Journal of Machine Learning for Modeling and Computing, 3(4):73–91 (2022)

GRADIENT-ENHANCED DEEP NEURAL NETWORK APPROXIMATIONS

Xiaodong Feng & Li Zeng*

LSEC, Institute of Computational Mathematics and Scientific/Engineering Computing, AMSS, Chinese Academy of Sciences, Beijing, China

*Address all correspondence to: Li Zeng, LSEC, Institute of Computational Mathematics and Scientific/Engineering Computing, AMSS, Chinese Academy of Sciences, Beijing, China, E-mail: zengli@lsec.cc.ac.cn

Original Manuscript Submitted: 11/8/2022; Final Draft Received: 12/10/2022

We propose in this work the gradient-enhanced deep neural network (DNN) approach for function approximations and uncertainty quantification. More precisely, the proposed approach adopts both the function evaluations and the associated gradient information to yield enhanced approximation accuracy. In particular, the gradient information is included as a regularization term in the gradient-enhanced DNN approach, for which we present posterior estimates (by the two-layer neural networks) similar to those in the path-norm regularized DNN approximations. We also discuss the application of this approach to gradient-enhanced uncertainty quantification, and present several numerical experiments to show that the proposed approach can outperform the traditional DNN approach in many cases of interest.

KEY WORDS: *deep neural networks, two-layer neural network, Barron space, uncertainty quantification*

1. INTRODUCTION

In recent years, deep neural networks (DNNs) have been widely used for dealing with scientific and engineering problems, such as function approximations (E et al., 2022; Schwab and Zech, 2019; Siegel and Xu, 2020), numerical partial differential equations (PDEs) (E and Yu, 2018; Raissi et al., 2019; Sirignano and Spiliopoulos, 2018), image classification (He et al., 2016; Litjens et al., 2017), and uncertainty quantification (Meng and Karniadakis, 2020; Qin et al., 2021; Yang et al., 2021), to name a few. Compared to traditional tools such as polynomials (DeVore and Lorentz, 1993), radial basis functions (Majdisova and Skala, 2017), and kernel methods (Liu et al., 2020), one of the main advantages of DNNs is their potential approximation capacity for high-dimensional problems. Unlike classic tools such as polynomial approximations (for which the relevant theoretical analysis results have been well studied), the associated theoretical analysis for DNNs is still in its infancy. Among others, we mention the seminal works by Barron (1993) and E et al. (2022) where the concept of "Barron space" was proposed, and some approximation results for DNNs were presented.

In this work, we shall present the gradient-enhanced DNN approach. More precisely, our approach adopts both the function evaluations and the associated gradient information. This is similar to the classic Hermite-type interpolation. Our main contributions are summarized as follows:

2689–3967/22/$35.00

- We present the gradient-enhanced DNN approach, where the gradient information is included as a regularization term.
- For the gradient-enhanced DNN approach, we present posterior estimates (via a two-layer neural network) similar to those in the path-norm regularized DNN approximations. More precisely, we show that the posterior generalization error can be bounded by

$$\mathscr{O}\left(d\|\theta\|_{\mathscr{P}}\sqrt{\frac{\ln(2d)}{n}}+\|\theta\|_{\mathscr{P}}\frac{\ln(\|\theta\|_{\mathscr{P}})}{\sqrt{n}}\right),$$

  where n is the number of training points, $\|\theta\|_{\mathscr{P}}$ is the path norm, and d is the dimension.
- We discuss the application of our approach to gradient-enhanced uncertainty quantification and present several numerical experiments to show that the gradient-enhanced DNN approach can outperform the traditional DNNs in many cases of interest.

We remark that gradient-enhanced polynomial approximations have been proposed for uncertainty quantification (Guo et al., 2018; Jakeman et al., 2015; Li et al., 2011; Lockwood and Mavriplis, 2013; Peng et al., 2016). Under some circumstances of uncertainty quantification, the cost of calculating derivatives can be inexpensive, e.g., by solving adjoint equations. These additional derivatives actually increase the existing data information, which may bring a better approximation. On the other hand, there are already many gradient-enhanced methods based on deep learning. Many researchers consider enhancing the gradient value back-propagated by the network to avoid the phenomenon of gradient vanishing or gradient explosion during training (He et al., 2016; Schmidhuber and Hochreiter, 1997; Yan et al., 2022). A gradient-enhanced damage model was used in Zhuang et al. (2020) to ensure the well-posedness of the boundary value and yields mesh-independent results in computational methods. Some also regularize the derivatives of the output of neural networks to improve the adversarial robustness and interpretability (Ross and Doshi-Velez, 2018). In addition, a gradient-enhanced physics-informed neural network (PINN) was proposed to improve the accuracy and training efficiency of PINNs in Yu et al. (2022), where the gradient information (i.e., the associated adjoint equation) is included to yield a modified PINN loss function. Different from these methods, our method is data driven, tackling the case that derivatives are given along with function evaluations. Moreover, theoretical analysis is provided to illustrate the feasibility and effectiveness of this method.

The rest of this paper is organized as follows. In Section 2, we set up the problem and present some preliminaries. In Section 3 we present the error estimations in two-layer neural networks for the gradient regularized approximation problem. Applications to uncertainty quantification are discussed in Section 4. Finally, we give some concluding remarks in Section 5.

2. PRELIMINARIES

2.1 Problem Setup

We begin the discussion by considering function approximations via DNNs with labeled data. We consider the target function $f^*:\mathbb{R}^d\to\mathbb{R}$, and we assume that the following data are available: $\{\boldsymbol{x}_i,y_i,\boldsymbol{y}_i'\}_{i=1}^n$. Here y_i and $\boldsymbol{y}_i'$ are the functional evaluations and gradient evaluations, respectively. Namely,

$$\begin{cases} y_i=f^*(\boldsymbol{x}_i), & i=1,\dots,n.\\ \boldsymbol{y}_i'=\nabla f^*(\boldsymbol{x}_i), & i=1,\dots,n.\end{cases} \tag{1}$$

For simplicity, we assume that the data $\{\boldsymbol{x}_i\}_i$ lie in $\mathscr{X} = [-1,1]^d$ and $0 \le f^* \le 1$. We shall show our analysis results via a two-layer neural network, for which the nonlinear function can be defined as

$$f(\boldsymbol{x};\theta) = \sum_{i=1}^{m} a_k \sigma(\boldsymbol{w}_k^T \boldsymbol{x}), \tag{2}$$

where $\boldsymbol{w}_k \in \mathbb{R}^d$, $\sigma : \mathbb{R} \to \mathbb{R}$ is a nonlinear activation function, and $\theta = \{(a_k, \boldsymbol{w}_k)\}_{k=1}^m$ is the unknown parameter. We define a truncated form of $f(\boldsymbol{x};\theta)$ through

$$Tf(\boldsymbol{x};\theta) = \max\big\{\min\{f(\boldsymbol{x};\theta),1\},0\big\}.$$

By an abuse of notation, in the following we still use $f(\boldsymbol{x};\theta)$ to denote $Tf(\boldsymbol{x};\theta)$. For the training $\{\boldsymbol{x}_i, y_i\}_{i=1}^n$, the population risk can be defined by

$$L(\theta) = \mathbb{E}_{\boldsymbol{x},y}\big[\ell(f(\boldsymbol{x};\theta), y)\big]. \tag{3}$$

The empirical risk with the training data yields

$$L_n(\theta) = \frac{1}{n}\sum_{i=1}^{n} \ell\big(f(\boldsymbol{x}_i;\theta), y_i\big), \tag{4}$$

where $\ell(f(\boldsymbol{x}), y) = (1/2)(f(\boldsymbol{x}) - y)^2$. For the gradient information, we consider

$$L'(\theta) = \mathbb{E}_{\boldsymbol{x},\boldsymbol{y}}\big[\tilde{\ell}\big(\nabla f(\boldsymbol{x};\theta), \boldsymbol{y}'\big)\big], \tag{5}$$

where $\tilde{\ell}\big(\nabla f(\boldsymbol{x};\theta), \boldsymbol{y}'\big) = \big\|\nabla f(\boldsymbol{x};\theta) - \boldsymbol{y}'\big\|_2$ and $\|\cdot\|_q$ indicates the ℓ_q norm of a vector. Similarly, the empirical risk with the training data $\{\boldsymbol{x}_i, y_i, \boldsymbol{y}_i'\}_{i=1}^n$ yields

$$L_n'(\theta) = \frac{1}{n}\sum_{i=1}^{n} \big[\tilde{\ell}(\nabla f(\boldsymbol{x}_i;\theta), \boldsymbol{y}_i')\big]^2. \tag{6}$$

Moreover, the path-norm of the two-layer neural network is defined as

$$\|\theta\|_{\mathscr{P}} := \sum_{k=1}^{m} |a_k| \|\boldsymbol{w}_k\|_1. \tag{7}$$

We are now ready to present the so-called gradient-enhanced DNN approach.

Definition 2.1 (Gradient-enhanced DNNs model)**.** For a two-layer neural network $f(\cdot;\theta)$ of width m, the gradient regularized risk is defined as follows:

$$J_{n,\beta}(\theta) := L_n(\theta) + \beta \cdot L_n'(\theta).$$

The corresponding regularized estimator is defined as

$$\theta_{n,\beta} = \arg\min J_{n,\beta}(\theta).$$

Note that the minimizers are not necessarily unique, and $\theta_{n,\beta}$ should be understood as any of the minimizers.

The above approach can be viewed as an extension of the classic Hermite interpolation (Spitzbart, 1960; Wu, 1992), and is motivated by applications such as gradient-enhanced uncertainty quantification (Guo et al., 2018; Jakeman et al., 2015; Li et al., 2011; Lockwood and Mavriplis, 2013; Peng et al., 2016).

