. Wiener, Projection) is recommended over those which reduce errors in the sampled space (Memorization) or other variations of it (eg. Memorization + regularization term). However, practical computational considerations and lack of prior knowledge may limit approximation ability to the extent obtained through memorization learning.

4.3 Optimally Generalizing Neural Network

A practical implementation of the Optimally Generalizing Neural Network (OGNN) can be realized by determining the number of hidden units N, the basis functions of the hidden units $\{u_n\}_{n=1}^N$ and the connection weights $\{w_n\}_{n=1}^N$ (refer Fig.1) as follows:

1. Fix N such that
$$N \geq dim(\mathcal{R}(XA)) = rank(XA). \tag{25}$$
2. Fix $\{u_n\}_{n=1}^N$ such that
$$\mathcal{L}(\{u_n\}_{n=1}^N) \supset \mathcal{R}(XA). \tag{26}$$
3. Compute the vector of weights $w \equiv (w_1 w_2 ... w_N)^T$ as
$$w = (T^*)^\dagger Xy + (I - TT^\dagger)z, \tag{27}$$
where $T = \sum_{n=1}^N (e_n \otimes \overline{u_n})$ and z is any N dimensional vector.

5 Incremental learning with optimal generalization

So far, the learning problem has been discussed under the basic framework of finding an optimum learning operator X from the given set of training data. In other words, the sampling operator A is fixed based on the training data as shown in eqs.(4) and (7), and then, the corresponding learning operator is optimized based on various criteria. If an additional training data is made available at a later stage, it would involve recalculating the learning operator from scratch to incorporate this new information, which is highly inefficient and computationally expensive.

However, we now devise an incremental learning scheme which utilizes the results of training computed so far and incorporates the newly added information in a computationally efficient manner. An important point to be noted here is that, unlike other methods, the generalization ability achieved using this incremental learning technique is exactly equal to that achieved while using *batch* learning for optimal generalization on the entire set of training data (including the newly added one) as discussed in the previous section; in other words, this form of incremental learning is exact rather than an approximation.

Since we will be dealing with a variable number of sample points and hence, a variable dimensional sampled space, we will use lower case letters to represent the number of training data and the dimension of the sampled space from now on. Let $\{e_n^{(m)}\}_{n=1}^m$ represent the natural basis of the sampled space $\mathbb{R}^m$. Let A_m be the sampling operator from H to $\mathbb{R}^m$ while considering m sample points, defined as

$$A_m = \sum_{n=1}^{m} (e_n^{(m)} \otimes \overline{\psi_n}). \tag{28}$$

Let $y^{(m)}$ be the vector consisting of elements $\{y_n\}_{n=1}^m$ and X_m be the corresponding learning operator for the m sample points. In the following subsections, we will derive the incremental results for the Wiener optimization criterion in detail, which can be extended for solving for other optimization criterion results shown in the next section.

5.1 Incremental computation of the learning operator and the learned function for the Wiener optimization criterion

First, we present a general result for the recursive computation of the generalized inverse which will be used in our incremental computation of the learning operator.

Let V_m be an $m \times m$ *positive semi-definite matrix*. Let V_{m+1} be represented in the form

$$V_{m+1} = \begin{pmatrix} V_m & s_m \\ s_m^T & \sigma \end{pmatrix}, \tag{29}$$

where s_m and σ represent an m-dimensional vector and scalar, respectively. We assume that $V_0 = 0$, $s_0 = 0$. Let t_m and α be defined as

$$t_m = V_m^\dagger s_m, \tag{30}$$

$$\alpha = \sigma - (t_m, s_m). \tag{31}$$

Lemma 1 *[1] If V_{m+1} is positive semi-definite, then, $\alpha \geq 0$ and the Moore-Penrose generalized inverse of V_{m+1} for $\alpha > 0$ is given as*

$$V_{m+1}^{\dagger} = \begin{bmatrix} V_m^{\dagger} + \frac{1}{\alpha}(t_m \otimes \overline{t_m}) & -\frac{1}{\alpha}t_m \\ -\frac{1}{\alpha}t_m^T & \frac{1}{\alpha} \end{bmatrix}. \tag{32}$$

The result for the condition $\alpha = 0$ has been omitted because this condition works out to be a trivial case in the context of the learning problem since it represents a redundant training data (refer to Vijayakumar and Ogawa[26] for the proof). Therefore, we will now deal with the case $\alpha > 0$ only.

We shall apply Lemma 1 to our incremental learning operator computation. Considering a particular solution for the Wiener learning, i.e., taking $W = 0$ in eq.(19), we have

$$X_{m+1} = RA_{m+1}^{*}U_{m+1}^{\dagger}, \tag{33}$$

where

$$U_{m+1} = A_{m+1}RA_{m+1}^{*}. \tag{34}$$

In order to express the new sampling operator A_{m+1} in terms of the old sampling operator A_m, we introduce an operator T defined as

$$T = \sum_{n=1}^{m}(e_n^{(m+1)} \otimes \overline{e_n^{(m)}}). \tag{35}$$

Now, we can write the new sampling operator A_{m+1} as

$$A_{m+1} = TA_m + (e_{m+1}^{(m+1)} \otimes \overline{\psi_{m+1}}) \tag{36}$$

because of eq.(28).

Let

$$\begin{aligned} s_m &= A_m R\psi_{m+1}, && (37)\\ \sigma &= (R\psi_{m+1})(x_{m+1}), && (38)\\ t_m &= U_m^{\dagger}s_m, && (39)\\ \alpha &= \sigma - (t_m, s_m). && (40) \end{aligned}$$

It follows from eqs.(34) and (36) that

$$U_{m+1} = TU_mT^{*} + (Ts_m \otimes \overline{e_{m+1}^{(m+1)}}) + (e_{m+1}^{(m+1)} \otimes \overline{Ts_m}) + \sigma(e_{m+1}^{(m+1)} \otimes \overline{e_{m+1}^{(m+1)}}). \tag{41}$$

Looking at different terms of the expansion, it is easy to show that the matrix U_{m+1} can be written in the form

$$U_{m+1} = \begin{pmatrix} U_m & s_m \\ s_m^T & \sigma \end{pmatrix}, \tag{42}$$

as shown in Fig.6, which also corresponds to the structure of eq.(29). It can also be proved that the matrix represented by the terms U_{m+1} is *positive semi-definite* because of eq.(34). Therefore, Lemma 1 can be applied to calculate the term $U_{m+1}^{\dagger}$ recursively for the condition $\alpha > 0$.

