Approximation with CNNs in Sobolev Space: with Applications to Classification

Anonymous Author(s)
Affiliation
Address
email

Abstract

We derive a novel approximation error bound with explicit prefactor for Sobolev-regular functions using deep convolutional neural networks (CNNs). The bound is non-asymptotic in terms of the network depth and filter lengths, in a rather flexible way. For Sobolev-regular functions which can be embedded into the Hölder space, the prefactor of our error bound depends on the ambient dimension polynomially instead of exponentially as in most existing results, which is of independent interest. We also establish a new approximation result when the target function is supported on an approximate lower-dimensional manifold. We apply our results to establish non-asymptotic excess risk bounds for classification using CNNs with convex surrogate losses, including the cross-entropy loss, the hinge loss (SVM), the logistic loss, the exponential loss and the least squares loss. We show that the classification methods with CNNs can circumvent the curse of dimensionality if input data is supported on a neighborhood of a low-dimensional manifold.

1 Introduction

Classification methods with hypothesis spaces specified through deep neural networks have achieved remarkable successes in a variety of machine learning tasks [36]. In particular, *convolutional neural networks* (CNNs) have demonstrated outstanding performance in many applications, including computer vision [35], natural language processing [58], and sequence analysis in bioinformatics [1, 66]. However, to the best of our knowledge, there has only been limited study on the approximation with CNNs and systematic studies on the properties of various classification methods using CNNs in terms of the non-asymptotic bounds for the excess risks.

1.1 Related work

There has been much effort devoted to understanding the approximation properties of deep neural networks in recent years. Many interesting results have been obtained concerning the approximation power of deep neural networks for multivariate functions; some examples of important recent works include [60, 61, 40, 50, 54, 55, 44, 13, 28]. These works focused on the approximation power for ReLU activated feedforward neural networks (FNNs) on different kinds of smooth functions.

The approximation power of CNNs has been studied in limited works. [2] studied the approximation power of the composited CNNs, where the network consists of fully-connected CNNs followed by a fully-connected layer. They proved that under suitable conditions, convolution neural networks can inherit the universal approximation property of its last fully connected layers. [49] showed that for translation equivariant functions, all upper and lower approximation bounds are equivalent between FNNs and CNNs with the same order of network length and size. [47] showed that ResNet-type

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

CNNs, a special CNN architecture with skip-layer connections (or identity connections between inconsecutive layers), can replicate the learning ability of FNNs having block-sparse structures. The ResNet-type CNN in [47], as claimed, can be dense, and its width, channel size, and filter size are constant with respect to sample size rather than the diverging one in [49].

In addition, a series of work studied the approximation power of deep CNNs with Toeplitz type convolutional matrices [67, 65, 64]. The universality of such CNNs was established in [65] for target functions restricted to Sobolev space $W^{\beta,2}(\mathbb{R}^d)$ with $\beta \geq d/2 + 2$. The approximation error is of order $O(L^{-1/2-1/d})$ (up to logarithms), where L is the number of layers of such CNN. But the CNN defined in [65] has width increasing linearly with respect to depth. To overcome the difficulty in theoretical analysis of such CNNs, [64] introduced a downsampling operator to reduce the widths and the approximation power of downsampled CNNs for a special class of functions (β-Hölder continuous ridge function) was studied. The approximate rate is shown to be $O(N^{-\beta})$ by downsampled CNNs with a uniform filter length $s = O(4N + 6)$ and width $\mathcal{W} = O(N)$ for positive integer N. Further, it was shown that a downsampled CNN can compute the same function as a FNN does, with the total number of free parameters of downsampled CNN being at most 8 times of that of FNN. For the CNNs considered by [67, 16, 64] and [65], each convolutional layer has only one filter with size s and the weight matrix is of a special Toeplitz type with shape $(d_{in} + s) \times d_{in}$ and no fully-connected layer is allowed in hidden layers. The paper [27] studied the approximation properties of convolutional architectures for the target functions defined on infinite time domains tailored to temporal sequence data.

To mitigate the curse of dimensionality and show the advantages of CNNs in classification problems, several works studied the approximation as well as the statistical learning theories under some low-dimensional assumptions on the target function or the data distribution. In [32, 33], the posterior probability function is assumed to be Hölder smooth satisfying hierarchical max-pooling model. In [34], the target function is assumed to be spatially rotation invariant. In [18] and [38], input data is assumed to be supported on spheres and manifolds, respectively. Under different structural assumptions on the target function, similar results have also been proved for FNNs [7, 52, 53]. If data is assumed to be supported on low-dimensional manifolds, the curse of dimensionality can also be mitigated for FNNs [51, 44, 12, 28].

1.2 Our contributions

We study the approximation power of CNNs for functions in Sobolev spaces, including the special case of Hölder class and apply our new approximation results to establish non-asymptotic error bounds for several important classification methods using CNNs. Our main contributions are as follows.

(i) We derive a novel error bound for approximating Sobolev-regular functions using deep CNNs, with the error bound explicitly expressed in terms the network parameters and model parameters. The error bound depends on the network depth and filter lengths in a rather flexible way. For functions which can be embedded into Hölder space, the prefactor of our error bound depends on the ambient dimension polynomially instead of exponentially as in existing results. We also establish a novel approximation result when the target function is supported on an approximate lower-dimensional manifold, which shows that CNNs are capable of mitigating the curse of dimensionality under such a distributional condition.

(ii) As an application of our CNN approximation results, we establish non-asymptotic bounds for the stochastic and approximation errors of classification methods with a general class of convex surrogate losses in Sobolev space using CNNs. Our main results are also applicable to other problems that use CNNs approximations.

(iii) We apply our general results to establish the non-asymptotic excess risk bounds for classification using CNNs with convex surrogate losses in Sobolev space, including the cross-entropy loss, the hinge loss (SVM), the logistic loss, the exponential loss, and the least squares loss. We show that classification methods with CNNs can circumvent the curse of dimensionality if the input data is supported on a neighborhood of a low-dimensional manifold embedded in the ambient space.

Table 1: A comparison of some recent convolutional neural network approximation results.

	Network	Target function	Flexible filter length	Explicit prefactor	Low-dimensional Result
[49]	CNN	FNN	✗	✗	✗
[47]	ConvResNet	FNN	✓	✗	✗
[65]	CNN	Sobolev	✓	✗	✗
[33]	CNN	Hölder	✗	✗	✓
[38]	ConvResNet	Besov	✓	✗	✓
This paper	CNN	Sobolev and Hölder	✓	✓	✓

In Table 1, we summarize and compare some recent and most related work on CNN approximation results for various function spaces.

The statistical convergence properties of the excess risk of the empirical risk minimizer (ERM) for the misclassification 0-1 loss have been studied extensively under various conditions. Different types of hypothesis spaces have been used in these approaches, including the linear space, the reproducing kernel Hilbert space and the class of tree-based models, see, for example, [9, 56, 42, 6, 29, 8] and [63]. However, there are limited studies worked on deep binary classifications by FNNs [46, 25, 30], ResNet [26, 45] and CNNs [32, 33, 38].

2 Approximation power of CNNs

2.1 Convolutional neural networks

Convolutional networks are a special type of structured sparse feedforward neural network (FNN) that use convolution in place of general matrix multiplication in at least one of their layers [21]. There are different formulations of CNNs in the literature [2, 67, 47, 37, 64, 65, 33, 32, 38]. In this paper, we consider ReLU activated downsampled CNNs with bias vectors in the convolutional layers defined in [64]. We also do not require the norm of network parameters (weight and bias) to be uniformly bounded.

We consider a general CNN function $f_{\text{CNN}} : \mathcal{X} \to \mathbb{R}$, where $\mathcal{X} \subset \mathbb{R}^d$. For simplicity, we take $\mathcal{X} = (0,1)^d$. Suppose f_{CNN} has L number of hidden layers, then it can be expressed as

$$f_{\text{CNN}}(x) = A_{L+1} \circ A_L \circ \cdots \circ A_2 \circ A_1(x), x \in \mathcal{X}, \tag{1}$$

where $\circ$ denotes the functional composition. The A_i's are either convolutional operators or downsampling operators. For convolutional layers, $A_i(x) = \sigma(W_i^c x + b_i^c)$, where $W_i^c \in \mathbb{R}^{d_i \times d_{i-1}}$ is the structured sparse Toeplitz type weight matrix induced by the convolutional filter $\{w_j^{(i)}\}_{j=0}^{s^{(i)}}$ with filter length and $s^{(i)} \in \mathbb{N}^+$, $b_i^c \in \mathbb{R}^{d_i}$ is a bias vector, and σ is the rectified linear unit (ReLU) activation function applying to each component of the input vector.

For downsampling layers, for any $x \in \mathbb{R}^{d_{i-1}}$, $A_i(x) = D_i(x) = (x_{jm_i})_{j=1}^{\lfloor d_{i-1}/m_i \rfloor}$, where $D_i : \mathbb{R}^{d_i \times d_{i-1}}$ is the downsampling operator with scaling parameter $m_i \leq d_{i-1}$ in the i-th layer. We introduce the convolutional operation and downsampling operation [64] in details in Appendix B.

We focus on the function class consisting of deep CNNs, denoted by $\mathcal{F}_{\text{CNN}}$ which is defined as

$$\mathcal{F}_{\text{CNN}} = \{f_{\text{CNN}} \text{ defined in (1) over all possible choiceof } \mathrm{A_i}, \mathrm{i} = 1, \ldots, \mathrm{L}+1\}. \tag{2}$$

For a general function class $\mathcal{F}$ and any $f \in \mathcal{F}$, define the $\|\cdot\|_\infty$ metric on $\mathcal{F}$ by $\sup_{x \in \mathcal{X}} \|f(x)\|_\infty$. Then let $\mathcal{B} := \sup_{f \in \mathcal{F}_{\text{CNN}}} \|f\|_\infty$ denote the bound of the functions in $\mathcal{F}_{\text{CNN}}$. Let L be th number of hidden layers and $\mathcal{S}$ be the total number of parameters for networks in $\mathcal{F}_{\text{CNN}}$ and let $s_{\min}$ and $s_{\max}$ be the minimum and maximum filter length over convolutional layers respectively.

2.2 Approximation in Sobolev space

We use the Sobolev and Hölder classes for the target functions in our approximation results since they are well-established formulations for describing smooth functions in the literature. Indeed,

earlier works on approximation theory of neural networks have been developed for Sobolev functions [60, 65, 24] and for Hölder functions [33, 52].

