Finite Neuron Method and Convergence Analysis

Jinchao Xu

Department of Mathematics, Pennsylvania State University, University Park, PA 16802, USA.

Received 27 September 2020; Accepted (in revised version) 21 October 2020

Abstract. We study a family of $H^m$-conforming piecewise polynomials based on the artificial neural network, referred to as the finite neuron method (FNM), for numerical solution of $2m$-th-order partial differential equations in $\mathbb{R}^d$ for any $m,d \geq 1$ and then provide convergence analysis for this method. Given a general domain $\Omega \subset \mathbb{R}^d$ and a partition $T_h$ of $\Omega$, it is still an open problem in general how to construct a conforming finite element subspace of $H^m(\Omega)$ that has adequate approximation properties. By using techniques from artificial neural networks, we construct a family of $H^m$-conforming functions consisting of piecewise polynomials of degree $k$ for any $k \geq m$ and we further obtain the error estimate when they are applied to solve the elliptic boundary value problem of any order in any dimension. For example, the error estimates that $\|u - u_N\|_{H^m(\Omega)} = O(N^{-\frac{d}{2} - \frac{1}{2}})$ is obtained for the error between the exact solution $u$ and the finite neuron approximation $u_N$. A discussion is also provided on the difference and relationship between the finite neuron method and finite element methods (FEM). For example, for the finite neuron method, the underlying finite element grids are not given a priori and the discrete solution can be obtained by only solving a non-linear and non-convex optimization problem. Despite the many desirable theoretical properties of the finite neuron method analyzed in the paper, its practical value requires further investigation as the aforementioned underlying non-linear and non-convex optimization problem can be expensive and challenging to solve. For completeness and the convenience of the reader, some basic known results and their proofs.

AMS subject classifications: 65D07, 65D15, 65N22, 65N30

Key words: Finite neuron method, finite element method, neural network, error estimate.

1 Introduction

This paper is devoted to the study of numerical methods for high-order partial differential equations in any dimension using appropriate piecewise polynomial function classes.

*Corresponding author. Email address: xu@math.psu.edu (J. Xu)
In this introduction, we will briefly describe a class of elliptic boundary value problems of any order in any dimension. We will then give an overview of some existing numerical methods for this model and other related problems. We will then explain the motivation and objective of this paper.

1.1 Model problem

Let \( \Omega \subset \mathbb{R}^d \) be a bounded domain with a sufficiently smooth boundary \( \partial \Omega \). For any integer \( m \geq 1 \), we consider the following model \( 2m \)-th-order partial differential equation with certain boundary conditions:

\[
\begin{aligned}
Lu &= f \quad \text{in } \Omega, \\
B^k(u) &= 0 \quad \text{on } \partial \Omega \quad (0 \leq k \leq m-1),
\end{aligned}
\]  

(1.1)

where \( L \) is the partial differential operator

\[
Lu = \sum_{|\alpha|=m} (-1)^{|\alpha|} \partial^\alpha (a_\alpha(x) \partial^\alpha u) + a_0(x)u,
\]

(1.2)

and \( \alpha \) denotes the \( n \)-dimensional multi-index \( \alpha = (\alpha_1, \cdots, \alpha_n) \) with

\[
|\alpha| = \sum_{i=1}^n \alpha_i, \quad \partial^\alpha = \frac{\partial^{|\alpha|}}{\partial x_1^{\alpha_1} \cdots \partial x_n^{\alpha_n}}.
\]

For simplicity, we assume that \( a_\alpha \) are strictly positive and smooth functions on \( \Omega \) for \( |\alpha| = m \) and \( \alpha = 0 \), namely, \( \exists \alpha_0 > 0 \), such that

\[
a_\alpha(x), a_0(x) \geq \alpha_0, \quad \forall x \in \Omega, \quad |\alpha| = m.
\]

(1.3)

Given a nonnegative integer \( k \) and a bounded domain \( \Omega \subset \mathbb{R}^d \), let

\[
H^k(\Omega) := \{ v \in L^2(\Omega), \partial^\alpha v \in L^2(\Omega), |\alpha| \leq k \}
\]

be standard Sobolev spaces with the norm and seminorm given respectively by

\[
\|v\|_k := \left( \sum_{|\alpha| \leq k} \|\partial^\alpha v\|_0^2 \right)^{1/2}, \quad |v|_k := \left( \sum_{|\alpha| = k} \|\partial^\alpha v\|_0^2 \right)^{1/2}.
\]

For \( k = 0 \), \( H^0(\Omega) \) is the standard \( L^2(\Omega) \) space with the inner product denoted by \( (\cdot, \cdot) \). Similarly, for any subset \( K \subset \Omega \), the \( L^2(K) \) inner product is denoted by \( (\cdot, \cdot)_0 \). We note that, by the well-known property of the Sobolev space, the assumption (1.3) implies that

\[
a(v,v) \geq \|v\|^2_{m, \Omega}, \quad \forall v \in H^m(\Omega).
\]

(1.4)

The boundary value problem (1.1) can be cast into an equivalent optimization or a variational problem as described below for some approximate subspace \( V \subset H^m(\Omega) \).
Minimization Problem M: Find \( u \in V \) such that
\[
J(u) = \min_{v \in V} J(v),
\] (1.5)
or

Variational Problem V: Find \( u \in V \) such that
\[
a(u,v) = \langle f,v \rangle, \quad \forall v \in V.
\] (1.6)

The bilinear form \( a(\cdot,\cdot) \) in (1.6), the objective functional \( J(\cdot) \) in (1.5), and the functional space \( V \) depend on the type of boundary condition in (1.1).

One popular type of boundary condition is the Dirichlet boundary condition when \( B^k = B^k_D \) are given by the following Dirichlet type trace operators:
\[
B^k_D(u) : = \frac{\partial^k u}{\partial \nu^k} \bigg|_{\partial \Omega} \quad (0 \leq k \leq m - 1),
\] (1.7)
with \( \nu \) being the outward unit normal vector of \( \partial \Omega \).

For the aforementioned Dirichlet boundary condition, the elliptic boundary value problem (1.1) is equivalent to (1.5) or (1.6) with \( V = \mathcal{H}_m^0(\Omega) \) and
\[
a(u,v) := \sum_{|\alpha|=m} (a_\alpha \partial^\alpha u, \partial^\alpha v)_{0, \Omega} + (a_0 u, v), \quad \forall u, v \in V,
\] (1.8)
and
\[
J(v) = \frac{1}{2} a(v,v) - \int_\Omega f v dx.
\] (1.9)

Other boundary conditions such as the Neumann boundary and mixed boundary conditions are a little bit complicated to describe for the general case when \( m \geq 2 \) and will be discussed later.

### 1.2 A brief overview of existing methods

Here we briefly review some classic finite elements and other relevant methods for the numerical solution of elliptic boundary value problems (1.1) for all \( d, m \geq 1 \).

Classic finite element methods use piecewise polynomial functions based on a given subdivision, namely a finite element grid, of the domain, to discretize the variational problem. We will mainly review three different types of finite element methods: (1) the conforming element method; (2) the nonconforming and discontinuous Galerkin method; and (3) the virtual element method.
**Conforming finite element method.** Given a finite element grid, this type of method is to construct \( V_h \subset V \) and find \( u_h \in V_h \) such that

\[
J(u_h) = \min_{v_h \in V_h} J(v_h).
\] (1.10)

It is well-known that a piecewise polynomial space \( V_h \subset H^m(\Omega) \) if and only if \( V_h \subset C^{m-1}(\bar{\Omega}) \). For \( m = 1 \), the piecewise linear finite element space \( V_h \subset H^1(\Omega) \) can be easily constructed on simplicial finite element grids in any dimension \( d \geq 1 \). The construction and analysis of the linear finite element method for \( m = 1 \) and \( d = 2 \) can be traced back to [29]. The situation becomes complicated when \( m \geq 2 \) and \( d \geq 2 \).

For example, it has been proved that the construction of an \( H^2 \)-conforming finite element space requires the use of polynomials of at least degree five in two dimensions [68] and degree nine in three dimensions [45]. We refer to [4] for the classic quintic \( H^2 \)-Argyris element in two dimensions and to [69] for the ninth-degree \( H^2 \)-element in three dimensions.

Many other efforts have been made in the literature to construct \( H^m \)-conforming finite element spaces. [11] proposed 2D simplicial \( H^m \) conforming elements \((m \geq 1)\) by using the polynomial spaces of degree \( 4m - 3 \), which are a generalization of the \( H^2 \) Argyris element (cf. [4, 19]) and the \( H^3 \) Čebyšev element (cf. [68]). Again, the degree of polynomials used is quite high. For (1.1), an alternative in 2D is to use mixed methods based on the Helmholtz decompositions for tensor-valued functions (cf. [60]). However, the general construction of \( H^m \)-conforming elements in any dimension is still an open problem.

We note that the construction of the conforming finite element space depends on the structure of the underlying grid. For example, one can construct relatively low-order \( H^2 \) finite elements on grids with special structures. Examples include the (quadratic) Powell-Sabin, (cubic) Clough-Tocher elements in two dimensions [20, 57], and the (quintic) Alfeld splits in three dimensions [2], where full-order accuracy, namely \( O(h), O(h^2) \), and \( O(h^4) \) accuracy can be estimated. For more recent developments on Alfeld splits, we refer to [31] and references cited therein. But these constructions do not apply to general grids. For example, de Boor-DeVore-Höllig [9, 10] showed that the \( H^2 \) element that consists of piecewise cubic polynomials on a uniform grid sequence would not provide full approximation accuracy. This gives us a hint that the structure of the underlying grid plays an important role in constructing the \( H^m \)-conforming finite element.

**Nonconforming finite element and discontinuous Galerkin methods.** Given a finite element grid \( T_h \), compared to the conforming method, the nonconforming finite element method does not require that \( V_h \subset V \), namely \( V_h \not\subset V \). We find \( u_h \in V_h \) such that

\[
J_h(u_h) = \min_{v_h \in V_h} J_h(v_h)
\] (1.11)

with

\[
J_h(v_h) = \sum_{k \in T_h} J_k(v_h) = \sum_{k \in T_h} \frac{1}{2} \int_k \left( \sum_{|\alpha|=m} a_{k,\alpha} \partial^\alpha v_h^2 + a_0 |v_h|^2 \right) dx - \int_k f v_h dx.
\]
One interesting example of the nonconforming element for (1.1) is the Morley element \([55]\) for \(m = d = 2\), which uses piecewise quadratic polynomials. For \(m \leq d\), Wang and Xu \([64]\) provided a universal construction and analysis for a family of nonconforming finite elements consisting of piecewise polynomials of minimal order for (1.1) on \(\mathbb{R}^d\) simplicial grids. The elements in \([64]\), now known as MWX-elements in the literature, gave a natural generalization of the classic Morley element to the general case \(1 \leq m \leq d\). Recently, there have been reported a number of results on the extension of MWX-elements. \([66]\) enriched the \(P_m\) polynomial space by \(P_{m+1}\) bubble functions to obtain a family of \(H^m\) nonconforming elements when \(m = d+1\). \([41]\) applied the full \(P_d\) polynomial space for the construction of the nonconforming element when \(m = 3, d = 2\), which has three more degrees of freedom locally than the element in \([66]\). They also used the full \(P_{2m-3}\) polynomial space for the nonconforming finite element approximations when \(m \geq 4, d = 2\).

In addition to the aforementioned conforming and nonconforming finite element methods, the discontinuous Galerkin (DG) method that makes use of piecewise polynomials but with globally discontinuous finite element functions have also been used for solving high-order partial differential equations, c.f. \([5]\). The DG method requires the use of many stabilization terms and parameters and the number of stabilization terms and parameters naturally grow as the order of PDE grows. To reduce the extent of stabilization needed, one approach is to introduce some continuity and smoothness in the discrete space to replace the totally discontinuous spaces. Examples for such an approach include the \(C^0\)-interior penalty DG methods for the fourth-order elliptic problem by Brener and Sung \([12]\) and for the sixth-order elliptic equation by Gudi and Neilan \([34]\). More recently, Wu and Xu \([65]\) provided a family of interior penalty nonconforming finite element methods for (1.1) in \(\mathbb{R}^d\), for any \(m \geq 0, d \geq 1\). This family of elements recover the MWX-elements in \([64]\) when \(m \leq d\), which does not require any stabilization.

**Virtual finite element.** The classic definition of finite element methods \([19]\) based on the finite element triple can be extended in many different ways. One successful extension is the virtual element method (VEM) in which general polygons or polyhedrons are used as elements and non-polynomial functions are used as shape functions. For \(m = 1\), we refer to \([7]\) and \([13]\). For \(m = 2\), we refer to \([14]\) on conforming virtual element methods for plate-bending problems, and to \([3]\) on nonconforming virtual element methods for biharmonic problems. For general \(m \geq 1\), we refer to \([16]\) for nonconforming elements that extend the MWX-elements in \([64]\) from simplicial elements to polyhedral elements.

### 1.3 Objectives

The deep neural network (DNN) is a tool developed for machine learning \([33]\). DNN provides a very special function class that has been used for the numerical solution of partial differential equations, c.f. \([44]\). By using different activation functions such as sigmoidal, the DNN can give rise to a very wide range of functional classes that can be drastically different from the piecewise polynomial of function classes used in the
classic finite element method. One advantage of the DNN approach is that it is quite easy to obtain smooth, namely $H^m$-conforming for any $m \geq 1$, DNN functions by simply choosing smooth activation functions. These function classes, however, do not usually form a linear vector space and hence the usual variational principle in the classic finite element method cannot be applied easily and instead collocation-type methods are often used. DNN is known to have much less “curse of dimensionality” than the traditional functional classes (such as polynomials or piecewise polynomials), so the DNN-based method is potentially efficient to high dimensional problems and has been studied, for example, in [24] and [63].