2.2 Barron Space

We now provide a brief overview of Barron space (E et al., 2019). Let $\mathbb{S}^{d-1} := \{\boldsymbol{w} \in \mathbb{R}^d \mid \|\boldsymbol{w}\|_1 = 1\}$. Let $\mathscr{F}$ be the Borel σ-algebra on $\mathbb{S}^{d-1}$ and let $\mathbb{P}(\mathbb{S}^{d-1})$ be the collection of the probability measures on $\left(\mathbb{S}^{d-1}, \mathscr{F}\right)$. Let $\mathscr{B}(\mathscr{X})$ be the collection of functions that admit the following integral representation:

$$f(\boldsymbol{x}) = \int_{\mathbb{S}^{d-1}} a(\boldsymbol{w})\sigma\big(\langle \boldsymbol{w}, \boldsymbol{x}\rangle\big)\mathrm{d}\boldsymbol{\pi}(\boldsymbol{w}), \quad \forall \boldsymbol{x} \in \mathscr{X}, \tag{8}$$

where $\boldsymbol{\pi} \in \mathbb{P}(\mathbb{S}^{d-1})$, and $a(\cdot)$ is a measurable function with respect to $(\mathbb{S}^d, \mathscr{F})$. For any $f \in \mathscr{B}(\mathscr{X})$ and $p \geq 1$, we define the following norm:

$$\gamma_p(f) := \inf_{(a,\boldsymbol{\pi})\in\Theta_f} \left(\int_{\mathbb{S}^{d-1}} |a(\boldsymbol{w})|^p \,\mathrm{d}\boldsymbol{\pi}(\boldsymbol{w})\right)^{1/p}, \tag{9}$$

where

$$\Theta_f = \left\{(a, \boldsymbol{\pi}) \,\middle|\, f(\boldsymbol{x}) = \int_{\mathbb{S}^{d-1}} a(\boldsymbol{w})\sigma\big(\langle \boldsymbol{w}, \boldsymbol{x}\rangle\big)\mathrm{d}\boldsymbol{\pi}(\boldsymbol{w})\right\}.$$

Definition 2.2 [Barron space (E et al., 2019)]**.** The Barron space is defined as

$$\mathscr{B}_p(\mathscr{X}) := \left\{f \in \mathscr{B}(\mathscr{X}) \,\middle|\, \gamma_p(f) < \infty\right\}.$$

We next present several assumptions.

Assumption 2.1. *Throughout the paper we assume that*

- $\mathscr{X} = [-1, 1]^d$ *and* $0 \leq f^* \leq 1$.
- *The derivative of* $f^*(\boldsymbol{x})$ *is bounded by a constant* D.
- *The activation function* σ *is scaling invariant, namely,* $\sigma(kz) = k\sigma(z)$*, and satisfies* $|\sigma(z)| \leq C_1|z|, |\sigma'| \leq C_2$ *and* σ' *is Lipschitz continuous with a positive constant* C_3*. Specifically, we use rectified linear unit (ReLU), i.e.,* $\sigma(z) = max\{z, 0\}$ *as activation function with constant* $C_1 = C_2 = C_3 = 1$.
- $\ln(2d) \geq 1$; *here* d *is the dimension of input data.*

Proposition 2.1 [Proposition of E et al. (2022)]**.** *Let* $f \in C(\mathscr{X})$*, the space of continuous functions on* $\mathscr{X}$*, and assume that* f *satisfies*

$$\gamma(f) := \inf_{\hat{f}} \int_{\mathbb{R}^d} \|w\|_1^2 \,|\hat{f}(\omega)|\,\mathrm{d}\omega \leq \infty, \tag{10}$$

where $\hat{f}$ *is the Fourier transform of an extension of* f *to* $\mathbb{R}^d$*. Then* f *admits an integral representation. Moreover,*

$$\gamma_p(f) \leq 2\gamma(f) + 2\|\nabla f(0)\|_1 + 2|f(0)|. \tag{11}$$

It is usually difficult to check condition (10). For further understanding, we recall the following relationship of Barron space and more classical function spaces, which can be found in E and Stephan (2021) and references therein for details.

Proposition 2.2 [Theorem 3.1 of E and Stephan (2021)]**.** *If* $f \in H^s(\mathbb{R}^d)$ *for* $s > d/2 + 2$*, then* $f \in \mathscr{B}(\mathbb{R}^d)$.

Proposition 2.2 states that every sufficient smooth function admits an integral representation.

3. ERROR ESTIMATES FOR THE GRADIENT-ENHANCED DNN APPROACH

In this section, we present the error estimates for the gradient-enhanced DNN approach. To this aim, we first present the following approximation theorem, which combines the relationship between Barron spaces and two-layer neural networks.

Theorem 3.1. *For any $f \in \mathscr{B}_2(\mathscr{X})$, there exists a two-layer neural network $f(\cdot;\tilde{\theta})$ of width m, such that*

$$\mathbb{E}_{\boldsymbol{x}}\Big[\big(f(\boldsymbol{x}) - f(\boldsymbol{x};\tilde{\theta})\big)^2\Big] \le \frac{3\gamma_2^2(f)}{m}, \tag{12}$$

$$\mathbb{E}_{\boldsymbol{x}}\left[\Big\|\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{x};\tilde{\theta})\Big\|_2^2\right] \le \frac{7\gamma_2^2(f)}{m}, \tag{13}$$

$$\big\|\tilde{\theta}\big\|_{\mathscr{P}} \le 2\gamma_2(f). \tag{14}$$

Proof. Let $(a,\boldsymbol{\pi})$ be the best representation of f, i.e., $\gamma_2^2(f) = \mathbb{E}_{\boldsymbol{\pi}}\big[|a(\boldsymbol{w})|^2\big]$, and let $U = \{\boldsymbol{w}_j\}_{j=1}^m$ be i.i.d. random variables drawn from $\boldsymbol{\pi}(\cdot)$. Define

$$\hat{f}_U(\boldsymbol{x}) := \frac{1}{m}\sum_{j=1}^m a(\boldsymbol{w}_j)\sigma\big(\langle \boldsymbol{w}_j, \boldsymbol{x}\rangle\big).$$

Then the derivative of f with respect to $\boldsymbol{x}$ is

$$\nabla \hat{f}_U(\boldsymbol{x}) = \frac{1}{m}\sum_{j=1}^m a(\boldsymbol{w}_j)\sigma'\big(\langle \boldsymbol{w}_j, \boldsymbol{x}\rangle\big)\boldsymbol{w}_j^T.$$

Let $L_U^1 = \mathbb{E}_{\boldsymbol{x}}\Big[\big|\hat{f}_U(\boldsymbol{x}) - f(\boldsymbol{x})\big|^2\Big]$, $L_U^2 = \mathbb{E}_{\boldsymbol{x}}\Big[\big\|\nabla \hat{f}_U(\boldsymbol{x}) - \nabla f(\boldsymbol{x})\big\|_2^2\Big]$. Then we have

$$\begin{aligned}
\mathbb{E}_U\big[L_U^1\big] &= \mathbb{E}_{\boldsymbol{x}}\mathbb{E}_U\Big[\big|\hat{f}_U(\boldsymbol{x}) - f(\boldsymbol{x})\big|^2\Big] \\
&= \mathbb{E}_{\boldsymbol{x}}\mathbb{E}_U\left[\left|\frac{1}{m}\sum_{j=1}^m a(\boldsymbol{w}_j)\sigma\big(\langle \boldsymbol{w}_j, \boldsymbol{x}\rangle\big) - f(\boldsymbol{x})\right|^2\right] \\
&= \frac{1}{m^2}\mathbb{E}_{\boldsymbol{x}}\mathbb{E}_U\left[\left(\sum_{j=1}^m \big(a(\boldsymbol{w}_j)\,\sigma\big(\langle \boldsymbol{w}_j, \boldsymbol{x}\rangle\big) - f(\boldsymbol{x})\big)\right)\left(\sum_{i=1}^m \big(a(\boldsymbol{w}_i)\,\sigma\big(\langle \boldsymbol{w}_i, \boldsymbol{x}\rangle\big) - f(\boldsymbol{x})\big)\right)\right] \\
&= \frac{1}{m^2}\mathbb{E}_{\boldsymbol{x}}\sum_{i,j=1}^m \mathbb{E}_{\boldsymbol{w}_i,\boldsymbol{w}_j}\Big[\big(a(\boldsymbol{w}_j)\,\sigma\big(\langle \boldsymbol{w}_j, \boldsymbol{x}\rangle\big) - f(\boldsymbol{x})\big)\big(a(\boldsymbol{w}_i)\,\sigma\big(\langle \boldsymbol{w}_i, \boldsymbol{x}\rangle\big) - f(\boldsymbol{x})\big)\Big] \\
&= \frac{1}{m}\mathbb{E}_{\boldsymbol{x}}\mathbb{E}_{\boldsymbol{w}}\Big[\big(a(\boldsymbol{w})\,\sigma\big(\langle \boldsymbol{w}, \boldsymbol{x}\rangle\big) - f(\boldsymbol{x})\big)^2\Big] \\
&\le \frac{1}{m}\mathbb{E}_{\boldsymbol{x}}\mathbb{E}_{\boldsymbol{w}}\Big[\big(a(\boldsymbol{w})\,\sigma\big(\langle \boldsymbol{w}, \boldsymbol{x}\rangle\big)\big)^2\Big] \\
&\le \frac{1}{m}\mathbb{E}_{\boldsymbol{x}}\mathbb{E}_{\boldsymbol{w}}\Big[|a(\boldsymbol{w})|^2\,\big|\sigma\big(\langle \boldsymbol{w}, \boldsymbol{x}\rangle\big)\big|^2\Big] \\
&\le \frac{1}{m}\mathbb{E}_{\boldsymbol{x}}\mathbb{E}_{\boldsymbol{w}}\Big[|a(\boldsymbol{w})|^2\Big] = \frac{1}{m}\mathbb{E}_{\boldsymbol{w}}\Big[|a(\boldsymbol{w})|^2\Big].
\end{aligned}$$