Let $\mathbb{N}_0$ be the set of non-negative integers and $\beta \in \mathbb{N}_0$. The Sobolev class of functions $W^{\beta,p}(\mathcal{X})$ with $\mathcal{X} = (0,1)^d$ is defined as

$$W^{\beta,p}(\mathcal{X}) = \Big\{ f \in L^p(\mathcal{X}) : D^\alpha f \in L^p(\mathcal{X}) \text{for all } \alpha \in \mathbb{N}_0^d \text{ with } \|\alpha\|_1 \le \beta \Big\}, \tag{3}$$

where $D^\alpha = \partial^{\alpha_1} \cdots \partial^{\alpha_d}$ with $\alpha = (\alpha_1, \ldots, \alpha_d)^\top \in \mathbb{N}_0^d$, $\|\alpha\|_1 = \sum_{i=1}^d \alpha_i$ and $1 \le p \le \infty$. For $f \in W^{\beta,p}(\mathcal{X})$ and $0 \le m \le \beta$, we define the norm $\|f\|_{W^{m,p}(\mathcal{X})} := \big(\sum_{0 \le \|\alpha\|_1 \le m} \|D^\alpha f\|^p_{L^p(\mathcal{X})} \big)^{1/p}$ for $1 \le p < \infty$, and define $\|f\|_{W^{m,\infty}(\mathcal{X})} := \max_{0 \le \|\alpha\|_1 \le m} \|D^\alpha f\|_{L^\infty(\mathcal{X})}$.

In the remainder of the paper, for any positive integers $M, N \in \mathbb{N}^+$, let $\mathcal{F}_{\text{CNN}}$ be the class of CNNs defined in (2) with depth L, size $\mathcal{S}$ and filter length specified as follows:

$$L \le 42(\lfloor \beta \rfloor + 1)^2 M \lceil \log_2(8M) \rceil \lceil \frac{\mathcal{W} - 1}{s_{\min} - 1} \rceil, \tag{4}$$

$$2 \le s_{\min} \le s_{\max} \le \mathcal{W}, \tag{5}$$

$$\mathcal{S} \le 8\mathcal{W}L, \tag{6}$$

where

$$\mathcal{W} = 38^2 (\lfloor \beta \rfloor + 1)^4 d^{2\lfloor \beta \rfloor + 2} N^2 \lceil \log_2(8N) \rceil^2, \tag{7}$$

denotes the maximum incremental width (number of neurons) for consecutive layers in the network and $\lceil a \rceil$ denotes the smallest integer no less than a.

Theorem 2.1. *Assume that $f \in W^{\beta,p}(\mathcal{X})$ with $1 \le \beta \in \mathbb{N}_0$, $1 \le p \le \infty$ and $\|f\|_{W^{\beta,p}(\mathcal{X})} \le B_0$ for $B_0 > 0$. For any $M, N \in \mathbb{N}^+$, there exists a function $f_{CNN} \in \mathcal{F}_{CNN}$ defined in (2) with depth L and filter lengths and size $\mathcal{S}$ specified in (4), (5) and (6), such that for $m = 0, 1$,*

$$\|f - f_{CNN}\|_{W^{m,p}(\mathcal{X})} \le C_0(d, \beta, p)(NM)^{-2(\beta - m)/d},$$

where $C_0(d, \beta, p) = 37 \cdot 2^{2\beta + 2d/p} B_0^2 (\beta + 1)^3 \times \{\pi^{-d/2} \Gamma(d/2 + 1)\}^{2/p+1} (1 + 2\sqrt{d})^d d^{4\beta}$. Here $\Gamma(\cdot)$ is the gamma function.

Our result is new for approximation of Sobolev-regular functions in $W^{\beta,p}(\mathcal{X})$ for a positive integer $\beta \in \mathbb{N}^+$ with respect to the Sobolev norms $\|\cdot\|_{W^{m,p}(\mathcal{X})}$ for $m = 0, 1$ in terms of explicitly defined approximation error prefactor, clearly defined and flexible network parameters, as well as an fast approximation rate comparable with [22]. A toy example is provided in Appendix D to illustrate the approximation power of CNNs and examine how the approximation error varies according to the filter size and depth of the networks.

In [24], approximation result on β-smooth functions by networks with squared ReLU activation function is also provided under the norm $\|\cdot\|_{W^{m,p}(\mathcal{X})}$ for positive integer $m < \beta$. The paper[22] analyzed the approximation rates of ReLU DNN for Sobolev-regular functions with respect to the weaker Sobolev norms $\|\cdot\|_{W^{s,p}(\mathcal{X})}$ for $s \in [0, 1]$, which can be seen as a generalization of the result in [60]. Approximation results on Sobolev-regular functions $W^{\beta,p}(\mathcal{X})$ by networks with general activation functions are derived in [23], where the error is with respect to $\|\cdot\|_{W^{m,p}(\mathcal{X})}$ for some integer $m < \beta$.

By the general Sobolev inequality, see Theorem 6 in Chapter 5 of [15], $W^{k,p}(\mathcal{X})$ with $kp > d$ can be embedded into a Hölder class $\mathcal{H}^\beta(\mathcal{X}, B_0)$ (with $\beta = k - \lfloor \frac{d}{p} \rfloor - 1$) defined as

$$\mathcal{H}^\beta(\mathcal{X}, B_0) = \Big\{ f : \mathcal{X} \to \mathbb{R}, \max_{\|\alpha\|_1 \le \lfloor \beta \rfloor} \|D^\alpha f\|_\infty \le B_0 \text{ and } \max_{\|\alpha\|_1 = s} \sup_{x \ne y} \frac{|D^\alpha f - D^\alpha f(x)|}{\|x - y\|_2^r} \le B_0 \Big\}. \tag{8}$$

where $\beta > 0$ and $\lfloor \beta \rfloor \in \mathbb{N}_0$ denotes the largest integer strictly smaller than β. Because of the extra regularity on the Hölder class, a much improved prefactor C_0 in the error bound of Theorem 2.1 can be obtained.

Theorem 2.2. *Let $f \in \mathcal{H}^\beta(\mathcal{X}, B_0)$ be defined in (8) and let $X \in \mathbb{R}^d$ be a random vector whose probability distribution is supported on $\mathcal{X} = (0,1)^d$ and absolutely continuous with respect to the Lebesgue measure. For any $M, N \in \mathbb{N}^+$, there exists a function $f_{CNN} \in \mathcal{F}_{CNN}$ defined in (2) with depth L and filter lengths and size $\mathcal{S}$ specified in (4), (5) and (6), such that*

$$\mathbb{E}|f(X) - f_{CNN}(X)| \leq C_0(d,\beta)(NM)^{-2\beta/d},$$

where $C_0(d,\beta) = 18B_0(\beta+1)^2 d^{\beta+(\beta\vee 1)/2}$. Here $a \vee b := \max\{a,b\}$ for $a, b \in \mathbb{R}$.

The convergence rate $(NM)^{-2\beta/d}$ in Theorems 2.1 and 2.2 with respect to the network depth and filter lengths specified by M and N, is in line with the nearly optimal rate of ReLU FNNs on smooth functions under Sobolev norm in [24] and on Hölder smooth functions under L^∞ norm in [28]. The prefactor $18B_0(\beta+1)^2 d^{\beta+(\beta\vee 1)/2}$ of approximation error depends on the dimension d polynomially, different from the exponential dependence in many existing neural network approximation results mentioned in Section 1.1.

2.3 Approximation with a lower-dimensional support

In many modern machine learning problems, the ambient dimension d of the input data could be very large, which results in an extremely slow convergence rate. This fact is known as the *curse of dimensionality*. Fortunately, many types of data have a low-dimensional latent structure, that is, although the ambient dimension d is large, the distribution of the data is approximately supported on a low-dimensional subset of $\mathbb{R}^d$, in which case the approximation error bound can be substantially improved. We establish an approximation result for CNNs in Sobolev spaces under a low-dimensional support assumption.

Assumption 2.3. The distribution of X is supported on $\mathcal{M}_\rho$, a ρ-neighborhood of $\mathcal{M} \subset \mathcal{X}$, where $\mathcal{M}$ is a compact $d_{\mathcal{M}}$-dimensional Riemannian submanifold and $\mathcal{M}_\rho = \{x \in \mathcal{X} : \inf\{\|x-y\|_2 : y \in \mathcal{M}\} \leq \rho\}$ for $\rho \in (0,1)$.

In real-world applications, data are hardly observed to locate on an *exact manifold*, instead they could be more realistically viewed as consisting of a latent part supported on a low-dimensional manifold $\mathcal{M}$ plus noises. Therefore, Assumption 2.3 is more realistic than the exact manifold assumption [51, 12, 38].

Define

$$d_\varepsilon = O(d_{\mathcal{M}}\varepsilon^{-2}\log(d/\varepsilon)),\ \varepsilon \in (0,1), \tag{9}$$

$$\rho_\varepsilon = C_2(NM)^{-2\beta/d_\varepsilon}(\beta+1)^2 d^{1/2} d_\varepsilon^{3\beta/2} \times [(d/d_\varepsilon)^{1/2} + 1 - \varepsilon]^{-1}(1-\varepsilon)^{1-\beta}. \tag{10}$$

In our error bound results below, it is the d_ε that will affect the convergence rate. Since it is often the case that $d_{\mathcal{M}} \ll d$ and therefore $d_\varepsilon \ll d$, the manifold assumption will lead to a better convergence rate than those in Theorems 2.1 and 2.2. In addition, we require $\rho \leq \rho_\varepsilon$.

Theorem 2.4. *Suppose that Assumption 2.3 holds, $f \in \mathcal{H}^\beta(\mathcal{X}, B_0)$ and the distribution of X is absolutely continuous with respect to the Lebesgue measure. Let d_ε and ρ_ε be defined in (9) and (10), respectively. Assume that $\rho \leq \rho_\varepsilon$. Then, for any $M, N \in \mathbb{N}^+$, there exists a CNN function $f_{CNN} \in \mathcal{F}_{CNN}$ with depth L and filter lengths and size $\mathcal{S}$ specified in (4), (5) and (6) with $\mathcal{W} = 38^2(\lfloor\beta\rfloor+1)^4 d_\varepsilon^{2\lfloor\beta\rfloor+2} N^2 \lceil\log_2(8N)\rceil^2$ such that*

$$\mathbb{E}|f(X) - f_{CNN}(X)| \leq C(d,\beta)(NM)^{-2\beta/d_\varepsilon},$$

where $C(d,\beta) = (18 + C_2)\frac{B_0}{(1-\varepsilon)^\beta}(\beta+1)^2 d^{1/2} d_\varepsilon^{3\beta/2}$.

Since the intrinsic dimension d_ε can be much smaller than the ambient dimension d, the rate $(NM)^{-2\beta/d_\varepsilon}$ in Theorem 2.4 is greatly improved compared with the rate $(NM)^{-2\beta/d}$ in Theorem 2.2 and the curse of dimensionality is mitigated. The result here is of independent interest and can be useful in other problems that involve the use of CNNs.

3 Excess risk in classification

In this section, we present the application of the approximation results Theorems 2.1 and 2.4 to the error analysis of classification with CNNs.