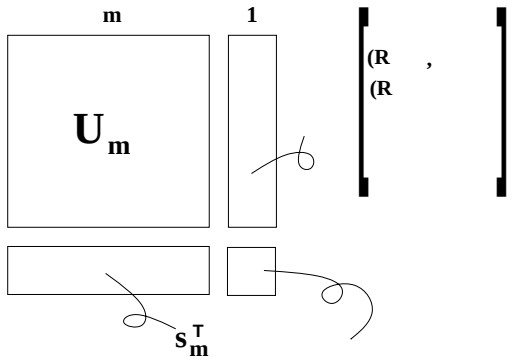


Figure 6: Structure of the matrix U_{m+1}

Theorem 1 *Let s_m, σ, t_m and α be defined as in eqs.(37), (38), (39) and (40), respectively. If $\alpha > 0$, then the new learning operator X_{m+1} can be computed incrementally from the previous learning operator X_m as*

$$X_{m+1} = (X_m + Y_m)T^* + (v_m \otimes \overline{e_{m+1}^{(m+1)}}), \tag{43}$$

where

$$v_m = \frac{1}{\alpha}(R\psi_{m+1} - RA_m^* t_m), \tag{44}$$

$$Y_m = -(v_m \otimes \overline{t_m}), \tag{45}$$

Proof: It follows from eqs.(33) and (36) that

$$X_{m+1} = R[TA_m + (e_{m+1}^{(m+1)} \otimes \overline{\psi_{m+1}})]^* U_{m+1}^\dagger. \tag{46}$$

Since U_{m+1} in eq.(46) is positive semi-definite and has the structure as shown in Fig.6, we can apply Lemma 1 to eq.(46). Hence, we have

$$X_{m+1} = R[A_m^* T^* + (\psi_{m+1} \otimes \overline{e_{m+1}^{(m+1)}})] \begin{bmatrix} U_m^\dagger + \Gamma & \rho \\ \rho^T & \tau \end{bmatrix}, \tag{47}$$

where Γ, ρ, and τ are defined as below.

$$\Gamma = \frac{1}{\alpha}(t_m \otimes \overline{t_m}), \tag{48}$$

$$\rho = -\frac{1}{\alpha} t_m, \tag{49}$$

$$\tau = \frac{1}{\alpha}. \tag{50}$$

From eq.(47), we have

$$\begin{aligned} X_{m+1} &= R[A_m^* T^* + (\psi_{m+1} \otimes \overline{e_{m+1}^{(m+1)}})][T(U_m^\dagger + \Gamma)T^* + (T\rho \otimes \overline{e_{m+1}^{(m+1)}}) \\ &\quad + (e_{m+1}^{(m+1)} \otimes \overline{T\rho}) + \tau(e_{m+1}^{(m+1)} \otimes \overline{e_{m+1}^{(m+1)}})]. \end{aligned} \tag{51}$$

Since $T^*T = I$ and $T^* e_{m+1}^{(m+1)} = 0$, it follows from eqs.(51)and (33) that

$$\begin{aligned} X_{m+1} &= RA_m^*(U_m^\dagger + \Gamma)T^* + (RA_m^*\rho \otimes \overline{e_{m+1}^{(m+1)}}) + (R\psi_{m+1} \otimes \overline{T\rho}) \\ &\quad + \tau(R\psi_{m+1} \otimes \overline{e_{m+1}^{(m+1)}}) \\ &= (X_m + RA_m^*\Gamma + (R\psi_{m+1} \otimes \overline{\rho}))T^* + (RA_m^*\rho + \tau R\psi_{m+1}) \otimes \overline{e_{m+1}^{(m+1)}}. \end{aligned} \tag{52}$$

Substituting eqs.(48),(49), and (50) into eq.(52), we have

$$\begin{aligned} X_{m+1} &= X_m T^* + [\frac{1}{\alpha}(RA_m^*(t_m \otimes \overline{t_m}) - R(\psi_{m+1} \otimes \overline{t_m}))]T^* u \\ &\quad + [\frac{-1}{\alpha}(RA_m^* t_m - R\psi_{m+1}) \otimes \overline{e_{m+1}^{(m+1)}}] \\ &= [X_m - (v_m \otimes \overline{t_m})]T^* + (v_m \otimes \overline{e_{m+1}^{(m+1)}}) \\ &= (X_m + Y_m)T^* + (v_m \otimes \overline{e_{m+1}^{(m+1)}}). \end{aligned} \tag{53}$$

Hence, the theorem holds.

Theorem 2 *When $\alpha > 0$, the learned function f_{m+1} due to $m+1$ sample points can be computed incrementally from the previous learned function f_m and the new sample value $f(x_{m+1})$ as*

$$f_{m+1} = f_m + Y_m y^{(m)} + f(x_{m+1})v_m, \tag{54}$$

where Y_m v_m and $Y_m$are given by eqs.(44) and (45), respectively.

Proof: From the general form of eq.(9), we can write the newly learned function f_{m+1} as

$$f_{m+1} = X_{m+1} y^{(m+1)}. \tag{55}$$

Using eq.(43) of Theorem 1 to compute X_{m+1} in eq.(55), we have

$$\begin{aligned} f_{m+1} &= (X_m + Y_m)T^* y^{(m+1)} + (v_m \otimes \overline{e_{m+1}^{(m+1)}})y^{(m+1)} \\ &= (X_m + Y_m)y^{(m)} + (y^{(m+1)}, e_{m+1}^{(m+1)})v_m, \end{aligned}$$

which yields eq.(54).

On reducing eq.(54), we can further clarify the relationship between f_m and f_{m+1}, which we will proceed to do in the next theorem.