Since $\|\boldsymbol{w}_j\|_1 = 1$ and $\boldsymbol{x} \in [-1,1]^d$, we have $\langle \boldsymbol{w}_j, \boldsymbol{x}\rangle \leq \langle \boldsymbol{w}_j, \boldsymbol{x}_0\rangle \leq \|\boldsymbol{w}_j\|_1 = 1$ where $\boldsymbol{x}_0 = (\boldsymbol{x}_{01}, \boldsymbol{x}_{02}, \ldots, \boldsymbol{x}_{0d})$ and $\boldsymbol{x}_{0i} = \operatorname{sgn}(\boldsymbol{w}_{ji})$, which implies the last inequality. Meanwhile,

$$
\begin{aligned}
\mathbb{E}_U[L_U^2] &= \mathbb{E}_{\boldsymbol{x}}\mathbb{E}_U\left[\left\|\nabla \hat{f}_U(\boldsymbol{x}) - \nabla f(\boldsymbol{x})\right\|_2^2\right] \\
&= \mathbb{E}_{\boldsymbol{x}} E_U\left[\left\|\frac{1}{m}\sum_{j=1}^m a(\boldsymbol{w}_j)\,\sigma'\big(\langle \boldsymbol{w}_j, \boldsymbol{x}\rangle\big)\boldsymbol{w}_j^T - \nabla f(\boldsymbol{x})\right\|_2\right]^2 \\
&= \frac{1}{m^2}\mathbb{E}_{\boldsymbol{x}}\mathbb{E}_U\left[\left(\sum_{j=1}^m \big(a(\boldsymbol{w}_j)\,\sigma'\big(\langle \boldsymbol{w}_j, \boldsymbol{x}\rangle\big)\boldsymbol{w}_j^T - \nabla f(\boldsymbol{x})\big)\right)\right. \\
&\left.\times\left(\sum_{i=1}^m \big(a(\boldsymbol{w}_i)\,\sigma'\big(\langle \boldsymbol{w}_i, \boldsymbol{x}\rangle\big)\boldsymbol{w}_i^T - \nabla f(\boldsymbol{x})\big)\right)\right] \\
&= \frac{1}{m^2}\mathbb{E}_{\boldsymbol{x}}\sum_{i,j=1}^m \mathbb{E}_{\boldsymbol{w}_i,\boldsymbol{w}_j}\Big[\big(a(\boldsymbol{w}_j)\,\sigma'\big(\langle \boldsymbol{w}_j, \boldsymbol{x}\rangle\big)\boldsymbol{w}_j^T - \nabla f(\boldsymbol{x})\big)^T \\
&\times\big(a(\boldsymbol{w}_i)\,\sigma'\big(\langle \boldsymbol{w}_i, \boldsymbol{x}\rangle\big)\boldsymbol{w}_i^T - \nabla f(\boldsymbol{x})\big)\Big] \\
&= \frac{1}{m}\mathbb{E}_{\boldsymbol{x}}\mathbb{E}_{\boldsymbol{w}}\Big[\big\|a(\boldsymbol{w})\,\sigma'\big(\langle \boldsymbol{w}, \boldsymbol{x}\rangle\big)\boldsymbol{w}^T - \nabla f(\boldsymbol{x})\big\|_2^2\Big] \\
&\leq \frac{1}{m}\mathbb{E}_{\boldsymbol{x}}\mathbb{E}_{\boldsymbol{w}}\Big[\big\|a(\boldsymbol{w})\,\sigma'\big(\langle \boldsymbol{w}, \boldsymbol{x}\rangle\big)\boldsymbol{w}^T\big\|_2^2\Big] \\
&\leq \frac{1}{m}\mathbb{E}_{\boldsymbol{x}}\mathbb{E}_{\boldsymbol{w}}\Big[|a(\boldsymbol{w})|^2\,\|\boldsymbol{w}\|_2^2\Big] \\
&\leq \frac{1}{m}\mathbb{E}_{\boldsymbol{x}}\mathbb{E}_{\boldsymbol{w}}\Big[|a(\boldsymbol{w})|^2\,\|\boldsymbol{w}\|_1^2\Big] = \frac{1}{m}\mathbb{E}_{\boldsymbol{x}}\mathbb{E}_{\boldsymbol{w}}\Big[|a(\boldsymbol{w})|^2\Big] = \frac{1}{m}\mathbb{E}_{\boldsymbol{w}}\Big[|a(\boldsymbol{w})|^2\Big].
\end{aligned}
$$

Denote the path-norm of $\hat{f}_U(\boldsymbol{x})$ by A_U; we have $\mathbb{E}[A_U] = \gamma_1(f) \leq \gamma_2(f)$.

Define events $E_1 = \left\{L_U^1 < \dfrac{3\gamma_2^2(f)}{m}\right\}$, $E_2 = \left\{L_U^2 \leq \dfrac{7\gamma_2^2(f)}{m}\right\}$, $E_3 = \big\{A_U < 2\gamma_2(f)\big\}$.

Using Markov's inequality, we have

$$
\begin{aligned}
\mathbb{P}(E_1) &= 1 - \mathbb{P}\left(\left\{L_U^1 \geq \frac{3\gamma_2^2(f)}{m}\right\}\right) \geq 1 - \frac{\mathbb{E}_U[L_U^1]}{3\gamma_2^2(f)/m} \geq 1 - \frac{\gamma_2(f)^2/m}{3\gamma_2^2(f)/m} = \frac{2}{3}, \\
\mathbb{P}(E_2) &= 1 - \mathbb{P}\left(\left\{L_U^2 \geq \frac{7\gamma_2^2(f)}{m}\right\}\right) \geq 1 - \frac{\mathbb{E}_U[L_U^2]}{7\gamma_2^2(f)/m} \geq 1 - \frac{\gamma_2(f)^2/m}{7\gamma_2^2(f)/m} = \frac{6}{7}, \\
\mathbb{P}(E_3) &= 1 - \mathbb{P}\big(\{A_U \geq 2\gamma_2(f)\}\big) \geq 1 - \frac{\mathbb{E}_U[A_U]}{2\gamma_2(f)} \geq 1 - \frac{\gamma_2(f)}{2\gamma_2(f)} = \frac{1}{2}.
\end{aligned}
$$

Therefore, let $E_4 = E_1 \cap E_3$, then

$$\mathbb{P}(E_1 \cap E_3) \geq \mathbb{P}(E_1) + \mathbb{P}(E_3) - 1 = \frac{1}{6}.$$

Hence,

$$\mathbb{P}(E_1 \cap E_2 \cap E_3) = \mathbb{P}(E_4 \cap E_2) \geq \mathbb{P}(E_4) + \mathbb{P}(E_2) - 1 \geq \frac{1}{6} + \frac{6}{7} - 1 = \frac{1}{42} > 0.$$

That is, $E_1 \cap E_2 \cap E_3 \neq \varnothing$, which completes the proof. □

According to Theorem 3.1, for a function in Barron space, we can use a two-layer neural network to simultaneously approximate its function value and derivative. Next we will investigate the posterior error estimation of the gradient-enhanced method.

3.1 *A Posteriori* Error Estimation

To give *a posteriori* error estimation, we need to introduce the definition of the Rademacher complexity and review some related conclusions.

Definition 3.1 [Rademacher complexity (Shalev-Shwartz and Ben-David, 2014)]**.** Let $S = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$ be n i.i.d samples, and let $\mathscr{F} \circ S$ be the set of all possible evaluations a function $f \in \mathscr{F}$ can achieve on a sample S, namely,

$$\mathscr{F} \circ S = \Big\{ \big(f(\boldsymbol{x}_1), f(\boldsymbol{x}_2), \ldots, f(\boldsymbol{x}_n)\big) \,\Big|\, f \in \mathscr{F} \Big\}.$$

Let each component of random variable ξ be i.i.d. according to $\mathbb{P}[\xi_i = 1] = \mathbb{P}[\xi_i = -1] = 1/2$. Then, the Rademacher complexity of $\mathscr{F}$ with respect to S is defined as follows:

$$\mathscr{R}_n(\mathscr{F} \circ S) := \frac{1}{n} \mathbb{E}_{\boldsymbol{\xi} \sim \{\pm 1\}^n} \left[\sup_{f \in \mathscr{F}} \sum_{i=1}^{n} \xi_i f(\boldsymbol{x}_i) \right]. \tag{15}$$

Lemma 3.1 [Lemma 26.11 of Shalev-Shwartz and Ben-David (2014)]**.** *Let* $S = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$ *be vectors in* $\mathbb{R}^d$, $\mathscr{H}_1 = \{\boldsymbol{x} \mapsto \langle \boldsymbol{w}, \boldsymbol{x} \rangle \,|\, \|\boldsymbol{w}\|_1 \leq 1\}$. *Then,*

$$\mathscr{R}_n(\mathscr{H}_1 \circ S) \leq \max_i \|\boldsymbol{x}_i\|_\infty \sqrt{\frac{2\ln(2d)}{n}}.$$

Lemma 3.2 [Lemma 26.9 of Shalev-Shwartz and Ben-David (2014)]**.** *For each* $i \in [n] = \{1, 2, \ldots, n\}$, *let* $\phi_i : \mathbb{R} \mapsto \mathbb{R}$ *be a* ρ*-Lipschitz continuous, namely, for all* $\alpha, \beta \in \mathbb{R}$ *we have* $|\phi_i(\alpha) - \phi_i(\beta)| \leq \rho|\alpha - \beta|$. *For* $\boldsymbol{a} \in \mathbb{R}^n$, *let* $\boldsymbol{\phi}(\boldsymbol{a})$ *denote the vector* $\big(\phi_1(a_1), \ldots, \phi_n(a_n)\big)$. *Let* $\boldsymbol{\phi} \circ A = \{\boldsymbol{\phi}(\boldsymbol{a}) : \boldsymbol{a} \in A\}$. *Then,*

$$\mathscr{R}_n(\boldsymbol{\phi} \circ A) \leq \rho \mathscr{R}_n(A).$$

Lemma 3.3 [Lemma 26.6 of Shalev-Shwartz and Ben-David (2014)]**.** *For any $A \subset \mathbb{R}^n$, scale $c \in \mathbb{R}$, and vector $\boldsymbol{a}_0 \in \mathbb{R}^n$, we have*

$$\mathscr{R}_n\big(\{c\boldsymbol{a} + \boldsymbol{a}_0 \mid \boldsymbol{a} \in A\}\big) \leq |c|\, \mathscr{R}_n(A).$$

Lemma 3.4. *For any $A, B \subset \mathbb{R}^n$, we have*

$$\mathscr{R}_n(A + B) \leq \mathscr{R}_n(A) + \mathscr{R}_n(B).$$

Here $A + B = \{a + b \mid a \in A, b \in B\}$.