Consider a binary classification problem with a predictor $X \in \mathcal{X} \subset \mathbb{R}^d$ and its binary label $Y \in \{1, -1\}$. We are interested in learning a classifier $h : \mathcal{X} \to \{1, -1\}$ from a class of functions, or a hypothesis space, denoted by $\mathcal{H}$. Let the joint distribution of (X, Y) be $\mathbb{P}$. The goal is to find a classifier that minimizes the misclassification error or the 0-1 risk: $R_*(h) = \mathbb{P}\big\{h(X) \neq Y\big\}, h \in \mathcal{H}$. Denote the misclassification risk minimizer at the population level by $h_0 = \operatorname{argmin}_{h \text{ measurable}} R_*(h)$. For any $h \in \mathcal{H}$, the excess risk of h is $R_*(h) - R_*(h_0)$, the difference between the misclassification errors of h and h_0.

Since the probability measure $\mathbb{P}$ is unknown in practice, the classifier h will be learned based on a random sample $S = \{(X_i, Y_i)\}_{i=1}^n$ from $\mathbb{P}$, where n is the sample size. The empirical risk minimizer (ERM) is defined by

$$\hat{h}_n \in \operatorname*{argmin}_{h \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^n \mathbb{1}(h(X_i) \neq Y_i). \tag{11}$$

However, the empirical risk function based on the 0-1 loss is non-continuous and non-convex, thus this minimization problem is typically computationally intractable.

Rather than minimizing the non-smooth 0-1 loss, many popular methods adopt a proper convex loss function to train classifiers with computational efficiency that can be done in polynomial time. Moreover, proper surrogate convex loss functions have been shown to be consistent with the 0-1 loss function for binary classification problems [62, 5].

3.1 Convex surrogate loss functions

Let ϕ be a given convex univariate loss function $\phi : \mathbb{R} \to [0, \infty)$. We consider the risk function with respect to the loss ϕ

$$R(f) \equiv R^\phi(f) = \mathbb{E}\{\phi(Yf(X))\}, \tag{12}$$

where for simplicity of notation and without causing confusion we omit the superscript ϕ in R, similarly for $\hat{f}_n$ and f_0 defined below. For a given random sample $S = \{(X_i, Y_i)\}_{i=1}^n$, we denote the ERM over $\mathcal{F}_{\text{CNN}}$ with a given loss ϕ by

$$\hat{f}_n \in \arg \min_{f \in \mathcal{F}_{\text{CNN}}} \frac{1}{n} \sum_{i=1}^n \phi(Yf(X_i)). \tag{13}$$

Based on the ERM $\hat{f}_n$, a classifier $\hat{h}_n(x) := \operatorname{sign}(\hat{f}_n(x))$ for $x \in \mathcal{X}$ can be defined. As shown in [62, 5], for a properly chosen ϕ, $\hat{f}_n$ can help reduce the excess risk $R(\hat{h}_n) - R_0$. Specifically, define the measurable minimizer of R in (12) as

$$f_0 = \operatorname*{argmin}_{f \text{ measurable}} \mathbb{E}\{\phi(Yf(X))\}, \tag{14}$$

and the corresponding minimal ϕ-risk as $R_0 = R(f_0)$. Then for a proper ϕ, we have $\psi(R(\hat{h}_n) - R_0) \leq R(\hat{f}_n) - R_0$, where $\psi : [-1, 1] \to [0, \infty)$ is a nonnegative continuous function, invertible on $[0, 1]$, and achieves its minimum at 0 with $\psi(0) = 0$. A variety of classification methods are based on this tactic. Generally, classification-calibrated ϕ considered in this paper is non-increasing and convex; details are given in the Supplementary Materials.

Define the conditional probability

$$\eta(x) = \mathbb{P}(Y = 1 \mid X = x), \; x \in \mathcal{X}. \tag{15}$$

Let the conditional ϕ-risk of f given $X = x$ be denoted by $R_f(x) := \mathbb{E}\{\phi(Yf(X)) \mid X = x\}$. SVM and cross entropy, two important examples of ϕ, and the corresponding f_0, $R(f_0)$, ψ and its inverse on $[0, 1]$ in Table 2 in the supplementary materials. For the form of ψ, Theorem 34 in [11] shows that if ϕ is convex, $\phi''(0) > 0$ exists and $\phi' < 0$, then $\psi(u) = u^2$.

Table 2: Minimizer and minimal conditional risk under SVM and cross entropy loss functions ϕ. Bound ϕ_B of ϕ, Lipschitz constant B_ϕ and $\Delta_\phi(T)$ for the truncated f_0 under SVM hinge loss function restricted to $[-\mathcal{B}, \mathcal{B}]$ for $1 \le T \le \mathcal{B}$ and cross entropy loss function restricted to $[-\mathcal{B}, \mathcal{B}]$ for $T \le \mathcal{B} < 0.5$.

	SVM	Cross entropy
$\phi(a)$	$\max\{1-a, 0\}$	$-\log\{0.5+a\}$
$f_0(x)$	$\mathrm{sign}(2\eta-1)$	$\eta - 0.5$
$R_{f_0}(x)$	$1-\lvert 2\eta-1\rvert$	$-\eta\log(\eta)-(1-\eta)\log(1-\eta)$
$\psi(\theta)$	$\lvert\theta\rvert$	θ^2
ϕ_B	$\mathcal{B}+1$	$-\log\{0.5-\mathcal{B}\}$
B_ϕ	**	$1/(0.5-\mathcal{B})$
$\Delta_\phi(T)$	0	$-\log\{1+(T-0.5)\}$

Note: The conditional probability $\eta(x)$ defined in (15) is written as η for notational simplicity. "**" stands for "does not exist"

4 Non-asymptotic error bounds

For the ERM $\hat{f}_n$ defined in (13), we first state a basic inequality for bounding the excess risk of $\hat{f}_n$.

Lemma 4.1. *For any loss ϕ and any random sample $S = \{(X_i, Y_i)\}_{i=1}^n$, the excess ϕ-risk of the ERM $\hat{f}_n$ satisfies $R(\hat{f}_n) - R(f_0) \le$ StoErr + AppErr, where*

$$\textit{StoErr} = 2\sup_{f\in\mathcal{F}} |R(f) - R_n(f)|,\ \textit{AppErr} = \inf_{f\in\mathcal{F}} R(f) - R(f_0). \tag{16}$$

Therefore, the excess risk of $\hat{f}_n$ is bounded by the sum of two terms: the stochastic error and the approximation error. For a given loss function ϕ, the upper bound no longer depends on the ERM $\hat{f}_n$, but the function class $\mathcal{F}$ and the random sample S. The stochastic error in (16) depends on the complexity of $\mathcal{F}$ and the approximation error in (16) depends on the approximation power of the class $\mathcal{F}$ for f_0.

4.1 Stochastic error

We bound the stochastic error in terms of the pseudo-dimension of the downsampled CNNs defined in (2); particularly, we further bound its pseudo-dimension and express the bound in terms of quantities related to CNNs.

Theorem 4.2. *[Stochastic error bound] Suppose that ϕ is convex and non-increasing. For any $M, N \in \mathbb{N}^+$, let $\mathcal{F}_{CNN}$ be the class of CNNs defined in (2) with $\mathcal{B} \ge 0$, depth L and size $\mathcal{S}$ and let $\phi_B := \sup_{|a|\le\mathcal{B}} \phi(a)$. Then, for any $\delta \in (0,1)$, with probability at least $1-\delta$, the stochastic error in (16) satisfies*

$$\textit{StoErr} \le \frac{2\phi_B}{\sqrt{n}}\Big(C_0\sqrt{\mathcal{S}L\log(S)\log(n)} + \sqrt{\log(1/\delta)}\Big),$$

where $C_0 > 0$ is a universal constant.

The quantity ϕ_B for common loss functions are presented in Table 2. It worth noting that the error bound here does not require the norm of the CNN parameters (of weight and bias) to be uniformly bounded. In comparison, those stochastic error bounds based on the covering number generally assume an uniformly bounded norm on CNN parameters, which may hinder the approximation power of network since most approximation results need the norm of network parameters to tend to infinity as the approximation error tends to zero.

4.2 Approximation error

We derive an upper bound of the approximation error by relating $\inf_{f\in\mathcal{F}_{\text{CNN}}} R(f) - R(f_0)$ to $\inf_{f\in\mathcal{F}_{\text{CNN}}} \|f - f_0\|$ under proper conditions on ϕ and η, where R is defined in (12). The target function f_0 may be non-smooth or unbounded, which poses extra difficulty. Most of the existing

studies on the approximation properties of networks assume smooth and bounded target functions. As shown in Table 2, the target function f_0 for SVM is non-continuous; for the cross entropy loss, the loss is unbounded on $[-0.5, 0.5]$. For logistic loss and exponential loss, we have $f_0(x) \in \{+\infty, -\infty\}$ once $\eta(x) \in \{0, 1\}$. We overcome these difficulties by imposing mild conditions on the conditional probability η and the loss function ϕ, and use the truncation technique to analyze the approximation.

Assumption 4.3. (a) The conditional probability $\eta(x) = \mathbb{P}\{Y = 1 \mid X = x\}$ is continuous on the support of $\mathcal{X}$ and the probability measure of X is absolutely continuous with respect to the Lebesgue measure. (b) The loss function $\phi : \mathbb{R} \to [0, \infty)$ is classification-calibrated, convex, non-increasing and continuously differentiable on its support.

Assumption 4.3 (a) and (b) are regular conditions ensuring that the target function f_0 is continuous and the loss ϕ is Lipschitz on bounded intervals. The absolute continuity assumption of the probability measure of X with respect to the Lebesgue measure is reasonable for the approximate low-dimensional manifold case but incompatible with the exact low-dimensional manifold condition. This assumption is for deriving better L_p approximation error bound, without which the error bound in term of the L_∞ norm can still be established, but at the price of a wider neural network and a larger prefactor. It can be verified that commonly used loss functions, such as the cross entropy, the logistic and the exponential loss functions, satisfy Assumption 4.3 (b).

To deal with the approximation of unbounded target function, we truncate the target f_0 by a constant T, where T may depend on n. Let $f_{0,T}$ be the truncated version of f_0 defined as

$$f_{0,T}(x) = \begin{cases} f_0(x), & \text{if } |f_0(x)| \le T, \\ T\text{sign}(f_0(x)), & \text{if } |f_0(x)| > T. \end{cases}$$

Denote the error of the loss function ϕ due to truncation by

$$\Delta_\phi(T) := \inf_{|a| \le T} \phi(a) - \inf_{a \in \text{Ran}(f_0)} \phi(a), \tag{17}$$

where $\text{Ran}(f_0)$ is the range of the target function f_0. This error decreases as T increases. Then, the approximation error can be decomposed into two terms that are easier to deal with.

Lemma 4.4. *[ϕ-approximation error] Suppose that Assumption 4.3 holds and $T \le \mathcal{B}$. Then, the ϕ-approximation error $\inf_{f \in \mathcal{F}_{CNN}} R(f) - R(f_0)$ with respect to the loss function ϕ satisfies*

$$\inf_{f \in \mathcal{F}_{CNN}} R(f) - R(f_0) \le B_\phi \inf_{f \in \mathcal{F}_{CNN}} \mathbb{E}|f(X) - f_{0,T}(X)| + \Delta_\phi(T).$$

where B_ϕ is defined as the Lipschitz constant of ϕ on the interval $[-\mathcal{B}, \mathcal{B}]$.