Let S_m and S_{mc} be defined as

$$S_m = \mathcal{R}(RA_m^*), \tag{56}$$

$$S_{mc} = \mathcal{R}(R) \cap \mathcal{N}(A_m), \tag{57}$$

where $\mathcal{R}(.)$ and $\mathcal{N}(.)$ refer to the range and the null space of an operator, respectively. The following lemma holds.

Lemma 2 *[26] For Wiener learning, the learned function $f_m = X_m A_m f$, for every $f \in \mathcal{R}(R)$, is an oblique projection of f onto S_m along S_{mc}, that is,* [5]

$$f_m = X_m A_m f = P_{S_m, S_{mc}} f. \tag{58}$$

Let f_{mc} be the complement projection of f_m, i.e. a projection of f onto S_{mc} along S_m. Now, consider a function ϕ defined as

$$\phi \equiv R\psi_{m+1}. \tag{59}$$

Let ϕ_m and ϕ_{mc} be the projection of ϕ onto S_m and S_{mc} , respectively. Then, it can be proved, as shown in Appendix A.1, that α of eq.(31) is equivalent to the value of the function ϕ_{mc} at the new sampled location x_{m+1}, that is,

$$\alpha = \phi_{mc}(x_{m+1}). \tag{60}$$

Using these relations and further reducing eq.(54), we have the following theorem.

Theorem 3 *[26] When $\alpha > 0$, it holds that*

$$f_{m+1} = f_m + \frac{f_{mc}(x_{m+1})}{\phi_{mc}(x_{m+1})}\phi_{mc}. \tag{61}$$

A proof of the theorem is provided in Appendix A.2.

A schematic diagram of the relationship is shown in Fig.7. It is clear from eq.(61) and Fig.7 that the newly learned function f_{m+1} can be expressed as the sum of f_m and a scalar multiple of the function ϕ_{mc}. Here, ϕ_{mc} is the projection of ϕ onto S_{mc}, which again depends on the location of the new sample due to eqs.(59) and (4). Another point to be noted is that the term $f_{mc}(x_{m+1})$ which appears in the scalar part of eq.(61) is, in fact, the error between the actual function and the learned function at the new sample point, i.e.,

$$f_{mc}(x_{m+1}) = (f - f_m)(x_{m+1}). \tag{62}$$

A closed form relationship between the newly sampled location x_{m+1}, the newly learned function and the previous function provides a basis for the work on active data selection for improving generalization.

5.2 Generalized incremental learning results for various optimization criteria

Using a similar approach as adopted for the Wiener criterion, we can find incremental closed form solutions for the general class of optimization criterion as follows.

Theorem 4 (Incremental learning operator : projection learning) *[11]*
The incremental learning operator for projection learning can be computed as

$$X_{m+1}^{(P)} = (X_m^{(P)} - v_m^{(P)} \otimes \overline{t_m^{(P)}})\Gamma^* + v_m^{(P)} \otimes \overline{e_{m+1}^{(m+1)}}, \tag{63}$$

[5] $P_{B,C}$ refers to the projection onto a subspace B along a subspace C.

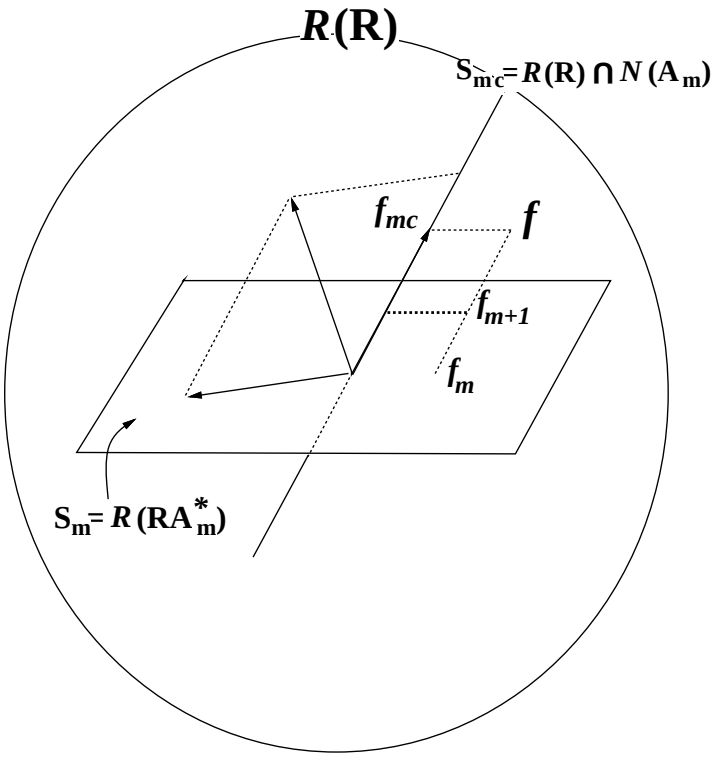


Figure 7: Relationship between f_m and f_{m+1} for Wiener optimization criterion.

where $v_m^{(P)}$, $t_m^{(P)}$ *and* $s_m^{(P)}$ *are defined as*

$$v_m^{(P)} = \frac{1}{\alpha}(\psi_{m+1} - A_m^* t_m^{(P)}), \tag{64}$$

$$t_m^{(P)} = (A_m A_m^*)^\dagger s_m^{(P)}, \tag{65}$$

$$s_m^{(P)} = A_m \psi_{m+1}. \tag{66}$$

Theorem 5 (Incrementally learned function : projection learning) *[11]*
The function learned by projection learning using $m+1$ *training data,* f_{m+1}*, can be incrementally computed as*

$$f_{m+1} = f_m + \frac{y_{m+1} - f_m(x_{m+1})}{\psi_{mc}(x_{m+1})}\psi_{mc}, \tag{67}$$

where ψ_{mc} *is the orthogonal projection of* ψ_m *onto* $\mathcal{N}(A_m)$*.*

Theorem 6 (Incremental learning operator : memorization learning) *[11]*
The incremental learning operator for memorization learning can be computed as

$$X_{m+1}^{(M)} = (X_m^{(M)} - \zeta_m \otimes \overline{h^{(m)}})\Gamma^* + \gamma_m \otimes \overline{e_{m+1}^{(m+1)}}, \tag{68}$$

where ζ_m, γ_m and $h^{(m)}$ are defined as

$$\begin{aligned} \zeta_m &= \frac{1}{\alpha}(\psi_m - A_m^* t_m^{(M)}), && (69) \\ \gamma_m &= \tilde{\eta}_m + (1 - \tilde{\eta}_m(x_m + 1))\zeta_m, && (70) \\ h^{(m)} &= t_m^{(M)} - K_m^\dagger K_m(\Omega_m^* t_m^{(M)} - q_m). && (71) \end{aligned}$$