One main motivation of this paper is to explore the DNN-type methods that are most closely related to the traditional finite element methods. Specifically we are interested in DNN function classes that consist of piecewise polynomials. By exploring the relationship between DNN and FEM, we hope, on the one hand, to expand or extend the traditional FEM approach by using new tools from DNN, and, on the other hand, to gain and develop theoretical insights into and algorithmic tools for DNN by combining the rich mathematical theories and techniques in FEM.

In an earlier work [37], we studied the relationship between DNN using ReLU as the activation function and continuous piecewise linear functions. One conclusion that can be drawn from [37] is that any ReLU-DNN function is an $H^1$-conforming linear finite element function, and vice versa. The current work can be considered an extension of [37] by considering using ReLU$^k$-DNN for high-order partial differential equations. One focus in the current work is to provide error estimates when ReLU$^k$-DNN is applied to solve high-order partial differential equations. More specifically, we will study a special class of $H^m$-conforming generalized finite element methods (consisting of piecewise polynomials) for (1.1) for any $m \geq 1$ and $d \geq 1$ based on the artificial neural network for the numerical solution of the arbitrarily high-order elliptic boundary value problem (1.1) and then provide convergence analysis for this method. For this type of method, the underlying finite element grids are not given a priori and the discrete solution can be obtained for solving a non-linear and non-convex optimization problem. In the case that the boundary of $\Omega \subset \mathbb{R}^d$, namely $\partial \Omega$, is curved, it is often an issue as to how to find a good finite element grid to accurately approximate $\partial \Omega$. As it turns out, this is not an issue for the finite neuron method, which is probably one of the advantages of the finite neuron method analyzed in this paper.

We note that the numerical method studied in this paper for elliptic boundary value problems is closely related to the classic finite element method, that is, the numerical method studied herein amounts to piecewise polynomials with respect to an implicitly defined grid. We can also argue that it can be viewed as a mesh-less or even vertex-less method. But in comparison with the popular meshless method, this method does correspond to some underlying grid, although this grid is not given a priori. This underlying grid is determined by the artificial neurons, which mathematically speaking refer to hyperplanes $\omega_1 \cdot x + b_1 = 0$, together with a given activation function. Combining the names for the finite element method and the artificial neural network, for convenience of
exposition, we will refer to the method studied in this paper as the finite neuron method.

The rest of the paper is organized as follows. In Section 2, we describe the Monte-Carlo sampling technique, the stratified sampling technique, and the spectral Barron space. In Section 3, we construct the finite neuron functions and prove their approximation properties. In Section 4, we analyze the adaptivity and spectral accuracy property for deep finite neuron networks. In Section 5, we propose the finite neuron method and provide the convergence analysis. Finally, in Section 6, we present a summary and discussion of the results in this paper.

Following [67], we will use the notation "\( x \lesssim y \)" to denote "\( x \leq Cy \)" for some constant \( C \) independent of crucial parameters such as mesh size.

2 Preliminaries

In this section, for clarity of exposition, we present some standard materials from statistics about Monte-Carlo sampling, stratified sampling, and their applications to analysis of the asymptotic approximation properties of neural network functions.

2.1 Monte-Carlo and stratified sampling techniques

Let \( \lambda \geq 0 \) be a probability density function on a domain \( G \subset \mathbb{R}^D (D \geq 1) \) such that

\[
\int_G \lambda(\theta) d\theta = 1. \tag{2.1}
\]

We define the expectation and variance as

\[
E_g := \int_G g(\theta) \lambda(\theta) d\theta, \quad \mathbb{V}_g := E((g - E_g)^2) = E(g^2) - (E_g)^2. \tag{2.2}
\]

We note that

\[
\mathbb{V}_g \leq \max_{\theta, \theta' \in G} \left( g(\theta) - g(\theta') \right)^2.
\]

For any subset \( G_i \subset G \), let

\[
\lambda(G_i) = \int_{G_i} \lambda(\theta) d\theta, \quad \lambda_i(\theta) = \frac{\lambda(\theta)}{\lambda(G_i)}.
\]

It holds that

\[
E_{G_i}g = \sum_{i=1}^{M} \lambda(G_i) E_{G_i}g.
\]

For any function \( h(\theta_1, \cdots, \theta_N) : G_1 \times G_2 \cdots G_N \rightarrow \mathbb{R} \), define

\[
E_{G_i}g = \int_{G_i} g(\theta) \lambda_i(\theta) d\theta
\]
Lemma 2.1. For any $g \in L^\infty(G)$, we have

$$
\mathbb{E}_N \left( \mathbb{E}g - \frac{1}{N} \sum_{i=1}^N g(\omega_i) \right)^2 = \begin{cases}
\frac{1}{N} \mathbb{V}(g) \leq \frac{1}{N} \sup_{\omega, \omega' \in G} |g(\omega) - g(\omega')|^2,

\frac{1}{N} \left( \mathbb{E}(g^2) - (\mathbb{E}(g))^2 \right) \leq \frac{1}{N} \mathbb{E}(g^2) \leq \frac{1}{N} \|g\|_*^2.
\end{cases}
$$

Stratified sampling [8] gives a more refined version of the Monte Carlo method.

Lemma 2.2. For any nonoverlapping decomposition $G = G_1 \cup G_2 \cup \cdots \cup G_M$ and positive integer $n$, let $n_i = \lceil \lambda(G_i)n \rceil$ be the smallest integer larger than $\lambda(G_i)n$ and $N = \sum_{i=1}^M n_i$. Let $\theta_{ij} \in G_i$ ($1 \leq j \leq n_i$) and

$$
g_n = \sum_{i=1}^M \lambda(G_i)g_{n_i} \quad \text{with} \quad g_{n_i} = \frac{1}{n_i} \sum_{j=1}^{n_i} g(\theta_{ij}).
$$

It holds that

$$
\mathbb{E}_N(\mathbb{E}g - g_n)^2 = \sum_{i=1}^M \frac{\lambda^2(G_i)}{n_i} \mathbb{E}_{G_i}(g - \mathbb{E}_{G_i}g)^2 \leq \frac{1}{n} \max_{1 \leq i \leq M, \theta, \theta' \in G_i} |g(\theta) - g(\theta')|^2.
$$

Lemma 2.1 and Lemma 2.2 represent two simple identities and subsequent inequalities that can be verified by a direct calculation. Actually, Lemma 2.1 is a special case of Lemma 2.2 with $M = 1$. Lemma 2.1 and Lemma 2.2 are the basis of Monte-Carlo sampling and stratified sampling in statistics. In the presentation of this paper, we choose not to use any concepts related to random sampling.

Given another domain $\Omega \subset \mathbb{R}^d$, we consider the case that $g(x, \theta)$ is a function of both $x \in \Omega$ and $\theta \in G$. Given any function $\rho \in L^1(G)$, we consider

$$
u(x) = \int_G g(x, \theta) \rho(\theta) d\theta
$$

with $\|\rho\|_{L^1(G)} < \infty$. Let $\lambda(\theta) = \frac{\rho(\theta)}{\|\rho\|_{L^1(G)}}$. Thus,

$$
u(x) = \|\rho\|_{L^1(G)} \int_G g(x, \theta) \lambda(\theta) d\theta
$$

with $\|\lambda(\theta)\|_{L^1(G)} = 1$.

We can apply the above two lemmas to the given function $\nu(x)$. 

Lemma 2.3. [Monte-Carlo Sampling] Consider \( u(x) \) in (2.8) with \( 0 \leq \rho(\theta) \in L^1(G) \). For any \( N \geq 1 \), there exist \( \{\theta^*_i\}_{i=1}^N \) such that

\[
\|u-u_N\|^2_{L^2(\Omega)} \leq \frac{1}{N} \int_G \|g(\cdot,\theta)\|^2_{L^2(\Omega)} \rho(\theta) d\theta = \frac{\|\rho\|^2_{L^1(G)}}{N} \mathbb{E}(\|g(\cdot,\theta)\|^2_{L^2(\Omega)}),
\]

where \( \|g(\cdot,\theta)\|^2_{L^2(\Omega)} = \int_\Omega |g(x,\theta)|^2 d\mu(x) \).

Similarly, if \( g(\cdot,\theta) \in H^m(\Omega), \) for any \( N \geq 1 \), there exist \( \{\theta^*_i\}_{i=1}^N \) with \( f_N \) given in (2.10) such that

\[
\|u-u_N\|^2_{H^m(\Omega)} \leq \int_G \|g(\cdot,\theta)\|^2_{H^m(\Omega)} \rho(\theta) d\theta = \frac{\|\rho\|^2_{L^1(G)}}{N} \mathbb{E}(\|g(\cdot,\theta)\|^2_{H^m(\Omega)}).
\]

Proof. Note that

\[
u(x) = \|\rho\|_{L^1(G)} \mathbb{E}(g).
\]

By Lemma 2.1,

\[
\mathbb{E}_n \left( \left( \mathbb{E}(g(x,\cdot)) - \frac{1}{N} \sum_{i=1}^N g(x,\theta_i) \right)^2 \right) \leq \frac{1}{N} \mathbb{E}(g^2).
\]

By taking integration w.r.t. \( x \) on both sides, we get

\[
\mathbb{E}_n (h(\theta_1,\theta_2,\ldots,\theta_N)) \leq \frac{1}{N} \mathbb{E} \left( \int_\Omega g^2 d\mu(x) \right),
\]

where

\[
h(\theta_1,\theta_2,\ldots,\theta_N) = \int_\Omega \left( \mathbb{E}(g(x,\cdot)) - \frac{1}{N} \sum_{i=1}^N g(x,\theta_i) \right)^2 d\mu(x).
\]

As \( \mathbb{E}_N(1) = 1 \) and \( \mathbb{E}_N(h) \leq \frac{1}{N} \mathbb{E}(\int_\Omega g^2 d\mu(x)) \), there exist \( \{\theta^*_i\}_{i=1}^N \) such that

\[
h(\theta^*_1,\theta^*_2,\ldots,\theta^*_N) \leq \frac{1}{N} \int_\Omega \mathbb{E}(g^2) d\mu(x).
\]

This implies that

\[
\mathbb{E}_n \|u-u_N\|^2_{L^2(\Omega)} \leq \frac{\|\rho\|^2_{L^1(G)}}{N} \int_\Omega \|g(\cdot,\theta)\|^2_{L^2(\Omega)} \lambda(\theta) d\theta.
\]

The proof for (2.11) is similar to the above analysis for the \( L^2 \)-error analysis, which completes the proof. \( \square \)
Lemma 2.4. [Stratified Sampling] For \( u(x) \) in (2.8) with positive \( \rho(\theta) \in L^1(G) \), given any positive integers \( n \) and \( M \leq n \), for any nonoverlapping decomposition \( G = G_1 \cup G_2 \cup \cdots \cup G_M \), there exist \( \{ \theta_i^* \}_{i=1}^N \) with \( n \leq N \leq 2n \) such that

\[
\| u - u_N \|_{L^2(\Omega)} \leq N^{-1/2} \| \rho \|_{L^1(G)} \max_{1 \leq i \leq M} \sup_{\theta_j, \theta_j' \in G_j} \| g(x, \theta_j) - g(x, \theta_j') \|_{L^2(\Omega)},
\]

(2.13)

where

\[
u_N(x) = \frac{2\| \rho \|_{L^1(G)}}{N} \sum_{i=1}^N \beta_i g(x, \theta_i^*)
\]

and \( \beta_i \in [0, 1] \).

Proof. Let \( n_j = \lceil \lambda(G_j)n \rceil \) and \( \theta_{ij} \in G_j \) \( (1 \leq i \leq n_j) \). Define \( N = \sum_{j=1}^M n_j \) and

\[
u_N(x) = \| \rho \|_{L^1(G)} \sum_{j=1}^M \lambda(G_j) g_{n_j} \text{ with } g_{n_j} = \frac{1}{n_j} \sum_{i=1}^{n_j} g(\theta_{ij}).
\]

As \( u(x) = \| \rho \|_{L^1(G)} \sum_{j=1}^M \lambda(G_j) E_{G_j} g_j \) by Lemma 2.1,

\[
\mathbb{E}_n \| u - u_N \|_{L^2(\Omega)}^2 = \| \rho \|_{L^1(G)}^2 \sum_{j=1}^M \frac{\lambda^2(G_j)}{n_j} \mathbb{E}_G \| g_j - g \|_{L^2(\Omega)}^2
\leq \| \rho \|_{L^1(G)}^2 \sum_{j=1}^M \frac{\lambda^2(G_j)}{n_j} \sup_{\theta_j, \theta_j' \in G_j} \| g(x, \theta_j) - g(x, \theta_j') \|_{L^2(\Omega)}^2.
\]

(2.14)

As \( \frac{\lambda(G_j)}{n_j} \leq \frac{1}{n} \) and \( \sum_{j=1}^M \lambda(G_j) = 1 \),

\[
\mathbb{E}_n \| u - u_N \|_{L^2(\Omega)}^2 \leq n^{-1} \| \rho \|_{L^1(G)}^2 \max_{1 \leq i \leq M} \sup_{\theta_j, \theta_j' \in G_j} \| g(x, \theta_j) - g(x, \theta_j') \|_{L^2(\Omega)}^2.
\]

(2.15)

There exists \( \{ \theta_{ij}^* \} \) such that \( \theta_{ij}^* \in G_i \) and

\[
\| u - u_N \|_{L^2(\Omega)}^2 \leq n^{-1} \| \rho \|_{L^1(G)}^2 \max_{1 \leq i \leq M} \sup_{\theta_j, \theta_j' \in G_j} \| g(x, \theta_j) - g(x, \theta_j') \|_{L^2(\Omega)}^2.
\]

(2.16)