We list B_ϕ and $\Delta_\phi(T)$ for hinge and cross entropy loss functions in Table 2. A detailed table including other loss functions are given in the supplementary materials.

Theorem 4.5. *[Approximation error bound] Suppose that Assumption 4.3 holds and $f_0 \in \mathcal{H}^\beta([0,1]^d, B_0)$. For any $M, N \in \mathbb{N}^+$, let $\mathcal{F}_{CNN}$ be the class of CNNs defined in (2) with $T \le \mathcal{B}$, depth L and filter lengths specified in (4) and (5). Then, the approximation error defined in (16) satisfies AppErr $\le C(d, \beta)(NM)^{-2\beta/d} + \Delta_\phi(T)$, where $C(d, \beta) = 18 B_\phi B_0 (\beta + 1)^2 d^{\beta + (\beta \vee 1)/2}$.*

There is a trade-off since $T \le \mathcal{B}$ and $\Delta_\phi(T)$ deceases in T but $C(d, \beta)$ increases in B_ϕ (or $\mathcal{B}$).

4.3 Non-asymptotic excess risk bound

Combining Theorems 4.2 and 4.5, we obtain the excess error bound.

Theorem 4.6. *[Non-asymptotic excess ϕ-risk bound] Suppose that Assumption 4.3 holds and $f_0 \in \mathcal{H}^\beta([0,1]^d, B_0)$. For any $M, N \in \mathbb{N}^+$, let $\mathcal{F}_{CNN}$ be the class of CNNs defined in (2) with $T \le \mathcal{B}$, depth L and filter lengths specified in (4) and (5). Then, for any $\delta \in (0, 1)$, with probability at least $1 - \delta$, the ERM $\hat{f}_n$ defined in (13) satisfies*

$$R(\hat{f}_n) - R(f_0) \le \textit{StoErr} + \textit{AppErr}, \tag{18}$$

with

$$\textit{StoErr} = \frac{2\phi_B}{\sqrt{n}} \left(C_0 \sqrt{\mathcal{S} L \log(S) \log(n)} + \sqrt{\log(1/\delta)} \right), \ \textit{AppErr} = C(d, \beta)(NM)^{-2\beta/d} + \Delta_\phi(T),$$

where $C(d, \beta) = 18 B_\phi B_0 (\beta + 1)^2 d^{\beta + (\beta \vee 1)/2}$ and $C_0 > 0$ is a universal constant.

The upper bound of the excess ϕ-risk $R(\hat{f}_n) - R(f_0)$ in Theorem 4.6 is a sum of two error terms, the stochastic error and the approximation error. To achieve the optimal error rate, we need to balance the trade-off between them. On one hand, the bound for the stochastic error *StoError* increases with the size and the depth of $\mathcal{F}_{\text{CNN}}$. On the other hand, the bound for the approximation error *AppError* decreases with the depth and the upper bound of filter lengths of $\mathcal{F}_{\text{CNN}}$.

4.4 Circumventing the curse of dimensionality

We state a theorem that provides a non-asymptotic excess risk bound under the approximate low-dimensional manifold assumption.

Theorem 4.7 (Circumventing the curse of dimensionality)**.** *Suppose that Assumptions 2.3, 4.3 hold, and $f_0 \in \mathcal{H}^\beta([0,1]^d, B_0)$. For any $M, N \in \mathbb{N}^+$, let $\mathcal{F}_{CNN}$ be the class of CNNs defined in (2) with $T \leq \mathcal{B}$, depth L and filter lengths and size $\mathcal{S}$ specified in (4), (5) and (6) with $\mathcal{W} = 38^2(\beta+1)^4 d_\varepsilon^{2\beta+2} N^2 \lceil \log_2(8N) \rceil^2$. Suppose ρ in Assumption 2.3 satisfies $\rho \leq \rho_\varepsilon$,, where ρ_ε is defined in (10). Then for any $\delta \in (0,1)$, with probability at least $1-\delta$, the ERM $\hat{f}_n$ defined in (13) satisfies*

$$R(\hat{f}_n) - R(f_0) \leq \textit{StoErr}_* + \textit{AppErr}_*, \tag{19}$$

with

$$\textit{StoErr}_* = \frac{2\phi_B}{\sqrt{n}}\Big(C_1\sqrt{\mathcal{S}L\log(S)\log(n)} + \sqrt{\log(1/\delta)}\Big),$$
$$\textit{AppErr}_* = \Big((18+C_2)\frac{B_\phi B_0}{(1-\varepsilon)^\beta}(\beta+1)^2 d^{1/2} d_\varepsilon^{3\beta/2}\Big) \times (NM)^{-2\beta/d_\varepsilon} + \Delta_\phi(T),$$

where $C_1, C_2 > 0$ are universal constants.

For a high-dimensional input X with a large d, d_ε satisfies $d_{\mathcal{M}} \leq d_\varepsilon < d$ for $\varepsilon \in (0,1)$. For the stochastic error *StoErr*$_*$, comparing with the bound (18) in Theorem 4.6, we see that *StoErr*$_* =$ *StoErr*, that is, the stochastic error does not change under the approximate low-dimensional manifold assumption. For the approximation error *AppErr*$_*$, we see that the convergence rate $(NM)^{-2\beta/d_\varepsilon}$ only depends on d_ε, which leads to a much faster convergence rate.

5 Examples

In this section, we illustrate the applications of Theorems 4.6 and 4.7 to obtaining non-asymptotic error bounds for the excess risk in classification. We apply the general excess risk bounds established in these theorems to several important classification methods with CNNs when a specific form of ϕ is given. We present the results for the cross entropy loss below. The non-asymptotic error bounds for the hinge, the logistic, the exponential and the least squares losses are given in Appendix C in the supplementary material.

For the cross entropy loss $\phi(a) = -\log(0.5+a)$, the Lipschitz constant B_ϕ on $[-0.5, 0.5]$ is not bounded. As the minimizer of ϕ-risk $\eta - 0.5$ is bounded, we can choose $T = \mathcal{B} = 0.5 - n^{\beta/(2d+4\beta)}$. Then $\Delta_\phi(T) \leq n^{-\beta/(2d+4\beta)}$ and $B_\phi = n^{\beta/(2d+4\beta)}$.

Denote $\xi_n = n^{d/(2d+4\beta)}$. Take $N = 1$ in (7) and $M = \lfloor \xi \rfloor$ in (4) and let $\mathcal{F}_{\text{CNN}}$ be the class of CNNs in (2) with depth $L \leq 378 \cdot 38^2(\lfloor\beta\rfloor+1)^6 d^{2\lfloor\beta\rfloor+2}\lfloor\xi_n\rfloor\lceil\log_2(8\lfloor\xi_n\rfloor)\rceil/(s_{\min}-1)$, filter lengths $2 \leq s_{\min} \leq s_{\max} \leq 9 \times 38^2(\lfloor\beta\rfloor+1)^4 d^{2\lfloor\beta\rfloor+2}$ and size $\mathcal{S} \leq 8\mathcal{W}L \leq 42 * 8 * 9^2 * 38^4(\lfloor\beta\rfloor+1)^{10} d^{4\lfloor\beta\rfloor+4}\lfloor\xi_n\rfloor\lceil\log_2(8\lfloor\xi_n\rfloor)\rceil/(s_{\min}-1)$. Theorem 4.6 implies that, with probability at least $1-\exp(-\xi_n^2)$, the excess ϕ-risk of the ERM $\hat{f}_n$ defined in (13) satisfies

$$R(\hat{f}_n) - R(f_0) \leq C n^{-\beta/(2d+4\beta)}(\log n)^2,$$

where $C = O(B_0(\beta+1)^8 d^{3\beta+3}/(s_{\min}-1))$ is a constant independent of n.

Let $T = \mathcal{B} = 0.5 - (\log n)^{-1}$. Using a modified cross entropy loss $\phi(a) = \max\{-\log(0.5+a), \tau\}$ with $\tau = -\log(1-(\log n)^{-1})$, the excess ϕ-risk of the ERM $\hat{f}_n$ can be improved to $O(n^{-\beta/(d+2\beta)}(\log n)^2)$.

If the approximate low-dimensional manifold Assumption 2.3 holds and for any $\varepsilon \in (0, 1)$, the radius of the neighborhood ρ in Assumption 2.3 satisfies $\rho \leq \rho_\varepsilon$, where ρ_ε is defined in (10), then Theorem 4.7 implies that the rate of convergence can be improved:

$$R(\hat{f}_n) - R(f_0) \leq C n^{-\beta/(2d_\varepsilon + 4\beta)} (\log n)^2,$$

where C is a constant independent of n.

The above discussion shows that deep binary classifications with CNNs are adaptive to the low dimensional structure of the data and the smoothness of the target. Moreover, the prefactor C depends on the ambient dimension d polynomially, which improves the prefactors depending on d exponentially in the existing excess risk bounds [46, 25, 30, 26, 45, 32, 33, 38]. Moreover, the filter lengths here are more flexible comparing with those requiring certain filter lengths in [32, 33, 49].

6 Conclusion

In this work, we derive new approximation error bounds with explicit prefactor in terms of network parameters for Sobolev-regular functions and Hölder smooth functions using deep convolutional networks. New approximation result when the target function is supported on an approximate lower-dimensional manifold is established. Different from existing results, the prefactor of our error bound depends on the ambient dimension polynomially instead of exponentially for Hölder smooth functions. The new approximation results are applied to establish non-asymptotic excess risk bounds for a class of classification methods using CNNs.

An important limitation of our work is that we only considered binary classification problems in applying our approximation results using deep convolutional networks. It would be interesting to apply the results in this work to other settings involving CNN approximation, such as the multiclass classification and regression problems. In addition, our work only partially explains the empirical successes of CNNs in practice. For image data, the assumption of approximate lower-dimensional support does not capture all the structural information. For example, spatial invariance is likely to be expected in some problems such as image classification. We believe that if such properties are taken into account, the theoretical bounds can be further improved. We hope to study this in the future.

References

[1] Babak Alipanahi, Andrew Delong, Matthew T Weirauch, and Brendan J Frey. Predicting the sequence specificities of dna-and rna-binding proteins by deep learning. *Nature biotechnology*, 33(8):831–838, 2015.

[2] Chenglong Bao, Qianxiao Li, Zuowei Shen, Cheng Tai, Lei Wu, and Xueshuang Xiang. Approximation analysis of convolutional neural networks. 2014.

[3] Richard G. Baraniuk and Michael B. Wakin. Random projections of smooth manifolds. *Found. Comput. Math.*, 9(1):51–77, 2009.

[4] Peter L. Bartlett, Nick Harvey, Christopher Liaw, and Abbas Mehrabian. Nearly-tight VC-dimension and pseudodimension bounds for piecewise linear neural networks. *J. Mach. Learn. Res.*, 20:Paper No. 63, 17, 2019.