Theorem 7 (Incrementally learned function : memorization learning) *[11] The function learned by projection learning using $m+1$ training data, f_{m+1}, can be incrementally computed as*

$$f_{m+1} = f_m + y_{m+1}\eta_{mc} + \frac{\beta - y_{m+1}\eta_{mc}}{\alpha}\psi_{mc} \tag{72}$$

where ψ_{mc} is the orthogonal projection of ψ_m onto $\mathcal{N}(A_m)$.

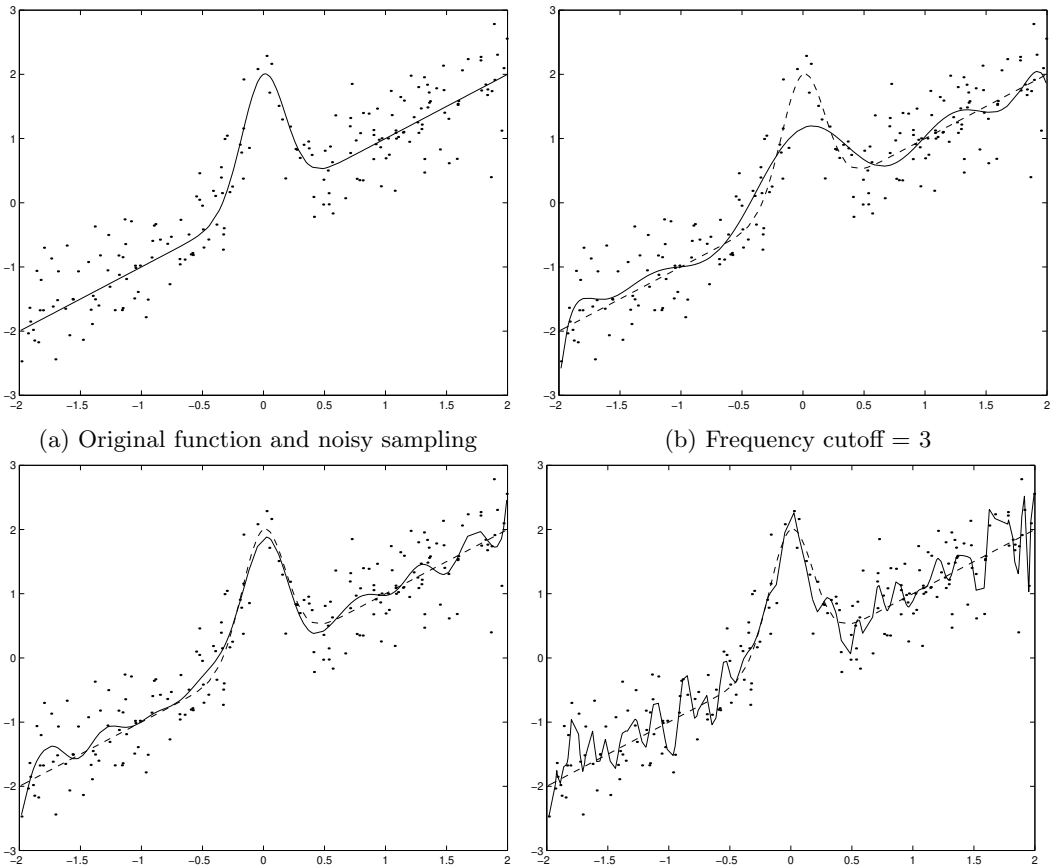
6 Empirical experiments

In this section, we will give the pseudocode for implementation of the algorithm introduced in the previous section and illustrate incremental function approximation in an artificial example.

6.1 Pseudocode of incremental learning algorithm

The pseudocode for implementation of incremental learning for optimal generalization for the Wiener optimization criterion is as follows.

1. Decide on optimal search space H based on model selection techniques described in Section 4.1.
2. Construct reproducing kernel.
   (a) Use existing reproducing kernel for H, if known. For example, for band-limited function spaces, use eq.(12).
   (b) Else, construct ONBs $\{\varphi_n\}_{n=1}^N$ for the Hilbert space H.
   Then, compute the reproducing kernel corresponding to the Hilbert space H as $K(x, x') = \sum_{n=1}^N \varphi_n(x)\varphi_n(x')$.
3. If using the Wiener criterion, approximate the correlation operator R.
4. Initialize to zero function : $f_0 = 0$.
5. For each new training data (x_{m+1}, y_{m+1}),
   (a) Use the reproducing kernel $K(x, x')$ to instantiate $\psi_{m+1} = K(x, x_{m+1})$.
   (b) Compute $\phi = R\psi_{m+1}$.
   (c) Compute projection $\phi_{mc} = P_{\mathcal{N}(A),\mathcal{R}(RA^*)}\phi$.
   (d) Compute $f_{mc}(x_{m+1})$ as
   $f_{mc}(x_{m+1}) = f(x_{m+1}) - f_m(x_{m+1}) = y_{m+1} - f_m(x_{m+1})$.