Lemma 3.5 [Lemma B.3 of E et al. (2019)]**.** *Let $\mathscr{F}_Q = \{f(\boldsymbol{x}; \theta) \mid \|\theta\|_{\mathscr{P}} \leq Q\}$ be the set of two-layer networks with path-norm bounded by Q; then we have*

$$\mathscr{R}_n(\mathscr{F}_Q \circ S) \leq 2Q\sqrt{\frac{2\ln(2d)}{n}}.$$

Lemma 3.6 [Vector valued Rademacher complexity (Maurer, 2016)]**.** *Let $\mathscr{X}$ be any set, $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) \in \mathscr{X}^n$, let $\mathscr{F}$ be a class of functions $\boldsymbol{f} : \mathscr{X} \to \ell_2$, and let $h_i : \ell_2 \to \mathbb{R}$ have Lipschitz norm Lip, where ℓ_2 denote the Hilbert space of square summable sequences of real numbers. Then*

$$\mathbb{E}\left[\sup_{f \in \mathscr{F}} \sum_{i=1}^{n} \xi_i h_i\big(\boldsymbol{f}(\boldsymbol{x}_i)\big)\right] \leq \sqrt{2}\,\mathrm{Lip} \cdot \mathbb{E}\left[\sup_{f \in \mathscr{F}} \sum_{i=1}^{n} \sum_{k=1}^{d} \xi_{ik} f_k(\boldsymbol{x}_i)\right], \tag{16}$$

where ξ_{ik} is an independent doubly indexed Rademacher sequence according to probability distribution $\mathbb{P}[\xi_{ik} = -1] = \mathbb{P}[\xi_{ik} = 1] = 1/2$ and $f_k(\boldsymbol{x}_i)$ is the kth component of $\boldsymbol{f}(\boldsymbol{x}_i)$.

Lemma 3.7. *Let $S = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$. $\{\boldsymbol{y}_i'\}_{i=1}^n$ denotes the gradient information, where $\boldsymbol{y}_i' \in \mathbb{R}^d\ \forall i \in [n]$.*

$$\mathscr{F}_{Q,j}' = \{\nabla_j f(\boldsymbol{x}; \theta) \mid \|\theta\|_{\mathscr{P}} \leq Q\}, \quad j = 1, \ldots, d, \quad \mathscr{F}_Q' = \Pi_{j=1}^d \mathscr{F}_{Q,j}'.$$

Define $\boldsymbol{g} : \mathscr{X} \to \ell_2$, $\boldsymbol{g}(\boldsymbol{x}; \theta) = \big(\partial_1 f(\boldsymbol{x}; \theta), \partial_2 f(\boldsymbol{x}; \theta), \ldots, \partial_d f(\boldsymbol{x}; \theta)\big)^{\mathrm{T}}$ and $\tilde{\ell}_j : \ell_2 \to \mathbb{R}$, $\tilde{\ell}_j(\boldsymbol{z}) = \|\boldsymbol{z} - \boldsymbol{y}_j'\|_2$ for each $j \in \{1, 2, \ldots, n\}$, where ℓ_2 denote the Hilbert space of square summable sequences of real numbers. Note that $\tilde{\ell}_j$ is 1-Lipschitz function; then we have

$$\mathbb{E}\left[\sup_{\|\theta\|_{\mathscr{P}} \leq Q} \sum_{i=1}^{n} \xi_i \tilde{\ell}_i\big(\boldsymbol{g}(\boldsymbol{x}_i; \theta)\big)\right] \leq \sqrt{2} \cdot \mathbb{E}\left[\sup_{\|\theta\|_{\mathscr{P}} \leq Q} \sum_{i=1}^{n} \sum_{k=1}^{d} \xi_{ik} g_k(\boldsymbol{x}_i; \theta)\right],$$

where $g_k(\boldsymbol{x}_i; \theta)$ is the kth component of $\boldsymbol{g}(\boldsymbol{x}_i; \theta)$. Moreover, denote $\tilde{\boldsymbol{\ell}} = (\tilde{\ell}_1, \tilde{\ell}_2, \ldots, \tilde{\ell}_n)$; we have

$$\mathscr{R}_n(\tilde{\boldsymbol{\ell}} \circ \mathscr{F}_Q' \circ S) \leq 2\sqrt{2}\, Qd\sqrt{\frac{2\ln(2d)}{n}}.$$

Proof. Without loss of generality, let $\|\boldsymbol{w}_j\|_1 = 1,\ \forall j = \{1, \ldots, m\}$. Applying Lemma 3.6, we immediately obtain

$$
\begin{aligned}
\mathscr{R}_n(\tilde{\boldsymbol{\ell}} \circ \mathscr{F}'_Q \circ S) &= \frac{1}{n}\mathbb{E}_{\boldsymbol{\xi}}\left[\sup_{\|\boldsymbol{\theta}\|_{\mathscr{P}} \le Q} \sum_{i=1}^{n} \xi_i \left\|\boldsymbol{g}(\boldsymbol{x}_i;\boldsymbol{\theta}) - \boldsymbol{y}'_i\right\|_2\right] \\
&\le \frac{\sqrt{2}}{n}\mathbb{E}_{\boldsymbol{\xi}}\left[\sup_{\|\boldsymbol{\theta}\|_{\mathscr{P}} \le Q} \sum_{k=1}^{d}\sum_{i=1}^{n} \xi_{ik} g_k(\boldsymbol{x}_i;\boldsymbol{\theta})\right] \\
&= \frac{\sqrt{2}}{n}\mathbb{E}_{\boldsymbol{\xi}}\left[\sup_{\|\boldsymbol{\theta}\|_{\mathscr{P}} \le Q} \sum_{k=1}^{d}\sum_{i=1}^{n} \xi_{ik} \sum_{j=1}^{m} a_j \sigma'\big(\boldsymbol{w}_j^T \boldsymbol{x}_i\big)\boldsymbol{w}_{jk}\right] \\
&= \frac{\sqrt{2}}{n}\mathbb{E}_{\boldsymbol{\xi}}\left[\sup_{\|\boldsymbol{\theta}\|_{\mathscr{P}} \le Q} \sum_{j=1}^{m} a_j \|\boldsymbol{w}_j\|_1 \sum_{k=1}^{d}\sum_{i=1}^{n} \xi_{ik} \sigma'\big(\boldsymbol{w}_j^T \boldsymbol{x}_i\big)\boldsymbol{w}_{jk}\right] \\
&\le \frac{\sqrt{2}}{n}\mathbb{E}_{\boldsymbol{\xi}}\left[\sup_{\|\boldsymbol{\theta}\|_{\mathscr{P}} \le Q} \sum_{j=1}^{m} |a_j| \|\boldsymbol{w}_j\|_1 \sup_{\|\boldsymbol{v}\|_1=1} \left|\sum_{k=1}^{d}\sum_{i=1}^{n} \xi_{ik} \sigma'\big(\boldsymbol{v}^T \boldsymbol{x}_i\big)\boldsymbol{v}_k\right|\right] \\
&\le \frac{\sqrt{2}Q}{n}\mathbb{E}_{\boldsymbol{\xi}}\left[\sup_{\|\boldsymbol{v}\|_1=1} \left|\sum_{k=1}^{d}\sum_{i=1}^{n} \xi_{ik} \sigma'\big(\boldsymbol{v}^T \boldsymbol{x}_i\big)\boldsymbol{v}_k\right|\right] \\
&= \frac{\sqrt{2}Q}{n}\mathbb{E}_{\boldsymbol{\eta}}\left[\sup_{\|\boldsymbol{v}\|_1=1} \left|\Big\langle \sum_{i=1}^{n} \boldsymbol{\eta}_i \sigma'\big(\boldsymbol{v}^T \boldsymbol{x}_i\big), \boldsymbol{v}\Big\rangle\right|\right] \quad \big(\boldsymbol{\eta}_i = (\xi_{i1}, \xi_{i2}, \ldots, \xi_{id})\big) \\
&\le \frac{\sqrt{2}Q}{n}\mathbb{E}_{\boldsymbol{\eta}}\left[\sup_{\|\boldsymbol{v}\|_1=1} \left\|\sum_{i=1}^{n} \boldsymbol{\eta}_i \sigma'\big(\boldsymbol{v}^T \boldsymbol{x}_i\big)\right\|_1\right] \\
&\le \frac{\sqrt{2}Qd}{n}\mathbb{E}_{\boldsymbol{\xi}}\left[\sup_{\|\boldsymbol{v}\|_1\le 1} \sum_{i=1}^{n} \xi_i \sigma'\big(\boldsymbol{v}^T \boldsymbol{x}_i\big)\right],
\end{aligned}
$$

where $\xi \in [-1,1]$ and $\boldsymbol{\eta} \in [-1,1]^d$. Due to the symmetry, we have

$$
\begin{aligned}
\mathbb{E}_{\boldsymbol{\xi}}\left[\sup_{\|\boldsymbol{u}\|_1\le 1} \left|\sum_{i=1}^{n} \xi_i \sigma'\big(\boldsymbol{u}^T \boldsymbol{x}_i\big)\right|\right] &\le \mathbb{E}_{\boldsymbol{\xi}}\left[\sup_{\|\boldsymbol{u}\|_1\le 1} \sum_{i=1}^{n} \xi_i \sigma'\big(\boldsymbol{u}^T \boldsymbol{x}_i\big) + \sup_{\|\boldsymbol{u}\|_1\le 1} \sum_{i=1}^{n} -\xi_i \sigma'\big(\boldsymbol{u}^T \boldsymbol{x}_i\big)\right] \\
&\le 2\mathbb{E}_{\boldsymbol{\xi}}\left[\sup_{\|\boldsymbol{u}\|_1\le 1} \sum_{i=1}^{n} \xi_i \sigma'\big(\boldsymbol{u}^T \boldsymbol{x}_i\big)\right].
\end{aligned}
$$

Then applying Lemma 3.2 and Lemma 3.1, we obtain

$$
\mathscr{R}_n(\tilde{\boldsymbol{\ell}} \circ \mathscr{F}'_Q \circ S) \le 2\sqrt{2}\, Qd\sqrt{\frac{2\ln(2d)}{n}}.
$$

□

Theorem 3.2 (Shalev-Shwartz and Ben-David, 2014)**.** *Assume that for all samples $\boldsymbol{x}$ and h in hypothesis space $\mathscr{H}$, the loss function $\ell : \mathscr{H} \times \mathscr{Y} \to \mathbb{R}$ satisfies $|\ell(h(\boldsymbol{x}), y)| \le B$. Then, with probability of at least $1-\delta$, for all $h \in \mathscr{H}$, and $S = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$,*

$$\left|\frac{1}{n}\sum_{i=1}^{n}\ell\big(h(\boldsymbol{x}_i),y_i\big)-\mathbb{E}_{\boldsymbol{x},y}\big[\ell(h(\boldsymbol{x}),y)\big]\right|\le 2\mathbb{E}_S\big[\mathscr{R}_n(\ell\circ\mathscr{H}\circ S)\big]+B\sqrt{\frac{2\ln(2/\delta)}{n}}. \quad (17)$$

We are now ready to present the main results of this section.

