Approximation Properties of Local Bases Assembled From Neural Network Transfer Functions

Andrew J. Meade, Jr.[1] and Boris A. Zeldin
Department of Mechanical Engineering and Materials Science
Mail Stop 321
William Marsh Rice University
Houston, Texas 77251-1892 , USA
Phone: (713) 527-8101 ext. 3590
E-mail: meade@rice.edu

Mathematical and Computer Modelling

---

1. Author to whom correspondence should be addressed. E-mail: meade@rice.edu
This work was supported under NASA grant NAG 9-719 and Office of Naval Research grant N00014-95-1-0741.

Approximation Properties of Local Bases Assembled From Neural Network Transfer Functions

Andrew J. Meade, Jr. and Boris A. Zeldin

William Marsh Rice University, Houston, Texas 77251-1892

Abstract

The adaptive data-driven emulation and control of mechanical systems are popular applications of artificial neural networks in engineering. However, multi-layer perceptron training is an ill-posed nonlinear optimization problem. This paper explores a method to constrain network parameters so that conventional computational techniques for function approximation can be used during training. This was accomplished by forming local basis functions which provide accurate approximation and stable evaluation of the network parameters. It is noted that this approach is quite general and does not violate the principles of network architecture. By employing the concept of shift invariant subspaces, this approach yields a new and more robust error condition for feedforward artificial neural networks and allows one to both characterize and control the accuracy of the local bases formed. The two methods used are: 1) adding bases while altering their shape and keeping their spacing constant and 2) adding bases while altering their shape and decreasing their spacing in a coupled fashion. Numerical examples demonstrate the usefulness of the proposed approximation of functions and their derivatives.

Nomenclature

A, B, C, a, r	constants
$\bar{b} = (b_1, \dots b_k)$	set of network parameters
C_Ω	hypercube $[-\Omega, \Omega]^d$
$C(\varphi)$	subset of C_Ω
c_k	parameters of a linear approximation
F	a normed space
$f(x)$	target function
$f_a(x, \bar{b})$	network approximation of the target function
$f_N(x)$	polynomial approximation of the target function
$f_\Omega(x)$	bandlimited function associated with $f(x)$
$\hat{f}$	Fourier transform of f
$\bar{\hat{f}}$	complex conjugate of $\hat{f}$
$H^\alpha(\boldsymbol{R}^d)$	Sobolev space
h	grid spacing
$L^2(\boldsymbol{R}^d)$	Banach space
$\boldsymbol{N}^d$	d-dimensional space of integers
$\boldsymbol{R}^d$	d-dimensional Euclidean space
S_i	non-linear transfer function of the i-th intermediate layer
$V^{\beta, \Delta}(\varphi)$	shift invariant subspace of $L^2(\boldsymbol{R}^d)$
$x = (x_1, \dots x_d, 1)^T$	$(d+1)$-dimensional input of the FFANN system
α, γ, υ	constants
$\beta, \theta, \mu, \kappa$	transfer function parameters
Δ	distance between the bell-shaped function centers

Nomenclature

ε	error criterion
η	dummy scalar variable
$\lambda(\xi)$	2π periodic function of all components of the argument ξ
ξ	dummy variable $\xi = (\xi_1, \ldots \xi_d)^T$
τ_i	$\tau_i = max(S(\beta x_i + \theta) - S(\beta x_i - \theta))$
$\varphi_j(x)$	basis function
Ω	domain of interest

Approximation Properties of Local Bases Assembled From Neural Network Transfer Functions

Andrew J. Meade, Jr.[1] and Boris A. Zeldin[2]
William Marsh Rice University, Houston, Texas 77251-1892

1. Introduction

At present, an area of increasing interest is the adaptive data-driven emulation and control of engineering systems by the multi-layer perceptron, also known as the feed-forward neural network (FFANN). It is known that neural networks can approximate functions and mathematical operators arbitrarily well as the number of neurons in the network tends to infinity [1-5]. In this regard, FFANNs can be considered to be "universal approximators" capable of describing the input-output relationships of mechanical systems. For engineers, the universal approximation capability of ANNs would be especially useful in the emulation of complex mechanical systems.

The supervised training of a FFANN system can be formulated as the procedure of selecting the network parameters so that the response of the neural network is "close" to the available data. This can be formalized by introducing a measure of the distance between the exact function, represented by available data, and its FFANN approximation with respect to a normed space F of functions

$$\varepsilon = \left\| f(x) - f_a(x, \bar{b}) \right\|_F = \left\| f(x) - \sum_i b_{k,i} S_{(k-1)}\left(\sum_j b_{(k-1),j} S_{(k-2)}\left(\ldots S_1\left(\sum_l b_{1,l} x_l \right)\right)\right)\right\|_F, \quad (1)$$

This paper considers only single-output FFANN systems. An extension to the multi-output case is straightforward.

1. Associate Professor, Department of Mechanical Engineering and Materials Science.
Author to whom correspondence should be addressed. E-mail: meade@rice.edu.
2. Research Associate, Department of Mechanical Engineering and Materials Science.

The network response $f_a(x, \bar{b})$ depends linearly on only a few parameters, which are shown in Eq. (1) as the coefficients $\{b_{k,i}\}$ of the expansion. The remaining parameters affect the network response in a non-linear manner. Correspondingly, supervised training of neural networks is based on non-linear optimization procedures. The backpropagation training algorithm [6] clearly resembles the steepest descent method for minimization of a multi-variate function. Other training algorithms can be associated with the conjugate gradient method, Levenberg-Marquardt method, and a variety of other non-linear optimization procedures. The application of the previously mentioned numerical algorithms as network training procedures were studied by Saarinen et al. [7]. It was found that even the most advanced optimization algorithms were susceptible to poor convergence in training.

The work of Saarinen et al. [7] demonstrated that the possibility of failure in training is an inherent feature of networks and is related to the internal architecture of the multi-layered FFANN, which yields ill-conditioned Jacobian and Hessian matrices. As a result of the highly parallel and redundant structure, a set of trained neural networks with differing parameter values are able to produce very similar outputs. The selection of optimal parameters through training is clearly an ill-posed problem.

In an effort to address the ill-posed feature of the supervised training of FFANN, Poggio and Girosi [8, 9] adopted the Tikhonov regularization method [10] by incorporating regularization functionals. This approach was further extended to diverse practical applications [11-14]. Unfortunately, regularization does not address all of the issues related to FFANN training. It merely provides requisite constraints so that an approximation exists for the considered class of neural networks. An alternate and straightforward remedy for network training can be sought through linearization in which all non-linear parameters are selected uniformly or by using a random sampling procedure [15], and the linear parameters are found to minimize the error criterion [16,17]. The response of this type of neural network can be seen as a linear combination of transfer functions. Specifically,

$$f_a(x, \bar{b}) = \sum c_j \varphi_j(x). \quad (2)$$

where $\{\varphi_j(x)\}$ are bases formed from linear combinations of the transfer functions. This approach possesses a number of theoretical advantages: 1) The "optimal" solution of Eq. (1) can be readily determined by solving a linear system of algebraic equations. Complicated non-linear optimization procedures associated with training by backpropagation are eliminated. 2) The theory of regularization of linear ill-posed problems can be readily exploited for neural network applications [18]. 3) The solution of the corresponding least-squares minimization problem, associated with evaluation of the linear parameters of Eq. (2), is unique and stable with respect to the error of measurements. 4) The cross-validation principles [19, 20] can be used to predict missing data and eliminate uncorrelated random noise of measurement as the number of observations grows sufficiently large.

From a practical standpoint, the most attractive advantage in constraining non-linear parameters is that the user has control in forming the bases. For engineering applications, the bases: 1) should be assembled in a straightforward manner from the FFANN architecture with parameters that can be evaluated uniquely and in a numerically stable manner, 2) should accurately approximate the desired function, 3) should allow for utilizing a variety of computational techniques familiar to the engineering community such as finite difference, finite elements and spectral methods, for evaluation of the linear coefficients, 4) should allow the user to control the accuracy of the network approximation and 5) should have the property of approximating desired functions using a small number of hidden layer neurons and of requiring only a modest increase in the number of neurons to improve modelling accuracy (i.e., good rate of convergence).

It is proposed in this paper that a local basis be formed from linear combinations of transfer functions and distributed in the input variable space. The linear coefficients associated with the basis (output weights) can then be evaluated by a linear optimization technique. This approach can be viewed by the connectionist community as an efficient method to estimate the initial values of network parameters for subsequent training. Sigmoid generated local bases have been investigated in the literature [23], but only to the extent of establishing the density of neural network response functions.

The technical approach presented in this paper appeals explicitly to the theory of shift-invariant space approximation [24]. Shift-invariant spaces are important for a large class of engineering applications and are the key elements for such computational methods as spline approximation, finite element analysis, and wavelet analysis. In this paper the orthogonal projection, in conjunction with shift-invariant subspace approximation [25], will be utilized to obtain the approximation from a set of local bell-shaped basis functions.

As mentioned previously, the formation of local bases should not exclude the use of regularization. Indeed, the use of any available information regarding the mechanical system of interest, such as the corresponding governing equations in the form of integral-differential equations, is highly advantageous. However, the modelling of governing equations may introduce additional restrictions in the determination of $\bar{b}$. In particular, it may be required for the network to accurately approximate not only a function, but also some of its derivatives. In this regard, Cardaliaguet and Euvrard [23] showed that local bases can be used to interpolate a function by using the function values as the linear coefficients. A new set of network parameters was then introduced to approximate the first derivative of the function. For our future efforts it is necessary that the developed bases accurately interpolate both known functions and some of their derivatives using only a single parameter set $\bar{b}$.