(a) Original function and noisy sampling

(b) Frequency cutoff = 3

(c) Frequency cutoff = 10

(d) Frequency cutoff = 40

Figure 8: Function approximation using the projection learning criterion and using original function spaces with frequency cut-off (b) $\sigma = 3$, (c) $\sigma = 10$ and (d) $\sigma = 40$

6. Compute the newly learned function f_{m+1} as $f_{m+1} = f_m + \frac{f_{mc}(x_{m+1})}{\phi_{mc}(x_{m+1})}\phi_{mc}$.

7. If no more training data to incorporate, STOP.
   Else, goto step 5.

6.2 Learning in Paley-Wiener spaces

First, we show an example of approximation using the Paley-Wiener space of band limited functions (refer Section 4.1). The original function to be approximated is a nearly linear function with a non-linear bump in the middle (See Fig.8(a)) given by

$$y = x + 2exp(-16x^2). \tag{73}$$

The function is sampled at 200 locations with an additive noise of the order of ten percent (Noise variance =0.1). The sampled locations are shown with dots in Fig.8. We show

results of approximation using the projection learning criterion. Fig.8 (b), (c) and (d) show approximations (solid line) when we consider a band limited function space (refer Section 4.1) with band frequency cut-off $\sigma = 3$, $\sigma = 10$ and $\sigma = 40$, respectively. As can be seen from the comparison with the original function, function space with $\sigma = 3$ has a large bias, hence, the high frequency components of the functions are smoothed out. On the other hand, $\sigma = 40$ results in a approximation which has very low bias but highly depends on the noisy data; i.e. has high variance. A suitable tradeoff between bias and variance results when we consider the function space with frequency cutoff $\sigma = 10$. Standard model selection techniques can be used to arrive at ideal values of the frequency cutoff for the search space.

6.3 Incremental learning in trigonometric spaces

Next, we consider an example in which we take the problem of approximating a trigonometric function f given by

$$f = -2sin(x) + 3cos(x) + sin(2x) + 4cos(2x) - 0.5sin(3x). \tag{74}$$

This function is represented by the solid line in Fig.9 (a) and (b). To show the effect of the model selection, we consider two Hilbert spaces H_1 and H_2, the former being the subspace of the latter. Let H_1 be a 6-dimensional function space spanned by the orthonormal functions

$$\{\varphi_i^{(1)}(x)\}_{i=1}^{6} \equiv \{sin(x), cos(x), sin(2x), cos(2x), sin(3x), cos(3x)\},$$

where the inner product in H_1 is defined as

$$(f, g) = \frac{1}{\pi}\int_{-\pi}^{\pi} f(x)g(x)dx \ ; \qquad \forall f, g \in H_1. \tag{75}$$

In general, for function spaces of the kind spanned by orthonormal functions $\{sin(nx), cos(nx)\}_{n=1}^{N}$, the reproducing kernel can be written as

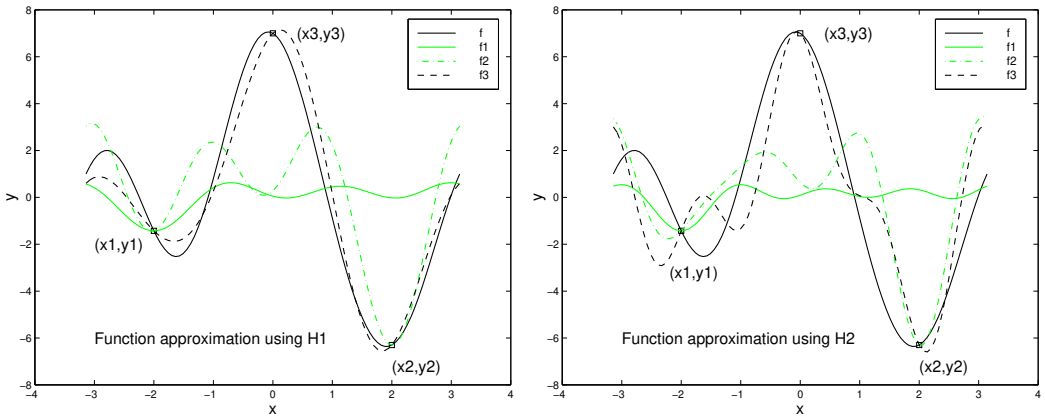
$$\begin{aligned} K(x, x') &= \sum_{n=1}^{N}(cos(nx)cos(nx') + sin(nx)sin(nx')) \\ &= \sum_{n=1}^{N} cos(n(x - x')) \\ &= \begin{cases} N & \text{if } x = x' \\ \frac{1}{2}[sin\frac{2N+1(x-x')}{2}/sin\frac{x-x'}{2} - 1] & \text{otherwise} \end{cases} \end{aligned} \tag{76}$$

which correspond to the Dirichlet kernel in Vapnik et al.[24]. Using the above result of eq.(76), the reproducing kernel for the Hilbert space H_1 can be written as

$$K^{(1)}(x, x') = \begin{cases} 3 & \text{if } x = x' \\ \frac{1}{2}[sin\frac{7(x-x')}{2}/sin\frac{x-x'}{2} - 1] & \text{otherwise} \end{cases} . \tag{77}$$

Similarly, consider H_2 to be a 8-dimensional function space spanned by the orthonormal functions

$$\{\varphi_n^{(2)}(x)\}_{n=1}^{8} \equiv \{sin(x), cos(x), sin(2x), cos(2x), sin(3x), cos(3x), sin(4x), cos(4x)\}$$



(a) (b)

Figure 9: Function approximation result for original function $f = -2sin(x) + 3cos(x) + sin(2x) + 4cos(2x) - 0.5sin(3x)$ with one (f_1), two (f_2) and three (f_3) sample points. (a) uses a 6-dimensional Hilbert space H_1 while (b) uses an 8-dimensional Hilbert space H_2

such that H_1 becomes a subspace of H_2. The reproducing kernel $K_m^{(2)}(x)$ for H_2, by similar computation, works out to be

$$K^{(2)}(x,x') = \begin{cases} 4 & \text{if } x = x' \\ \frac{1}{2}[sin\frac{9(x-x')}{2}/sin\frac{x-x'}{2} - 1] & \text{otherwise} \end{cases} . \tag{78}$$

Noiseless training data is obtained from the original function at locations $x_1 = -2$, $x_2 = 2$ and $x_3 = 0$ and is marked by tiny squares on the graphs. Here, for the sake of simplicity, we assume R, the correlation operator, to be an identity operator; in other words, all possible functions have an equal and independent probability of occurance.

Fig.9(a) shows the incremental learning results when learning with Hilbert space H_1. Here, f_1 refers to learned function using one sample point, f_2 with two sample points and so on. We can observe the improved generalization as the number of sample points increase. On the other hand, Fig.9(b) shows the approximation results for the same problem when we change the original search space to H_2. We observe, especially for f_3, that the generalization capability drops when we use a more general search space with more parameters (as in H_2) compared to one with a stronger and correct bias (as in H_1).