[5] Peter L. Bartlett, Michael I. Jordan, and Jon D. McAuliffe. Convexity, classification, and risk bounds. *J. Amer. Statist. Assoc.*, 101(473):138–156, 2006.

[6] Peter L Bartlett and Shahar Mendelson. Empirical minimization. *Probability theory and related fields*, 135(3):311–334, 2006.

[7] Benedikt Bauer and Michael Kohler. On deep learning as a remedy for the curse of dimensionality in nonparametric regression. *Ann. Statist.*, 47(4):2261–2285, 2019.

[8] Shai Ben-David and Ruth Urner. The sample complexity of agnostic learning under deterministic labels. In *Conference on Learning Theory*, pages 527–542. PMLR, 2014.

[9] Anselm Blumer, Andrzej Ehrenfeucht, David Haussler, and Manfred K Warmuth. Learnability and the vapnik-chervonenkis dimension. *Journal of the ACM (JACM)*, 36(4):929–965, 1989.

[10] Susanne C Brenner and L Ridgway Scott. *The mathematical theory of finite element methods*, volume 3. Springer, 2008.

[11] Di-Rong Chen, Qiang Wu, Yiming Ying, and Ding-Xuan Zhou. Support vector machine soft margin classifiers: error analysis. *The Journal of Machine Learning Research*, 5:1143–1175, 2004.

[12] Minshuo Chen, Haoming Jiang, Wenjing Liao, and Tuo Zhao. Nonparametric regression on low-dimensional manifolds using deep relu networks. *arXiv preprint arXiv:1908.01842*, 2019.

[13] Minshuo Chen, Haoming Jiang, and Tuo Zhao. Efficient approximation of deep relu networks for functions on low dimensional manifolds. *Advances in Neural Information Processing Systems*, 2019.

[14] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[15] Lawrence C Evans. Partial differential equations. *Graduate studies in mathematics*, 19(4):7, 1998.

[16] Zhiying Fang, Han Feng, Shuo Huang, and Ding-Xuan Zhou. Theory of deep convolutional neural networks ii: Spherical analysis. *Neural Networks*, 131:154–162, 2020.

[17] Charles Fefferman. Whitney's extension problem for c^m. *Annals of Mathematics.*, 164(1):313–359, 2006.

[18] Han Feng, Shuo Huang, and Ding-Xuan Zhou. Generalization analysis of cnns for classification on spheres. *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[19] Yoav Freund, Robert E Schapire, et al. Experiments with a new boosting algorithm. In *icml*, volume 96, pages 148–156. Citeseer, 1996.

[20] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Additive logistic regression: a statistical view of boosting. *Ann. Statist.*, 28(2):337–407, 2000.

[21] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, Cambridge, MA, 2016.

[22] Ingo Gühring, Gitta Kutyniok, and Philipp Petersen. Error bounds for approximations with deep relu neural networks in w s, p norms. *Analysis and Applications*, 18(05):803–859, 2020.

[23] Ingo Gühring and Mones Raslan. Approximation rates for neural networks with encodable weights in smoothness spaces. *Neural Networks*, 134:107–130, 2021.

[24] Sean Hon and Haizhao Yang. Simultaneous neural network approximations in sobolev spaces. *arXiv preprint arXiv:2109.00161*, 2021.

[25] Tianyang Hu, Zuofeng Shang, and Guang Cheng. Sharp rate of convergence for deep neural network classifiers under the teacher-student setting. *arXiv preprint arXiv:2001.06892*, 2020.

[26] Furong Huang, Jordan Ash, John Langford, and Robert Schapire. Learning deep resnet blocks sequentially using boosting theory. In *International Conference on Machine Learning*, pages 2058–2067. PMLR, 2018.

[27] Haotian Jiang, Zhong Li, and Qianxiao Li. Approximation theory of convolutional architectures for time series modelling. In *International Conference on Machine Learning*, pages 4961–4970. PMLR, 2021.

[28] Yuling Jiao, Guohao Shen, Yuanyuan Lin, and Jian Huang. Deep nonparametric regression on approximately low-dimensional manifolds, 2021.

[29] Matti Kääriäinen. Active learning in the non-realizable case. In *International Conference on Algorithmic Learning Theory*, pages 63–77. Springer, 2006.

[30] Yongdai Kim, Ilsang Ohn, and Dongha Kim. Fast convergence rates of deep neural networks for classification. *Neural Networks*, 138:179–197, 2021.

[31] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[32] Michael Kohler, A Krzyzak, and Benjamin Walter. On the rate of convergence of image classifiers based on convolutional neural networks. *arXiv preprint arXiv:2003.01526*, 2020.

[33] Michael Kohler and Sophie Langer. Statistical theory for image classification using deep convolutional neural networks with cross-entropy loss. *arXiv preprint arXiv:2011.13602*, 2020.

[34] Michael Kohler and Benjamin Walter. Analysis of convolutional neural network image classifiers in a rotationally symmetric model. *arXiv preprint arXiv:2205.05500*, 2022.

[35] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

[36] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[37] Shan Lin and Jingwei Zhang. Generalization bounds for convolutional neural networks. *arXiv preprint arXiv:1910.01487*, 2019.

[38] Hao Liu, Minshuo Chen, Tuo Zhao, and Wenjing Liao. Besov function approximation and binary classification on low-dimensional manifolds using convolutional residual networks. In *International Conference on Machine Learning*, pages 6770–6780. PMLR, 2021.

[39] Jianfeng Lu, Zuowei Shen, Haizhao Yang, and Shijun Zhang. Deep network approximation for smooth functions. *SIAM Journal on Mathematical Analysis*, 53(5):5465–5506, 2021.

[40] Zhou Lu, Hongming Pu, Feicheng Wang, Zhiqiang Hu, and Liwei Wang. The expressive power of neural networks: A view from the width. *arXiv preprint arXiv:1709.02540*, 2017.

[41] Enno Mammen and Alexandre B Tsybakov. Smooth discrimination analysis. *The Annals of Statistics*, 27(6):1808–1829, 1999.

[42] Pascal Massart and Élodie Nédélec. Risk bounds for statistical learning. 34(5):2326 – 2366, 2006.

[43] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of Machine Learning*. MIT Press, Cambridge, MA, 2018.

[44] Ryumei Nakada and Masaaki Imaizumi. Adaptive approximation and estimation of deep neural network with intrinsic dimensionality. *arXiv preprint arXiv:1907.02177*, 2019.

[45] Atsushi Nitanda and Taiji Suzuki. Functional gradient boosting based on residual network perception. In *International Conference on Machine Learning*, pages 3819–3828. PMLR, 2018.

[46] Ilsang Ohn and Yongdai Kim. Smooth function approximation by deep neural networks with general activation functions. *Entropy*, 21(7):627, 2019.

[47] Kenta Oono and Taiji Suzuki. Approximation and non-parametric estimation of resnet-type convolutional neural networks. In *International Conference on Machine Learning*, pages 4922–4931. PMLR, 2019.

[48] Philipp Petersen and Felix Voigtlaender. Optimal approximation of piecewise smooth functions using deep relu neural networks. *Neural Networks*, 108:296–330, 2018.

[49] Philipp Petersen and Felix Voigtlaender. Equivalence of approximation by convolutional neural networks and fully-connected networks. *Proceedings of the American Mathematical Society*, 148(4):1567–1581, 2020.

[50] Maithra Raghu, Ben Poole, Jon Kleinberg, Surya Ganguli, and Jascha Sohl-Dickstein. On the expressive power of deep neural networks. In *International Conference on Machine Learning*, pages 2847–2854. PMLR, 2017.

[51] Johannes Schmidt-Hieber. Deep relu network approximation of functions on a manifold. *arXiv preprint arXiv:1908.00695*, 2019.

[52] Johannes Schmidt-Hieber. Rejoinder: "Nonparametric regression using deep neural networks with ReLU activation function" [MR4134775; MR4134776; MR4134777; 4134778; MR4134774]. *Ann. Statist.*, 48(4):1916–1921, 2020.

[53] Guohao Shen, Yuling Jiao, Yuanyuan Lin, Joel L Horowitz, and Jian Huang. Deep quantile regression: Mitigating the curse of dimensionality through composition. *arXiv preprint arXiv:2107.04907*, 2021.

[54] Zuowei Shen, Haizhao Yang, and Shijun Zhang. Nonlinear approximation via compositions. *Neural Networks*, 119:74–84, 2019.

[55] Zuowei Shen, Haizhao Yang, and Shijun Zhang. Deep network approximation characterized by number of neurons. *Commun. Comput. Phys.*, 28(5):1768–1811, 2020.

[56] Alexandre B. Tsybakov. Optimal aggregation of classifiers in statistical learning. *Ann. Statist.*, 32(1):135–166, 2004.

[57] Qiang Wu, Yiming Ying, and Ding-Xuan Zhou. Multi-kernel regularized classifiers. *Journal of Complexity*, 23(1):108–134, 2007.

[58] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016.

[59] Yuhong Yang. Minimax nonparametric classification. i. rates of convergence. *IEEE Transactions on Information Theory*, 45(7):2271–2284, 1999.

[60] Dmitry Yarotsky. Error bounds for approximations with deep relu networks. *Neural Networks*, 94:103–114, 2017.

[61] Dmitry Yarotsky. Optimal approximation of continuous functions by very deep relu networks. In *Conference on Learning Theory*, pages 639–649. PMLR, 2018.

[62] Tong Zhang. Statistical behavior and consistency of classification methods based on convex risk minimization. *Ann. Statist.*, 32(1):56–85, 2004.

[63] Nikita Zhivotovskiy and Steve Hanneke. Localization of vc classes: Beyond local rademacher complexities. *Theoretical Computer Science*, 742:27–49, 2018.

[64] Ding-Xuan Zhou. Theory of deep convolutional neural networks: Downsampling. *Neural Networks*, 124:319–327, 2020.

[65] Ding-Xuan Zhou. Universality of deep convolutional neural networks. *Appl. Comput. Harmon. Anal.*, 48(2):787–794, 2020.

[66] Jian Zhou and Olga G Troyanskaya. Predicting effects of noncoding variants with deep learning–based sequence model. *Nature methods*, 12(10):931–934, 2015.

[67] Pan Zhou and Jiashi Feng. Understanding generalization and optimization performance of deep cnns. In *International Conference on Machine Learning*, pages 5960–5969. PMLR, 2018.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A] This may not apply to our experiment on function approximation.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [No] Toy example is involved in our experiment which is not computationally expensive.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [N/A]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

Appendix

In the appendix, we include detailed descriptions on convex surrogate losses,convolutional neural networks, non-asymptotic error bounds for commonly used loss functions, and prove Theorems 2.1,2.2, 2.4, 4.4, 4.6, and 4.7, as well as the error bounds for cross entropy, SVM, logistic, exponential and least squares loss examples. A toy example on the numerical performance of CNN approximation is presented in Appendix D.

A Convex surrogate loss

We next give a brief review of the convex surrogate loss functions and discuss in details on the connection between the excess risk with respect to the ϕ-loss and that of 0-1 loss [62, 5].