Note that \( n \leq N \leq n + M \leq 2n \),

\[
u_N(x) = \frac{2\| \rho \|_{L^1(G)}}{N} \sum_{j=1}^M \frac{\lambda(G_j)}{2n_j} \sum_{i=1}^{n_j} g(x, \theta_{ij}^*) = \frac{2\| \rho \|_{L^1(G)}}{N} \sum_{j=1}^M \sum_{i=1}^{n_j} \beta_{ij} g(x, \theta_{ij}^*)
\]

with

\[
\beta_{ij} = \frac{\lambda(G_j)}{2n_j} \leq \frac{2\lambda(G_j)n}{2\lambda(G_j)n} \leq 1,
\]

(2.17)

which completes the proof. \( \square \)
2.2 Spectral Barron space

Let us use a simple example to motivate the spectral Barron space. Consider a bounded domain $\Omega \subset \mathbb{R}^d$ and a real function $u \in L^1(\Omega)$. The Fourier transform of $u$ is

$$\hat{u}(\omega) = (2\pi)^{-d/2} \int_{\Omega} e^{-i\omega \cdot x} u(x) dx,$$

where $u \in L^1(\mathbb{R})$. This gives the following integral representation of $u$ in terms of cosine function

$$u(x) = \text{Re} \int_{\mathbb{R}^d} e^{i\omega \cdot x} \hat{u}(\omega) d\omega = \int_{\mathbb{R}^d} \cos(\omega \cdot x + b(\omega)) |\hat{u}(\omega)| d\omega,$$

where $\hat{u}(\omega) = e^{ib(\omega)}|\hat{u}(\omega)|$. Let

$$g(x,\omega) = \cos(\omega \cdot x + b(\omega)) \quad \text{and} \quad \rho(\omega) = |\hat{u}(\omega)|.$$

Thus,

$$u(x) = \int_{\mathbb{R}^d} g(x,\omega) \rho(\omega) d\omega.$$

If

$$\int_{\mathbb{R}^d} |\hat{u}(\omega)| d\omega < \infty,$$

then $\|\rho\|_{L^1} < \infty$. With the application of the Lemma 2.3, there exists $\omega_i \in \mathbb{R}^d$ such that

$$\|u - u_N\|_{0,\Omega} \leq N^{-1/2} \|\hat{u}\|_{L^1(\mathbb{R}^d)},$$

where

$$u_N(x) = \frac{\|\hat{u}\|_{L^1(\mathbb{R}^d)}}{N} \sum_{i=1}^{N} \cos(\omega_i \cdot x + b(\omega_i)).$$

More generally, we consider the approximation property in the $H^m$-norm. By (2.19),

$$\partial^\alpha u(x) = \int_{\mathbb{R}^d} \cos|\alpha| (\omega \cdot x + b(\omega)) \omega^\alpha |\hat{u}(\omega)| d\omega, \quad \forall \ |\alpha| \leq m.$$

For any positive integer $m$, let

$$g_m(x,\omega) = \frac{\cos(\omega \cdot x + b(\omega))}{(1 + \|\omega\|)^m} \quad \text{and} \quad \rho_m(\omega) = (1 + \|\omega\|)^m |\hat{u}(\omega)|,$$

where

$$\|\rho_m\|_{L^1(\mathbb{R}^d)} = \int_{\mathbb{R}^d} (1 + \|\omega\|)^m |\hat{u}(\omega)| d\omega < \infty.$$

Then, $u(x) = \int_{\mathbb{R}^d} g_m(x,\omega) \rho_m d\omega = \|\rho_m\|_{L^1(\mathbb{R}^d)} \mathbb{E} g_m(x,\omega)$. Define

$$u_N(x) = \frac{\|\rho_m\|_{L^1(\mathbb{R}^d)}}{N} \sum_{i=1}^{N} g_m(x,\omega_i) = \frac{\|\rho_m\|_{L^1(\mathbb{R}^d)}}{N} \sum_{i=1}^{N} \frac{\cos(\omega_i \cdot x + b(\omega_i))}{(1 + \|\omega_i\|)^m}.$$
It holds that
\[ \partial^\alpha (u(x) - u_N(x)) = \frac{\| \rho_m \|_{L^1(\mathbb{R}^d)}}{N} \sum_{i=1}^{N} \mathbb{E} \partial^{\alpha} (g_m(x, \omega) - g_m(x, \omega_i)). \]

By Lemma 2.1,
\[ \mathbb{E}_N \sum_{|\alpha| \leq m} \| \partial^\alpha (u(x) - u_N(x)) \|^2_{0, \Omega} \]
\[ \leq \| \rho_m \|_{L^1(\mathbb{R}^d)}^2 \mathbb{E}_N \sum_{|\alpha| \leq m} \frac{1}{N^2} \sum_{i=1}^{N} (\mathbb{E} \partial^{\alpha} (g_m(x, \omega) - g_m(x, \omega_i)))^2 \]
\[ \leq \frac{\| \rho_m \|_{L^2}^2}{N} \sum_{|\alpha| \leq m} \mathbb{E} (\partial^{\alpha} g_m(x, \omega))^2. \] (2.27)

Note that the definitions of \( g_m \) and \( \rho_m \) in (2.25) guarantee that \( |\partial^\alpha g_m(x, \omega)| \leq 1. \)

Thus,
\[ \mathbb{E}_N \sum_{|\alpha| \leq m} \| \partial^\alpha (u(x) - u_N(x)) \|^2_{0, \Omega} \leq \frac{\| \rho_m \|_{L^2}^2}{N}. \]

This implies that there exists \( \omega_i \in \mathbb{R}^d \) such that
\[ \| u - u_N \|_{H^m(\Omega)} \lesssim N^{-1/2} \int_{\mathbb{R}^d} (1 + \| \omega \|)^m |\hat{u}(\omega)| d\omega. \] (2.28)

Given \( v \in L^2(\Omega) \), consider all the possible extensions \( v_E : \mathbb{R}^d \mapsto \mathbb{R} \) with \( v_E|_{\Omega} = v \) and define the spectral Barron norm for any \( s \geq 1 \):
\[ \| v \|_{B^s (\Omega)} = \inf_{v_E|_{\Omega} = v} \int_{\mathbb{R}^d} (1 + \| \omega \|)^s |\partial_E(\omega)| d\omega \] (2.29)

and spectral Barron space
\[ B^s (\Omega) = \{ v \in L^2 (\Omega) : \| v \|_{B^s (\Omega)} < \infty \}. \] (2.30)

In summary, we have
\[ \| u - u_N \|_{H^m(\Omega)} \lesssim N^{-1/2} \| u \|_{B^m(\Omega)}. \] (2.31)

The estimate of (2.22), first obtained in [42] using a slightly different technique, appears to be the first asymptotic error estimate for the artificial neural network. [6] extended Jones’s estimate (2.22) to sigmoidal-type activation functions in place of cosine.

The above short discussions reflect the core idea in the analysis of the approximation property of artificial neural networks. Namely, represent \( f \) as an expectation of some
probability distribution as in (2.21) and then a simple application of Monte-Carlo sampling leads to an error estimate like (2.28) for a special neural network function given by (2.23) using $\sigma = \cos$ as an activation function. For a more general activation function $\sigma$, we just need to derive a corresponding representation like (2.21) with $g$ in terms of $\sigma$. Quantitative estimates on the order of approximation are obtained for sigmoidal activation functions in [6] and for periodic activation functions in [51, 52]. Error estimates in Sobolev norms for general activation functions can be found in [39]. A review of a variety of known results, especially for networks with one hidden layer, can be found in [56]. More recently, these results have been improved by a factor of $n^{1/d}$ in [43] using the idea of stratified sampling, based in part on the techniques in [49]. [62] provides an analysis for general activation functions under very weak assumptions, which applies to essentially all activation functions used in practice. In [25–27], a more refined definition of the spectral Barron norm is introduced to give sharper approximation error bounds of neural networks.

The following lemma shows some relationship between the Sobolev norm and the spectral Barron norm.

**Lemma 2.5.** Let $m \geq 0$ be an integer and $\Omega \subset \mathbb{R}^d$ a bounded domain. Then for any Schwartz function $v$, we have

$$
\|v\|_{H^m(\Omega)} \lesssim \|v\|_{B^m(\Omega)} \lesssim \|v\|_{H^{m+d/2+\epsilon}((\Omega))},
$$

where $\epsilon$ is positive.

**Proof.** The first inequality in (2.32) and its proof can be found in [62]. A version of the second inequality in (2.32) and its proof can be found in [6]. Below is a proof, by definition and Cauchy-Schwarz inequality,

$$
\|v\|_{B^m(\Omega)} = \inf_{\varphi \in \mathcal{P}} \left( \int_{\mathbb{R}^d} (1 + \|\omega\|)^m |\hat{\varphi}(\omega)|^2 d\omega \right)^{1/2} \\
\leq \int_{\mathbb{R}^d} (1 + \|\omega\|)^{-d-2\epsilon} d\omega \inf_{\varphi \in \mathcal{P}} \int_{\mathbb{R}^d} (1 + \|\omega\|)^{d+2m+2\epsilon} |\hat{\varphi}(\omega)|^2 d\omega \\
\lesssim \inf_{\varphi \in \mathcal{P}} \int_{\mathbb{R}^d} (1 + \|\omega\|)^{d+2m+2\epsilon} |\hat{\varphi}(\omega)|^2 d\omega \lesssim \|v\|_{H^{m+d/2+\epsilon}((\Omega))}.
$$

This completes the proof. \[\square\]

3 Finite neuron functions and approximation properties

As mentioned before, for $m = 1$, the finite element for (5.1) can be given by the piecewise linear function in any dimension $d \geq 1$. As shown in [37], the linear finite element function can be represented by deep neural networks with ReLU as activation functions. Here,

$$
\text{ReLU}(x) = x_+ = \max(0,x).
$$

(3.1)
In this paper, we will consider the power of ReLU as activation functions

$$\text{ReLU}^k(x) = [x_+]^k = \left[ \max(0, x) \right]^k.$$  \hfill (3.2)

We will use the short-hand notation $x^k_+ = [x_+]^k$ in the rest of the paper.

We consider the following neuron network function class with one hidden layer:

$$V^k_N = \left\{ \sum_{i=1}^{N} a_i (w_i \cdot x + b_i)^k_+, \ a_i, b_i \in \mathbb{R}^1, \ w_i \in \mathbb{R}^{1 \times d} \right\}. \hfill (3.3)$$

We note that $V^k_N$ is not a linear vector space. The definition of the neural network function class such as (3.3) can be traced back to [50], and its early mathematical analysis can be found in [22, 32, 38].

The functions in $V^k_N$ as defined in (3.3) will be known as finite neuron functions in this paper.

**Lemma 3.1.** For any $k \geq 1$, $V^k_N$ consists of functions that are piecewise polynomials of degree $k$ with respect to a grid whose boundaries are given by the intersection of the hyperplanes

$$w_i x + b_i = 0, \quad 1 \leq i \leq N,$$

see Fig. 1 and Fig. 3. Furthermore, if $k \geq m$,

$$V^k_N(\Omega) \subset H^k(\Omega) \subset H^m(\Omega),$$

where $\Omega$ is a bounded domain in $\mathbb{R}^d$.

The main goal of this section is to prove that the following type of error estimate holds, for some $\delta \geq 0$,

$$\inf_{v_N \in V^k_N} \|u - v_N\|_{H^m(\Omega)} \lesssim N^{-\frac{k}{2} - \delta}. \hfill (3.4)$$
We will use two different approaches to establish (3.4). The first approach, presented in Section 3.1, mainly follows [39] and [62] to give error estimates for a general class of activation functions. The second approach, presented in Section 3.2, follows [43] to give error estimates specifically for the ReLU activation function.

We assume that $\Omega \subset \mathbb{R}^d$ is a given bounded domain. Thus,

$$T = \max_{x \in \Omega} \|x\| < \infty. \quad (3.5)$$

### 3.1 B-spline as activation functions

The activation function $[\text{ReLU}]^k$ (3.2) is related to cardinal B-splines. A cardinal B-spline of degree $k \geq 0$ denoted by $b^k$ is defined by convolution as

$$b^k(x) = (b^{k-1} \ast b^0)(x) = \int_{\mathbb{R}} b^{k-1}(x-t)b^0(t)dt, \quad (3.6)$$

where

$$b^0(x) = \begin{cases} 1, & x \in [0,1), \\ 0, & \text{otherwise}. \end{cases} \quad (3.7)$$

More explicitly, see [23], for any $x \in [0,k+1]$ and $k \geq 1$, we have

$$b^k(x) = \frac{x}{k} b^{k-1}(x) + \frac{k+1-x}{k} b^{k-1}(x-1), \quad (3.8)$$

or

$$b^k(x) = (k+1) \sum_{i=0}^{k+1} w_i (i-x)_+^k \quad \text{and} \quad w_i = \prod_{j=0, j \neq i}^{k+1} \frac{1}{l-1}. \quad (3.9)$$

We note that all $b^k$ are locally supported (see Fig. 2 for their plots).