Theorem 3.3. *If Assumption 2.1 holds, then with probability at least* $1-\delta$ *we have,*

$$\begin{aligned}&\sup_{\|\theta\|_{\mathscr{P}}\le Q}\Big|L(\theta)+\beta L'(\theta)-\big(L_n(\theta)+\beta L_n'(\theta)\big)\Big|\\&\le 4\big(1+\sqrt{2}\beta d\big)Q\sqrt{\frac{2\ln(2d)}{n}}+\left(\frac{1}{2}+\beta(Q+D)\right)\sqrt{\frac{2\ln(2/\delta)}{n}}.\end{aligned} \quad (18)$$

Proof. Using the triangular inequality, we have

$$\begin{aligned}&\sup_{\|\theta\|_{\mathscr{P}}\le Q}\Big|L(\theta)+\beta L'(\theta)-\big(L_n(\theta)+\beta L_n'(\theta)\big)\Big|\\&\le\sup_{\|\theta\|_{\mathscr{P}}\le Q}\big|L(\theta)-L_n(\theta)\big|+\sup_{\|\theta\|_{\mathscr{P}}\le Q}\beta\big|L'(\theta)-L_n'(\theta)\big|.\end{aligned} \quad (19)$$

Define $\mathscr{H}_Q=\Big\{\ell\big(f(\boldsymbol{x};\theta),y\big)\,\big|\,f(\boldsymbol{x};\theta)\in\mathscr{F}_Q\Big\}$ and $\mathscr{H}_Q'=\Big\{\tilde{\ell}\big(\nabla f(\boldsymbol{x};\theta),\boldsymbol{y}'\big)\mid f(\boldsymbol{x};\theta)\in\mathscr{F}_Q\Big\}$, where $\ell\big(f(\boldsymbol{x};\theta),y\big)=(1/2)\big(f(\boldsymbol{x};\theta)-y\big)^2$ and $\tilde{\ell}\big(\nabla f(\boldsymbol{x};\theta),\boldsymbol{y}'\big)=\big\|\nabla f(\boldsymbol{x};\theta)-\boldsymbol{y}'\big\|_2$.

Note that $f(\boldsymbol{x};\theta)\in[0,1]$, $\ell(\cdot,y)$ is 1-Lipschitz continuous,

$$\begin{aligned}&\frac{1}{2}\Big|\big(f(\boldsymbol{x}_i;\theta_f)-y_i\big)^2-\big(g(\boldsymbol{x}_i;\theta_g)-y_i\big)^2\Big|\\&=\frac{1}{2}|f(\boldsymbol{x}_i;\theta_f)-g(\boldsymbol{x}_i;\theta_g)|\cdot|f(\boldsymbol{x}_i;\theta_f)+g(\boldsymbol{x}_i;\theta_g)-2y_i|\le|f(\boldsymbol{x}_i;\theta_f)-g(\boldsymbol{x}_i;\theta_g)|.\end{aligned}$$

Following Lemma 3.5 and Lemma 3.2, we have

$$\mathscr{R}_n(\mathscr{H}_Q)=\frac{1}{n}\mathbb{E}_{\boldsymbol{\xi}}\left[\sup_{f\in\mathscr{F}_Q}\sum_{i=1}^{n}\xi_i\frac{1}{2}|f(\boldsymbol{x}_i;\theta)-y_i|^2\right]\le 2Q\sqrt{\frac{2\ln(2d)}{n}}, \quad (20)$$

and ℓ is bounded, $\ell\big(f(\boldsymbol{x};\theta),y\big)=(1/2)(f(\boldsymbol{x};\theta)-y)^2\le 1/2$. Hence applying Theorem 3.2, we can obtain,

$$\sup_{\|\theta\|_{\mathscr{P}}\le Q}\big|L(\theta)-L_n(\theta)\big|\le 4Q\sqrt{\frac{2\ln(2d)}{n}}+\frac{1}{2}\sqrt{\frac{2\ln(2/\delta)}{n}}. \quad (21)$$

Similarly, $\tilde{\ell}\big(\nabla f(\boldsymbol{x};\theta),\boldsymbol{y}'\big)=\big\|\nabla f(\boldsymbol{x};\theta)-\boldsymbol{y}'\big\|_2$ is 1-Lipschitz with respect to $\nabla f(\boldsymbol{x};\theta)$,

$$\Big|\big\|\nabla f(\boldsymbol{x}_i;\theta_f)-\boldsymbol{y}_i'\big\|_2-\big\|\nabla g(\boldsymbol{x}_i;\theta_g)-\boldsymbol{y}_i'\big\|_2\Big|\le\big\|\nabla f(\boldsymbol{x}_i;\theta_f)-\nabla g(\boldsymbol{x}_i;\theta_g)\big\|_2.$$

Let $\tilde{\boldsymbol{\ell}}=(\tilde{\ell}_1,\ldots,\tilde{\ell}_n)$, in which $\tilde{\ell}_i(\cdot)=\big\|\cdot-\boldsymbol{y}_i'\big\|_2$ for $i=1,\ldots,n$. We can obtain the estimation of $\mathscr{R}_n(\mathscr{H}_Q')$ directly from Lemma 3.7,

$$\mathscr{R}_n(\mathscr{H}_Q')=\frac{1}{n}\mathbb{E}_{\boldsymbol{\xi}}\left[\sup_{f\in\mathscr{F}_Q}\sum_{i=1}^{n}\xi_i\Big\|\nabla f(\boldsymbol{x}_i;\theta)-\boldsymbol{y}_i'\Big\|_2\right]\le 2\sqrt{2}\,Qd\sqrt{\frac{2\ln(2d)}{n}}, \quad (22)$$

and $\tilde{\ell}\big(\nabla f(\boldsymbol{x};\theta),\boldsymbol{y}'\big)$ is bounded since we assume that the gradient of the objective function is bounded by a positive constant D:

$$\begin{aligned} f(\boldsymbol{x};\theta) &= \sum_{k=1}^{m} a_k \sigma\big(\boldsymbol{w}_k^T \boldsymbol{x}\big), \\ \big\|\nabla f(\boldsymbol{x};\theta)\big\|_2^2 &= \left\|\sum_{k=1}^{m} a_k \sigma' \boldsymbol{w}_k^T\right\|_2^2 \le \left\|\sum_{k=1}^{m} a_k \sigma' \boldsymbol{w}_k^T\right\|_1^2 \le \left\|\sum_{k=1}^{m} a_k \boldsymbol{w}_k^T\right\|_1^2 \le \big\|\theta_f\big\|_{\mathscr{P}}^2, \end{aligned} \tag{23}$$

$$\tilde{\ell}\big(\nabla f(\boldsymbol{x};\theta),\boldsymbol{y}'\big) = \big\|\nabla f(\boldsymbol{x};\theta) - \boldsymbol{y}'\big\|_2 \le \big\|\nabla f(\boldsymbol{x};\theta)\big\|_2 + \big\|\boldsymbol{y}'\big\| \le \big\|\theta_f\big\|_{\mathscr{P}} + D \le Q + D. \tag{24}$$

Thus,

$$\sup_{\|\theta\|_{\mathscr{P}} \le Q} \big|L'(\theta) - L_n'(\theta)\big| \le 4\sqrt{2}\, Q d \sqrt{\frac{2\ln(2d)}{n}} + (Q+D)\sqrt{\frac{2\ln(2/\delta)}{n}}. \tag{25}$$

The desired result follows by plugging Eqs. (21) and (25) into Eq. (19), and the proof is completed. □

We next present a posterior generalization bound by relaxing such restrictions.

Theorem 3.4 (A posterior generalization bound)**.** *Assume that Assumption 2.1 holds, then for any $\delta > 0$, with probability at least $1-\delta$ over the choice of the training set S, we have, for any two-layer network $f_m(\cdot,\theta)$,*

$$\begin{aligned} \Big|L(\theta) + \beta L'(\theta) - \big(L_n(\theta) + \beta L_n'(\theta)\big)\Big| &\le 4\big(1+\sqrt{2}\beta d\big)\big(\|\theta\|_{\mathscr{P}} + 1\big)\sqrt{\frac{2\ln(2d)}{n}} \\ &\quad + \left(\frac{1}{2} + \beta\big(\|\theta\|_{\mathscr{P}} + 1 + D\big)\right)\sqrt{\frac{2\ln\Big(2c\big(\|\theta\|_{\mathscr{P}}+1\big)^2/\delta\Big)}{n}}, \end{aligned} \tag{26}$$

where $c = \sum\limits_{k=1}^{\infty} 1/k^2$.

Proof. Consider the decomposition $\mathscr{F} = \cup_{k=1}^{\infty}\mathscr{F}_k$, where $\mathscr{F}_k = \big\{f(\boldsymbol{x};\theta) \mid \|\theta\|_{\mathscr{P}} \le k\big\}$. Let $\delta_k = \delta/(ck^2)$ where $c = \sum_{k=1}^{\infty} 1/k^2$. According to Theorem 3.3, if we fix k in advance, then with probability at least $1-\delta_k$ over the choice of S, we have

$$\begin{aligned} &\Big|L(\theta) + \beta L'(\theta) - \big(L_n(\theta) + \beta L_n'(\theta)\big)\Big| \\ &\le 4\big(1+\sqrt{2}\beta d\big)k\sqrt{\frac{2\ln(2d)}{n}} + \left(\frac{1}{2} + \beta\big(k+D\big)\right)\sqrt{\frac{2\ln(2/\delta_k)}{n}}. \end{aligned} \tag{27}$$

Therefore $\mathbb{P}\big(\{\text{inequality (27) unholds}\}\big) \le \sum_{k=1}^{\infty}\delta_k =: \delta$, namely, $\mathbb{P}\big(\{\text{inequality (27) holds for all } k\}\big) \ge 1-\delta$. In other words, with probability at least $1-\delta$, the inequality (27) holds for all k.