It is the objective of this paper to develop bases that satisfy all of the previously mentioned properties by establishing appropriate constraints on the non-linear network parameters. It is the opinion of the authors that progress in this area may aid connectionist efforts and allow FFANNs to be reliably applied to mechanics problems. In Section 2, establishment of appropriate constraints will begin by constructing bell-shaped functions from the interconnections of FFANN neurons. Section 3 will review requisite definitions of functional spaces and bases. Section 4 will deal with the approximation properties of the bell-shaped functions and will explore practical methods that can be used to reduce the error of approximation. Section 5 will discuss the effect of established constraints on the error estimate as the number of neurons is varied and will compare the accuracy of the proposed network approximation with the corresponding estimates available in the literature. This section will also complete the establishment of appropriate constraints on the non-linear parameters. Numerical examples will be included in Section 6 to illustrate the utility of the parameter constraints on network approximation. It will be shown that the bell-shaped functions can accurately approximate functions and their derivatives using only a single parameter set $\bar{b}$. Though the numerical examples will be restricted to the approximation of one-dimensional functions by hyperbolic tangents for the sake of brevity, the presented material is kept at a sufficient level of applicability for general FFANNs. Finally, Section 7 will summarize the findings of this paper. To avoid additional mathematical burdens in our discussion, the proofs of formulated propositions and equations are included in the appendix.

2. Feed-Forward Neural Networks and Bell-Shaped Functions

Modern feed-forward neural networks consist of at least three layers of processing elements. The internal layers of FFANN systems are formed by neurons, each using a non-linear transfer function $S(\eta)$. However, in general $S(\eta)$ is often not amenable to certain analytical investigations. In particular, $S(\eta)$ may not be an integrable function. To address this difficulty, several authors have introduced a procedure for generating local functions from non-local transfer functions [3, 23]. Hornik [3] introduced the concept of l-finite functions. A function $S(\eta)$ is called l-finite if $S(\eta)$ is l times continuously differentiable and its l-th derivative is an absolutely integrable function. That is, for $d = 1$

$$0 < \int_{-\infty}^{\infty} \left| \frac{d^l}{dx^l} S(x) \right| dx < \infty . \tag{3}$$

Then, it can be shown that the properties of the network with the transfer function $\frac{d^l}{dx^l}S(x)$ are those of the network with the local bell-shaped transfer function given by the following equation

$$\tilde{S}(x) = D^l S(x) , \tag{4}$$

where D is the finite-difference-like approximation of the differential operator, $D^1 S(x) = S(x+1) – S(x-1)$. Note that the function $\tilde{S}(x)$ can be readily realized in a network as a linear combination of several neurons.

Often in network applications, the function $S(x)$ is taken as a continuous sigmoidal function which is one-finite in the sense of Hornik [3]. Utilizing the results of the *l*-finite function concept, a single bell-shaped function can be constructed from two sigmoidal one-finite functions using a modification of Eq. (4) with $l = 1$,

$$\varphi(\beta x) \equiv \varphi(b_1^T x) = \frac{S(\beta x + \theta) - S(\beta x - \theta)}{\tau_1} , \tag{5}$$

where $b_1 = (\beta, \theta)^T$ for $d = 1$. The parameters β and θ act as input and bias weights, respectively, and are introduced to control the shape of the sigmoidal function and its location in the input space.

To further illustrate the preceding developments, the hat function shown in Figure 1 can be formed from the sigmoidal function

$$S(x) = \begin{cases} 1, & x \geq 1 \\ x, & x \in [-1, 1] \\ -1, & x \leq 1 \end{cases} \tag{6}$$

using Eq. (5) with $\beta = \theta = 1$. Similarly, an infinitely smooth bell-shaped function can be obtained from the sigmoidal hyperbolic tangent function

$$S(x) = \tanh(x) \tag{7}$$

also using Eq. (5). Figure 2 shows such a function for the case $\beta = \theta = 1$.

The authors have developed two methods for constructing multi-dimensional local bases. In the first method, a local function in the multi-dimensional case can be formed from a neural network with two hidden-layers of neurons. Specifically, with $\psi = (\psi_1, \ldots, \psi_d, 1)^T$

$$\varphi(b_2^T\psi) = \sigma(\rho\psi_1 + \ldots + \rho\psi_d - \kappa), \qquad d-1 \leq \kappa < d \quad , \tag{8}$$

for a general value of d where

$$\psi_i = \frac{S(\beta x_i + \theta) - S(\beta x_i - \theta)}{\tau_i}, \tag{9}$$

$b_2 = (\rho, \ldots, \rho, -\kappa)^T$, $S(\eta)$ is one-finite sigmoidal function, and $\sigma(\eta)$ is equal to zero for $\eta \leq 0$. Figure 3 shows a two-dimensional example for $\rho = \kappa = \beta = \theta = 1$, where $S(\eta)$ is given by Eq. (7) and

$$\sigma(\eta) = \begin{cases} 0, & \eta \leq 0 \\ \exp(-1/\eta), & \eta > 0 \end{cases}. \tag{10}$$

In the second method the well known method of polynomial approximation, commonly used in data analysis and computational mechanics applications, can be mapped into a FFANN architecture. Specifically,

$$S_{\Im}(b_1^T x) = \frac{\partial^{|\Im|}}{\partial b_{1,1}^{p_1} \dots \partial b_{1,d}^{p_d}} S(b_1^T x) = x^{\Im} S^{(|\Im|)}(b_1^T x), \tag{11}$$

for a general value of d where $x^{\Im} = (x_1^{p_1}, \dots, x_d^{p_d})^T$, $\Im$ is a multi-index such that $|\Im| = p_1 + \dots + p_d$ and superscript ( ) represents the order of the ordinary derivative of S. Equation (11) shows that the polynomial $x^{\Im}$ can be readily expressed as

$$x^{\Im} = \frac{S_{\Im}(0, x)}{S^{(|\Im|)}(\theta)} \text{ where } S_{\Im}(0, x) = S^{(|\Im|)}(b_1^T x)\Big|_{b_1 = 0}. \tag{12}$$

By replacing the partial derivative $S_{\Im}(0, x)$ by an adequate finite difference approximation, one can accurately approximate the polynomial $x^{\Im}$ by a finite linear combination of transfer functions. For our purpose we can approximate a tensor product of basis functions $\psi(b_1, x_i)$ as

$$\varphi(\beta x_1)\varphi(\beta x_2) =$$

$$\frac{[S(w_1, w_2) + S(-w_1, -w_2) - S(-w_1, w_2) - S(w_1, -w_2)]}{4 w_1 w_2 \tau_1 \tau_2 S^{(2)}(\mu)} + O(w_1 w_2) \quad , \tag{13}$$

where $S(w_1, w_2) \equiv S(\delta b_2^T \psi) = S(w_1 \psi_1 + w_2 \psi_2 + \mu)$, $\delta b_2 = (w_1, \dots, w_d, \mu)^T$ is a perturbation of the b_2 vector, and μ is selected to maximize $S^{(2)}(\mu)$. An example is shown in

Figure 4 for $\beta = \theta = 1$, $w_1 = w_2 = 1 \times 10^{-3}$ and $\mu = 0.6585$ using the basis of Figure 2. Note that this product approximation technique requires a greater than one-finite transfer function. Consequently, this technique will not work with piecewise linear basis functions alone, but can work with a FFANN using smooth functions like Eq. (7). Note also that the functions of Figures 3 and 4 are infinitely smooth and decay exponentially to zero at infinity. Elements of the second construction method has been previously utilized in references [21] and [22] for establishing the approximation properties of some types of neural network architectures.

The following section shows that by the appropriate selection of network parameters, the neural network approximation can be associated with the approximation by a shift-invariant subspace of bell-shaped functions. Cardaliaguet and Euvrard [23] used this approach to establish the density of neural network response functions. We extend this approach and will show that the bell-shaped functions associated with neural networks can be used to determine the accuracy of network approximation. The properties of corresponding local basis functions are addressed in subsequent sections.

3. Properties of Shift-Invariant Subspaces

To study the approximation properties of neural networks, one must introduce the notion of distance between two functions. Since this research is related to mechanics problems, it is appropriate to use the Sobolev space $H^{\alpha}(\boldsymbol{R}^d)$, $\alpha \ge 0$, [26]. In this paper, the Fourier transform of an integrable function $f(x)$ is defined as

$$\hat{f}(\xi) = \int_{\boldsymbol{R}^d} f(x) e^{-i\langle x, \xi\rangle} dx \tag{14}$$

$$f(x) = \frac{1}{(2\pi)^{d/2}} \int_{\boldsymbol{R}^d} \hat{f}(\xi) e^{i\langle x, \xi\rangle} d\xi \tag{15}$$

where $\langle x, \xi\rangle = x_1\xi_1 + \ldots + x_d\xi_d$.

Any Hilbert space H has a basis. That is, any element $f(x)$ of the Hilbert space can be uniquely represented as a finite linear combination of the basis functions $\{\varphi_j(x), j \in \boldsymbol{N}\}$,

$$f(x) = \sum c_j \varphi_j(x). \tag{16}$$

However, not every basis is a useful vehicle of computational analysis. The basis $\{\varphi_j(x), j \in \boldsymbol{N}\}$ must also provide an efficient and stable mechanism for representing and reconstructing elements of H. This requirement can be formalized by the following inequalities

$$A\|f\|^2 \leq \sum_{j=1}^{\infty} |c_j|^2 \leq B\|f\|^2, \tag{17}$$

where the constants A and B are the Riesz bounds. A basis satisfying Eq. (17) is known as a Riesz, or unconditional, basis.