7 Discussion

A functional analytic framework based on Reproducing Kernel Hilbert Spaces (RKHS) for theorizing on the generalization ability of the trained network is provided and based on this, a scheme for efficiently incorporating new training data is presented. This framework is applied to compute the new learning operator X_{m+1} from the intermediate results of the previous stage like X_m, A_m, and the next sample point x_{m+1}. This has provided an efficient computation strategy to implement sequential learning with optimal generalization ability. An analytic relationship between the newly learned function and the previous function is de-

rived, hence, opening possibilities of modifying sampling strategy to optimize generalization. The framework offers the following advantages:

- The incremental learning is *exact* and not an approximation to the batch learning results.
- The results are extendable to any standard kind of optimization criteria including special cases of regularization.
- The Hilbert space framework provides a *systematic* tool for introduction of apriori knowledge and bias.
- Novel model selection strategies based on stratified search spaces can be easily implemented using this framework.

Another interesting observation is that the RKHS kernel based approximation derived here is closely related to the regularization networks and Support Vector Machine approximations. It is clear from the analysis that our final approximation has the kernel functions $K(x, x_i)$, one for each training data, as the basis functions[6]. The exact linear combinations of these basis functions depend upon the optimization criterion we are minimizing. It can be shown, as demonstrated in Appendix B, on the lines of Kimeldorf and Wahba[7] and Girosi[4], that the solution for regularization networks and SVM approximation is also a linear superposition of the kernel function $K(x, x_i)$ irrespective of the cost function used. Therefore, the results on incremental learning derived here can be expected to provide a useful hint in performing regression using SVM based methods in an online setting. Moreover, controlling the smoothness in regularization networks can be interpreted as changing the rate of decrease of the sequence $\{\lambda_n\}_{n=1}^{\infty}$ (refer Appendix B), which corresponds to using kernels with different frequency components[4], an automatic implementation of which was described in Section 4.1.

One of the drawbacks of this method - and in general, of most kernel based approximation methods - is that exact incremental learning in this setting is relatively computationally expensive, increasing in the order of the square of training set size. However, using cost functions which allow selection of subset of effective training data (e.g. support vectors) or active selection of non-redundant training data [27] can circumvent this problem to some extent.

As a follow up, results have already been derived for *incremental update in the weight space* for effective hardware implementation. Future research work will focus on extending these results for incremental learning in the presence of additive white noise. Avenues of using these results for the task of optimal training data selection using intermediate learning results, often referred to as active learning, is currently being explored.

8 Acknowledgments

This work was funded in parts by a fellowship grant to S.Vijayakumar provided by the Ministry of Education, Science and Culture, Japan (Mombusho) and the Mombusho Grant-in-Aid for Scientific Research (B) No. 08458076 to H.Ogawa. The author acknowledges the contribution of S.Nakazawa in extending the incremental framework to the memorization and projection learning criterion.

[6] This is evident from the fact that the learned function lies in the space $\mathcal{R}(A^*)$

A Exact Incremental Update Formula: rigorous proofs

A.1 Proof of eq.(60)

From eqs.(40) and (38), we have

$$\alpha = (R\psi_{m+1})(x_{m+1}) - (t_m, s_m). \tag{79}$$

Looking at the second term of the expansion in eq.(79) and using eqs.(37) and (39), we have

$$\begin{aligned}(t_m, s_m) &= (U_m^\dagger A_m R\psi_{m+1}, A_m R\psi_{m+1}) \\ &= (RA_m^* U_m^\dagger A_m R\psi_{m+1}, \psi_{m+1}).\end{aligned} \tag{80}$$

Using eq.(33) and Lemma 2, we can write eq.(80) as

$$\begin{aligned}(t_m, s_m) &= (X_m A_m R\psi_{m+1}, \psi_{m+1}) \\ &= (P_{S_m, S_{mc}} R\psi_{m+1}, \psi_{m+1}) \\ &= (P_{S_m, S_{mc}} R\psi_{m+1})(x_{m+1}).\end{aligned} \tag{81}$$

Substituting eq.(81) in eq.(79) and using the definition of ϕ,ϕ_m and ϕ_{mc}, we have

$$\begin{aligned}\alpha &= (R\psi_{m+1})(x_{m+1}) - (P_{S_m, S_{mc}} R\psi_{m+1})(x_{m+1}) \\ &= \phi(x_{m+1}) - \phi_m(x_{m+1}) \\ &= \phi_{mc}(x_{m+1}).\end{aligned} \tag{82}$$

A.2 Proof of Theorem 3

From eqs.(44) and (39),

$$\begin{aligned}v_m &= \frac{1}{\alpha}(R\psi_{m+1} - RA_m^* t_m) \\ &= \frac{1}{\alpha}(R\psi_{m+1} - RA_m^* U_m^\dagger s_m).\end{aligned} \tag{83}$$

Applying eqs.(33) and (37) to eq.(83), we have

$$\begin{aligned}v_m &= \frac{1}{\alpha}(R\psi_{m+1} - X_m s_m) \\ &= \frac{1}{\alpha}(R\psi_{m+1} - X_m A_m R\psi_{m+1}) \\ &= \frac{1}{\alpha}(I - X_m A_m) R\psi_{m+1}.\end{aligned} \tag{84}$$

Using the result of Lemma 2, we write eq.(84) as

$$\begin{aligned}v_m &= \frac{1}{\alpha}(I - P_{S_m, S_{mc}}) R\psi_{m+1} \\ &= \frac{1}{\alpha} P_{S_{mc}, S_m} R\psi_{m+1} = \frac{1}{\alpha} P_{S_{mc}, S_m}\phi.\end{aligned} \tag{85}$$