Given an arbitrary set of parameters θ, denote $k_0 = \min\{k \mid \|\theta\|_{\mathscr{P}} \le k\}$, then $k_0 \le \|\theta\|_{\mathscr{P}} + 1$. Inequality (27) implies that

$$\begin{aligned}
&\Big|L(\theta) + \beta L'(\theta) - \big(L_n(\theta) + \beta L_n'(\theta)\big)\Big| \\
&\quad\le 4\big(1+\sqrt{2}\beta d\big)k_0\sqrt{\frac{2\ln(2d)}{n}} + \left(\frac{1}{2} + \beta\Big(k_0 + D\Big)\right)\sqrt{\frac{2\ln(2ck_0^2/\delta)}{n}}. \\
&\quad\le 4\big(1+\sqrt{2}\beta d\big)\big(\|\theta\|_{\mathscr{P}} + 1\big)\sqrt{\frac{2\ln(2d)}{n}} \\
&\quad+ \left(\frac{1}{2} + \beta\big(\|\theta\|_{\mathscr{P}} + 1 + D\big)\right)\sqrt{\frac{2\ln\Big(2c\big(\|\theta\|_{\mathscr{P}}+1\big)^2/\delta\Big)}{n}}.
\end{aligned}$$

This completes the proof. □

We note that the generalization gap is bounded roughly by

$$d\|\theta\|_{\mathscr{P}}\sqrt{\frac{\ln(2d)}{n}} + \|\theta\|_{\mathscr{P}}\frac{\ln(\|\theta\|_{\mathscr{P}})}{\sqrt{n}},$$

which shares a similar convergence rate with the method without gradient information (E et al., 2019), indicating that the gradient-enhanced method would not destroy the original function approximation algorithm.

3.2 An Upper Bound for the Empirical Risk

Recalling the approximation property, there exists a two-layer neural network $f(\cdot;\tilde{\theta})$ whose path-norm is independent of the network width, while achieving the optimal approximation error. Furthermore, this path-norm can also be used to bound the generalization gap (Theorem 3.4). In this section, we want to estimate the gradient regularized risk of $\tilde{\theta}$. To this end, we first assume that the norm of the gradient of the target function can be bounded by a constant D.

Let $\hat{\gamma}_p(f) = \max\{1, \gamma_p(f)\}$, where d is the dimension of input data.

Theorem 3.5. *Let $\tilde{\theta}$ be the network mentioned in Theorem 3.1; then with probability at least $1-\delta$, we have*

$$J_{n,\beta}(\tilde{\theta}) \lesssim \frac{\gamma_2^2(f^*)}{m} + \beta\frac{1}{\sqrt{m}}\gamma_2(f^*) + \beta\hat{\gamma}_2(f^*)\big(\hat{\gamma}_2(f^*)+d\big)\sqrt{\frac{2\ln(2d)}{n}} + \beta\hat{\gamma}_2(f^*)\sqrt{\frac{\ln(2c/\delta)}{n}}, \tag{28}$$

where

$$J_{n,\beta}(\tilde{\theta}) = L_n(\tilde{\theta}) + \beta L_n'(\tilde{\theta}) \quad \text{and} \quad c = \sum_{k=1}^{\infty}\frac{1}{k^2}.$$

Proof. According to the definition of a regularized model, the properties that

$$\big\|\tilde{\theta}\big\|_{\mathscr{P}} \le 2\gamma_2(f^*), \quad L(\tilde{\theta}) \le \frac{3\gamma_2^2(f^*)}{m}, \quad \big(L'(\tilde{\theta})\big)^2 \le \mathbb{E}_x\Big[\big\|\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{x};\tilde{\theta})\big\|_2^2\Big] \le \frac{7\gamma_2^2(f^*)}{m},$$

and the posteriori error bound, the regularized risk of $\tilde{\theta}$ satisfies

$$\begin{aligned}
J_{n,\beta}(\tilde{\theta}) &= L_n(\tilde{\theta}) + \beta L'_n(\tilde{\theta}) \\
&\le L(\tilde{\theta}) + \beta L'(\tilde{\theta}) + 4\big(1+\sqrt{2}\beta d\big)\big(\|\tilde{\theta}\|_{\mathscr{P}} + \big)\sqrt{\frac{2\ln(2d)}{n}} \\
&\quad + \left(\frac{1}{2} + \beta\big(\|\tilde{\theta}\|_{\mathscr{P}} + 1 + D\big)\right)\sqrt{\frac{2\ln\Big(2c\big(\|\tilde{\theta}\|_{\mathscr{P}} + 1\big)^2/\delta\Big)}{n}} \\
&\le L(\tilde{\theta}) + \beta L'(\tilde{\theta}) + 4\big(1+\sqrt{2}\beta d\big)\big(2\gamma_2(f^*) + 1\big)\sqrt{\frac{2\ln(2d)}{n}} \qquad (29) \\
&\quad + \left(\frac{1}{2} + \beta\big(2\gamma_2(f^*) + 1 + D\big)\right)\sqrt{\frac{2\ln\Big(2c\big(1+2\gamma_2(f^*)\big)^2/\delta\Big)}{n}}. \qquad (30)
\end{aligned}$$

The last term can be simplified by using $\sqrt{a+b} \le \sqrt{a} + \sqrt{b}$ and $\ln(a) \le a$ for $a \ge 0$, $b \ge 0$. Thus we have

$$\begin{aligned}
\sqrt{2\ln\Big(2c\big(1+2\gamma_2(f^*)\big)^2/\delta\Big)} &= \sqrt{2\ln(2c/\delta) + 2\ln\big(1+2\gamma_2(f^*)\big)^2} \\
&\le \sqrt{2\ln(2c/\delta)} + \sqrt{2\ln\big(1+2\gamma_2(f^*)\big)^2} \\
&\le \sqrt{2\ln(2c/\delta)} + \sqrt{2\ln\big(3\hat{\gamma}_2(f^*)\big)^2} \\
&\le \sqrt{2\ln(2c/\delta)} + 3\sqrt{2}\,\hat{\gamma}_2(f^*).
\end{aligned}$$

Plugging it into Eq. (30), we obtain

$$\begin{aligned}
J_{n,\beta}(\tilde{\theta}) \le L(\tilde{\theta}) + \beta L'(\tilde{\theta}) &+ 4\big(1+\sqrt{2}\,\beta d\big)\big(2\gamma_2(f^*) + 1\big)\sqrt{\frac{2\ln(2d)}{n}} \\
&+ \left(\frac{1}{2} + \beta\big(2\gamma_2(f^*) + 1 + D\big)\right)\left(\sqrt{\frac{2\ln(2c/\delta)}{n}} + 3\sqrt{\frac{2}{n}}\hat{\gamma}_2(f^*)\right).
\end{aligned}$$

Thus after some simplifications, we have

$$J_{n,\beta}(\tilde{\theta}) \lesssim \frac{\gamma_2^2(f^*)}{m} + \beta\frac{1}{\sqrt{m}}\gamma_2(f^*) + \beta\hat{\gamma}_2(f^*)\big(\hat{\gamma}_2(f^*) + d\big)\sqrt{\frac{2\ln(2d)}{n}} + \beta\hat{\gamma}_2(f^*)\sqrt{\frac{\ln(2c/\delta)}{n}}.$$

This completes the proof. □

According to the definition of $\theta_{n,\beta}$, we have $J_{n,\beta}(\theta_{n,\beta}) \le J_{n,\beta}(\tilde{\theta})$. Thus the above theorem gives an upper bound for $J_{n,\beta}(\theta_{n,\beta})$. Such an upper bound for the empirical risk verifies the feasibility of our method that we can obtain a good approximation by increasing the number of neurons in the hidden layer and samples.

4. APPLICATIONS TO UNCERTAINTY QUANTIFICATION

We now consider the application of the gradient-enhanced DNN approach for uncertainty quantification.

4.1 Gradient-Enhanced Uncertainty Quantification

In complex engineering systems, mathematical models can only serve as simplified and reduced representations of true physics, and the effect of some uncertainties, such as boundary/initial conditions and parameter values, can be significant. Uncertainty quantification (UQ) aims to develop numerical tools that can accurately predict quantities of interest (QoI) and facilitate the quantitative validation of the simulation model. Generally, we use differential equations to model complex systems on a domain Ω, in which the uncertainty sources are represented by Ξ. The solution u is governed by the PDEs

$$\begin{aligned} \mathscr{L}\big(\boldsymbol{x}, \Xi; u(\boldsymbol{x}, \Xi)\big) &= 0, \qquad \boldsymbol{x} \in \Omega, \\ \mathscr{B}\big(\boldsymbol{x}, \Xi; u(\boldsymbol{x}, \Xi)\big) &= 0, \qquad \boldsymbol{x} \in \partial\Omega, \end{aligned} \tag{31}$$

where $\mathscr{L}$ and $\mathscr{B}$ are differential and boundary operators, respectively. Our goal is to approximate the QoI $u(\boldsymbol{x}_0, \Xi)$ for some fixed spatial location $\boldsymbol{x}_0$. To reduce the notation, we simply write $u(\Xi)$. In many applications, the dimension of random variable Ξ is very high and can be characterized by a d-dimensional random variable. Hence DNNs are good candidates for such problems.

We consider inclusion of gradient measurements with respect to random variables Ξ, i.e., $\partial u/\partial \Xi_k, k = 1, 2, \ldots, d$. The gradient measurements can usually be obtained in a relatively inexpensive way via the adjoint method (Luchini and Bottaro, 2014).

4.2 Numerical Examples

We compare the performance between original neural networks without gradient input and gradient-enhanced neural networks. For simplicity, we say that a method is $X\%$ gradient-enhanced if $X\%$ of samples contain derivative information with respect to all dimensions. In our tests, each neural network contains two layers with 1000 hidden neurons. The hyperparameter β, which is used to balance the two part losses introduced by the function values and derivative information, is set to 10. We initialize all trainable parameters using the Glorot normal scheme. For the training procedure, we use the Adam optimizer. To quantitatively evaluate the accuracy of the numerical solution, we shall consider the relative L^2 error $\|u_\theta - u\|_2/\|u\|_2$, where u and u_θ denote the ground truth and predicted solution. All numerical tests are implemented in PyTorch.