Of particular importance for the analysis approach of this paper is the approximation ability of bases constructed by shifting a single bell-shaped function $\varphi(\beta x)$. Let $V^{\beta,\Delta}(\varphi)$ denote the space defined as the $L^2(\boldsymbol{R}^d)$-closure of finite linear combinations of $\{\varphi_k(\beta x) = \varphi(\beta x - k\Delta), k \in \boldsymbol{N}^d\}$. Here, the parameter Δ specifies the distance between the centers of the bell-shaped functions. Note that the space $V^{\beta,\Delta}(\varphi)$ is a shift invariant subspace of $L^2(\boldsymbol{R}^d)$ since if $f(x) \in V^{\beta,\Delta}(\varphi)$ then $f(x - k\Delta) \in V^{\beta,\Delta}(\varphi)$ for all $k \in \boldsymbol{N}^d$. In this regard, the set of functions $\{\varphi_k(\beta x),\ k \in \boldsymbol{N}^d\}$ can be associated with an FFANN system with the l-finite transfer function. As a result, the approximation properties of FFANNs can be assessed by studying how the shift-invariant subspace $V^{\beta,\Delta}(\varphi)$ generated by $\varphi(\beta x)$ approximates the spaces $L^2(\boldsymbol{R}^d)$ and $H^{\alpha}(\boldsymbol{R}^d)$. Also, the conditions that allow the set $\{\varphi_k(\beta x), k \in \boldsymbol{N}^d\}$ to form Riesz bases for $V^{\beta,\Delta}(\varphi)$ can be determined to assure the numerical stability of the network during training.

Note that if $f_a(x, \bar{b})$ is a network approximation of the function $f(x)$ from the subspace $V^{\beta, \Delta}(\varphi) \subset L^2(\boldsymbol{R}^d)$, then the Fourier transform of the integrable function $f_a(x, \bar{b})$ is

$$\hat{f}_a(\xi, \bar{b}) = \int_{\boldsymbol{R}^d} \left(\sum c_k \varphi(\beta x - k\Delta)\right) e^{-i\langle x, \xi\rangle} dx =$$

$$\sum c_k e^{-i\frac{\Delta}{\beta}\langle k, \xi\rangle} \frac{1}{\beta^d} \hat{\varphi}\left(\frac{\xi}{\beta}\right) = \frac{1}{\beta^d} \lambda\left(\frac{\Delta}{\beta}\xi\right) \hat{\varphi}\left(\frac{\xi}{\beta}\right) \qquad , \tag{18}$$

where

$$\lambda(\xi) = \sum c_k e^{-i\langle k, \xi\rangle} . \tag{19}$$

In fact, it can be shown [25] that $f_a(x, \bar{b}) \in V^{\beta, \Delta}(\varphi)$ if and only if $\hat{f}_a(\xi, \bar{b}) = \frac{1}{\beta^d}\lambda\left(\frac{\Delta}{\beta}\xi\right)\hat{\varphi}\left(\frac{\xi}{\beta}\right)$ for some 2π periodic function $\lambda(\xi)$ and $\lambda(\xi)\hat{\varphi}(\xi) \in L^2(\boldsymbol{R}^d)$. This fact characterizes all functions from the shift-invariant subspace $V^{\beta, \Delta}(\varphi)$ in terms of the generating function $\varphi(\beta x)$.

Next, the stability of this characterization can be determined by evaluating the corresponding Riesz bounds:

$$A = \left(\left(\frac{2\pi}{\beta\Delta}\right)^d sup\left(\sum_l \left|\hat{\varphi}\left(\frac{\xi}{\beta} + 2\pi\frac{l}{\Delta}\right)\right|^2\right)\right)^{-1} , \tag{20}$$

and

$$B = \left(\left(\frac{2\pi}{\beta\Delta}\right)^d inf\left(\sum_l \left|\hat{\varphi}\left(\frac{\xi}{\beta} + 2\pi\frac{l}{\Delta}\right)\right|^2\right)\right)^{-1} . \tag{21}$$

Clearly, the basis $\{\varphi_k(\beta x), k \in \boldsymbol{N}^d\}$ is a Riesz basis for $V^{\beta,\Delta}(\varphi)$, if and only if,

$$0 < \sum_l \left|\hat{\varphi}\left(\frac{\xi}{\beta} + 2\pi\frac{l}{\Delta}\right)\right|^2 < \infty \tag{22}$$

for all $\xi \in C(\varphi) \subset \left[-\pi\frac{\beta}{\Delta}, \pi\frac{\beta}{\Delta}\right]^d$, where $\sum_l \left|\hat{\varphi}\left(\frac{\xi}{\beta} + 2\pi\frac{l}{\Delta}\right)\right|^2 \neq 0$ on $C(\varphi)$. Note that if $\sum_l \left|\hat{\varphi}\left(\frac{\xi_0}{\beta} + 2\pi\frac{l}{\Delta}\right)\right|^2 = 0$ for some $\xi_0 \in \left[-\pi\frac{\beta}{\Delta}, \pi\frac{\beta}{\Delta}\right]^d$, then the set of functions $\{\varphi_k(\beta x), k \in \boldsymbol{N}^d\}$ is not a set of linearly independent elements, at least in the sense of distributions.

Equation (22) is especially important for numerical applications. For example, if the coefficients $\{c_k\}$ that approximate the function $f(x)$ in an optimal way are determined by using the least-squares method, then the largest and smallest eigenvalues of associated matrices approach the values $1/A$ and $1/B$. As a result, the condition number of the associated matrices approach the ratio A/B which is equal to zero if the conditions of Eq. (22) are not met. Thus, the approximation of the function $f(x)$ can be determined numerically using the linear optimization procedure if and only if Eq.(22) holds.

The preceding analysis can be used to determine how the bell-shaped functions can be made to form a Riesz basis of the corresponding shift invariant spaces. For example, evaluating the Fourier transform of the hat function (Figure 1) one readily finds that the function is a Riesz basis of $V^{\beta,\Delta}(\varphi)$ if $\Delta \neq \frac{1}{n}$, $n = 2, 3, \ldots$. Similarly, the bell-shape function of Eq. (5) and Eq. (7) generates a Riesz basis of $V^{\beta,\Delta}(\varphi)$ if $\Delta \neq \frac{\theta}{n}$, $n = 1, 2, 3, \ldots$.

4. Approximation Properties of Bell-Shaped Functions.

In this section the performance of feedforward neural networks with *l*-finite transfer functions is addressed by studying the approximation properties of the shift-invariant subspaces generated by the corresponding bell-shaped bases.

Instead of approximating the function $f(x) \in H^{\alpha}(\boldsymbol{R}^{d})$ directly, the neural network can approximate the corresponding band-limited function $f_{\Omega}(x)$ such that

$$\hat{f}_{\Omega}(\xi) = \begin{cases} \hat{f}(\xi), & \xi \in C_{\Omega} = [-\Omega, \Omega]^{d} \\ 0, & \xi \notin C_{\Omega} = [-\Omega, \Omega]^{d} \end{cases} . \tag{23}$$

The total error of the network approximation can then be decomposed into two parts:

$$\varepsilon = \|f - f_{a}\|_{H^{\nu}} \le \|f - f_{\Omega}\|_{H^{\nu}} + \|f_{\Omega} - f_{a}\|_{H^{\nu}} \tag{24}$$

The first term of the right hand side of Eq. (24) depends only on the function $f(x)$ and is not affected by the properties of the neural network. The bound on its value is obtained as

$$\|f - f_{\Omega}\|_{H^{\nu}} \le \frac{1}{\Omega^{\alpha-\nu}}\|f\|_{H^{\alpha}} e(\Omega), \qquad 0 \le \nu \le \alpha \tag{25}$$

where $e(\Omega) \to 0$ as $\Omega \to \infty$. The proof of this formula is given in the Appendix. The accuracy of approximation of a general function from $H^{\alpha}(\boldsymbol{R}^{d})$ can be found by studying the approximation of the corresponding band-limited function. To illustrate, substituting Eq. (25) into Eq. (24) one finds,

$$\varepsilon = \|f - f_{a}\|_{H^{\nu}} \le \|f - f_{\Omega}\|_{H^{\nu}} + \|f_{\Omega} - f_{a}\|_{H^{\nu}} \le C\Omega^{-(\alpha-\nu)} + \|f_{\Omega} - f_{a}\|_{H^{\nu}} . \tag{26}$$

Next, let $f_{a}(x, \bar{b}) \in V^{\beta, \Delta}(\varphi)$ be the approximation of the function $f(x)$. It has been previously shown that $\hat{f}_a(\xi, \bar{b}) = \frac{1}{\beta^d}\lambda\left(\frac{\Delta}{\beta}\xi\right)\hat{\varphi}\left(\frac{\xi}{\beta}\right)$ for some 2π periodic function $\lambda(\xi)$. The distance between the space $V^{\beta,\Delta}(\varphi)$ and the function $f(x)$, in the sense of the space $H^{\nu}(\boldsymbol{R}^d)$, can be found by minimizing the approximation error,

$$\varepsilon_v^2 = \int_{\boldsymbol{R}^d} q(\xi)\left|\hat{f}(\xi) - \frac{1}{\beta^d}\lambda\left(\frac{\Delta}{\beta}\xi\right)\hat{\varphi}\left(\frac{\xi}{\beta}\right)\right|^2 d\xi, \tag{27}$$

where $q(\xi) = (1 + |\xi|^2)^{\nu}$. If the function $f(x)$ is band-limited to the frequency $\Omega \le \pi\frac{\beta}{\Delta}$, Eq. (27) can be simplified as

$$\varepsilon_v^2 = \int_{C_\Omega} q(\xi)|\hat{f}(\xi)|^2\left(1 - \left|\hat{\psi}_v\left(\frac{\xi}{\beta}\right)\right|^2\right)d\xi, \tag{28}$$

where

$$\hat{\psi}_v\left(\frac{\xi}{\beta}\right) = \frac{\sqrt{q(\xi)}\hat{\varphi}\left(\frac{\xi}{\beta}\right)}{\sqrt{\sum_k q\left(\xi + 2\pi\frac{\beta}{\Delta}k\right)\left|\hat{\varphi}\left(\frac{\xi}{\beta} + \frac{2\pi}{\Delta}k\right)\right|^2}}. \tag{29}$$

Proof of Eq. (28) and Eq. (29) can be found in the Appendix.