For a uniform grid with mesh size $h = \frac{1}{n+1}$, we define

$$b^k_{j,h}(x) = b^k\left(\frac{x}{h} - j\right). \quad (3.10)$$

Then the cardinal B-Spline series of degree $k$ on the uniform grid is

$$S^k_N = \left\{ v(x) = \sum_{j=-k}^{N} c_j b^k_{j,h}(x) \right\}. \quad (3.11)$$

**Lemma 3.2.** For $V^k_N$ and $S^k_N$ defined by (3.3) and (3.11), we have

$$S^k_N \subset V^k_{N+k+1}. \quad (3.12)$$

As a result, for any bounded domain $\Omega \subset \mathbb{R}^1$, we have

$$\inf_{v \in V^k_N} \|u - v\|_{m,\Omega} \leq \inf_{v \in S^k_{N+k+1}} \|u - v\|_{m,\Omega} \lesssim N^{m-(k+1)}\|u\|_{k+1,\Omega}. \quad (3.13)$$
Given an activation function \( \sigma \in L^1(\mathbb{R}) \), consider its Fourier transformation:

\[
\hat{\sigma}(a) = \frac{1}{2\pi} \int_{\mathbb{R}} \sigma(t) e^{-iat} dt. \tag{3.14}
\]

For any \( a \neq 0 \) with \( \hat{\sigma}(a) \neq 0 \), with a change of variables \( t = a^{-1} \omega \cdot x + b \) and \( dt = db \), we have

\[
\hat{\sigma}(a) = \frac{1}{2\pi} \int_{\mathbb{R}} \sigma(a^{-1} \omega \cdot x + b) e^{-ia(a^{-1} \omega \cdot x + b)} db = e^{-ia} \frac{1}{2\pi} \int_{\mathbb{R}} \sigma(a^{-1} \omega \cdot x + b) e^{-iab} db. \tag{3.15}
\]

This implies that

\[
e^{ia \omega \cdot x} = \frac{1}{2\pi \hat{\sigma}(a)} \int_{\mathbb{R}} \sigma(a^{-1} \omega \cdot x + b) e^{-iab} db. \tag{3.16}\]

We write \( \hat{u}(\omega) = e^{-ib(\omega)} |\hat{u}(\omega)| \) and then obtain this integral representation:

\[
u(x) = \int_{\mathbb{R}^d} e^{i\omega \cdot x} \hat{u}(\omega) d\omega = \int_{\mathbb{R}^d} \int_{\mathbb{R}} \frac{1}{2\pi \hat{\sigma}(a)} \sigma(a^{-1} \omega \cdot x + b) |\hat{u}(\omega)| e^{-i(ab + \theta(\omega))} db d\omega. \tag{3.17}\]

Now we consider activation function \( \sigma(x) = b^k(x) \) and \( \hat{\sigma} \) as the Fourier transformation of \( \sigma(x) \). Note that, by (2),

\[
\hat{\sigma}(a) = \left(1 - e^{-ia} \right)^{k+1} = \left(\frac{2}{a} \sin \frac{a}{2}\right)^{k+1} e^{-\frac{(a(k+1))}{2}}. \tag{3.18}\]

We first take \( a = \pi \) in (3.18). Thus,

\[
\hat{\sigma}(\pi) = \left(\frac{2}{\pi}\right)^{k+1} e^{-\frac{(\pi(k+1))}{2}}. \tag{3.19}\]

Combining (3.17) and (3.19), we obtain that

\[
u(x) = \frac{1}{4} \left(\frac{\pi}{2}\right)^k \int_{\mathbb{R}^d} \int_{\mathbb{R}} \sigma(\pi^{-1} \omega \cdot x + b) |\hat{u}(\omega)| e^{-i(\pi b + \pi(k+1) + \theta(\omega))} db d\omega. \tag{3.20}\]

An application of the Monte Carlo method in Lemma 2.1 to the integral representation (3.20) leads to the following estimate.
Theorem 3.1. For any $0 \leq m \leq k$, if $u \in B^{m+1}(\Omega)$, there exist $\omega_i \in \mathbb{R}^d$, $b_i \in \mathbb{R}$ such that
\[
\|u - u_N\|_{H^m(\Omega)} \lesssim N^{-\frac{1}{2}} \|u\|_{B^{m+1}(\Omega)} \quad (3.21)
\]
with
\[
u_N(x) = \sum_{i=1}^{N} \beta_i b^k \left( \pi^{-1} \omega_i \cdot x + b_i \right). \quad (3.22)
\]

Based on the integral representation (3.20), a stratified analysis similar to the one in [62] leads to the following result.

Theorem 3.2. For any $0 \leq m \leq k$ and positive $\epsilon$, if $u \in B^{m+1+\epsilon}(\Omega)$, there exist $\omega_i \in \mathbb{R}^d$, $b_i \in \mathbb{R}$ such that
\[
\|u - u_N\|_{H^m(\Omega)} \leq N^{-\frac{1}{2} - \epsilon} \|u\|_{B^{m+1+\epsilon}(\Omega)} \quad (3.23)
\]
with
\[
u_N(x) = \sum_{i=1}^{N} \beta_i b^k \left( \bar{\omega}_i \cdot x + b_i \right). \quad (3.24)
\]

Next, we try to improve the estimate (3.23). Again, we will use (3.17). Let $a_\omega = 4\pi \left( \frac{\|\omega\|}{4\pi} \right) + \pi$ in (3.18) and $\bar{\omega} = \frac{\omega}{a_\omega}$. We have
\[
\hat{\sigma}(a_\omega) = \left( \frac{2}{a_\omega} \right)^{k+1}, \quad \|\omega\| + \pi \leq a_\omega \leq \|\omega\| + 5\pi, \quad \|\bar{\omega}\| \leq 1, \quad (3.25)
\]
which, together with (3.17), indicates that
\[
u(x) = \int_{\mathbb{R}^d} \int_{\mathbb{R}} \frac{1}{2\pi} \sigma (\bar{\omega} \cdot x + b) \left( \frac{a_\omega}{2} \right)^{k+1} \hat{u}(\omega) e^{-i\omega (b + \frac{k+1}{2})} db d\omega. \quad (3.26)
\]

Theorem 3.3. If $u \in B^{k+1}(\Omega)$, there exist $\|\bar{\omega}_i\| \leq 1$, $|b_i| \leq T + k + 1$ such that
\[
\|u - u_N\|_{H^{k+1}(\Omega)} \lesssim N^{-\frac{1}{2}} \|u\|_{B^{k+1}(\Omega)} \quad (3.27)
\]
with
\[
u_N(x) = \sum_{i=1}^{N} \beta_i b^k \left( \bar{\omega}_i \cdot x + b_i \right). \quad (3.28)
\]

Proof. We write (3.26) as
\[
u(x) = \int_{\mathbb{R}^d} \int_{\mathbb{R}} g(x, b, \omega) \rho(b, \omega) db d\omega
\]
with
\[
\hat{u}(\omega) = e^{-i\hat{\theta}(\omega)} |\hat{u}(\omega)|, \quad \hat{\theta}(\omega) = \theta(\omega) + a_\omega \left( b + \frac{k+1}{2} \right)
\]
\[
g(x,b,\omega) = \sigma(\tilde{\omega} \cdot x + b) \text{sgn}(\cos \tilde{\theta}(\omega)),
\]
(3.29)
\[
\rho(b,\omega) = \frac{1}{(2\pi)^d} \left( \frac{a_\omega}{2} \right)^{k+1} |\tilde{u}(\omega)||\cos \tilde{\theta}(\omega)|.
\]
(3.30)

Note that
\[
||\tilde{\omega}|| \leq 1, \quad |b| \leq T + k + 1.
\]
(3.31)

Let
\[
G = \{(\omega,b) : \omega \in \mathbb{R}^d, |b| \leq T + k + 1\}, \quad \tilde{G} = \{(\tilde{\omega},b) : ||\tilde{\omega}|| \leq 1, |b| \leq T + k + 1\}.
\]

For any positive integer \(n\), divide \(\tilde{G}\) into \(\tilde{M}\) \((\tilde{M} \leq \frac{n}{2})\) nonoverlapping subdomains, say \(\tilde{G} = \tilde{G}_1 \cup \tilde{G}_2 \cup \ldots \cup \tilde{G}_{\tilde{M}}\), such that
\[
|b-b'| \lesssim n^{-\frac{1}{2d}}, \quad |\tilde{\omega}-\tilde{\omega}'| \lesssim n^{-\frac{1}{2d}}, \quad (\tilde{\omega},b), (\tilde{\omega}',b') \in \tilde{G}_i, \quad 1 \leq i \leq \tilde{M}.
\]
(3.32)

Define \(M = 2\tilde{M}\) and for \(1 \leq i \leq \tilde{M}\),
\[
G_i = \{(\omega,b) : (\tilde{\omega},b) \in \tilde{G}_i, \cos \tilde{\theta}(\tilde{\omega}) \geq 0\}, \quad G_{\tilde{M}+i} = \{(\omega,b) : (\tilde{\omega},b) \in \tilde{G}_i, \cos \tilde{\theta}(\tilde{\omega}) \leq 0\}.
\]

Thus, \(G = G_1 \cup G_2 \cup \ldots \cup G_M\) with \(G_i \cap G_j = \emptyset\) if \(i \neq j\), and
\[
|b-b'| \lesssim n^{-\frac{1}{2d}}, \quad |\tilde{\omega}-\tilde{\omega}'| \lesssim n^{-\frac{1}{2d}}, \quad \text{sgn}(\cos \tilde{\theta}(\omega)) = \text{sgn}(\cos \tilde{\theta}(\omega')).
\]
(3.33)

Let \(n_i = \lceil \lambda(G_i)n \rceil\), \(N = \sum_{i=1}^{M} n_i\) and
\[
u_N(x) = \|\rho\|_{L^1(G)} \sum_{i=1}^{M} \frac{\lambda(G_i)}{n_i} \sum_{j=1}^{n_i} \varphi(x,\theta_{ij}).
\]
(3.34)

It holds that
\[
\mathbb{E}\left(\|u-u_N\|^2_{H^m(\Omega)}\right) \leq \|\rho\|_{L^1(G)} \sum_{i=1}^{M} \frac{\lambda^2(G_i)}{n_i} \sup_{\theta,\theta' \in G_i} \|\varphi(x,\theta) - \varphi(x,\theta')\|^2_{H^m(\Omega)}
\]
(3.35)

with \(\theta = (b,\omega)\). For any \((b,\omega) \in G_i, 1 \leq i \leq M\), if \(k \geq m+1\),
\[
|\varphi(x,\theta) - \varphi(x,\theta')| \lesssim |b-b'| + |\omega-\omega'| \lesssim n^{-\frac{1}{2d}}.\]
(3.36)

Thus,
\[
\sum_{i=1}^{M} \frac{\lambda^2(G_i)}{n_i} \sup_{\theta,\theta' \in G_i} \|\varphi(x,\theta) - \varphi(x,\theta')\|^2_{H^m(\Omega)} \lesssim n^{-\frac{2d}{2d}}|\Omega|.
\]
(3.37)
Thus,
\[ \mathbb{E} \left( \|u-u_N\|^2_{H^m(\Omega)} \right) \lesssim n^{-1-\frac{2}{d}} |\Omega| \|\rho\|_{L^1(G)}. \quad (3.38) \]

As \( a \leq \|\omega\| + 5\pi \),
\[ \|\rho\|_{L^1(G)} \lesssim \int_G (\|\omega\| + 1)^{k+1} |\hat{u}(\omega)| \, d\omega \, db \lesssim \|u\|_{B^{k+1}(\Omega)}. \quad (3.39) \]

Note that \( n \leq N \leq 2n \). Thus, there exist \( \omega_i \in \mathbb{R}^d, \beta_i, b_i \in \mathbb{R} \) such that
\[ \|u-u_N\|_{H^m(\Omega)} \lesssim N^{-\frac{1}{2}} - \frac{1}{d} \|u\|_{B^{k+1}(\Omega)}, \quad (3.40) \]

which completes the proof. \( \square \)

The above analysis can also be applied to more general activation functions with compact support.

**Theorem 3.4.** Suppose that \( \sigma \in W^{m+1,\infty}(\mathbb{R}) \) that has compact support. If for any \( a > 0 \), there exists \( \hat{a} > 0 \) such that
\[ \hat{a} \gtrsim a, \quad |\hat{\sigma}(\hat{a})| \gtrsim a^{-\ell}, \quad (3.41) \]

and \( u \in B^{\ell}(\Omega) \), then, there exist \( \omega_i \in \mathbb{R}^d \) and \( b_i \in \mathbb{R} \) such that
\[ \|u-u_N\|_{H^m(\Omega)} \lesssim N^{-\frac{1}{2}} - \frac{1}{d} \|u\|_{B^{k+1}(\Omega)}, \quad (3.42) \]

where
\[ u_N(x) = \sum_{i=1}^{N} \beta_i \sigma(\hat{\omega}_i \cdot x + b_i). \quad (3.43) \]

### 3.2 [ReLU]\(^k\) as activation functions

Rather than using the general Fourier transformation as in (3.16) to represent \( e^{i\omega \cdot x} \) in terms of \( \sigma(\omega \cdot x + b) \), [43] gave a different method to represent \( e^{i\omega \cdot x} \) in terms of \( (\omega \cdot x + b)^k \) for \( k=1 \) and \( k=2 \). The following lemma gives a generalization of this representation for all \( k \geq 0 \).