Let ϕ be a given convex univariate function $\phi : \mathbb{R} \to [0, \infty)$. Instead of minimizing the excess risk R over $\mathcal{H}$, we consider minimizing the risk with respect to the loss ϕ (ϕ-risk)

$$R(f) := \mathbb{E}\{\phi(Yf(X))\}$$

over a certain class of functions $\mathcal{F}$, where $\phi : \mathbb{R} \to [0, \infty)$ is some generic loss function. For the special case when $\mathcal{H} = \{h : h(x) = \text{sign}(f(x)), f \in \mathcal{F}\}$ and $\phi(\cdot)$ is a step function, i.e., $\phi(x) = 1$ if $x < 0$ and $\phi(x) = 0$ if $x \geq 0$, then to minimize R over $\mathcal{H}$, it suffices to minimize R over $\mathcal{F}$. Given $S = \{(X_i, Y_i)\}_{i=1}^n$, let $R_n(f) := \sum_{i=1}^n \phi(Y_i f(X_i))/n$ be the empirical risk of f w.r.t the ϕ-loss, and define the empirical risk estimator (ERM) as

$$\hat{f}_n \in \arg\min_{f \in \mathcal{F}} R_n(f). \tag{20}$$

Based on the ERM $\hat{f}_n$, a classification rule or a classifier $\hat{h}_n(x) := \text{sign}(\hat{f}_n(x))$ for $x \in \mathcal{X}$ can be induced which aims at minimizing the 0-1 risk. As shown in [62] and [5], for a properly chosen ϕ, $\hat{f}_n$ can indeed help reduce the excess risk $R(\hat{h}_n) - R_0$. More precisely, let $R_0 := \inf_{f \text{ measurable}} R(f)$, then for a proper ϕ, we have

$$\psi(R(\hat{h}_n) - R_0) \leq R(\hat{f}_n) - R_0,$$

where $\psi : [-1, 1] \to [0, \infty)$ is a nonnegative continuous function, invertible on $[0, 1]$, and achieves its minimum at 0 with $\psi(0) = 0$. A wide variety of popular classification methods are based on this tactic. For instance, when $\phi(a) = \log\{1 + \exp(-a)\}$, it is the logistic regression [20]; when $\phi(a) = \max\{1-a, 0\}$, it becomes the SVM [14]; when $\phi(a) = \exp(-a)$ is the exponential loss, it is the AdaBoost algorithm [19]; when $\phi(a) = (1 - a)^2$, it is the least squares method for classification, and so on.

Define the measurable minimizer of R as

$$f_0 = \arg \min_{f \text{ measurable}} \mathbb{E}\{\phi(Yf(X))\}, \tag{21}$$

and the corresponding minimal ϕ-risk as $R_0 = R(f_0)$. Clearly, the optimal f_0 depends on the loss function ϕ. If ϕ is not properly chosen, the resulting f_0 can be poor and thus the classifier $h_0(x) = \text{sign}(f_0(x))$ based on f_0 can be invalid. To study the basic conditions imposed on ϕ, so as to produce a valid classifier towards minimizing the 0-1 loss, we need a thorough understanding of the risk R.

Let the conditional ϕ-risk of f given $X = x$ be denoted by $R_x(f) := \mathbb{E}\{\phi(Yf(X)) \mid X = x\}$. Recall that $\eta(x) = \mathbb{P}\{Y = 1 \mid X = x\}$ as defined in (15). We have

$$R_x(f) = \eta(x)\phi(f(x)) + \{1 - \eta(x)\}\phi(-f(x)),\ x \in \mathcal{X}.$$

For a good classifier, if $\eta(x) > 1/2$, it is naturally expected that $f_0(x) > 0$ and $h_0(x) = \text{sign}(f_0(x)) = 1$ (correct sign); thus to encourage $f_0(x) > 0$, we should at least require $\phi(f_0(x)) < \phi(-f_0(x))$ to minimize the conditional risk R_x. Similarly, if $\eta(x) < 1/2$, we expect that the contrary happens. A rigorous definition for such an ideal ϕ is given in Definition 1 of [5].

Definition A.1 (Classification-calibrated)**.** For $\eta \in [0, 1]$ and $a \in \mathbb{R}$, define $H(\eta, a) = \eta\phi(a) + (1 - \eta)\phi(-a)$. Then, we say that ϕ is *classification-calibrated* if for any $\eta \neq 1/2$,

$$\inf_{a \in \mathbb{R}} H(\eta, a) < \inf_{a \in \mathbb{R}: a(2\eta - 1) \leq 0} H(\eta, a).$$

With a classification-calibrated loss ϕ, it is guaranteed that the "incorrect" label $\text{sign}(f(X))$ in the sense that it is inconsistent with the Bayes estimator $\text{sign}(2\eta(X)-1))$ results in a strictly larger loss under ϕ. It was shown by [5] that, the surrogate loss function ϕ is able to produce the optimal Bayes classifier. For ease of reference, we state this result as Lemma A.2 below.

Lemma A.2 (Theorem 1 of [5])**.** *For any nonnegative loss function $\phi : \mathbb{R} \to [0, \infty)$, any measurable $f : \mathcal{X} \to \mathbb{R}$, its induced classifier $h_f = \text{sign}(f) : \mathcal{X} \to \{\pm 1\}$, and any probability measure $\mathbb{P}$ of (X, Y) on $\mathcal{X} \times \{\pm 1\}$,*

$$\psi(R(h_f) - R_0) \leq R(f) - R_0,$$

where $\psi : [-1, 1] \to [0, \infty)$ is the Fenchel-Legendre biconjugate of $\tilde{\psi} : [-1, 1] \to \mathbb{R}$, and

$$\tilde{\psi}(\theta) = \inf_{a \in \mathbb{R}: a(2\eta-1) \leq 0} H(\frac{1+\theta}{2}) - \inf_{a \in \mathbb{R}} H(\frac{1+\theta}{2}).$$

Besides, if ϕ is classification-calibrated, then for any sequence $\{a_m\}$ in $[0, 1]$, $\psi(a_m) \to 0$, if and only if $a_m \to 0$.

Lemma A.2 has some important implications. First, for any nonnegative ϕ, ψ is simply the functional convex hull (the greatest convex minorant) of $\tilde{\psi}$. Both ψ and $\tilde{\psi}$ are continuous on $[-1, 1]$, and ψ is nonnegative that attains its minimum at 0 with $\psi(0) = 0$. Second, if the loss ϕ is nonnegative and classification-calibrated, then $\psi(\theta) > 0$ for all $\theta \in (0, 1]$ and ψ is invertible on $[0, 1]$. In this case, we have $R(h_f) - R_0 \leq \psi^{-1}(R(f) - R_0)$. Third, if ϕ is nonnegative convex and classification-calibrated, then $\psi(\theta) = \phi(0) - \inf_{a \in \mathbb{R}} H((1+\theta)/2, a)$, which gives an easy way to compute the function ψ. Note that for the formulation of ψ, Theorem 34 in [11] shows that if ϕ is convex, $\phi''(0) > 0$ exists and $\phi' < 0$, then $\psi(\theta) = \theta^2$.

Next, we present several examples of ϕ, and the corresponding f_0, $R(f_0)$, ψ and its inverse on $[0, 1]$ in Table 3.

Table 3: Minimizer and minimal conditional risk under different loss functions ϕ.

	Least squares	SVM	Exponential	Logistic	Cross entropy
$\phi(a)$	$(1-a)^2$	$\max\{1-a, 0\}$	$\exp(-a)$	$\log\{1+\exp(-a)\}$	$-\log\{0.5+a\}$
$f_0(x)$	$2\eta - 1$	$\text{sign}(2\eta-1)$	$\frac{1}{2}\log(\frac{\eta}{1-\eta})$	$\log(\frac{\eta}{1-\eta})$	$\eta - 0.5$
$R_x(f_0)$	$4\eta(1-\eta)$	$1 - \lvert 2\eta - 1\rvert$	$2\sqrt{\eta(1-\eta)}$	$-\eta\log(\eta) - (1-\eta)\log(1-\eta)$	$-\eta\log(\eta) - (1-\eta)\log(1-\eta)$
$\psi(\theta)$	θ^2	$\lvert\theta\rvert$	$1 - \sqrt{1-\theta^2}$	θ^2	θ^2
$\psi^{-1}(\theta)$	$\sqrt{\theta}$	$\lvert\theta\rvert$	$\sqrt{1-(1-\theta)^2}$	$\sqrt{\theta}$	$\sqrt{\theta}$

Note: $\eta(x)$ is written as η for notational simplicity.

We also list the Lipschitz constant B_ϕ and $\Delta_\phi(T)$ for commonly-used SVM, cross entropy, least squares, exponential and logistic loss functions in Table 4.

Table 4: Bound ϕ_B of ϕ, Lipschitz constant B_ϕ and $\Delta_\phi(T)$ for the truncated f_0 under least squares, logistic, exponential and SVM hinge loss function restricted to $[-\mathcal{B}, \mathcal{B}]$ for $1 \leq T \leq \mathcal{B}$; and cross entropy loss function restricted to $[-\mathcal{B}, \mathcal{B}]$ for $T \leq \mathcal{B} \leq 0.5$.

	Least squares	SVM	Exponential	Logistic	Cross entropy
$\phi(a)$	$(1-a)^2$	$\max\{1-a, 0\}$	$\exp(-a)$	$\log\{1+\exp(-a)\}$	$-\log\{0.5+a\}$
$f_0(x)$	$2\eta - 1$	$\text{sign}(2\eta-1)$	$\frac{1}{2}\log(\frac{\eta}{1-\eta})$	$\log(\frac{\eta}{1-\eta})$	$\eta - 0.5$
ϕ_B	$(\mathcal{B}+1)^2$	1	$\exp(\mathcal{B})$	$\log(1+\exp(\mathcal{B}))$	$-\log(0.5-\mathcal{B})$
B_ϕ	$2\mathcal{B}+2$	***	$\exp(\mathcal{B})$	$1/\{\exp(-\mathcal{B})+1\}$	$1/(0.5+\mathcal{B})$
$\Delta_\phi(T)$	0	0	$\exp(-T)$	$\log\{1+\exp(-T)\}$	$-\log\{1+(0.5-T)\}$

Note: $\eta(x)$ is written as η for notational simplicity and "***" stands for not applicable.

B Convolutional neural networks

As indicated by their name, CNNs employ a mathematical operation called convolution. Convolutional networks are a specialized type of structured sparse feedforward neural network (FNN) that use

convolution in place of general matrix multiplication in at least one of their layers [21]. There are different formulations of CNNs in the literature [2, 67, 47, 37, 64, 65].

For a general neural network, let L denote the number of layers and $(\sigma_1, \ldots, \sigma_L)$ denote the activation functions, such as the rectified linear unit (ReLU) and max pooling function. Besides, let d_i denote the width (the number of neurons or computational units) of the i-th layer, $i = 1, \ldots, L+1$, and $\mathcal{W} = \max\{d_1, \ldots, d_L\}$ denote the maximum width among layers. In our present classification problem, $d_0 = d$ (the dimension of the input X) and $d_{L+1} = 1$ (the dimension of the output Y).