According to Eq. (26) and Eq. (28) the error of the approximation depends on how well the function $\left|\hat{\psi}_v\left(\frac{\xi}{\beta}\right)\right|^2$ approximates 1 in the hypercube $C_\Omega = [-\Omega, \Omega]^d$. Two methods of controlling the error of approximation can be pursued in practice, both related to manipulating the values of β and Δ. The properties of these error control methods are examined next.

4.1 First Method: $\beta \to \infty$ and Δ is Fixed

In the first method the performance of neural networks can be improved, and the error of approximation can be reduced in several cases, by taking increasingly larger values of the parameter β. Note that this method is similar to the spline approximation used extensively in computational mechanics for the solution of differential equations. Carrying the similarity further, the parameter β can be associated with the mesh size of the finite element discretization $h \sim 1/\beta$. The conditions which must be imposed on the bell-shaped function to insure the convergence can be assessed by the following proposition.

Proposition 1. If $\hat{\varphi}(\xi)$ is continuous, $\hat{\varphi}(0) \neq 0$, and

$$\left|\hat{\varphi}\left(\xi + 2\pi\frac{l}{\Delta}\right)\right| \leq c_l |\xi|^{\gamma}, \qquad l \in \boldsymbol{N}^d, \quad l \neq (0, 0, \ldots 0), \tag{30}$$

where γ is a positive constant which indicates the order of zeros of the function $\hat{\varphi}(\xi)$ at the points $\xi = 2\pi\frac{l}{\Delta}$, and c_l are such that

$$\sum_{l \neq (0, 0, \ldots 0)} l^{2\nu} |c_l|^2 < \infty, \tag{31}$$

then

$$\|f - f_a\|_{H^{\nu}} < C_1 \frac{1}{\beta^{\alpha - \nu}} + C_2 \frac{1}{\beta^{\gamma - \nu}} \tag{32}$$

for any function $f(x) \in H^{\alpha}(\boldsymbol{R}^d)$, $\alpha > \nu, \gamma$.

The proof of this proposition is given in the Appendix. A similar proposition was proven by de Boor et al. [25]. Note that Eq. (30) resembles the Strang-Fix condition which is commonly used to establish the approximation order for shift-invariant spaces [27]. The relationship between the conditions of Proposition 1 and the Strang-Fix condition has also been investigated by de Boor et al. [25].

4.2 Second Method: $\beta \to \infty$ and $\Delta \to 0$

Only a small class of bases satisfy Eq. (30). Therefore, a second method of controlling the network performance in conjunction with the bell-shaped function approximation is investigated. A convergence result can be derived by decreasing the value of the parameter Δ and simultaneously increasing the parameter β. This can be viewed in the connectionist sense as varying the input and bias weights in a coupled fashion, rather than independently as is done in most training algorithms.

Proposition 2. If $\hat{\varphi}(0) \neq 0$, $\hat{\varphi}(\xi)$ is continuous and decays to zero at infinity as

$$|\hat{\varphi}(\xi)| \leq \frac{C}{|\xi|^r}, \qquad r > \frac{1}{2} + \nu, \tag{33}$$

where ν is a positive constant, then for any $f(x) \in H^{\alpha}(\boldsymbol{R}^d)$, $\alpha > \nu$

$$\|f - f_a\|_{H^\nu} \leq C_1 \beta^{-(\alpha - \nu)} + C_2 \beta^{\nu} \Delta^{(r - \nu)} \quad . \tag{34}$$

Moreover, if

$$|\hat{\varphi}(\xi)| \leq C e^{-\upsilon|\xi|}, \tag{35}$$

then

$$\|f - f_a\|_{H^\nu} \leq C_1 \beta^{-(\alpha - \nu)} + C_2 \left(\frac{\beta}{\Delta}\right)^{\nu} a^{1/\Delta}, \qquad 0 < a < 1\,. \tag{36}$$

Though this second method provides convergence for a much larger (virtually arbitrary) class of transfer functions than the first method, the associated computational procedures can be inefficient since the basis $\{\varphi(\beta x - k\Delta), k \in \boldsymbol{N}^d\}$ eventually becomes a non-Riesz basis. This can be seen by first substituting Eq. (33) into Eq. (20) and Eq. (21). The ratio of the Riesz bounds is then determined to be

$$\frac{A}{B} \le C_1 \sum_k \left(\frac{\Delta}{|k|}\right)^{2r} = C\Delta^{2r} \tag{37}$$

Similarly, Eq. (35) yields the following ratio of the Riesz bounds

$$\frac{A}{B} \le C_1 \sum_k e^{-2\upsilon\pi|k|/\Delta} = C_2 a^{2/\Delta}, \qquad 0 < a < 1\,. \tag{38}$$

Consequently, the problem of approximating a function in an optimal manner becomes a numerically unstable problem as $\Delta \to 0$.

5. Convergence of the Network Approximation

The class of functions that can be represented by f_a using non-linear transfer functions is dense with respect to a large family of functional spaces, including the space $L^2(\boldsymbol{R}^d)$. However, to the authors' knowledge, few results can be found in the connectionist literature regarding the dependence of the network approximation error on the number of neurons [28-32].

These results can be divided into two types. The first [28-31], using the principles of the convex hull approximation, show that if the function $f(x)$ lies in the closure of the convex hull $\wp$ of a collection of elements, then

$$\|f(x) - f_N(x)\|_{L^2(\boldsymbol{R}^d)} \le \frac{C}{\sqrt{N}}, \tag{39}$$

where $f_N(x)$ is a linear combination of N elements that span the set $\wp$. The estimate given by

Eq. (39) has drawn the interest of the connectionist community since it does not explicitly depend on the dimensionality of the approximation problem. That is, the convex hull approximation of neural networks is not affected by the "curse of dimensionality." However, Eq. (39) possesses significant restrictions on the class of multi-dimensional functions that can be approximated by the convex hull scheme. Specifically, these restrictions can be manifested as smoothness requirements on the target functions. For example, functions from the space $H^{\alpha}(\boldsymbol{R}^d)$, must have $\alpha \geq \frac{d}{2} + 1$, to use the convex hull approximation with 3-layered perceptron networks [28].

The second type of result is the estimate of reference [32]

$$\|f(x) - f_N(x)\|_{L^2(\boldsymbol{R}^d)} \leq C N^{-\alpha/d} \tag{40}$$

which is obtained by associating the FFANN approximation with the approximation by a polynomial basis. The transfer functions of the corresponding FFANN systems are required to be infinitely smooth and satisfy additional constraints.

The linear approximation by bell-shaped functions developed in the preceding section can also be used to establish the efficiency of neural networks. For the sake of brevity, we will restrict our discussion to infinitely smooth bell-shaped functions that satisfy Eq. (35). By selecting the parameter $\beta = C a^{-1/(\alpha\Delta)}$, where C is a constant and $0 < a < 1$, the error of the bell-shaped network approximation of a function $f(x)$ from the space $H^{\alpha}(\boldsymbol{R}^d)$ can be bounded as

$$\|f - f_a\|_{L^2} < C a^{1/\Delta}, \qquad 0 < a < 1 . \tag{41}$$

Further, the number of bell-shaped functions *N* corresponding to the selected parameters β and Δ is determined by the following equation

$$N = C_B\left(\frac{\beta}{\Delta}\right)^d = C_B\left(\frac{a^{-1/(\alpha\Delta)}}{\Delta}\right)^d, \tag{42}$$

where C_B is the constant that in practice depends only on the size of the approximation domain $B \subset \boldsymbol{R}^d$ and N is directly proportional to the number of neurons. Combining Eq. (41) and Eq. (42) yields the error estimate

$$\|f - f_a\|_{L^2} \le CN^{-\alpha/d}\ln N. \tag{43}$$

Note that apart from the factor $\ln N$ in Eq. (43), the linear approximation by infinitely smooth bell-shaped bases attains the optimal rate of convergence for a linear technique [33].

It should be pointed out that the error estimate of Eq. (43) is more robust than the corresponding result of the convex hull approximation of Eq. (39). For $\alpha = \frac{d}{2} + 1$ the rate of convergence of the network approximation by bell-shaped functions is, apart from the slowly increasing factor $\ln N$, equal to $\frac{1}{2} + \frac{1}{d}$ which is a higher rate of convergence than that provided by Eq. (39). Moreover, the class of functions which can be approximated by bell-shaped functions is significantly larger and include all square integrable functions. In addition, the bell-shaped basis approximation is applicable for a large class of sigmoidal transfer functions and is not restricted to only infinitely smooth functions, as it is required in reference [32].