**Lemma 3.3.** For any \( k \geq 0 \) and \( x \in \Omega \),
\[ e^{i\omega \cdot x} = \sum_{j=0}^{k} \frac{(i\omega \cdot x)^j}{j!} + \frac{t^{k+1}}{k!} \|\omega\|^{k+1} \int_0^T \left[ (\omega \cdot x - t)^k e^{i|\omega|t} + (-1)^{k-1} (\omega \cdot x - t)^k e^{-i|\omega|t} \right] dt. \quad (3.44) \]
Proof. For $|z| \leq c$, by the Taylor expansion with integral remainder,

$$e^{iz} = \sum_{j=0}^{k} \frac{(iz)^j}{j!} + \frac{i^{k+1}}{k!} \int_{0}^{z} e^{iu}(z-u)^k du. \quad (3.44)$$

Note that

$$(z-u)^k = (z-u)_{+}^k - (u-z)_{+}^k.$$  

It follows that

$$\begin{align*}
\int_{0}^{c} (z-u)^k e^{iu} du &= \int_{0}^{z} (z-u)^k e^{iu} du + \int_{z}^{c} (-1)^k (u-z)^k e^{iu} du \\
&= \int_{0}^{z} (z-u)^k e^{iu} du + \int_{0}^{-z} (-1)^{k-1} (-u-z)^k e^{-iu} du \\
&= \int_{0}^{c} (z-u)^k e^{iu} du + (-1)^{k-1} (-u-z)^k e^{-iu} du. \quad (3.45)
\end{align*}$$

Thus,

$$e^{iz} - \sum_{j=0}^{k} \frac{(iz)^j}{j!} = \frac{i^{k+1}}{k!} \int_{0}^{c} \left[ (z-u)^k e^{iu} + (-1)^{k-1} (-z-u)^k e^{-iu} \right] du. \quad (3.46)$$

Let

$$z = \omega \cdot x, \quad u = \|\omega\| t, \quad \hat{\omega} = \frac{\omega}{\|\omega\|}. \quad (3.47)$$

As $\|x\| \leq T$ and $|\hat{\omega} \cdot x| \leq T$, we obtain

$$e^{i\omega \cdot x} - \sum_{j=0}^{k} \frac{(i\omega \cdot x)^j}{j!} = \frac{i^{k+1}}{k!} \|\omega\|^k \int_{0}^{T} \left[ (\omega \cdot x - t)^k e^{i\|\omega\| t} + (-1)^{k-1} (-\omega \cdot x - t)^k e^{-i\|\omega\| t} \right] dt, \quad (3.48)$$

which completes the proof. \hfill \Box

As $u(x) = \frac{1}{(2\pi)^d} \int_{\mathbb{R}^d} e^{i\omega \cdot x} \hat{u}(\omega) d\omega$ and $\partial^a u(x) = \int_{\mathbb{R}^d} i^{\|\omega\|} \omega^a e^{i\omega \cdot x} \hat{u}(\omega) d\omega$,

$$\partial^a u(0) x^a = \int_{\mathbb{R}^d} i^{\|\omega\|} \omega^a x^a \hat{u}(\omega) d\omega. \quad (3.49)$$

Note that $(\omega \cdot x)^j = \sum_{|a| = j} \frac{j!}{a!} \omega^a x^a$. It follows that

$$\sum_{|a| = j} \frac{1}{a!} \partial^a u(0) x^a = i^j \sum_{|a| = j} \frac{1}{a!} \int_{\mathbb{R}^d} \omega^a x^a \hat{u}(\omega) d\omega = \frac{1}{j!} \int_{\mathbb{R}^d} (i\omega \cdot x)^j \hat{u}(\omega) d\omega. \quad (3.50)$$
Let \( \hat{u}(\omega) = |\hat{u}(\omega)|e^{i\varphi(\omega)} \). Then, \( e^{i\varphi(\omega)} \|\hat{u}(\omega)\| = |\hat{u}(\omega)|e^{i(\varphi(\omega)+b(\omega))} \). By Lemma 3.3,

\[
    u(x) - \sum_{|a|\leq k} \frac{1}{a!} \partial^a u(0) x^a
    \]

\[
    = \int_{\mathbb{R}^d} \left( e^{i\omega \cdot x} - \sum_{j=0}^k \frac{1}{j!} (i\omega \cdot x)^j \right) \hat{u}(\omega) d\omega.
    \]

\[
    = \text{Re} \left( \frac{k+1}{k!} \int_{\mathbb{R}^d} \int_0^T \left( (\omega \cdot x - t)^k e^{i\varphi(\omega) + b(\omega)} + (-1)^{k-1} (-\omega \cdot x - t)^k e^{-i\varphi(\omega) - b(\omega)} \right) \hat{u}(\omega) \|\omega\|^{k+1} dt d\omega \right)
    \]

\[
    = \frac{1}{k!} \int_{-1,1} \int_{\mathbb{R}^d} \int_0^T (z\omega \cdot x - t)^k s(zt,\omega) |\hat{u}(\omega)||\omega|^{k+1} dt d\omega dz
    \tag{3.51}
    \]

with \( \int_{-1,1} r(z) dz = r(-1) + r(1) \) and

\[
    s(zt,\omega) = \begin{cases} 
    (-1)^{k+1} \cos(z\|\omega\|), & k \text{ is odd}, \\
    (-1)^{k+2} \sin(z\|\omega\|), & k \text{ is even}.
    \end{cases}
    \tag{3.52}
    \]

Define \( G = \{-1,1\} \times [0,T] \times \mathbb{R}^d, \theta = (z,t,\omega) \in G, \)

\[
    g(x,\theta) = (z\omega \cdot x - t)^k \text{sgn}(zt,\omega),
    \]

\[
    \rho(\theta) = \frac{1}{(2\pi)^d} |s(zt,\omega)||\hat{u}(\omega)||\omega|^{k+1}, \quad \lambda(\theta) = \frac{\rho(\theta)}{\|\rho\|_{L^1(G)}}.
    \tag{3.53}
    \]

Then (3.51) can be written as

\[
    u(x) = \sum_{|a|\leq k} \frac{1}{a!} D^a u(0) x^a + \frac{V}{k!} \int_G g(x,\theta) \lambda(\theta) d\theta,
    \tag{3.54}
    \]

with \( v = \int_G \rho(\theta) d\theta \). In summary, we have the following lemma.

**Lemma 3.4.** It holds that

\[
    u(x) = \sum_{|a|\leq k} \frac{1}{a!} \partial^a u(0) x^a + \frac{v}{k!} r_k(x), \quad x \in \Omega
    \tag{3.55}
    \]

with \( v = \int_G \rho(\theta) d\theta \) and

\[
    r_k(x) = \int_G g(x,\theta) \lambda(\theta) d\theta, \quad G = \{-1,1\} \times [0,T] \times \mathbb{R}^d,
    \tag{3.56}
    \]

and \( g(x,\theta), \rho(\theta) \) and \( \lambda(\theta) \) defined in (3.53).

According to (3.53), the main ingredient \( (z\omega \cdot x - t)^k \) of \( g(x,\theta) \) includes only the direction \( \hat{\omega} \) of \( \omega \), which belongs to a bounded domain \( S^{d-1} \). Thanks to the continuity of \( (z\omega \cdot x - t)^k \) with respect to \( (z,\hat{\omega},t) \) and the boundedness of \( S^{d-1} \), the application of the stratified sampling to the residual term of the Taylor expansion leads to the approximation property in Theorem 3.5.
Theorem 3.5. Assume $u \in B^{k+1}(\Omega)$. There exist $\beta_j \in [-1,1]$, $\|\bar{\omega}\| = 1$, $t_j \in [0,T]$ such that
\[
u = \sum_{|\alpha| \leq k} \frac{1}{\alpha!} \partial^\alpha u(0)x^\alpha + \frac{\nu}{k!N} \sum_{j=1}^{N} \beta_j (\bar{\omega} \cdot x - t_j)\}_{j}^k \}
\] (3.57)
with $\nu = \int_{\Omega} \rho(\theta) d\theta$ and $\rho(\theta)$ defined in (3.53) satisfies the following estimate
\[
u - u \|_{H^m(\Omega)} \lesssim \begin{cases} N^{-\frac{1}{2} - \frac{1}{2}} \| u \|_{B^{k+1}(\Omega)}, & m < k, \\ N^{-\frac{1}{2}} \| u \|_{B^{k+1}(\Omega)}, & m = k. \end{cases}
\] (3.58)

Proof. Let
\[
u = \sum_{|\alpha| \leq k} \frac{1}{\alpha!} \partial^\alpha u(0)x^\alpha + \frac{\nu}{k!} \sum_{j=1}^{N} \beta_j (\bar{\omega} \cdot x - t_j)\}_{j}^k \}
\] (3.59)
By Lemma 2.4, for any decomposition $G = \bigcup_{i=1}^{N} G_i$, there exist $\{\theta_i\}_{i=1}^{N}$ and $\{\beta_i\}_{i=1}^{N} \in [0,1]$ such that
\[
u = \sum_{|\alpha| \leq k} \frac{1}{\alpha!} \partial^\alpha u(0)x^\alpha + \frac{\nu}{k!} \sum_{j=1}^{N} \beta_j (\bar{\omega} \cdot x - t_j)\}_{j}^k \}
\] (3.60)
Consider a $\epsilon$-covering decomposition $G = \bigcup_{i=1}^{N} G_i$ such that
\[
u = \sum_{|\alpha| \leq k} \frac{1}{\alpha!} \partial^\alpha u(0)x^\alpha + \frac{\nu}{k!} \sum_{j=1}^{N} \beta_j (\bar{\omega} \cdot x - t_j)\}_{j}^k \}
\] (3.61)
where $\bar{\omega}$ is defined in (3.47). For any $\theta_i, \theta'_j \in G_i$,
\[
u = \sum_{|\alpha| \leq k} \frac{1}{\alpha!} \partial^\alpha u(0)x^\alpha + \frac{\nu}{k!} \sum_{j=1}^{N} \beta_j (\bar{\omega} \cdot x - t_j)\}_{j}^k \}
\] (3.62)
As
\[
u = \sum_{|\alpha| \leq k} \frac{1}{\alpha!} \partial^\alpha u(0)x^\alpha + \frac{\nu}{k!} \sum_{j=1}^{N} \beta_j (\bar{\omega} \cdot x - t_j)\}_{j}^k \}
\] (3.63)
Thus, by Lemma 2.4, if \( m = |\alpha| < k \),

\[
\| \partial^{\alpha}_x (u - u_N) \|_{L^2(\Omega)} \leq \frac{|\Omega|^{1/2}}{(k - |\alpha|)!} (2T)^{k - |\alpha| - 1} \left( (k - |\alpha|)(T + 1) + 2T|\alpha| \right) N^{-\frac{k}{2}}. \tag{3.64}
\]

Note that \( \epsilon \sim N^{-\frac{1}{2}} \). There exists \( \theta_{i,j} \) such that for any \( 0 \leq k < m \),

\[
\| u - u_N \|_{H^m(\Omega)} \leq C(m, k, \Omega) \nu N^{-\frac{1}{2}} \tag{3.65}
\]

with \( \nu \leq \| u \|_{B^k(\Omega)} + 1 \) \((\Omega)\) and

\[
C(m, k, \Omega) = |\Omega|^{1/2} \left( \sum_{|\alpha| \leq k} \frac{1}{(k - |\alpha|)!} (2T)^{k - |\alpha| - 1} ((k - |\alpha|)(T + 1) + 2T|\alpha|) \right)^{1/2}. \tag{3.66}
\]

If \( m = |\alpha| = k \),

\[
\max_{1 \leq i \leq M, \theta, \theta' \in \mathcal{G}_i} \sup_{1 \leq j \leq M} \| D^{\alpha}_x (g(x, \theta_j) - g(x, \theta'_j)) \|_{L^2(\Omega)} \lesssim 1.
\]

This leads to

\[
\| u - u_N \|_{H^m(\Omega)} \leq C(m, k, \Omega) \nu N^{-\frac{1}{2}} \quad \text{for } k = m. \tag{3.67}
\]

Note that \( u_N \) defined above can be written as

\[
u_N(x) = \sum_{|\alpha| \leq k} \frac{1}{\alpha!} \partial^{\alpha} u(0) x^{\alpha} + \frac{1}{k!N} \sum_{j=1}^{N} \beta_j (\bar{\omega}_j \cdot x - t_j)^k
\]

with \( \beta_j \in [-1, 1] \), which completes the proof. \( \square \)

**Lemma 3.5.** There exist \( \alpha_i, \omega_i, b_i \) and

\[
N \leq 2 \left( \frac{k + d}{k} \right)
\]

such that

\[
\sum_{|\alpha| \leq m} \frac{1}{\alpha!} \partial^{\alpha} u(0) x^{\alpha} = \sum_{i=1}^{N} \alpha_i (\omega_i \cdot x + b_i)^k.
\]

with \( x^\alpha = x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_d^{\alpha_d} \), \( \alpha! = \alpha_1! \alpha_2! \cdots \alpha_d! \).

The above result can be found in [35].

A combination of Theorem 3.5 and the above lemma gives the following estimate in Theorem 3.6.
Theorem 3.6. Suppose \( u \in B_{k+1}(\Omega) \). There exist \( \beta_j, t_j \in \mathbb{R} \), \( \omega_j \in \mathbb{R}^d \) such that
\[
u_N(x) = \sum_{j=1}^{N} \beta_j (\omega_j \cdot x - t_j)_+^k
\]
(3.68)
satisfies the following estimate
\[
\|u - u_N\|_{H^m(\Omega)} \lesssim \begin{cases} N^{-\frac{1}{2} - \frac{1}{2}k} \|u\|_{B^{k+1}(\Omega)}, & k > m, \\ N^{-\frac{1}{2}} \|u\|_{B^{k+1}(\Omega)}, & k = m, \end{cases}
\]
(3.69)
where \( \omega \) is defined in (3.47).

Remark 3.1. We make the following comparisons between the results in Sections 3.1 and 3.2.

1. The results in Section 3.1 are for activation functions \( \sigma = b_k \), while the results in Section 3.2 are for activation functions \( \sigma = \text{ReLU}^k \).

2. By (3.9), the following relation obviously holds
\[
V_N(b_k) \subset V_{N+k}(\text{ReLU}^k),
\]
where \( V_{N+k}(\text{ReLU}^k) \) is the \( V_{N+k} \) defined in (3.3) and \( V_N(b_k) \) is the one hidden layer neuron network function class with activation function \( b_k \). Thus, asymptotically speaking, the results that hold for \( \sigma = b_k \) also hold for \( \sigma = \text{ReLU}^k \).

3. The results in Section 3.2 are in some cases asymptotically better than those in Section 3.1, but require more regularity assumptions on \( u \). For example, Theorem 3.1 requires only \( u \in B^m \), but Theorem 3.5 requires only \( u \in B^{k+1} \) even for \( m = 0 \).