By definition of ϕ_{mc} and α(eq.(60)), we can reduce eq.(85) to

$$v_m = \frac{\phi_{mc}}{\phi_{mc}(x_{m+1})}. \tag{86}$$

On the other hand, on substituting the value of Y_m from eq.(45) into eq.(54), we have

$$\begin{aligned} f_{m+1} &= f_m - (v_m \otimes \overline{t_m})y^{(m)} + f(x_{m+1})v_m \\ &= f_m - (y^{(m)}, t_m)v_m + f(x_{m+1})v_m \\ &= f_m + [f(x_{m+1}) - (y^{(m)}, t_m)]v_m. \end{aligned} \tag{87}$$

Now, let us look at the term $(y^{(m)}, t_m)$ in the above equation. It can be written as

$$(y^{(m)}, t_m) = (A_m f, t_m) = (f, A_m^* t_m). \tag{88}$$

Assume $f \in \mathcal{R}(R)$, which holds true for Wiener learning. Then, $f = Ru$ for some u. Hence, eq.(88) can be written as

$$(y^{(m)}, t_m) = (Ru, A_m^* t_m) = (u, RA_m^* t_m). \tag{89}$$

Expanding for t_m using eqs.(39) and (37) in eq.(89),

$$\begin{aligned} (y^{(m)}, t_m) &= (u, RA_m^* U_m^\dagger A_m R\psi_{m+1}) \\ &= (u, X_m A_m R\psi_{m+1}). \end{aligned} \tag{90}$$

Using Lemma 2, eq.(90) can be written as

$$\begin{aligned} (y^{(m)}, t_m) &= (u, PR\psi_{m+1}) \\ &= (Pu, R\psi_{m+1}) \\ &= (RPu, \psi_{m+1}), \end{aligned} \tag{91}$$

where P is the projection on S_m along S_{mc}. It can be easily shown that $PR = RP$. Using this result in eq.(91), we have

$$\begin{aligned} (y^{(m)}, t_m) &= (PRu, \psi_{m+1}) \\ &= (Pf, \psi_{m+1}) \\ &= (f_m, \psi_{m+1}). \end{aligned} \tag{92}$$

Using the property of the function ψ_{m+1}, eq.(92) becomes

$$(y^{(m)}, t_m) = f_m(x_{m+1}). \tag{93}$$

Now, using eqs.(86) and (93) in eq.(87), we have

$$\begin{aligned} f_{m+1} &= f_m + [f(x_{m+1}) - f_m(x_{m+1})]\frac{\phi_{mc}}{\phi_{mc}(x_{m+1})} \\ &= f_m + \frac{f_{mc}(x_{m+1})}{\phi_{mc}(x_{m+1})}\phi_{mc}. \end{aligned} \tag{94}$$

B Regularization, SVMs and kernel approximation

Let us consider a Hilbert Space H which contains functions that can be represented as

$$f(x) = \sum_{n=1}^{\infty} c_n \varphi_n(x) \tag{95}$$

where $\{\varphi_n(x)\}_{n=1}^{\infty}$ is a set of given linearly independent basis functions and c_n are the scalar parameters to be estimated. Let the inner product in H be defined as

$$(f, g)_H = (\sum_{n=1}^{\infty} c_n \varphi_n(x), \sum_{n=1}^{\infty} d_n \varphi_n(x))_H = \sum_{n=1}^{\infty} \frac{c_n d_n}{\lambda_n}, \tag{96}$$

where $\{\lambda_n\}_{n=1}^{\infty}$ is a positive, decreasing sequence. The reproducing kernel K of this RKHS can then be written as

$$K(x, x') = \sum_{n=1}^{\infty} \lambda_n \varphi_n(x) \varphi_n(x'). \tag{97}$$

Now, let us look at the solution of the variational problem for regularization given by the functional:

$$\min_{f \in H} J[f] = C \sum_{i=1}^{m} V(y_i - f(x_i)) + \frac{1}{2} \Phi[f] \tag{98}$$

where V is some cost function, C is a positive number controlling the tradeoff between the two terms and $\Phi[f]$ is the regularizing term, which we take here as

$$\Phi[f] = \|f\|_H^2 = \sum_{n=1}^{\infty} \frac{c_n^2}{\lambda_n}. \tag{99}$$

We can think of functional $J[f]$ as a function of coefficients c_n. In order to minimize $J[f]$, we take the it's derivative with respect to c_n and equate to zero obtaining the following:

$$-C \sum_{i=1}^{m} V'(y_i - f(x_i)) \varphi_n(x_i) + \frac{c_n}{\lambda_n} = 0 \tag{100}$$

If we define unknowns $a_i \equiv CV'(y_i - f(x_i))$, then, using eq.(100), we can write c_n as:

$$c_n = \lambda_n \sum_{i=1}^{m} a_i \varphi_n(x_i). \tag{101}$$

From eqs.(95), (101) and (97), we know that the solution of the variational problem has the form :

$$f(x) = \sum_{n=1}^{\infty} c_n \varphi_n(x) = \sum_{n=1}^{\infty} \sum_{i=1}^{m} a_i \lambda_n \varphi_n(x_i) \varphi_n(x) = \sum_{i=1}^{m} a_i K(x, x_i). \tag{102}$$

This shows that irrespective of the cost function V used, the solution of the regularization functional of eq.(98) is always a superposition of kernel functions, one for each data point. The cost function affects the computation of the coefficients a_i. When the cost function V is of the form

$$V(x) = |x|_\epsilon \equiv \begin{cases} 0 & \text{if } |x| < \epsilon \\ |x| - \epsilon & \text{otherwise} \end{cases} \tag{103}$$

then, the solution of the functional yields the Support Vector Machine approximation scheme.

References

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

[2] N. Aronszajn. Theory of reproducing kernels. *Trans. Amer. Math. Soc.*, 68:337–404, 1950.

[3] N. Dunkin, J. Shawe-Taylor, and P. Koiran. A new incremental learning technique. In *8th Italian Workshop on Neural Nets, WIRN Vietri-96*, pages 112–118. Springer-Verlag, 1997.

[4] F. Girosi. An equivalence between sparse approximation and support vector machines. *Neural Computation*, 10(6):155–1480, 1998.

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

[6] V. Kadirkamanathan and M. Niranjan. A function estimation approach sequential learning with neural network. *Neural Computation*, 5(6):954–975, 1993.