6. Numerical Examples of the Network Approximation

As mentioned earlier, a simple linear spline, or hat function, can be obtained from the sigmoidal function of Eq. (6). A basis developed by shifting the hat function is commonly used in computational mechanics in conjunction with the finite element method. Meade and Fernandez [34] used this fact to show the similarity between the approximation introduced by neural networks and computational methods in engineering. It is well known that interpolation error introduced by the hat basis functions of Figure 1 is proportional to $h^2 = \left(\frac{1}{\beta}\right)^2$, where h is the mesh size. This result can also be obtained by employing Proposition 1 of Section 4.1 and noting that the Fourier transform of the hat function of Figure 1 has zeros of order 2 at points $\xi_l = 2\pi l$, $l \neq 0$. That is, the hat function satisfies Eq. (30) with $\gamma = 2$.

Alternatively, the Fourier transform of the bell-shaped function $\varphi(x)$ of Figure 2 has only a simple zero at $\xi_l = \frac{\pi}{\theta}l$. That is, in this case Eq. (30) is satisfied with $\gamma = 1$ if $\Delta = \frac{2}{n}\theta$. Figure 5(a) shows the approximation of the infinitely smooth function

$$f(x) = e^{-x^2} \tag{44}$$

by a collection of the bell-shaped functions of Figure 2 using $\theta = 1$, $\Delta = 2$, and $\beta = 20$. The linear parameters were evaluated by the least-squares method. As the parameter β increases, the function of Eq. (44) and its approximation becomes almost indistinguishable. The numerically calculated rate of convergence in the square mean sense is shown in Figure 5(b). The slope of the curve in this figure is approximately equal to 1. This result is consistent with the theoretical estimates of Proposition 1.

However, Figure 5(c) shows that though the function of Eq. (44) and its approximation match well, their first derivatives do not. This deficiency cannot be reduced by increasing the value of the parameter β. The approximation of Eq. (44) with $\theta = 0.8$, $\Delta = 2$, and $\beta = 20$ is shown in Figure 5(d). In this case, the bell-shaped function of Eq. (5) and Eq. (7) does not satisfy the conditions of Eq. (30) and the shift-invariant space $V^{\beta,\Delta}(\varphi)$ does not properly approximate the functions of the space $H^{\alpha}(\boldsymbol{R}^d)$ as $\beta \to \infty$.

Recalling the second method outlined in Section 4.2, Figure 6(a) shows the function $f(x)$ of Eq. (44) and its network approximation using the bell-shaped functions of Figure 2 with $\theta = 1$, $\Delta = 0.26$, and $\beta = 20$. Figures 6(b) and (c) compare the first and second derivatives of the network approximation against the respective derivatives of Eq. (44). The error of approximation is sufficiently low in all three cases. The dependence of the approximation error on the number of basis functions *N* is shown in Figure 6(d); the values of both β and Δ were varied appropriately to ensure the best rate of convergence. Note that since the target function of Eq. (44) is infinitely smooth the approximation error decreases exponentially with the number of basis functions, N.

7. Conclusions

The adaptive data-driven emulation and control of mechanical systems are popular applications of artificial neural networks in engineering. However, because FFANN training is an ill-posed nonlinear optimization problem it is difficult to apply this attractive modeling tool to engineering applications. The use of regularization functionals has been proposed in the literature to increase the numerical robustness of FFANN training. This paper has explored an alternate method of network training that can be combined with regularization. Specifically, the network parameters were constrained so that a host of conventional computational techniques for function approximation, such as least-squares, can be used to evaluate the network parameters. This was accomplished by forming local Riesz bases which provide accurate approximation and stable evaluation of the network parameters.

The accuracy of the local bases was analyzed by employing the concept of shift invariant spaces. Two methods were explored in controlling the error of network approximation: 1) adding bases while altering their shape and keeping their spacing constant ($\beta \to \infty$ and Δ is fixed), and 2) adding bases while altering their shape and decreasing Δ in a coupled fashion ($\beta \to \infty$ and $\Delta \to 0$). It has been shown that the first method imposes stringent Strang-Fix-type conditions on the bell-shaped bases in decreasing the approximation error. Consequently, the bell-shaped functions associated with the hyperbolic tangents provide only a linear convergence rate.

The second method approximated the function and its derivatives for the bell-shaped functions associated with the hyperbolic tangents. Theoretically, the approximation capabilities are also possible for a large (virtually arbitrary) class of transfer functions, which should be welcome news for engineers pursuing digital or other hardware implementations. Also, this large class of transfer functions should be especially useful in the approximation of function derivatives that may be required for modelling integral-differential equations associated with complex engineering systems of interest. These modelling equations can be readily used as a regularization functional for network training. The second method provides the optimal rate of convergence for the linear technique. In addition, the class of target functions which can be approximated by bell-shaped bases is significantly larger than the first method and include all square integrable functions, making engineering applications with FFANNs more tractable. The convergence rate for the second method is exponential when using infinitely smooth transfer functions and target functions. However, the second method suffers from an increasingly ill-conditioned matrix associated with the evaluation of the linear parameters. The source of the ill-conditioning has been found to be directly related to the magnitude of Δ through the ratio of the Riesz bounds in Eq. (37) and Eq. (38).

To ensure that a FFANN can form local bases exhibiting properties that would be advantageous to engineering applications, the user can first constrain the sigmoidal transfer functions using Eq. (9). Utilizing the results of the previous sections, the relation between the remaining nonlinear parameters β and Δ, which would complete the constraints, can be found through Eq. (34) or Eq. (36) depending on how the local basis function decays with ξ in the frequency domain. For the infinitely smooth bell-shaped function formed from the popular hyperbolic tangent, one would use $\beta = Ca^{-1/(\alpha\Delta)}$ where $0 < a < 1$. Using Eq.(13) the results of this paper can be extended to multi-dimensional local bases in a straightforward manner.

Future work will include an investigation of embedding governing equations of physical systems in a FFANN through the use of regularization functionals and application of multi-dimensional local bases.

Acknowledgments

This work was supported under NASA Johnson Space Center grant NAG 9-719 and Office of Naval Research grant N00014-95-1-0741.

References

1. Cybenko, G., "Approximation by Superposition of a Sigmoidal Function," *Mathematics of Control, Signals, and Systems*, Vol. 2, 1989, pp. 303-314.

2. Girosi, F., and Poggio, T., "Networks and the Best Approximation Property," *Biological Cybernetics*, 63, 1990, pp. 169-176.

3. Hornik, K., "Some New Results on Neural Network Approximation," *Neural Networks*, Vol. 6, 1993, pp. 1069-1072.

4. Hornik, K., Stinchcombe, M., and White, H., "Universal Approximation of an Unknown Mapping and its Derivatives Using Multilayer Feedforward Networks," *Neural Networks*, Vol. 3, 1990, pp. 551-560.

5. Ito, Y., "Representation of Functions by Superpositions of a Step or Sigmoidal Function and Their Applications to Neural Network Theory," *Neural Networks*, Vol. 4, 1991, pp. 385-394.

6. Rumelhart, D., Hinton, G.E., and Williams, R.J., "Learning Internal Representations by Error Propagation", in *Parallel Distributed Processing: Exploration in the Microstructure of Cognition*, edited by D. Rumelhart, J.L. McClelland, and the PDP Research Group, MIT Press, Cambridge, MA, 1986, Ch. 8, pp. 318-362.

7. Saarinen, S., Bramley, R., and Cybenko, G., "Ill-Conditioning in Neural Network Training Problems," *SIAM J. Sci. Comput*., Vol. 14, No. 3, 1993, pp. 693-714.

8. Poggio, T., and Girosi, F., “Regularization Algorithms for Learning that Are Equivalent to Multilayer Networks,” *Science*, Vol. 247, 1990, pp. 978-982.

9. Poggio, T., and Girosi, F., “Networks for Approximation and Learning,” *Proceedings of the IEEE*, Vol. 78, No. 9, 1990, pp. 1481-1497.

10. Tikhonov, A. N., and Arsenin, V.Y., “Solution of Ill-Posed Problems,” W.H. Winston, Washington, D.C., 1977.

11. Bishop, C.M., “Curvature-Driven Smoothing: A Learning Algorithm for Feedforward Networks,” *IEEE Transactions on Neural Networks*, Vol. 4, No. 5, 1993, pp. 882-884.

12. Bishop, C.M., “Training with Noise is Equivalent to Tikhonov Regularization,” *Neural Computation*, Vol. 7, 1995, pp. 108-116.

13. Webb, A.R., “Functional Approximation by Feed-Forward Networks: A Least Square Approach to Generalization,” *IEEE Transactions on Neural Networks*, Vol. 5, No. 3, 1994, pp. 363-371.

14. Reed, R., Marks, R.J., and Oh, S., “Similarity of Error Regularization, Sigmoid Gain Scaling, Target Smoothing, and Training with Jitter,” *IEEE Transactions on Neural Networks*, Vol. 6, No. 3, 1995, pp. 529-538.

15. Igelnik, B., and Pao, Y.-H., “Stochastic Choice of Basis Functions in Adaptive Function Approximation and the Functional-Link Net,” *IEEE Transactions on Neural Networks*, Vol. 6, No. 6, 1995, pp. 1320-1329.

16. Chen, S., Cowan, C.F.N., and Grant, P.M., “Orthogonal Least Square Algorithm for Radial Basis Function Networks,” *IEEE Transactions on Neural Networks*, Vol. 2, No. 2, 1991, pp. 302-309.