4. The computational efficiency for the solution of the optimization problems (5.25) or (5.28) below, may be different with a different choice of activation function, namely, \( \sigma = b_k \) or \( \text{ReLU}^k \). For example, we refer to [48] for comparative numerical studies of applications of neural networks that use \( \sigma = b_2 \) and \( \text{ReLU}^2 \) as activation functions.

4 Deep finite neuron functions, adaptivity and spectral accuracy

In this section, we will study deep finite neuron functions through the framework of deep neural networks and then discuss its adaptive and spectral accuracy properties.
4.1 Deep finite neuron functions

Given $d, \ell \in \mathbb{N}^+, n_1, \cdots, n_\ell \in \mathbb{N}$ with $n_0 = d, n_{\ell+1} = 1$,

$$\theta^i(x) = \omega^i \cdot x + b^i, \quad \omega^i \in \mathbb{R}^{n_{i+1} \times n_i}, \quad b^i \in \mathbb{R}^{n_i},$$  \hspace{1cm} (4.1)

and the activation function ReLU$^k$, define a deep finite neuron function $u(x)$ from $\mathbb{R}^d$ to $\mathbb{R}$ as

$$f^0(x) = \theta^0(x),$$

$$f^i(x) = (\theta^i \circ \sigma)(f^{i-1}(x)), \quad i = 1: \ell,$$

$$f(x) = f^\ell(x).$$

The following more concise notation is often used in computer science literature:

$$f(x) = \theta^\ell \circ \sigma \circ \theta^{\ell-1} \circ \cdots \circ \theta^1 \circ \sigma \circ \theta^0(x),$$ \hspace{1cm} (4.2)

where $\theta^i : \mathbb{R}^{n_i} \to \mathbb{R}^{n_{i+1}}$ are linear functions as defined in (4.1). Such a deep neuron network has $(\ell+1)$-layer DNN, namely $\ell$-hidden layers. The size of this deep neuron network is $n_1 + \cdots + n_\ell$.

Based on these notations and connections, define deep finite neuron functions with the activation function $\sigma = \text{ReLU}^k$ by

$$n\mathcal{N}^k(n_1, n_2, \cdots, n_\ell) = \left\{ f^\ell(x) = \theta^\ell(x'), \text{ with } W^i \in \mathbb{R}^{n_{i+1} \times n_i}, b^i \in \mathbb{R}^{n_i}, i = 0: \ell, n_0 = d, n_{\ell+1} = 1 \right\}. \hspace{1cm} (4.3)$$

Generally, we can define the $\ell$-hidden layer neuron network as

$$n\mathcal{N}_\ell := \bigcup_{n_1, n_2, \cdots, n_\ell \geq 1} n\mathcal{N}(n_1, n_2, \cdots, n_\ell). \hspace{1cm} (4.4)$$

For $\ell=2$, functions in $n\mathcal{N}_2^k$ consist of piecewise polynomials of degree $k^2$ on a finite neuron grid, whose boundaries are level sets of quadratic polynomials (see Fig. 3).

4.2 Reproduction of polynomials and spectral accuracy

One interesting property of the ReLU$^k$-DNN is that it reproduces polynomials of degree $k$.

**Lemma 4.1.** Given $k \geq 2, q \geq 2$, there exist $\ell \geq 1, n_1, \cdots, n_\ell$ such that

$$\mathcal{P}_q \subset n\mathcal{N}^k(n_1, \cdots, n_\ell),$$

where $\mathcal{P}_q$ is the set of all polynomials with degree not larger than $q$. 
For a proof of the above result, we refer to [46].

**Theorem 4.1.** Let $\text{ReLU}^\ell$ be the activation function, and $\mathcal{N}_N^\ell(N)$ be the DNN model with $\ell$ hidden layers. There exists some $\ell$ such that

$$\inf_{v_N \in \mathcal{N}_N^\ell(N)} \|u - v_N\|_{H^m(\Omega)} \lesssim \inf_{v_N \in \mathcal{P}_N^\ell} \|u - v_N\|_{H^m(\Omega)}.$$  

(4.5)

Estimate (4.5) indicates that the deep finite neuron function may provide spectral approximate accuracy.

### 4.3 Reproduction of linear finite element functions and adaptivity

The deep neural network with ReLU activation function have been much studied in the literature and widely used in practice. One interesting fact is that ReLU-DNN is simply piecewise linear functions. More specifically, from [36], we have the following result:

**Lemma 4.2.** Assume that $\mathcal{T}_h$ is a simplicial finite element grid of $N$ elements, in which any union of simplexes that share a vertex is convex, any linear finite element function on this grid can be written as a ReLU-DNN with at most $O(d)$ hidden layers. The number of neurons is at most $O(\kappa^d N)$ for some constant $\kappa \geq 2$ depending on the shape-regularity of $\mathcal{T}_h$. The number of non-zero parameters is at most $O(d \kappa^d N)$.

The above result indicates that the deep finite neuron functions can reproduce any linear finite element function. Given the adaptive feature and capability of finite element methods, we see that the finite neuron method can be at least as adaptive as the finite element method.

## 5 The finite neuron method for boundary value problems

In this section, we apply the finite neuron functions for numerical solutions of (1.1). In Section 5.1, we first present some analytic results for (1.1). In Section 5.2, we obtain error estimates for the finite neuron method for (1.1) for both the Neumann and Dirichlet boundary conditions.
5.1 Elliptic boundary value problems of order $2m$

As discussed in the introduction, let us rewrite the Dirichlet boundary value problem as

$$
\begin{cases}
  Lu = f, & \text{in } \Omega, \\
  B^k_D(u) = 0, & \text{on } \partial \Omega, \quad (0 \leq k \leq m - 1).
\end{cases}
$$

(5.1)

Here $B^k_D(u)$ is given by (1.7). We next discuss the pure Neumann boundary conditions for the general PDE operator (1.2) when $m \geq 2$. We begin our discussion with the following simple result.

**Lemma 5.1.** For each $k = 0, 1, \cdots, m - 1$, there exists a bounded linear differential operator of order $2m - k - 1$:

$$B^k_N : H^{2m}(\Omega) \rightarrow L^2(\partial \Omega)$$

(5.2)

such that the following identity holds:

$$(Lu, v) = a(u, v) - \sum_{k=0}^{m-1} (B^k_N(u), B^k_D(v))_{0,\partial \Omega}.$$  

Namely,

$$\sum_{|\alpha|=m} (-1)^m(\partial^\alpha (a_\alpha \partial^\alpha u), v)_{0,\Omega} = \sum_{|\alpha|=m} (a_\alpha \partial^\alpha u, \partial^\alpha v)_{0,\Omega} - \sum_{k=0}^{m-1} (B^k_N(u), B^k_D(v))_{0,\partial \Omega}$$

(5.3)

for all $u \in H^{2m}(\Omega), v \in H^m(\Omega)$. Furthermore,

$$\sum_{k=0}^{m-1} \|B^k_D(u)\|_{L^2(\partial \Omega)} + \sum_{k=0}^{m-1} \|B^k_N(u)\|_{L^2(\partial \Omega)} \lesssim \|u\|_{2m,\Omega}.$$  

(5.4)

Lemma 5.1 can be proved by induction with respect to $m$. We refer to [47, Chapter 2] and [16] for a proof on a similar identity.

In general, the explicit expression of $B^k_N$ can be quite complicated. Let us get an idea by looking at some simple examples with the special operator

$$Lu = (-\Delta)^m u + u,$$  

(5.5)

and

$$a(u, v) = \sum_{|\alpha|=m} (a_\alpha \partial^\alpha u, \partial^\alpha v)_{0,\Omega} + (a_0 u, v), \quad \forall u, v \in V.$$  

(5.6)

- For $m = 1$, it is easy to see that $B^0_N u = \frac{\partial u}{\partial \nu}|_{\partial \Omega}$. 

For \( m = 2 \) and \( d = 2 \), see [18]:

\[
B^0_N u = \frac{\partial}{\partial \nu} \left( \Delta u + \frac{\partial^2 u}{\partial \tau^2} \right) - \frac{\partial}{\partial \tau} \left( \kappa \tau \frac{\partial u}{\partial \tau} \right) \bigg|_{\partial \Omega} \quad \text{and} \quad B^1_N u = \frac{\partial^2 u}{\partial \nu^2} \bigg|_{\partial \Omega},
\]

with \( \tau \) being the anti-clockwise unit tangential vector, and \( \kappa \tau \) the curvature of \( \partial \Omega \).

We are now in a position to state the pure Neumann boundary value problems for PDE operator (1.2) as follows:

\[
\begin{cases}
  Lu = f, \quad \text{in } \Omega, \\
  B_k N(u) = 0, \quad \text{on } \partial \Omega \quad (0 \leq k \leq m-1).
\end{cases} \tag{5.7}
\]

Combining the trace theorem for \( H^m(\Omega) \), see [1], and Lemma (5.1), it is easy to see that (1.5) is equivalent to (5.7) with \( V = H^m(\Omega) \).

For a given parameter \( \delta > 0 \), we next consider the following problem with the mixed boundary condition:

\[
\begin{cases}
  Lu_\delta = f, \quad \text{in } \Omega, \\
  B_k D(u_\delta) + \delta B_k N(u_\delta) = 0, \quad 0 \leq k \leq m-1.
\end{cases} \tag{5.8}
\]

It is easy to see that (5.8) is equivalent to the following problem: Find \( u_\delta \in H^m(\Omega) \), such that

\[
J_\delta(u_\delta) = \min_{v \in H^m(\Omega)} J_\delta(v), \tag{5.9}
\]

where

\[
J_\delta(v) = \frac{1}{2} a_\delta(v,v) - (f,v) \tag{5.10}
\]

and

\[
a_\delta(u,v) = a(u,v) + \delta^{-1} \sum_{k=0}^{m-1} \langle B_k D(u), B_k D(v) \rangle_{0,\partial \Omega}. \tag{5.11}
\]

In summary, we have the following lemma.

**Lemma 5.2.** The following equivalences hold:

1. \( u \) solves (5.1) or (5.7) if and only if \( u \) solves

\[
J(u) = \min_{v \in V} J(v) \tag{5.12}
\]

with \( V = H^m_0(\Omega) \) or \( V = H^m(\Omega) \).

2. \( u_\delta \) solves (5.8) if and only if \( u_\delta \) solves

\[
J_\delta(u_\delta) = \min_{v \in V^\delta} J_\delta(v)
\]

with \( V = H^m(\Omega) \).
Lemma 5.3. Assume that \( u \in V \) is the solution of (5.1) or (5.7) and \( u_{\delta} \in V \) is the solution of (5.9), then the following identities hold:

\[
\| v - u \|_a^2 = J(v) - J(u), \quad \forall v \in V, \quad (5.13)
\]

and

\[
\| v - u_{\delta} \|_{a,\delta}^2 = J_{\delta}(v) - J_{\delta}(u_{\delta}), \quad \forall v \in V. \quad (5.14)
\]

Here

\[
\| v \|_a^2 = a(v,v), \quad \| v \|_{a,\delta}^2 = a_{\delta}(v,v). \quad (5.15)
\]

Proof. Let \( u \) be the solution of (1.5). Given \( v \in V \), consider the quadratic function of \( t \):

\[
g(t) = J(u + t(v - u)).
\]

It is easy to see that

\[
0 = \arg\min \limits_t g(t), \quad g'(0) = 0
\]

and

\[
J(v) - J(u) = g(1) - g(0) = g'(0) + \frac{1}{2}g''(0) = \| v - u \|_a^2.
\]

This completes the proof of (5.13). The proof of (5.14) is similar. \( \square \)

Lemma 5.4. Let \( u \) be the solution of (5.1) and \( u_{\delta} \) be the solution of (5.8). Then,

\[
\| u - u_{\delta} \|_{a,\delta} \lesssim \sqrt{\delta} \| u \|_{2m,\Omega}. \quad (5.16)
\]

Proof. Let \( w = u - u_{\delta} \) and we have

\[
\begin{cases}
Lw = 0, & \text{in } \Omega, \\
B_D^k(w) + \delta B_N^k(w) = \delta B_N^k(u), & 0 \leq k \leq m-1.
\end{cases} \quad (5.17)
\]

By Lemma 5.1 and (5.17), we have

\[
0 = (Lw,w) = \sum_{|\alpha|=m} (a_{\alpha} \partial^\alpha w, \partial^\alpha w) - \sum_{k=0}^{m-1} \int_{\partial \Omega} B_N^k(w)B_D^k(w)ds + (a_0 w, w) = \sum_{|\alpha|=m} (a_{\alpha} \partial^\alpha w, \partial^\alpha w) + \sum_{k=0}^{m-1} \int_{\partial \Omega} (\delta^{-1}B_D^k(w) - B_N^k(u))B_D^k(w)ds + (a_0 w, w), \quad (5.18)
\]

implying

\[
a(w,w) + \delta^{-1} \sum_{k=0}^{m-1} \int_{\partial \Omega} B_D^k(w)^2 ds = \sum_{k=0}^{m-1} \int_{\partial \Omega} B_N^k(u)B_D^k(w)ds. \quad (5.19)
\]
By Cauchy inequality, we have
\[
\begin{align*}
  a(w, w) + \delta^{-1} \sum_{k=0}^{m-1} ||B_{D}^{k}(w)||_{L^2(\partial\Omega)}^2 &\leq \sum_{k=0}^{m-1} ||B_{N}^{k}(u)||_{L^2(\partial\Omega)} ||B_{D}^{k}(w)||_{L^2(\partial\Omega)} \\
  &\leq 2\delta \sum_{k=0}^{m-1} ||B_{N}^{k}(u)||_{L^2(\partial\Omega)}^2 + \frac{1}{2} \delta^{-1} \sum_{k=0}^{m-1} ||B_{D}^{k}(w)||_{L^2(\partial\Omega)}^2,
\end{align*}
\]
which implies
\[
\begin{align*}
  a(w, w) + \frac{1}{2} \delta^{-1} \sum_{k=0}^{m-1} ||B_{D}^{k}(w)||_{L^2(\partial\Omega)}^2 &\leq 2\delta \sum_{k=0}^{m-1} ||B_{N}^{k}(u)||_{L^2(\partial\Omega)}^2. 
\end{align*}
\]
By the definition of \(\|\cdot\|_{a, \delta}\) and noting that \(w = u - u_{\delta}\), we have
\[
\|u - u_{\delta}\|_{a, \delta}^2 \leq 4\delta \sum_{k=0}^{m-1} ||B_{N}^{k}(u)||_{L^2(\partial\Omega)}^2.
\]
In combination with (5.4), this completes the proof.