In the following, two common types of neural networks including FNNs or MLPs and CNNs with downsampling [64] are shown.

- The architecture of a MLP $f_{MLP} : \mathbb{R}^d \to \mathbb{R}$ can be expressed as a composition of a series of linear transformations: for any $x \in \mathbb{R}^d$,

$$f_{MLP}(x) = W_{L+1}\sigma_L(W_L\sigma_{L-1}(\cdots\sigma_2(W_2\sigma_1(W_1x + b_1) + b_2)\cdots) + b_L) + b_{L+1}, \quad (22)$$

  where $W_i \in \mathbb{R}^{d_{i+1}\times d_i}$ is a fully connected weight matrix, and $b_i \in \mathbb{R}^{d_{i+1}}$ is a bias vector in the i-th linear transformation. Usually, the activation functions $(\sigma_1, \ldots, \sigma_L)$ are the ReLU activation, where $\mathrm{ReLU}(x) = \max\{0, x\}$ for any $x \in \mathbb{R}$ (defined for each component of x if x is a vector).

- The formulation of a downsampled CNN $f_{\text{CNN}} : \mathbb{R}^d \to \mathbb{R}$ is essentially a specially structured architectures of MLP, and here we use the formulation defined in [67, 64, 65]

$$f_{\text{CNN}} = A_{L+1} \circ A_L \circ \cdots \circ A_2 \circ A_1, \quad (23)$$

  where $\circ$ denotes the functional composition and A_i is a linear operation, $i = 1, \ldots, L+1$. The A_i's are either convolutional operators or downsampling operators. For convolutional layers,

$$A_i = \sigma(W_i^c x + b_i^c),$$

  where σ is the rectified linear unit (ReLU) activation function applying to each component of the input vector, $W_i^c \in \mathbb{R}^{d_i\times d_{i-1}}$ is the structured sparse Toeplitz type weight matrix induced by the convolutional filter $\{w^{(i)_j}\}_{j=0}^{s^{(i)}}$ with filter length $s^{(i)} \in \mathbb{N}$ involving only $s^{(i)} + 1$ free parameters. The corresponding $(d_{i-1} + s^{(i)}) \times d_{i-1}$Toeplitz type weight matrix W_i^c is given explicitly by

$$W_i^c = \begin{bmatrix} w_0^{(i)} & 0 & 0 & 0 & \cdots & \cdots & 0 \\ w_1^{(i)} & w_0^{(i)} & 0 & 0 & \cdots & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ w_{s^{(i)}}^{(i)} & w_{s^{(i)}-1}^{(i)} & \cdots & w_0^{(i)} & 0 & \cdots & 0 \\ 0 & w_{s^{(i)}}^{(i)} & \cdots & w_1^{(i)} & w_0^{(i)} & 0\cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & w_{s^{(i)}}^{(i)} & \cdots & w_1^{(i)} & w_0^{(i)} \\ 0 & \cdots & 0 & 0 & w_{s^{(i)}}^{(i)} & \cdots & w_1^{(i)} \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 0 & w_{s^{(i)}}^{(i)} \end{bmatrix} \in \mathbb{R}^{(d_{i-1}+s^{(i)})\times d_{i-1}}.$$

  It can be seen that by convolutional operations here, the network has linearly increasing widths $d_i = d_{i-1} + s^{(i)}$ for convolutional layers. On the contrary, the downsampling layers decrease the width of the networks.

  For downsampling layers,

$$A_i(x) = D_i(x) = W_i^D x,$$

  where $D_i : \mathbb{R}^{d_{i-1}} \to \mathbb{R}^{[d_{i-1}/m_i]}$ is the downsampling operator with a scaling parameter $m_i \le d_{i-1}$ and $W^D \in \mathbb{R}^{d_{i-1}\times[d_{i-1}/m_i]}$ is corresponding structured sparse weight matrix induced by D_i in the i-th layer. Here the downsampling operator [64] is given by

$$D_i(x) = (x_{jm_i})_{j=1}^{[d_{i-1}/m_i]}, \qquad x \in \mathbb{R}^{d_{i-1}},$$

which takes a real vector $x \in \mathbb{R}^{d_{i-1}}$ as input and outputs a subvector of x with length $\mathbb{R}^{[d_{i-1}/m_i]}$ where $[a]$ denotes the integer part of $a \in \mathbb{R}$. It can be seen that by downsampling operation D_i here, the network width is scaled down by about m_i times. For the function class of the downsampled CNNs formulated as in (23), we let L be th number of hidden layers and $\mathcal{S}$ be the total number of parameters for networks in $\mathcal{F}_{\text{CNN}}$ and let $\mathcal{W}$ be the maximum filter length of convolutional layers.

A downsampled CNN f_{CNN} is called uniform if the length of its filters $\{s^{(i)}\}$ in the convolutional layers are the same. [64] showed that any multilayer perceptron can be represented by a uniform downsampled CNN with parameters no more than 8 times as that of the MLP.

C Non-asymptotic error bounds for commonly used loss functions

In this section, we show more examples of the non-asymptotic error bounds for commonly used loss functions including the hinge loss, the logistic loss, the exponential loss and the least square loss.

C.1 SVM: the hinge loss

For SVM, the loss function $\phi(a) = \max\{1-a, 0\}$ is not differentiable at $a = 1$ and thus the ϕ-risk minimizer $f_0(x) = \text{sign}(2\eta(x) - 1)$ may not be continuous even though η is continuous, as $f_0(x)$ is discontinuous when $\eta(x) = 1/2$. To tackle this problem, we additionally impose the low noise condition on η [41, 56], i.e., there exist $c_{\text{noise}} > 0$ and $q \in [0, \infty]$ such that for any $t > 0$, $\mathbb{P}(|2\eta(X) - 1| \le t) \le c_{\text{noise}} t^q$, where the constant q is called *the noise exponent*.

Suppose that Assumptions 4.3 holds $f_0 \in \mathcal{H}^\beta([0,1]^d, B_0)$. Denote $\zeta_n = n^{d/\{2d+4\beta(q+1)\}}$. Let $\mathcal{F}_{\text{CNN}}$ be the class of CNNs defined in (2) with $\mathcal{B} \ge 1$, depth $L \le 378 \cdot 38^2(\lfloor\beta\rfloor + 1)^6 d^{2\lfloor\beta\rfloor+2}\lfloor\zeta_n\rfloor\lceil\log(8\zeta_n)\rceil/(s_{\min}-1)$, filter lengths $2 \le s_{\min} \le s_{\max} \le 9 \times 38^2(\lfloor\beta\rfloor+1)^4 d^{2\lfloor\beta\rfloor+2}$ and size $\mathcal{S} \le 8\mathcal{W}L \le 42 * 8 * 9^2 * 38^4(\lfloor\beta\rfloor+1)^{10} d^{4\lfloor\beta\rfloor+4}\lfloor\zeta_n\rfloor\lceil\log(8\zeta_n)\rceil/(s_{\min}-1)$. Theorem 4.6 implies that, with probability at least $1 - \exp(-\zeta_n^2)$, the excess ϕ-risk of the ERM $\hat{f}_n$ defined in (13) satisfies

$$R(\hat{f}_n) - R(f_0) \le C n^{-\beta(q+1)/\{d+2\beta(q+1)\}} (\log n)^2,$$

where $C = O(c_{\text{noise}} 4^q B_0^{q+1} (\beta+1)^{2q+8} d^{(q+1)(3\beta+3)})$. For the induced classifier $\hat{h}_n = \text{sign}(\hat{f}_n)$, the excess misclassification error satisfies

$$R(\hat{h}_n) - R(h_0) \le C n^{-\beta(q+1)/\{d+2\beta(q+1)\}} (\log n)^2.$$

The excess risk bound depends on q, the noise exponent. When $q = 0$ (high noise), the convergence rate is $n^{-\beta/(d+2\beta)}$; when $q = +\infty$ (no noise), the rate will be significantly improved to $n^{-1/2}$. A similar result can be found in Theorem 3.3 of [30].

Suppose Assumption 2.3 holds and for any $\varepsilon \in (0,1)$, the radius of the neighborhood ρ in this assumption satisfies $\rho \le \rho_\varepsilon$, then Theorem 4.7 implies that

$$R(\hat{f}_n) - R(f_0) \le C n^{-\beta(q+1)/\{0.75 d_\varepsilon + 2\beta(q+1)\}} (\log n)^2,$$

where C is a constant independent of n.

C.2 The logistic loss

For the logistic loss function $\phi(a) = \log\{1+\exp(-a)\}$, the minimizer of ϕ-risk $\log\{\eta(x)/1-\eta(x)\}$ is unbounded with the Lipschitz constant $B_\phi = 1/\{\exp(-\mathcal{B}) + 1\} \le 1$ on $[-\mathcal{B}, \mathcal{B}]$, and $\phi_B = \log(1+\exp(\mathcal{B})) \le \mathcal{B} + \infty$ and the truncation error $\Delta_\phi(T) = \log\{1+\exp(-T)\}$. A feasible choice is $T = \mathcal{B} = \log(n)$ where n is the sample size, in which case $\Delta_\phi(T) = \log\{1 + 1/n\} \le 1/n$.

Suppose Assumptions 4.3 (in main context) holds and $f_0 \in \mathcal{H}^\beta([0,1]^d, B_0)$. Take $N = 1$ in (7) and $M = \lfloor\xi\rfloor$ in (4) and let $\mathcal{F}_{\text{CNN}}$ be the class of CNNs in (2) with depth $L \le 378 \cdot 38^2(\lfloor\beta\rfloor + 1)^6 d^{2\lfloor\beta\rfloor+2}\lfloor n^{d/\{2d+4\beta\}}\rfloor\lceil\log(8n)\rceil/(s_{\min}-1)$, filter lengths $2 \le s_{\min} \le s_{\max} \le 9 \times 38^2(\lfloor\beta\rfloor + 1)^4 d^{2\lfloor\beta\rfloor+2}$ and size $\mathcal{S} \le 8\mathcal{W}L \le 42*8*9^2*38^4(\lfloor\beta\rfloor+1)^{10} d^{4\lfloor\beta\rfloor+4}\lfloor n^{d/\{2d+4\beta\}}\rfloor\lceil\log(8n)\rceil/(s_{\min}-1)$.