17. Lehtokangas, M., Saarinen J., Kaski, K., and Huuntanen, P., “Initializing Weights of a Multilayer Perceptron Network by Using the Orthogonal Least Squares Algorithm,” *Neural Computation*, Vol. 7, 1995, pp. 982-999.

18. Bertero, M., Poggio, T., and Torre, V., "Ill-Posed Problems in Early Vision," *Proceeding of the IEEE*, Vol. 76, 1988, pp. 869-889.

19. Cox, D.D., "Multivariate Smoothing Spline Functions," *SIAM J. Numer. Anal*. Vol. 21, No. 4, 1984, pp. 789-813.

20. Wahba, G., "Spline Models for Observational Data", SIAM, Philadelphia, 1990.

21. Leshno, M., Lin, V.Y., Pinkus, A., and Schocken, S., "Multilayer Feedforward Networks with a Nonpolynomial Activation Function Can Approximate Any Function," *Neural Networks*, Vol. 6, 1993, pp. 861-867.

22. Mhaskar, H.N., "Neural Networks for Optimal Approximation of Smooth and Analytic Functions," *Neural Computation*, Vol. 8, 1995, pp. 164-177.

23. Cardaliaguet, P., and Euvrard, G., "Approximation of a Function and Its Derivative with a Neural Network," *Neural Networks*, Vol. 5, 1992, pp. 207-220.

24. Jetter, K., "Multivariate Approximation from the Cardinal Interpolation Point of View," *Approximation Theory VII*, edited by E.W. Cheney, C.K.Chui, and L.L. Schumaker, 1992, pp. 131-161.

25. de Boor, C., DeVore, R.A., and Ron, A., "Approximation from Shift-Invariant Subspaces of $L_2(\boldsymbol{R}^d)$," *Transaction of the American Mathematical Society*, Vol. 341, No. 2, 1994, pp. 787-806.

26. Grisvard, P., "Elliptic Problems in Nonsmooth Domains," Pitman Advanced Publishing Program, Boston, 1985.

27. Jia, R.-Q., and Lei, J., "Approximation by Multiinteger Translates of Functions Having Global Support," *Journal of Approximation Theory*, Vol. 72, 1993, pp. 2-23.

28. Barron, A.R., "Universal Approximation Bounds for Superpositions of a Sigmoidal Function," *IEEE Transactions on Information Theory*, Vol. 39, No. 3, 1993, pp. 930-945.

29. Jones, L.K., “Constructive Approximations for Neural Networks by Sigmoidal Functions,” *Proceedings of the IEEE*, Vol. 78, No. 10, 1990, pp. 1586-1589.

30. Jones, L.K., “A Simple Lemma on Greedy Approximation in Hilbert Space and Convergence Rates for Projection Pursuit Regression and Neural Network Training,” *The Annals of Statistics*, Vol. 20, No. 1, 1992, pp. 608-613.

31. Girosi, F., and Anzellotti, G., “Convergence Rates of Approximation by Translates,” Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Report AI 1288, 1992.

32. Mhaskar, H.N., “Neural Network for Optimal Approximation of Smooth and Analytic Functions,” *Neural Computation,* Vol. 8, 1996, pp. 164-177.

33. Lorentz, G.G., “Approximation of Functions,” Chelsea Publishing Co., New York, 1986.

34. Meade, A.J., and Fernandez, A.A., “Solution of Non-Linear Ordinary Differential equations by Feedforward Neural Networks,” *Mathl. Comput. Modeling*, Vol. 20, No. 9, 1994, pp. 19-44.

Appendix A

Proof of Eq. (25). One can find

$$\|f - f_\Omega\|^2_{H^\nu} = \int_{\boldsymbol{R}^d / C_\Omega} (1 + |\xi|^2)^\nu |\hat{f}(\xi)|^2 d\xi = \Omega^d \int_{\boldsymbol{R}^d / C_1} (1 + |\Omega\xi|^2)^\nu |\hat{f}(\Omega\xi)|^2 d\xi =$$

$$\Omega^d \int_{\boldsymbol{R}^d / C_1} \frac{(1 + |\Omega\xi|^2)^\alpha |\hat{f}(\Omega\xi)|^2}{(1 + |\Omega\xi|^2)^{\alpha - \nu}} d\xi = \Omega^{d - 2(\alpha - \nu)} \int_{\boldsymbol{R}^d / C_1} \frac{(1 + |\Omega\xi|^2)^\alpha |\hat{f}(\Omega\xi)|^2}{(1/\Omega^2 + |\xi|^2)^{\alpha - \nu}} d\xi \le$$

$$\Omega^{d - 2(\alpha - \nu)} \int_{\boldsymbol{R}^d / C_1} (1 + |\Omega\xi|^2)^\alpha |\hat{f}(\Omega\xi)|^2 d\xi = \Omega^{-2(\alpha - \nu)} \int_{\boldsymbol{R}^d / C_\Omega} (1 + |\xi|^2)^\alpha |\hat{f}(\xi)|^2 d\xi =$$

$$\frac{1}{\Omega^{2(\alpha-\nu)}}\|f\|^2_{H^\alpha}e^2(\Omega), \tag{45}$$

where $e(\Omega) = \left(\dfrac{\int\limits_{\boldsymbol{R}^d/C_\Omega}(1+|\xi|^2)^\alpha|\hat{f}(\xi)|^2 d\xi}{\int\limits_{\boldsymbol{R}^d}(1+|\xi|^2)^\alpha|\hat{f}(\xi)|^2 d\xi}\right)^{1/2} \to 0$, as $\Omega \to \infty$. □

Proof of Eq. (28) and Eq. (29). The problem of minimizing the error of Eq. (27) is a constrained minimization problem since the function $\lambda(\xi)$ is a periodic function. The following transformation reduces the problem to an unconstrained minimization problem

$$\varepsilon_\nu^2 = \int\limits_{\boldsymbol{R}^d} q(\xi)\left(|\hat{f}(\xi)|^2 - \hat{f}(\xi)\overline{\frac{1}{\beta^d}\lambda\left(\frac{\Delta}{\beta}\xi\right)\hat{\varphi}\left(\frac{\xi}{\beta}\right)} - \overline{\hat{f}(\xi)}\frac{1}{\beta^d}\lambda\left(\frac{\Delta}{\beta}\xi\right)\hat{\varphi}\left(\frac{\xi}{\beta}\right) + \left|\frac{1}{\beta^d}\lambda\left(\frac{\Delta}{\beta}\xi\right)\right|^2\left|\hat{\varphi}\left(\frac{\xi}{\beta}\right)\right|^2\right)d\xi =$$

$$\sum_k \int\limits_{C_{\pi\beta/\Delta}} q\left(\xi + 2\pi\frac{\beta}{\Delta}k\right)\left(\left|\hat{f}\left(\xi + 2\pi\frac{\beta}{\Delta}k\right)\right|^2 - \hat{f}\left(\xi + 2\pi\frac{\beta}{\Delta}k\right)\overline{\frac{1}{\beta^d}\lambda\left(\frac{\Delta}{\beta}\xi\right)\hat{\varphi}\left(\frac{\xi}{\beta} + \frac{2\pi}{\Delta}k\right)} -\right.$$

$$\left.\overline{\hat{f}\left(\xi + 2\pi\frac{\beta}{\Delta}k\right)}\frac{1}{\beta^d}\lambda\left(\frac{\Delta}{\beta}\xi\right)\hat{\varphi}\left(\frac{\xi}{\beta} + \frac{2\pi}{\Delta}k\right) + \left|\frac{1}{\beta^d}\lambda\left(\frac{\Delta}{\beta}\xi\right)\right|^2\left|\hat{\varphi}\left(\frac{\xi}{\beta} + \frac{2\pi}{\Delta}k\right)\right|^2\right)d\xi. \tag{46}$$

The minimization of Eq. (46) is straightforward and the solution is equal to

$$\frac{1}{\beta^d}\lambda\left(\frac{\Delta}{\beta}\xi\right) = \frac{\sum_k q\left(\xi + 2\pi\frac{\beta}{\Delta}k\right)\hat{f}\left(\xi + 2\pi\frac{\beta}{\Delta}k\right)\overline{\hat{\varphi}\left(\frac{\xi}{\beta} + \frac{2\pi}{\Delta}k\right)}}{\sum_k q\left(\xi + 2\pi\frac{\beta}{\Delta}k\right)\left|\hat{\varphi}\left(\frac{\xi}{\beta} + \frac{2\pi}{\Delta}k\right)\right|^2}. \tag{47}$$

By substituting Eq. (47) into Eq. (46), one finds the distance between $V^{\beta,\Delta}(\varphi)$ and the function $f(x)$,

$$\varepsilon_v^2 = \int\limits_{C_{\pi\beta/\Delta}} \left(\sum_k q\left(\xi + 2\pi\frac{\beta}{\Delta}k\right)\left|\hat{f}\left(\xi + 2\pi\frac{\beta}{\Delta}k\right)\right|^2 - \frac{\left|\sum_k q\left(\xi + 2\pi\frac{\beta}{\Delta}k\right)\hat{f}\left(\xi + 2\pi\frac{\beta}{\Delta}k\right)\overline{\hat{\varphi}\left(\frac{\xi}{\beta} + \frac{2\pi}{\Delta}k\right)}\right|^2}{\sum_k q\left(\xi + 2\pi\frac{\beta}{\Delta}k\right)\left|\hat{\varphi}\left(\frac{\xi}{\beta} + \frac{2\pi}{\Delta}k\right)\right|^2} \right) d\xi \tag{48}$$