**Lemma 5.5.** For any \(s \geq -m\) and \(f \in H^{s}(\Omega)\), the solution \(u\) of (5.1) or (5.7) satisfies \(u \in H^{2m+s}(\Omega)\) and
\[
\|u\|_{2m+s, \Omega} \lesssim \|f\|_{s, \Omega}.
\]
We refer to [47] (Chapter 2, Theorem 5.1 therein) for a detailed proof.

**Lemma 5.6.** For any \(s \geq -m, \epsilon > 0\) and \(f \in H^{s}(\Omega)\), the solution \(u\) of (5.1) or (5.7) satisfies
\[
\|u\|_{B^{m+1}(\Omega)} \leq \|f\|_{-m+\frac{d}{2}+1+\epsilon, \Omega}.
\]

### 5.2 The finite neuron method for (1.1) and error estimates

Let \(V_{N} \subset V\) be a subset of \(V\) defined by (3.3), which may not be a linear subspace. Consider the discrete problem of (5.12):
\[
\text{Find } u_{N} \in V_{N} \text{ such that } J(u_{N}) = \min_{v_{N} \in V_{N}} J(v_{N}).
\]
It is easy to see that the solution to (5.25) always exists (for deep neural network functions as defined below), but may not be unique.

**Theorem 5.1.** Let \(u \in V\) and \(u_{N} \in V_{N}\) be solutions to (5.7) and (5.25), respectively. Then
\[
\|u - u_{N}\|_{a} = \inf_{v_{N} \in V_{N}} \|u - v_{N}\|_{a}.
\]
Proof. By Lemma 5.3, we have
\[ \|u_N - u\|_a^2 = J(u_N) - J(u) \leq f(v_N) - f(u) = \|v_N - u\|_a^2, \quad \forall v \in V_N. \]
The proof is complete. \(\Box\)

We obtain the following result.

**Theorem 5.2.** Let \( u \in V \) and \( u_N \in V_N \) be solutions to (5.7) and (5.25), respectively. Then for arbitrary \( \epsilon > 0 \), we have
\[ \|u - u_N\|_a \lesssim (\|f\|_2^2 + \|f\|_{k+1/2+1/\epsilon}) \left\{ \begin{array}{ll} N^{-1/2}, & m < k, \\ N^{-1/2}, & m = k. \end{array} \right. \] (5.27)

By (5.26), Theorem 3.5 and the embedding of the spectral Barron space into the Sobolev space, namely Lemma 2.5, the regularity result (5.23), we get the proof.

Next we consider the discrete problem of (5.9):

Find \( u_N \in V_N \) such that \( J_\epsilon(u_N) = \min_{v_N \in V_N} J_\epsilon(v_N). \) (5.28)

**Lemma 5.7.** For any given number \( \delta \), let \( u_\delta \) be the solution of (5.8) and \( u_N \) be the solution of (5.28), respectively. We have
\[ \|u_N - u_\delta\|_{a,\delta} \lesssim (1 + \delta^{-1/2}) \inf_{v_N \in V_N} \|v_N - u_\delta\|_{m,\Omega}. \] (5.29)

**Proof.** First of all, by Lemma 5.3 and the variational property, it holds that
\[ \|u_N - u_\delta\|_{a,\delta} = J_\epsilon(u_N) - J_\epsilon(u_\delta) \leq J_\epsilon(v_N) - J_\epsilon(u_\delta) = \|v_N - u_\delta\|_{a,\delta}^2, \quad \forall v_N \in V_N. \] (5.30)

Further, for any \( v_N \in V_N \), by the definition of \( \|\cdot\|_{a,\delta} \) and trace inequality, we have
\[ \|v_N - u_\delta\|_{a,\delta} \lesssim \|v_N - u_\delta\|_{m,\Omega} + \delta^{-1/2} \|v_N - u_\delta\|_{a,\Omega} \lesssim (1 + \delta^{-1/2}) \|v_N - u_\delta\|_{m,\Omega}. \]
This completes the proof. \(\Box\)

**Theorem 5.3.** Let \( u \) be the solution of (5.1) and \( u_N \) be the solution of (5.28). We have
\[ \|u - u_N\|_{a,\delta} \lesssim (1 + \delta^{-1/2}) \inf_{v_N \in V_N} \|v_N - u\|_{m,\Omega} + \sqrt{\delta} \|f\|_{L^2(\Omega)}. \] (5.31)

**Proof.** First, by triangle inequality and (5.30), for any \( v_N \in V_N \), we have
\[ \|u_N - u\|_{a,\delta} \leq \|u_N - u_\delta\|_{a,\delta} + \|u_\delta - u\|_{a,\delta} \leq \|v_N - u_\delta\|_{a,\delta} + \|u_\delta - u\|_{a,\delta} \leq \|v_N - u\|_{a,\delta} + 2 \|u_\delta - u\|_{a,\delta}. \] (5.32)

Then by the definition of \( \|\cdot\|_{a,\delta} \), trace inequality and (5.16), for any \( v_N \in V_N \), we have
\[ \|u_N - u\|_{a,\delta} \lesssim (1 + \delta^{-1/2}) \|v_N - u\|_{m,\Omega} + \sqrt{\delta} \|f\|_{L^2(\Omega)}. \] (5.33)
This completes the proof. \(\Box\)
Theorem 5.4. Let $u$ be the solution of (5.1) and $u_N$ be the solution of (5.28) with $\delta \sim N^{-1/2-1/d}$, respectively. Then
\[
\|u - u_N\|_a \lesssim (\|f\|_{L^2(\Omega)} + \|f\|_{-k+\frac{d}{2}+1+\epsilon}) \begin{cases} N^{-\frac{1}{4}-\frac{d}{2}}, & m < k, \\ N^{-\frac{1}{4}}, & m = k. \end{cases} \tag{5.34}
\]

Proof. Let us consider only the case that $k > m$. By Theorem 3.5,
\[
\inf_{v_N \in V_N} \|u - v_N\|_{m,\Omega} \lesssim N^{-\frac{1}{2}-\frac{1}{2}} \|u\|_{B^{m+1}(\Omega)}.
\]
Thus, by (5.29),
\[
\|u - u_N\|_{m,\Omega} \lesssim \|u_N - u\|_a \lesssim \delta^{-\frac{1}{2}} N^{-\frac{1}{2}-\frac{1}{2}} \|u\|_{B^{m+1}(\Omega)} + \delta^\frac{1}{2} \|f\|_{L^2(\Omega)} \leq (\delta^{-\frac{1}{2}} N^{-\frac{1}{2}-\frac{1}{2}} + \delta^\frac{1}{2}) (\|u\|_{B^{m+1}(\Omega)} + \|f\|_{L^2(\Omega)}), \quad \forall \delta > 0. \tag{5.35}
\]
Set $\delta \sim N^{-1/2-1/d}$, and it follows that
\[
\|u - u_N\|_{m,\Omega} \lesssim N^{-\frac{1}{4}-\frac{1}{4}} (\|u\|_{B^{m+1}(\Omega)} + \|f\|_{L^2(\Omega)}). \tag{5.36}
\]
Now by embedding the spectral Barron space into the Sobolev space, namely Lemma 2.5, the regularity result (5.23), the proof is complete. \hfill \Box

Remark 5.1. Although the analysis in this section is mainly for activation function ReLU$^k$, similar results are valid for other activations. For example, Theorem 5.1 and Theorem 5.3 naturally hold for any activation function.

Remark 5.2. We also note that analysis in this section naturally generalizes to problems other than (1.1). For example, Theorem 5.1 can be generalized to the case when $J(\cdot)$ is replaced by a general strictly convex function.

Remark 5.3. We note that (5.28) was studied in [28] for $m = 1$ and $k = 3$. Convergence analysis for (5.25) and (5.28) seems to be new in this paper. For other convergence analyses of DNN for numerical PDE, we refer to [61] and [53, 54] for convergence analysis of the Physics Informed Neural Network (PINN).

6 Summary and discussions

In this paper, we consider a very special class of neural network function based on ReLU$^k$ as the activation function. This function class consists of piecewise polynomials that closely resemble finite element functions. By considering elliptic boundary value problems of $2m$-th-order in any dimensions, it is still unknown how to construct the $H^{m-\cdot}$ conforming finite element space in general in the classic finite element setting. In contrast, it is rather straightforward to construct $H^{m-\cdot}$-conforming piecewise polynomials using neural networks, known as the finite neuron method, and we further proved that the finite neuron method provides good approximation properties.
It is still a subject of debate and requires further investigation as to whether it is practically efficient to use artificial neural networks for the numerical solution of partial differential equations. One major challenge for this type of method is that the resulting optimization problem is hard to solve, as we shall discuss below.

6.1 Solution of the non-convex optimization problem

(5.25) or (5.28) each constitutes a highly nonlinear and non-convex optimization problem with respect to parameters defining the functions in $V_N$, see (3.3). How to solve this type of optimization problem efficiently is a topic of intensive research in deep learning. For example, the stochastic gradient method is used in [28] to solve (5.28) for $m = 1$ and $k = 3$. The multi-scale deep neural network (MscaleDNN) [48] and phase shift DNN (PhaseDNN) [15] are developed to convert the high-frequency solution to a low frequency one before training. Randomized Newton’s method is developed to train the neural network from a nonlinear computation point of view [17]. More refined algorithms still need to be developed to solve (5.25) or (5.28) with high accuracy so that the convergence order, (5.27) or (5.34), of the finite neuron method cannot be achieved.

6.2 Competition between locality and global smoothness

One insight gained from the studies in the paper is that the challenges in constructing the classic $H^m$-finite element subspace seems to lie in the competition between local degree of freedom (d.o.f.) and global smoothness. In the classic finite element, it is necessary to define d.o.f. on each element and then glue the local d.o.f. together to obtain a globally $H^m$-smooth function. This process has proven to be very difficult to realize in general when $m \geq 2$. But, if we relax the locality, as in the Powell-Sabine element [57], we can use piecewise polynomials of a lower degree to construct globally smooth function. The neuron network approach studied in this paper can be considered a global construction without any use of a grid in the first place (even though an implicitly defined grid exists). As a result, it is quite easy to construct globally smooth functions that are piecewise polynomials. It is quite remarkable that such a global construction leads to a function class that has very good approximation properties. This is an attractive property of the function classes from the artificial neural network. One important question to ask pertains to whether it is possible to develop finite element construction technique that are more global than the classic finite element but more local than the finite neuron method, which is an interesting topic for further research. We observe that more global d.o.f. leads to easier construction of conforming elements for high-order PDEs.

6.3 Piecewise $P_m$ for $H^m(\Omega)$: From finite element to finite neuron method

As is noted above, in the classic finite element setting, it is challenging to construct $H^m$-conforming finite element spaces for any $m,d \geq 1$. But if we relax the conformity,
as shown in [64], it is possible to give a universal construction of the convergent \( H^m \)-nonconforming finite element consisting of piecewise polynomials of degree \( m \). In the finite neuron method setting, by relaxing the constraints from the a priori finite element grid, the construction of \( H^m \)-conforming piecewise polynomials of degree \( m \) becomes straightforward. In fact, the finite neuron method can be considered mesh-less method, or even, a vertex-less method although there is a hidden grid for any finite neuron function. This raises the question as to whether it is possible to develop some “in-between” method that has the advantages of both the classic finite element method and the finite neuron method.

### 6.4 Adaptivity and spectral accuracy

One of the important properties in the traditional finite element method is its ability to locally adapt the finite element grids to provide accurate approximation of a PDE solution that may have local singularities (such as corner singularities and interface singularities). In contrast, the traditional spectral method (using high-order polynomials) can provide very high-order accuracy for solutions that are globally smooth. The finite neuron method analyzed in this paper seems to possess both the adaptivity feature as in the traditional finite element method and also global spectral accuracy as in the traditional spectral methods. The finite neuron method is adaptive as shown in Section 4, the deep finite neuron method can recover locally adaptive finite element spaces for \( m = 1 \). The spectral feature of the finite neuron method is illustrated in Theorem 4.1. As a result, it is conceivable that the finite neuron method may have both the local and also global adaptive feature, or perhaps even adaptive features in all different scales. Nevertheless, these highly adaptive features of the finite neuron method come with a potentially high price, namely the solution of nonlinear and non-convex optimization problems.
6.5 Comparison with PINN

One important class of methods related to the finite neuron method analyzed in this paper is the method of physical-informed neural networks (PINN) introduced in [58]. By minimizing certain norms of the PDE residual together with penalizations of boundary conditions and other relevant quantities, PINN is a very general approach that can be directly applied to a wide range of problems. In comparison, the finite neuron method can be applied only to some special class of problems that admit some special physical laws such as principle of energy minimization or principle of least action, see [30]. Because of the special physical law associated with our underlying minimization problems, the Neumann boundary conditions are naturally enforced in the minimization problem and, unlike in the PINN method, no penalization is needed to enforce boundary conditions of this type.