4.2.1 Function Approximations

Before applying the gradient-enhanced method to uncertainty quantification, we first demonstrate the effectiveness of our approach in approximating high-dimensional functions. More precisely, we consider the Gaussian function,

$$f_1(\boldsymbol{x}) = \exp\left(-\sum_{i=1}^{d} x_i^2\right), \quad \boldsymbol{x} = (x_1, x_2, \ldots, x_d) \in [-1, 1]^d,$$

and the polynomial function,

$$f_2(\boldsymbol{x}) = \sum_{i=1}^{d/2} x_i x_{i+1}, \quad \boldsymbol{x} = (x_1, x_2, \ldots, x_d) \in [-1, 1]^d.$$

For these two test functions, we assume that samples $\{\boldsymbol{x}_i\}_{i=1}^n$ are uniformly distributed in $[-1,1]^d$, y_i is the observation of target function at $\boldsymbol{x}_i$, and $\boldsymbol{y}_i'$ is the corresponding derivative. Thus $\{\boldsymbol{x}_i, y_i\}_{i=1}^n$ compose the training data for the original DNN. $\{\boldsymbol{x}_i, y_i, \boldsymbol{y}_i'\}_{i=1}^n$ compose the training data for 100% gradient-enhanced DNN. And $\{\boldsymbol{x}_i, y_i\}_{i=1}^n \cup \{\hat{\boldsymbol{x}}_j, \hat{\boldsymbol{y}}_j'\}_{j=1}^m$ compose the training data for 20% gradient-enhanced DNN where m is the rounding-off of $20\%n$ and $\{\hat{\boldsymbol{x}}_j\}_{j=1}^m$ is randomly chosen from $\{\boldsymbol{x}_i\}_{i=1}^n$. The learning rate for the Adam optimizer is set to 0.01 with 20% decay each 500 steps.

For Gaussian function $f_1(\boldsymbol{x})$, we consider the cases that $d = 2$, 4, and 8. The relative L^2 errors against the number of samples n are presented in top row of Fig. 1. The use of gradient information can indeed improve the accuracy, and furthermore, the more gradient information is included, the better accuracy is obtained. Moreover, we investigate the loss functions of different models for $d = 2$ with 400 samples, $d = 4$ with 1600 samples, and $d = 8$ with 3200 samples, which are depicted in the bottom row of Fig. 1.

For the polynomial function $f_2(\boldsymbol{x})$, we set the dimension to 4, 8, and 16. Similar to the Gaussian function, we present the relative L^2 errors for different dimensions in Fig. 2, which again shows that the gradient information regularized term can greatly enhance the approximation accuracy. The loss functions for $d = 4$ with 400 samples and $d = 8$ and 16 with 3200 samples are provided in the bottom row of Fig. 2. It can be observed that the loss function of gradient-enhanced methods may be smaller than the original DNN as the iteration number increases, verifying the strength of the gradient-enhanced methods.

4.2.2 Elliptic PDE with Random Inputs

We now consider the following stochastic elliptic PDE problem,

$$\begin{cases} -\nabla \cdot (a(\boldsymbol{x}, \omega)\nabla u(\boldsymbol{x}, \omega)) = f(\boldsymbol{x}, \omega) & \text{in } \mathscr{D} \times \Omega, \\ u(\boldsymbol{x}, \omega) = 0 & \text{on } \partial\mathscr{D} \times \Omega, \end{cases} \tag{32}$$

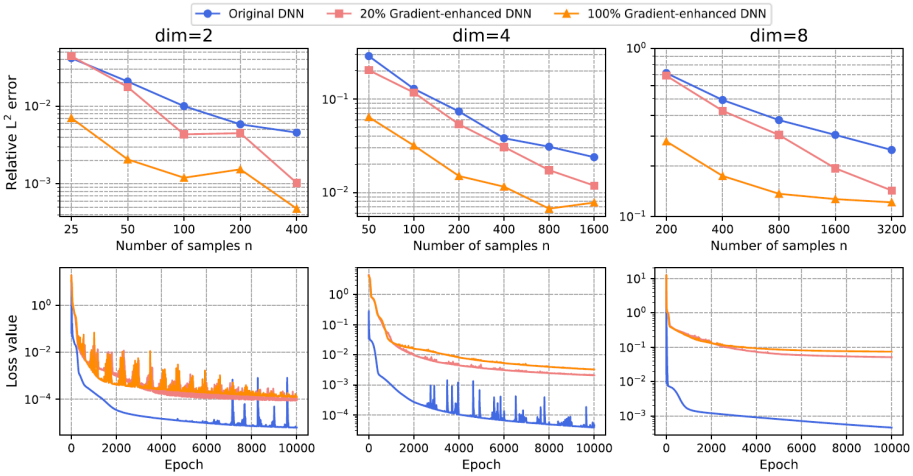

FIG. 1: Example 1. Approximation of $f_1(\boldsymbol{x})$. Top: The relative L^2 errors against the number of samples. Bottom: The loss functions against increasing epochs with the number of samples 400 for $d = 2$, 1600 for $d = 4$, and 3200 for $d = 8$.

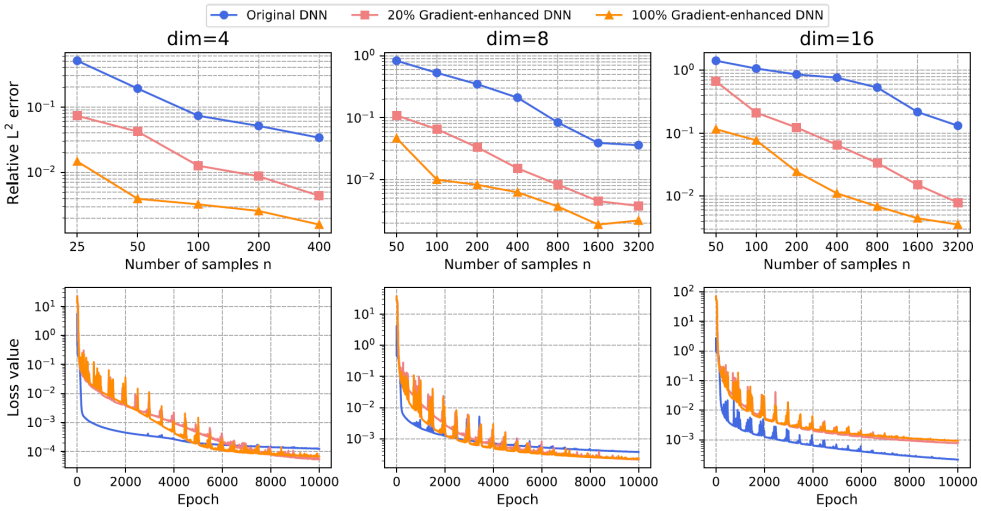

FIG. 2: Example 2. Approximation of $f_2(\boldsymbol{x})$. Top: The relative L^2 errors against the number of samples. Bottom: The loss functions against increasing epochs with the number of samples 400 for dim $=4$ and 3200 for dim $=8$ and 16.

where $\mathscr{D} = [0,1]^2, \boldsymbol{x} = (x_1, x_2)$ is a spatial coordinate and $f(\boldsymbol{x}, \omega)$ is a deterministic force term $f(\boldsymbol{x}, \omega) = \cos(x_1)\sin(x_2)$. The random diffusion coefficient $a(\boldsymbol{x}, \omega) = a_d(\boldsymbol{x}, \omega)$ with one-dimensional spatial dependence takes the form (Babuka et al., 2010),

$$\log\big(a_d(\boldsymbol{x}, \omega) - 0.5\big) = 1 + Y_1(\omega)\left(\frac{\sqrt{\pi}L}{2}\right)^{1/2} + \sum_{k=2}^{d} \zeta_k \phi_k(\boldsymbol{x}) Y_k(\omega), \tag{33}$$

where

$$\zeta_k := (\sqrt{\pi}L)^{1/2} \exp\left(\frac{-\big(\lfloor \frac{k}{2} \rfloor \pi L\big)^2}{8}\right) \quad \text{if } k > 1 \text{ and } L = \frac{1}{12},$$

and $\phi_k(\boldsymbol{x})$ only depends on the first component of $\boldsymbol{x}$,

$$\phi_k(\boldsymbol{x}) := \begin{cases} \sin\big(\lfloor \frac{k}{2} \rfloor \pi x_1\big) & \text{if } k \text{ even}, \\ \cos\big(\lfloor \frac{k}{2} \rfloor \pi x_1\big) & \text{if } k \text{ odd}. \end{cases} \tag{34}$$

Here $\{Y_k(\omega)\}_{k=1}^d$ are independent random variables uniformly distributed in the interval $[-1, 1]$. In the following we approximate the QoI q defined by $q(\omega) = u((0.5, 0.5), \omega)$, which is the solution of Eq. (32) at location $\boldsymbol{x} = (0.5, 0.5)$. Denote $\Psi(\omega) = (Y_1(\omega), \ldots, Y_d(\omega))$. The derivatives $\mathrm{d}q/\mathrm{d}\Psi = \partial u(\boldsymbol{x}, \omega)/\partial\Psi$ are computed by the adjoint sensitivity method. Both forward and adjoint solvers are implemented in the finite element method (FEM) project FEniCS (Logg et al., 2012). In numerical tests, $\{\Psi(\omega_i)\}_{i=1}^n$ are generated from a uniform distribution in $[-1, 1]^d$, and we solve the forward PDE 320 times for $d = 5$ and 1600 times for $d = 10$. Notice that each partial derivative leads to an adjoint equation; then the number of adjoint equations needed to solve is d times that of the forward equations. It is worth mentioning that the cost of generating derivatives of $q(\omega)$ in elliptic PDE (32) is negligible since they share the same stiff matrix with q.

After obtaining the function values as well as the corresponding gradient information, we apply the original DNN, 20% gradient-enhanced DNN, and 100% gradient-enhanced DNN to approximate the QoI $q(\omega)$. The learning rate for the Adam optimizer is 0.001 with half decay each 1000 steps. The relative L^2 errors for $d = 5$ and 10 are presented at the top row of Fig. 3. We also provide the loss functions of different models for $d = 5$ with the number of samples 320 and $d = 10$ with the number of samples 1600 in the bottom row of Fig. 3. All cases verify that gradient-enhanced methods significantly outperform the original approach. We can achieve the same accuracy using much fewer training samples.