If the function $f(x)$ is band-limited to the frequency $\Omega \le \pi\frac{\beta}{\Delta}$, Eq. (48) simplifies to Eq. (28) and Eq. (29) □

Proof of Proposition 1. To establish Eq. (32) one needs to evaluate $\left(1 - \left|\hat{\psi}_v\left(\frac{\xi}{\beta}\right)\right|^2\right)$, which is always a positive function less than or equal to one. It is then necessary to find its bound only in a neighborhood of the origin. Since $\hat{\varphi}(0) \ne 0$ and $\hat{\varphi}(\xi)$ is a continuous function, there is a neighborhood of the origin $\xi \in S$ where

$$|\hat{\varphi}(\xi)| \ge \eta > 0\,. \tag{49}$$

Next, for $\xi \in \beta S$

$$1 - \left|\hat{\psi}_v\left(\frac{\xi}{\beta}\right)\right|^2 = \frac{\sum\limits_{l \ne (0,\ldots 0)} \left(1 + \left|\xi + 2\pi l\frac{\beta}{\Delta}\right|^2\right)^{\nu} \left|\hat{\varphi}\left(\frac{\xi}{\beta} + 2\pi\frac{l}{\Delta}\right)\right|^2}{\sum\limits_{l}\left(1 + \left|\xi + 2\pi l\frac{\beta}{\Delta}\right|^2\right)^{\nu}\left|\hat{\varphi}\left(\xi + 2\pi\frac{l}{\Delta}\right)\right|^2} \le$$

$$\frac{C}{\eta^2(1 + |\xi|^2)^{\nu}} \sum_{l \ne (0,\ldots 0)} \left|2\pi l\frac{\beta}{\Delta}\right|^{2\nu}\left|\hat{\varphi}\left(\frac{\xi}{\beta} + 2\pi\frac{l}{\Delta}\right)\right|^2. \tag{50}$$

Substituting Eq. (30) into Eq. (50) yields

$$1 - \left|\hat{\psi}_v\left(\frac{\xi}{\beta}\right)\right|^2 \le \frac{C}{\eta^2(1+|\xi|^2)^\nu}\left(2\pi\frac{\beta}{\Delta}\right)^{2\nu}\left|\frac{\xi}{\beta}\right|^{2\gamma}\sum_{l\neq 0} l^{2\nu}|c_l|^2 . \tag{51}$$

Therefore, selecting $\Omega = \pi\frac{\beta}{\Delta}$ and substituting Eq. (51) into Eq. (28) with $q(\xi) = (1+|\xi|^2)^\nu$ gives

$$\left\|f_{\pi\frac{\beta}{\Delta}} - f_a\right\|^2_{H^\nu} \le \int_{C_{\pi\beta/\Delta}} |\hat{f}_\Omega(\xi)|^2 \frac{C}{\eta^2}\left(2\pi\frac{\beta}{\Delta}\right)^{2\nu}\left|\frac{\xi}{\beta}\right|^{2\gamma}\sum_{l\neq 0} l^{2\nu}|c_l|^2 d\xi \le C\frac{\|f\|^2_{H^\gamma}}{\beta^{2(\gamma-\nu)}} . \tag{52}$$

Combining Eq. (26) and Eq. (52) proves the proposition. □

Proof of Proposition 2. By selecting $\Omega = \pi\frac{\beta}{\Delta_0}$, $\Delta_0 > \Delta$, Eq. (26) can be rewritten as

$$\varepsilon = \|f - f_a\|_{H^\nu} \le C\beta^{-(\alpha-\nu)} + \left\|f_{\pi\frac{\beta}{\Delta_0}} - f_a\right\|_{H^\nu} . \tag{53}$$

Since the function $\hat{\varphi}(\xi)$ is assumed to be continuous and $\hat{\varphi}(0) \neq 0$, one can find Δ_0 large enough so that

$$\left|\hat{\varphi}\left(\frac{\xi}{\beta}\right)\right| \ge \eta > 0 \tag{54}$$

for all $\xi \in \left[-\pi\frac{\beta}{\Delta_0}, \pi\frac{\beta}{\Delta_0}\right]^d$. Taking $q(\xi) = (1+|\xi|^2)^\nu$ one can obtain the following bound

$$1 - \left|\hat{\psi}_v\left(\frac{\xi}{\beta}\right)\right|^2 = \frac{\sum_{l\neq(0,\ldots0)}\left(1+\left|\xi+2\pi l\frac{\beta}{\Delta}\right|^2\right)^\nu\left|\hat{\varphi}\left(\frac{\xi}{\beta}+2\pi\frac{l}{\Delta}\right)\right|^2}{\sum_l\left(1+\left|\xi+2\pi l\frac{\beta}{\Delta}\right|^2\right)^\nu\left|\hat{\varphi}\left(\frac{\xi}{\beta}+2\pi\frac{l}{\Delta}\right)\right|^2} \le$$

$$\frac{\sum_{l \neq (0,\ldots 0)} \left(1 + \left|\xi + 2\pi l\frac{\beta}{\Delta}\right|^2\right)^{\nu} \left|\hat{\varphi}\left(\frac{\xi}{\beta} + 2\pi\frac{l}{\Delta}\right)\right|^2}{(1+|\xi|^2)^{\nu} min\left|\hat{\varphi}\left(\frac{\xi}{\beta}\right)\right|^2} \le \frac{max\left(\sum_{l \neq 0} \left(1 + \left|\xi + 2\pi l\frac{\beta}{\Delta}\right|^2\right)^{\nu} \left|\hat{\varphi}\left(\frac{\xi}{\beta} + 2\pi\frac{l}{\Delta}\right)\right|^2\right)}{(1+|\xi|^2)^{\nu}\eta^2} . \quad (55)$$

The assumption of Eq. (33) yields

$$\sum_{l \neq (0,\ldots 0)} \left(1 + \left|\xi + 2\pi l\frac{\beta}{\Delta}\right|^2\right)^{\nu} \left|\hat{\varphi}\left(\frac{\xi}{\beta} + 2\pi\frac{l}{\Delta}\right)\right|^2 \le C \sum_{l \neq (0,\ldots 0)} \frac{\left|3\pi l\frac{\beta}{\Delta}\right|^{2\nu}}{\left|\pi\frac{2l}{\Delta}\right|^{2r}} = C'\beta^{2\nu}\Delta^{2(r-\nu)} . \quad (56)$$

Thus the second term of Eq. (53) can be bounded as

$$\left\| f_{\pi\frac{\beta}{\Delta_0}} - f_a \right\|^2_{H^{\nu}} = \int_{C_{\pi\frac{\beta}{\Delta_0}}} q(\xi)\left|\hat{f}(\xi)\right|^2 \left(1 - \left|\hat{\psi}_v\left(\frac{\xi}{\beta}\right)\right|^2\right) d\xi \le C'\beta^{2\nu}\Delta^{2(r-\nu)}\|f\|^2_{L^2} . \quad (57)$$

Similarly, Eq. (35) yields that

$$\sum_{l \neq (0,\ldots 0)} \left(1 + \left|\xi + 2\pi l\frac{\beta}{\Delta}\right|^2\right)^{\nu} \left|\hat{\varphi}\left(\frac{\xi}{\beta} + 2\pi\frac{l}{\Delta}\right)\right|^2 \le C\sum_{l} \left|3\pi l\frac{\beta}{\Delta}\right|^{2\nu} e^{-2\upsilon\pi\frac{|2l|}{\Delta}} = C'\left(\frac{\beta}{\Delta}\right)^{2\nu} a^{2/\Delta} , \quad (58)$$

where $0 < a < 1$. In this case, the second term of Eq. (53) can be bounded as

$$\left\| f_{\pi\frac{\beta}{\Delta_0}} - f_a \right\|^2_{H^{\nu}} = \int_{C_{\pi\frac{\beta}{\Delta_0}}} q(\xi)\left|\hat{f}(\xi)\right|^2 \left(1 - \left|\hat{\psi}_v\left(\frac{\xi}{\beta}\right)\right|^2\right) d\xi \le C'\left(\frac{\beta}{\Delta}\right)^{2\nu} a^{2/\Delta}\|f\|^2_{L^2} . \ \square \quad (59)$$

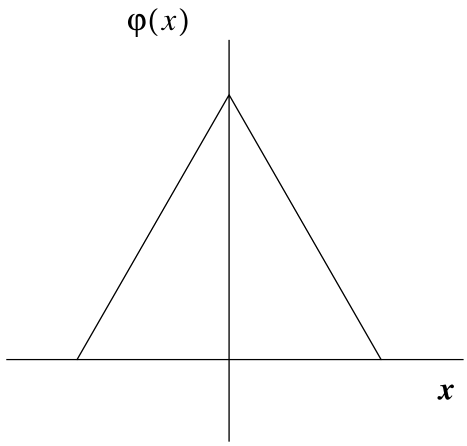

Figure 1: Hat function associated with piece-wise linear sigmoidal functions

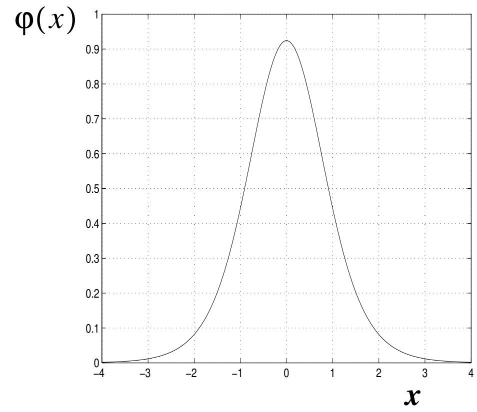

Figure 2: Bell-shaped function associated with a hyperbolic tangent sigmoidal network