[7] G.S. Kimeldorf and G. Wahba. A correspondence between bayesian estimation on stochastic processes and smoothing by splines. *Ann. Math. Statist.*, 2:495–502, 1971.

[8] S.P. Luttrell. The use of transinformation in the design of data sampling schemes for inverse problems. *Inverse Problems*, 1(1):199–218, 1985.

[9] D. Mackay. Information-based objective functions for active data selection. *Neural Computation*, 4(4):590–604, 1992.

[10] V.A. Morozov. *Methods for Solving Incorrectly Posed Problems*. Springer-Verlag, 1984.

[11] S. Nakazawa. A study on incremental learning in optimally generalizing neural networks. Master's thesis, Tokyo Institute of Technology, 1997.

[12] H. Ogawa. A unified approach to generalised sampling theorems. In *Proceedings, IEEE-IEICE-ASJ International Joint Conference on ASSP*, pages 1657–1660. IEEE-IEICE-ASJ, 1986.

[13] H. Ogawa. Neural network theory as an inverse problem. *Journal of IEICE, Japan*, 73(7):690–695, 1990. In Japanese.

[14] H. Ogawa. Neural network learning, generalization and over learning. In *Proceedings, ICIIPS'92 (Beijing)*, volume 2, pages 1–6, 1992.

[15] H. Ogawa and N. Nakamura. Projection filter restoration of degraded images. In *Proceedings, IEEE International Conference on Pattern Recognition (Montreal)*, pages 601–603. IEEE, 1984.

[16] H. Ogawa and E. Oja. Projection filter, Wiener filter and Karhunen-Loève subspaces in digital image restoration. *Journal of Mathematical Analysis and Applications*, 114(1):37–51, 1986.

[17] E. Parzen. Regression analysis of continuous parameter time series. In *Proceedings, Fourth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 469–489, 1961.

[18] T. Poggio and F. Girosi. Networks for approximation and learning. *Proceedings of the IEEE*, 78(9):1481–1497, 1990.

[19] S. Saitoh. *Integral Transforms, Reproducing Kernels and their Applications*. Pitman Research Notes in Mathematics No. 369. Longman Publishers, 1997.

[20] D. Schuurmans. A new metric based approach to model selection. In *Proceedings, Fourteenth National Conference on Artificial Intelligence*. AAAI, 1997. In press.

[21] A.N. Tikhonov and V.Y. Arsenin. *Solution of Ill-Posed Problems*. Winston, Washinton,D.C., 1977.

[22] M. Unser, A. Aldroubi, and M. Eden. Fast B-spline transforms for continuous image representation and interpolation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(3):277–285, 1991.

[23] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, New York, 1995.

[24] V. Vapnik, S.E. Golowich, and A. Smola. Support vector method for function approximation, regression estimation and signal processing. In Michael C. Mozer, Michael I. Jordan, and Thomas Petsche, editors, *Advances in Neural Information Processing Systems*, volume 9, pages 281–287. The MIT Press, 1997.

[25] S. Vijayakumar. An incremental approach to training data selection in neural networks. Master's thesis, Tokyo Institute of Technology, 1995.

[26] S. Vijayakumar and H. Ogawa. A functional analytic approach to incremental learning in optimally generalizing neural networks. In *Proceedings, IEEE International Conference on Neural Networks ICNN'95 (Perth)*, pages 777–782, Perth, Australia, 1995. IEEE.

[27] S. Vijayakumar, M. Sugiyama, and H. Ogawa. Training data selection for optimal generalization and noise variance reduction in neural networks. In *Proceedings, 10th Italian Workshop on Neural Nets WIRN'98*, Salerno, Italy, 1988. (In Press).

[28] G. Wahba. *Spline Models for Observational Data*, volume 59 of *Series in Applied Mathematics*. SIAM, Philadelphia, 1990.

[29] C.K.I. Williams. Prediction with Gaussian processes: From linear regression to linear prediction and beyond. In M.I.Jordan, editor, *Learning and Inference in Graphical Models*. Kluwer Academic Press, 1998.

[30] Y. Yamashita and H. Ogawa. Mutual relations among optimum image restoration filters. *Transactions of IEICE Japan*, J75-D-II(5):890–898, 1992. In Japanese.

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

[32] B.T. Zhang. An incremental algorithm that optimizes network size and sample size in one trial. In *Proceedings of the International Conference on Neural Networks ICNN'94*, pages 215–220, Orlando, Florida, 1994. IEEE.

Profiles

Sethu Vijayakumar was born in 1970 in Kerala, India. He received the B.E.(Comp.Sc.) degree from the Regional Engineering College, Tiruchirapalli, India in 1992 and the M.E. and Ph.D degrees from the Tokyo Institute of Technology, Japan in 1995 and 1998, respectively. He is currently a researcher with the Information Synthesis Laboratory of the RIKEN Brain Science Institute, Japan and holds a part time affiliation with the CLMC Lab, USC, California. His research interests includes statistical and machine learning, neural networks and computational neuroscience. He received the ICNN'95 Best Student Paper Award in 1995, the IEEE Vincent Bendix Award in 1991 and the IEEE R.K.Wilson RAB Award in 1996. Dr.Vijayakumar is also a member of the International Neural Network Society, and the IEEE.

Hidemitsu Ogawa was born in 1942 in Hiroshima Prefecture, Japan. He received the B.E. and D.E. degrees from the Tokyo Institute of Technology (TIT), Japan, in 1965 and 1977, respectively. From 1965 to 1972 he was with the Electrotechnical Laboratory. In 1972 he joined the faculty of the TIT, where he is now a professor of the Department of Computer Science, Graduate School of Information Science and Engineering. During the academic year 1984-1985 he was a visiting professor of the Department of Technical Physics at the Helsinki University of Technology, Finland. His research interests include pattern recognition, neural networks, and image processing. He received the Yonezawa Memorial Award in 1969 and Paper Awards in 1976, 1985, 1990, 1993, and 1994, respectively, from the Institute of Electronics, Information and Communication Engineers of Japan. Dr. Ogawa is also a member of the Information Processing Society of Japan, the Japanese Society for Artificial Intelligence, the American Mathematical Society, the International Neural Network Society, and the IEEE.