5. CONCLUSION

We have proposed gradient-enhanced deep neural network (DNN) approximations for function approximations and uncertainty quantification. In our approach, the gradient information is included as a regularization term. For this approach, we present posterior estimates (by the two-layer neural networks) similar to those in the path-norm regularized DNN approximations. We also discuss the application of this approach to gradient-enhanced uncertainty quantification, and numerical experiments show that the proposed approach can outperform the traditional DNN approach in many cases of interest. The discussion in this work is limited to supervised learning where labeled data are available, and in our future work, we will consider applying this gradient-enhanced idea to unsupervised learning where the physical equation is considered to yield the loss function.

ACKNOWLEDGMENT

We would like to thank Professor Tao Zhou of the Chinese Academy of Sciences for bringing this topic to our attention and for his encouragement and helpful discussion.

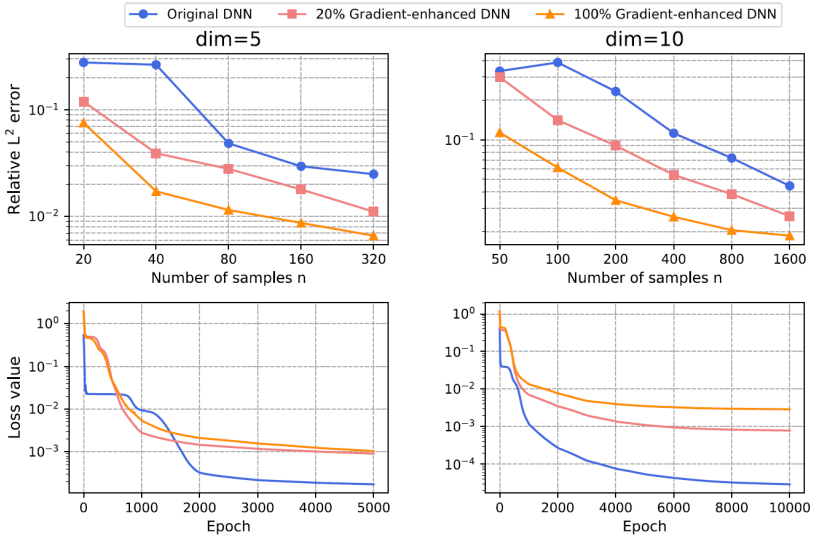

FIG. 3: Example 3. Top: The relative L^2 errors against number of samples for $N = 5, 10$. Bottom: The loss functions against increasing epochs for $d = 5$ with 320 samples and $d = 10$ with 1600 samples.

REFERENCES

Barron, A., Universal Approximation Bounds for Superpositions of a Sigmoidal Function, *IEEE Trans. Inf. Theor.*, vol. **39**, no. 3, pp. 930–945, 1993.

Babuka, I., Nobile, F., and Tempone, R., A Stochastic Collocation Method for Elliptic Partial Differential Equations with Random Input Data, *SIAM J. Numer. Anal.*, vol. **52**, no. 2, pp. 317–355, 2010.

DeVore, R.A. and Lorentz, G.G., *Constructive Approximation*, vol. **303**, Berlin: Springer Science & Business Media, 1993.

E, W., Ma, C., and Wu, L., *A Priori* Estimates of the Population Risk for Two-Layer Neural Networks, *Commun. Math. Sci.*, vol. **17**, no. 5, pp. 1407–1425, 2019.

E, W., Ma, C., and Wu, L., The Barron Space and the Flow-Induced Function Spaces for Neural Network Models, *Constr. Approx.*, vol. **55**, no. 1, pp. 369–406, 2022.

E, W. and Stephan, W., Representation Formulas and Pointwise Properties for Barron Functions, arXiv: 2006.05982, 2020.

E, W. and Yu, B., The Deep Ritz Method: A Deep Learning-Based Numerical Algorithm for Solving Variational Problems, *Commun. Math. Stat.*, vol. **1**, no. 6, pp. 1–12, 2018.

Guo, L., Narayan, A., and Zhou, T., A Gradient Enhanced ℓ_1-Minimization for Sparse Approximation of Polynomial Chaos Expansions, *J. Comput. Phys.*, vol. **367**, pp. 49–64, 2018.

He, K., Zhang, X., Ren, S., and Sun, J., Deep Residual Learning for Image Recognition, in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Las Vegas, NV, pp. 770–778, 2016.

Jakeman, J.D., Eldred, M.S., and Sargsyan, K., Enhancing ℓ_1-Minimization Estimates of Polynomial Chaos Expansions Using Basis Selection, *J. Comput. Phys.*, vol. **289**, pp. 18–34, 2015.

Li, Y., Anitescu, M., Roderick, O., and Hickernell, F., Orthogonal Bases for Polynomial Regression with Derivative Information in Uncertainty Quantification, *Vis. Mech. Processes: Int. Online J.*, vol. **1**, no. 4, pp. 297–320, 2011.

Litjens, G., Kooi, T., Bejnordi, B.E., Setio, A.A.A., Ciompi, F. Ghafoorian, M., Van Der Laak, J.A., Van Ginneken, B., and Sánchez, C.I., A Survey on Deep Learning in Medical Image Analysis, *Med. Image Anal.*, vol. **42**, pp. 60–88, 2017.

Liu, F., Huang, X., Chen, Y., and Suykens, J.A., Random Features for Kernel Approximation: A Survey on Algorithms, Theory, and Beyond, arXiv: 2004.11154, 2020.

Lockwood, B. and Mavriplis, D., Gradient-Based Methods for Uncertainty Quantification in Hypersonic Flows, *Comput. Fluids*, vol. **85**, pp. 27–38, 2013.

Logg, A., Mardal, K.A., and Wells, G., *Automated Solution of Differential Equations by the Finite Element Method: The FEniCS Book*, Berlin: Springer Science & Business Media, vol. **84**, 2012.

Luchini, P. and Bottaro, A., Adjoint Equations in Stability Analysis, *Annu. Rev. Fluid Mech.*, vol. **46**, pp. 493–517, 2014.

Majdisova, Z. and Skala, V., Radial Basis Function Approximations: Comparison and Applications, *Appl. Math. Model.*, vol. **51**, pp. 728–774, 2017.

Maurer, A., A Vector-Contraction Inequality for Rademacher Complexities, in *Int. Conf. on Algorithmic Learning Theory*, Bari, Italy, pp. 3–17, 2016.

Meng, X. and Karniadakis, G.E., A Composite Neural Network That Learns from Multi-Fidelity Data: Application to Function Approximation and Inverse PDE Problems, *J. Comput. Phys.*, vol. **401**, p. 109020, 2020.

Peng, J., Hampton, J., and Doostan, A., On Polynomial Chaos Expansion via Gradient-Enhanced ℓ_1-Minimization, *J. Comput. Phys.*, vol. **310**, pp. 440–458, 2016.

Qin, T., Chen, Z., Jakeman, J.D., and Xiu, D., Deep Learning of Parameterized Equations with Applications to Uncertainty Quantification, *Int. J. Uncertain. Quantif.*, vol. **11**, no. 2, pp. 63–82, 2021.

Raissi, M., Perdikaris, P., and Karniadakis, G.E., Physics-Informed Neural Networks: A Deep Learning Framework for Solving Forward and Inverse Problems Involving Nonlinear Partial Differential Equations, *J. Comput. Phys.*, vol. **378**, pp. 686–707, 2019.

Ross, A. and Doshi-Velez, F., Improving the Adversarial Robustness and Interpretability of Deep Neural Networks by Regularizing Their Input Gradients, *Proc. of the AAAI Conf. on Artificial Intelligence*, New Orlean, LA, vol. **32**, no. 1, 2018.

Schmidhuber, J. and Hochreiter, S., Long Short-Term Memory, *Neural Comput.*, vol. **9**, no. 8, pp. 1735–1780, 1997.

Schwab, C. and Zech, J., Deep Learning in High Dimension: Neural Network Expression Rates for Generalized Polynomial Chaos Expansions in UQ, *Anal. Appl.*, vol. **17**, no. 1, pp. 19–55, 2019.

Shalev-Shwartz, S. and Ben-David, S., *Understanding Machine Learning: From Theory to Algorithms*, Cambridge, UK: Cambridge University Press, 2014.

Siegel, J.W. and Xu, J., Approximation Rates for Neural Networks with General Activation Functions, *Neural Networks*, vol. **128**, pp. 313–321, 2020.

Sirignano, J. and Spiliopoulos, K., DGM: A Deep Learning Algorithm for Solving Partial Differential Equations, *J. Comput. Phys.*, vol. **375**, pp. 1339–1364, 2018.

Spitzbart, A., A Generalization of Hermite's Interpolation Formula, *Am. Math. Mon.*, vol. **67**, no. 1, pp. 42–46, 1960.

Wu, Z., Hermite–Birkhoff Interpolation of Scattered Data by Radial Basis Functions, *Approx. Theory Appl.*, vol. **8**, no. 2, pp. 1–10, 1992.

Yan, M., Yang, J., Chen, C., Zhou, J., Pan, Y., and Zeng, Z., Enhanced Gradient Learning for Deep Neural Networks, *IET Image Process.*, vol. **16**, no. 2, pp. 365–377, 2022.

Yang, L., Meng, X., and Karniadakis, G.E., B-PINNs: Bayesian Physics-Informed Neural Networks for Forward and Inverse PDE Problems with Noisy Data, *J. Comput. Phys.*, vol. **425**, p. 109913, 2021.

Yu, J., Lu, L., Meng, X., and Karniadakis, G.E., Gradient-Enhanced Physics-Informed Neural Networks for Forward and Inverse PDE Problems, *Comput. Methods Appl. Mech. Eng.*, vol. **393**, p. 114823, 2022.

Zhuang, X., Nguyen, L.C., Nguyen-Xuan, H., Alajlan, N., and Rabczuk, T., Efficient Deep Learning for Gradient-Enhanced Stress Dependent Damage Model, *Appl. Sci.*, vol. **10**, no. 7, p. 2556, 2020.