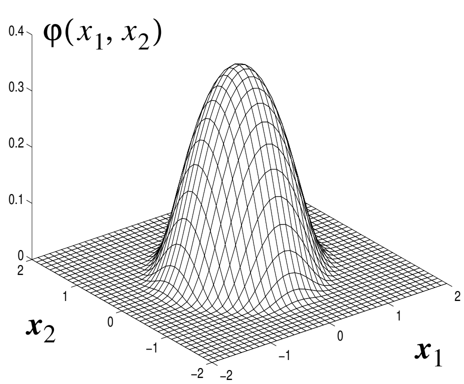

Figure 3: Multi-dimensional bell-shaped function associated with a 4-layer perceptron

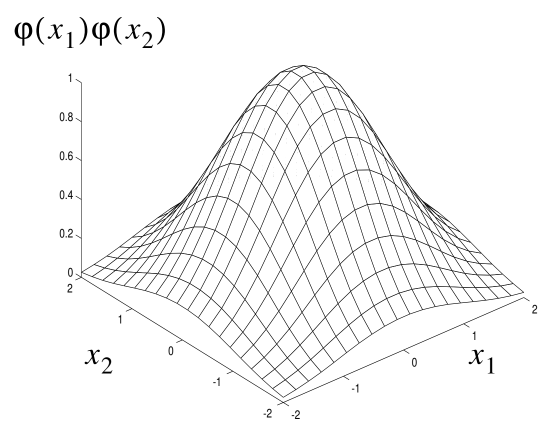

Figure 4: Tensor product of a bell-shaped function using a 4-layer perceptron

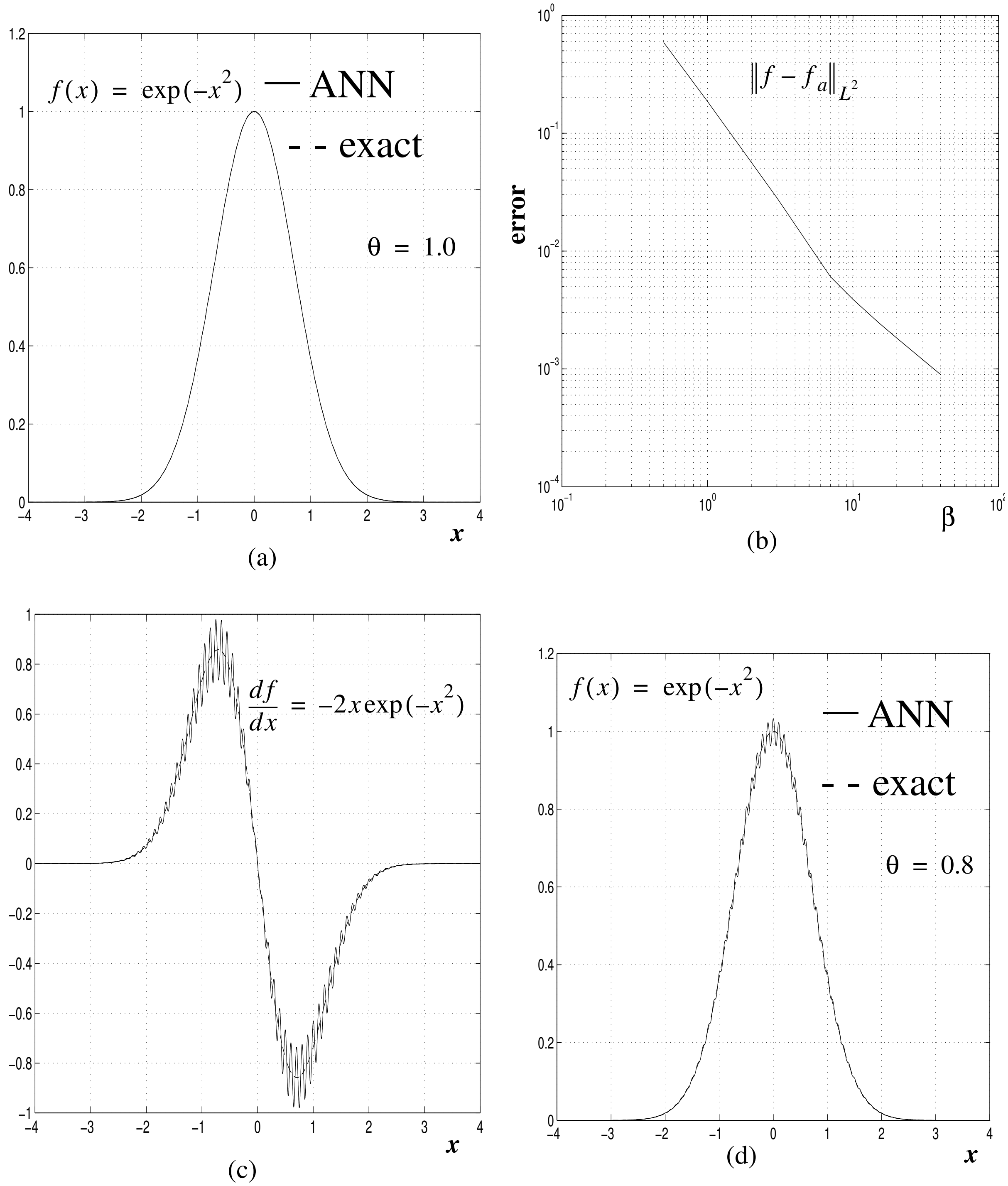

Figure 5: Analytical function and its approximation
for $\beta \to \infty$ and fixed Δ.

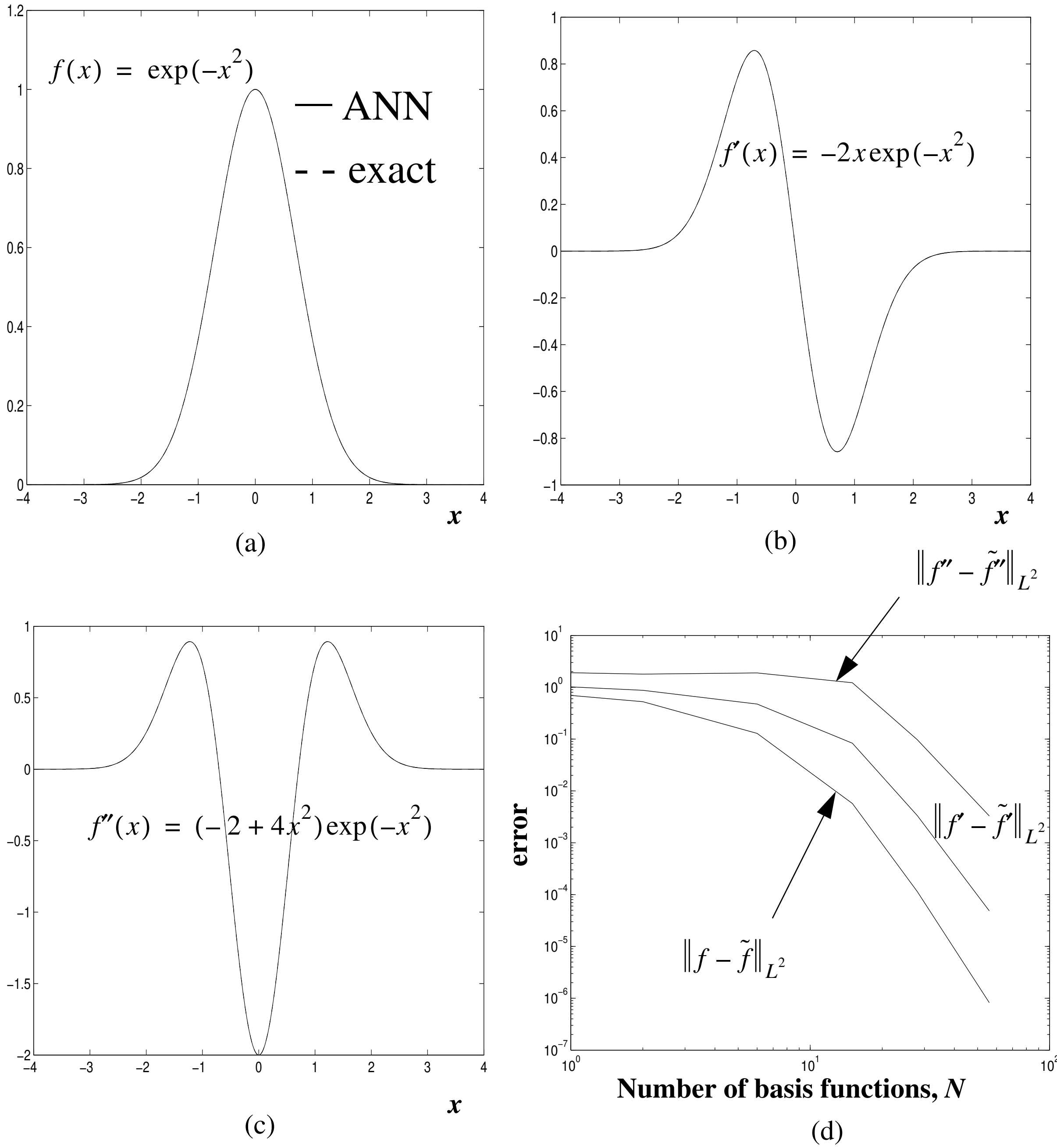

Figure 6: Analytical function and its approximation for $\beta \to \infty$ and $\Delta \to 0$.