6.6 On the sharpness of the error estimates

In this paper, we provide a number of error estimates for our finite neuron method such as (3.23), (3.27) and (3.69), which give increasingly better asymptotic order but also require more regularities. Even for the sufficiently regular solution \( u \), the best asymptotic estimate (3.69) may still not be optimal. In the finite element method, the piecewise polynomial of degree \( k \) usually gives rise to increasingly better asymptotic error when \( k \) increases. But the asymptotic rate in the estimate of (3.69) does not improve as \( k \) increases. On the other hand, if \( k > j \), ReLU\(^k\)-DNN should conceivably give better accuracy than ReLU\(^j\)-DNN since ReLU\(^j\) can be approximated to be arbitrarily accurate by a certain finite difference of ReLU\(^k\). How to obtain better asymptotic estimates than (3.69) remains a subject for further investigation.

We also note that our error estimate (5.34) for the Dirichlet boundary condition is not as good as the one (5.27) for the Neumann boundary condition. This is undesirable and may not be optimal. In comparison, the Nitsche trick does not suffer a loss of accuracy when used in the traditional finite element method.

6.7 Neural splines in multi-dimensions

The spline functions described in Section 3.1 are widely used in scientific and engineering computing, but their generalization to multiple dimensions is non-trivial, especially when \( \Omega \) has a curved boundary. In [40], using the tensor product, the authors extended the 1D spline to multi-dimensions on rectangular grids. Some others involve rational functions such as NURBS [21]. But the generalization of \( nN_1^k \) or \( nN_1(b^e) \) to multi-dimensions is straightforward and also the resulting (nonlinear) space has very good approximate properties. It is conceivable that the neural network extension of B-spline to multiple dimensions that are locally polynomials and globally smooth, may find useful applications in computer aided design (CAD) and isogeometric analysis [21]. This is a potentially an interesting research direction.
Acknowledgments

Main results in this manuscript were prepared for and reported in the “International Conference on Computational Mathematics and Scientific Computing” (August 17-20, 2020, http://lsec.cc.ac.cn/~iccmsc/Home.html), and the author is grateful to the conference organizers for their invitation to present and also to the audience for their helpful feedback. The author also wishes to thank Limin Ma, Qingguo Hong and Shuo Zhang for their help in preparing this manuscript and Jonathan Siegel and Juncai He for helpful discussions. This work was partially supported by the Verne M. William Professorship Fund from Penn State University and the National Science Foundation (Grant No. DMS-1819157).

References

[1] Adams, R. A. and Fournier, J. (2003). Sobolev spaces, volume 140. Academic press.
[2] Alfeld, P. (1984). A trivariate clough-tocher scheme for tetrahedral data. Computer Aided Geometric Design, 1(2):169–181.
[3] Antonietti, P. F., Manzini, G., and Verani, M. (2018). The fully nonconforming virtual element method for biharmonic problems. Mathematical Models and Methods in Applied Sciences, 28(02):387–407.
[4] Argyris, J. H., Fried, I., and Scharpf, D. W. (1968). The tuba family of plate elements for the matrix displacement method. The Aeronautical Journal, 72(692):701–709.
[5] Baker, G. A. (1977). Finite element methods for elliptic equations using nonconforming elements. Mathematics of Computation, 31(137):45–59.
[6] Barron, A. R. (1993). Universal approximation bounds for superpositions of a sigmoidal function. IEEE Transactions on Information Theory, 39(3):930–945.
[7] Beirão da Veiga, L., Brezzi, F., Cangiani, A., Manzini, G., Marini, L. D., and Russo, A. (2013). Basic principles of virtual element methods. Mathematical Models and Methods in Applied Sciences, 23(01):199–214.
[8] Bickel, P. J. and Freedman, D. A. (1984). Asymptotic normality and the bootstrap in stratified sampling. The Annals of Statistics, pages 470–482.
[9] Boor, C. D. and Devore, R. (1983). Approximation by smooth multivariate splines. Transactions of the American Mathematical Society, 276:775–788.
[10] Boor, C. D. and Höllig, K. (1983). Approximation order from bivariate C1-Cubics: A counterexample. Proceedings of the American Mathematical Society, 87(4):649–655.
[11] Bramble, J. H. and Zlámal, M. (1970). Triangular elements in the finite element method. Mathematics of Computation, 24(112):809–820.
[12] Brenner, S. C. and Sung, L. (2005). C0 interior penalty methods for fourth order elliptic boundary value problems on polygonal domains. Journal of Scientific Computing, 22(1-3):83–118.
[13] Brezzi, F., Falk, R. S., and Marini, L. D. (2014). Basic principles of mixed virtual element methods. ESAIM: Mathematical Modelling and Numerical Analysis, 48(4):1227–1240.
[14] Brezzi, F. and Marini, L. D. (2013). Virtual element methods for plate bending problems. Computer Methods in Applied Mechanics and Engineering, 253:455–462.
[15] Cai, W., Li, X., and Liu, L. (2019). A phase shift deep neural network for high frequency wave equations in inhomogeneous media. arXiv preprint arXiv:1909.11759.
[16] Chen, L. and Huang, X. (2020). Nonconforming virtual element method for 2nth order partial differential equations in $R^n$. Mathematics of Computation, 89(324):1711–1744.
[17] Chen, Q. and Hao, W. (2019). A randomized Newton’s method for solving differential equations based on the neural network discretization. arXiv preprint arXiv:1912.03196.
[18] Chien, W. Z. (1980). Variational methods and finite elements.
[19] Ciarlet, P. G. (1978). The finite element method for elliptic problems. North-Holland.
[20] Clough, R. and Tocher, J. (1965). Finite-element stiffness analysis of plate bending. In Proc. First Conf. Matrix Methods in Struct. Mech., Wright-Patterson AFB, Dayton (Ohio).
[21] Cottrell, J. A., Hughes, T. J., and Bazilevs, Y. (2009). Isogeometric analysis: toward integration of CAD and FEA. John Wiley & Sons.
[22] Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. Mathematics of Control, Signals and Systems, 2(4):303–314.
[23] de Boor, C. (1971). Subroutine package for calculating with B-splines. Los Alamos Scient. Lab. Report LA-4728-MS.
[24] E, W., Han, J., and Jentzen, A. (2017). Deep learning-based numerical methods for high-dimensional parabolic partial differential equations and backward stochastic differential equations. Communications in Mathematics and Statistics, 5(4):349–380.
[25] E, W., Ma, C., and Wu, L. (2019a). Barron spaces and the compositional function spaces for neural network models. arXiv preprint arXiv:1906.08039.
[26] E, W., Ma, C., and Wu, L. (2019b). A priori estimates of the population risk for two-layer neural networks. Communications in Mathematical Sciences, 17(5):1407–1425.
[27] E, W. and Wojtowytsch, S. (2020). Representation formulas and pointwise properties for Barron functions. arXiv preprint arXiv:2006.05982.
[28] E, W. and Yu, B. (2018). The deep Ritz method: A deep learning-based numerical algorithm for solving variational problems. Communications in Mathematics and Statistics, 6(1):1–12.
[29] Feng, K. (1965). Finite difference schemes based on variational principles. Appl. Math. Comput. Math, 2:238–262.
[30] Feynman, R., Leighton, R., and Sands, M. (1965). The Feynman lectures on physics, volume 3: Quantum mechanics. In Library of Congress, Catalog Card, number 63-20717.
[31] Fu, G., Guzmán, J., and Neilan, M. (2020). Exact smooth piecewise polynomial sequences on Alfeld splits. Mathematics of Computation, 89(323):1059–1091.
[32] Funahashi, K.-I. (1989). On the approximate realization of continuous mappings by neural networks. Neural Networks, 2(3):183–192.
[33] Goodfellow, I., Bengio, Y., Courville, A., and Bengio, Y. (2016). Deep Learning, volume 1. MIT Press, Cambridge.
[34] Gudi, T. and Neilan, M. (2011). An interior penalty method for a sixth-order elliptic equation. IMA Journal of Numerical Analysis, 31(4):1734–1753.
[35] He, J., Li, L., and Xu, J. (2020a). DNN with Heaviside, ReLU and ReQU activation functions. preprint.
[36] He, J., Li, L., Xu, J., and Zheng, C. (2018). ReLU deep neural networks and linear finite elements. arXiv preprint arXiv:1807.03973.
[37] He, J., Li, L., Xu, J., and Zheng, C. (2020b). ReLU deep neural networks and linear finite elements. Journal of Computational Mathematics, 38(3):502–527.
[38] Hornik, K., Stinchcombe, M., and White, H. (1989). Multilayer feedforward networks are universal approximators. Neural Networks, 2(5):359–366.
[39] Hornik, K., Stinchcombe, M., White, H., and Auer, P. (1994). Degree of approximation results for feedforward networks approximating unknown mappings and their derivatives. *Neural Computation*, 6(6):1262–1275.

[40] Hu, J. and Zhang, S. (2015). The minimal conforming $H^k$ finite element spaces on $\mathbb{R}^n$ rectangular grids. *Mathematics of Computation*, 84(292):563–579.

[41] Hu, J. and Zhang, S. (2017). A canonical construction of $H^m$-nonconforming triangular finite elements. *Annals of Applied Mathematics*, 33(3):266–288.

[42] Jones, L. K. (1992). A simple lemma on greedy approximation in Hilbert space and convergence rates for projection pursuit regression and neural network training. *The Annals of Statistics*, 20(1):608–613.

[43] Klusowski, J. M. and Barron, A. R. (2016). Uniform approximation by neural networks activated by first and second order ridge splines. *arXiv preprint arXiv:1607.07819*.

[44] Lagaris, I. E., Likas, A., and Fotiadis, D. I. (1998). Artificial neural networks for solving ordinary and partial differential equations. *IEEE Transactions on Neural Networks*, 9(5):987–1000.

[45] Lai, M. and Schumaker, L. L. (2007). *Spline functions on triangulations*. Number 110. Cambridge University Press.

[46] Li, B., Tang, S., and Yu, H. (2019). Better approximations of high dimensional smooth functions by deep neural networks with rectified power units. *arXiv preprint arXiv:1903.05858*.

[47] Lions, J. L. and Magenes, E. (2012). *Non-homogeneous boundary value problems and applications*, volume 1. Springer Science & Business Media.

[48] Liu, Z., Cai, W., and Xu, Z. (2020). Multi-scale deep neural network (mscalednn) for solving Poisson-Boltzmann equation in complex domains. *arXiv preprint arXiv:2007.11207*.

[49] Makovoz, Y. (1996). Random approximants and neural networks. *Journal of Approximation Theory*, 85(1):98–109.

[50] McCulloch, W. S. and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, 5(4):115–133.

[51] Mhaskar, H. and Micchelli, C. A. (1995). Degree of approximation by neural and translation networks with a single hidden layer. *Advances in Applied Mathematics*, 16(2):151–183.

[52] Mhaskar, H. N. and Micchelli, C. A. (1994). Dimension-independent bounds on the degree of approximation by neural networks. *IBM Journal of Research and Development*, 38(3):277–284.

[53] Mishra, S. and Molinaro, R. (2020). Estimates on the generalization error of physics informed neural networks (PINNs) for approximating PDEs ii: A class of inverse problems. *arXiv preprint arXiv:2007.01138*.

[54] Mishra, S. and Rusch, T. K. (2020). Enhancing accuracy of deep learning algorithms by training with low-discrepancy sequences. *arXiv preprint arXiv:2005.12564*.

[55] Morley, L. S. D. (1967). *The triangular equilibrium element in the solution of plate bending problems*. RAE.

[56] Pinkus, A. (1999). Approximation theory of the MLP model in neural networks. *Acta Numerica*, 8:143–195.

[57] Powell, M. J. and Sabin, M. A. (1977). Piecewise quadratic approximations on triangles. *ACM Transactions on Mathematical Software (TOMS)*, 3(4):316–325.

[58] Raissi, M., Perdikaris, P., and Karniadakis, G. E. (2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707.

[59] Rubinstein, R. Y. and Kroese, D. P. (2016). *Simulation and the Monte Carlo method*, volume 10. John Wiley & Sons.
[60] Schedensack, M. (2016). A new discretization for \( m \)-th-Laplace equations with arbitrary polynomial degrees. *SIAM Journal on Numerical Analysis*, 54(4):2138–2162.

[61] Shin, Y., Darbon, J., and Karniadakis, G. E. (2020). On the convergence and generalization of physics informed neural networks. *arXiv preprint arXiv:2004.01806*.

[62] Siegel, J. W. and Xu, J. (2020). Approximation rates for neural networks with general activation functions. *Neural Networks*.

[63] Sirignano, J. and Spiliopoulos, K. (2018). DGM: A deep learning algorithm for solving partial differential equations. *Journal of Computational Physics*, 375:1339–1364.

[64] Wang, M. and Xu, J. (2013). Minimal finite element spaces for \( 2m \)-th-order partial differential equations in \( \mathbb{R}^n \). *Mathematics of Computation*, 82(281):25–43.

[65] Wu, S. and Xu, J. (2017). \( P_m \) interior penalty nonconforming finite element methods for \( 2m \)-th order PDEs in \( \mathbb{R}^n \). *arXiv preprint arXiv:1710.07678*.

[66] Wu, S. and Xu, J. (2019). Nonconforming finite element spaces for \( 2m \)-th order partial differential equations on \( \mathbb{R}^n \) simplicial grids when \( m = n + 1 \). *Mathematics of Computation*, 88(316):531–551.

[67] Xu, J. (1992). Iterative methods by space decomposition and subspace correction. *SIAM Review*, 34(4):581–613.

[68] Ženíšek, A. (1970). Interpolation polynomials on the triangle. *Numerische Mathematik*, 15(4):283–296.

[69] Zhang, S. (2009). A family of 3D continuously differentiable finite elements on tetrahedral grids. *Applied Numerical Mathematics*, 59(1):219–233.