Theorem 4.6 implies that, with probability at least $1-\exp\{-n^{d/(d+2\beta)}\}$, the excess ϕ-risk of the ERM $\hat{f}_n$ defined in (13) satisfies

$$R(\hat{f}_n) - R(f_0) \leq n^{-1} + (\log n) n^{-\beta/(d+2\beta)}$$
$$\times\{c(\lfloor\beta\rfloor+1)^8 d^{3\lfloor\beta\rfloor+3}(\log n)^2/(s_{\min}-1) + 18B_0(\lfloor\beta\rfloor+1)^2 d^{\lfloor\beta\rfloor+(\lfloor\beta\rfloor\vee 1)/2}\},$$

or simply

$$R(\hat{f}_n) - R(f_0) \leq C(d,\beta,B_0,s_{\min})(\log n)^3 n^{-\beta/(d+2\beta)},$$

where $C(d,\beta,B_0,s_{\min}) = O(B_0(\lfloor\beta\rfloor+1)^8 d^{3\lfloor\beta\rfloor+3}/(s_{\min}-1))$ is a constant independent of n. For the induced classifier $\hat{h}_n = \text{sign}(\hat{f}_n)$, the excess misclassification error satisfies

$$R(\hat{h}_n) - R(h_0) \leq \sqrt{C(d,\beta,B_0,s_{\min})}(\log n)^{3/2} n^{-\beta/(2d+4\beta)}.$$

In addition, suppose the approximate low-dimensional manifold Assumption 2.3 also holds and for any $\varepsilon \in (0,1)$, the radius of the neighborhood ρ in Assumption 2.3 satisfies $\rho \leq C_2(NM)^{-2\beta/d_\varepsilon}(\lfloor\beta\rfloor+1)^2 d^{1/2} d_\varepsilon^{3\lfloor\beta\rfloor}(\sqrt{d/d_\varepsilon}+1-\varepsilon)^{-1}(1-\varepsilon)^{1-\beta}$, where $d_\varepsilon = O(d_{\mathcal{M}}\log(d/\varepsilon)/\varepsilon^2)$ is an integer satisfying $d_{\mathcal{M}} \leq d_\varepsilon \ll d$. Then Theorem 4.7 shows that the rate of convergence can be improved to

$$R(\hat{f}_n) - R(f_0) \leq C(d,d_\varepsilon,\beta,B_0,s_{\min})(\log n)^2 n^{-\beta/(d_\varepsilon+2\beta)},$$

where $C(d,d_\varepsilon,\varepsilon,\beta,B_0,s_{\min}) = O(B_0(1-\varepsilon)^{-\beta}(\lfloor\beta\rfloor+1)^8 d^{1/2} d_\varepsilon^{3\lfloor\beta\rfloor+3}/(s_{\min}-1))$ is a constant independent of n.

C.3 The exponential loss

For the exponential loss function $\phi(a) = \exp(-a)$, the minimizer of ϕ-risk $\log\{\eta(x)/1-\eta(x)\}/2$ is unbounded, the truncation error $\Delta_\phi(T) = \exp(-T)$, the Lipschitz constant $B_\phi = \exp(\mathcal{B})$ and $\phi_B = \exp(\mathcal{B})$. A feasible choice is $T = \mathcal{B} = (\beta/\{2d+4\beta\})\log(n)$ where n is the sample size.

Suppose Assumptions 4.3 (in main context) holds and $f_0 \in \mathcal{H}^\beta([0,1]^d, B_0)$. Take $N=1$ in (7) and $M = \lfloor\xi\rfloor$ in (4) and let $\mathcal{F}_{\text{CNN}}$ be the class of CNNs in (2) with depth $L \leq 378\cdot 38^2(\lfloor\beta\rfloor+1)^6 d^{2\lfloor\beta\rfloor+2}\lfloor n^{d/\{2d+4\beta\}}\rfloor\lceil\log(8n)\rceil/(s_{\min}-1)$, filter lengths $2 \leq s_{\min} \leq s_{\max} \leq 9\times 38^2(\lfloor\beta\rfloor+1)^4 d^{2\lfloor\beta\rfloor+2}$ and size $\mathcal{S} \leq 8\mathcal{W}L \leq 42*8*9^2*38^4(\lfloor\beta\rfloor+1)^{10} d^{4\lfloor\beta\rfloor+4}\lfloor n^{d/\{2d+4\beta\}}\rfloor\lceil\log(8n)\rceil/(s_{\min}-1)$. Theorem 4.6 implies that, with probability at least $1-\exp\{-n^{d/(d+2\beta)}\}$, the excess ϕ-risk of the ERM $\hat{f}_n$ defined in (13) satisfies

$$R(\hat{f}_n) - R(f_0) \leq n^{-\beta/(2d+4\beta)} + n^{-\beta/(2d+4\beta)}$$
$$\times\{c(\lfloor\beta\rfloor+1)^8 d^{3\lfloor\beta\rfloor+3}(\log n)^2/(s_{\min}-1) + 18B_0(\lfloor\beta\rfloor+1)^2 d^{\lfloor\beta\rfloor+(\lfloor\beta\rfloor\vee 1)/2}\},$$

or simply

$$R(\hat{f}_n) - R(f_0) \leq C(d,\beta,B_0,s_{\min})(\log n)^2 n^{-\beta/(2d+4\beta)},$$

where $C(d,\beta,B_0,s_{\min}) = O(B_0(\lfloor\beta\rfloor+1)^8 d^{3\lfloor\beta\rfloor+3}/(s_{\min}-1))$ is a constant independent of n.

In addition, suppose the approximate low-dimensional manifold Assumption 2.3 also holds and for any $\varepsilon \in (0,1)$, the radius of the neighborhood ρ in Assumption 2.3 satisfies $\rho \leq C_2(NM)^{-2\beta/d_\varepsilon}(\lfloor\beta\rfloor+1)^2 d^{1/2} d_\varepsilon^{2\lfloor\beta\rfloor}(\sqrt{d/d_\varepsilon}+1-\varepsilon)^{-1}(1-\varepsilon)^{1-\beta}$, where $d_\varepsilon = O(d_{\mathcal{M}}\log(d/\varepsilon)/\varepsilon^2)$ is an integer satisfying $d_{\mathcal{M}} \leq d_\varepsilon \ll d$. Then Theorem 4.7 shows that the rate of convergence can be improved to

$$R(\hat{f}_n) - R(f_0) \leq C(d,d_\varepsilon,\varepsilon,\beta,B_0,s_{\min})(\log n)^2 n^{-\beta/(2d_\varepsilon+4\beta)},$$

where $C(d,d_\varepsilon,\varepsilon,\beta,B_0,s_{\min}) = O(B_0(1-\varepsilon)^{-\beta}(\lfloor\beta\rfloor+1)^8 d^{1/2} d_\varepsilon^{3\lfloor\beta\rfloor+3}/(s_{\min}-1))$ is a constant independent of n.

To improve the rate of convergence, one can use a modified exponential loss function, i.e. $\phi(a) = \max\{\exp(-a),\tau\}$ for some small $\tau > 0$. The minimum of modified ϕ can be achieved at $a^*(\tau) = -\log(\tau)$, and the minimizer $f_0(x)$ will be a truncated version of $\log(\eta/(1-\eta))/2$ (truncated by $a^*(\tau) = -\log(\tau)$). In light of this, under the modified exponential loss with $\tau = \exp(-T)$, the

corresponding measurable minimizer defined in (14) is $f_{0,T}$, the truncated version of the minimizer f_0 under the original exponential loss; see Table 4. Then $\Delta_\phi(T) = 0$ by its definition since the infimum of the modified loss can be achieved within $[-T, T]$. By choosing $T = \mathcal{B} = \log\log n$ and $\tau = \exp(-T)$, with the modified exponential loss, Theorem 4.6 implies that with probability at least $1 - \exp\{-n^{d/(d+2\beta)}\}$, the excess ϕ-risk of the ERM $\hat{f}_n$ defined in (13) satisfies

$$R(\hat{f}_n) - R(f_0) \leq C(d, \beta, B_0, s_{\min})(\log n)^3 n^{-\beta/(d+2\beta)},$$

where $C(d, \beta, B_0, s_{\min}) = O(B_0(\lfloor\beta\rfloor + 1)^8 d^{3\lfloor\beta\rfloor+3}/(s_{\min} - 1))$ is a constant independent of n. For the induced classifier $\hat{h}_n = \text{sign}(\hat{f}_n)$, the excess misclassification error satisfies

$$R(\hat{h}_n) - R(h_0) \leq \sqrt{C(d, \beta, B_0, s_{\min})}(\log n)^{3/2} n^{-\beta/(2d+4\beta)}.$$

C.4 The least squares loss

For the least squares loss $\phi(a) = (1 - a)^2$, we first prove that the approximation error is of a special form. For any $f \in \mathcal{F}_{\text{CNN}}$

$$\begin{aligned}
R(f) - R(f_0) &= \mathbb{E}\{\phi(Yf(X)) - \phi(Yf_0(X))\} \\
&= \mathbb{E}\{(1 - Yf(X))^2 - (1 - Yf_0(X))^2\} \\
&= \mathbb{E}[f(X)^2 - f_0(X)^2 - 2Y\{f(X) - f_0(X)\}] \\
&= \mathbb{E}[(f(X) - f_0(X))\{f(X) + f_0(X) - 2\mathbb{E}(Y \mid X)\}] \\
&= \mathbb{E}[(f(X) - f_0(X))(f(X) + f_0(X) - 2f_0(X))] \\
&= \mathbb{E}|f(X) - f_0(X)|^2.
\end{aligned}$$

Ans we have

$$\inf_{f \in \mathcal{F}_{\text{CNN}}} R(f) - R(f_0) = \inf_{f \in \mathcal{F}_{\text{CNN}}} \mathbb{E}|f(X) - f_0(X)|^2.$$

Besides, note that for least squares $f_0 = 2\eta - 1$ and $\|f_0\|_\infty = 1$, thus taking the truncation $T = 1$ is sufficient for the error control analysis. Let $\mathcal{F}_{\text{CNN}}$ be a class of CNNs defined in (2) with $\mathcal{B} \geq T = 1$.

By Theorem 2.2, there exists a function f implemented by CNN with depth $L \leq 378 \cdot 38^2(\lfloor\beta\rfloor + 1)^6 d^{2\lfloor\beta\rfloor+2} M\lceil\log(8M)\rceil/(s_{\min} - 1)$, filter lengths $2 \leq s_{\min} \leq s_{\max} \leq 9 \times 38^2(\lfloor\beta\rfloor + 1)^4 d^{2\lfloor\beta\rfloor+2}$ and size $\mathcal{S} \leq 8\mathcal{W}L \leq 42 * 8 * 9^2 * 38^4(\lfloor\beta\rfloor + 1)^{10} d^{4\lfloor\beta\rfloor+4} N\lceil\log(8N)\rceil/(s_{\min} - 1)$ such that

$$\mathbb{E}\|f(X) - f_0(X)\|_2^2 \leq 18^2 B_0^2(\lfloor\beta\rfloor + 1)^4 d^{2\lfloor\beta\rfloor+(\beta\vee 1)}(NM)^{-4\beta/d},$$

and based on Lemma 4.1 and Theorem 4.2, the empirical ϕ-risk minimizer $\hat{f}_n$ defined in (13) satisfies, for any $\delta > 0$, with probability at least $1 - \delta$,

$$\begin{aligned}
R(\hat{f}_n) - R(f_0) \leq \frac{\phi_B}{\sqrt{n}}\Big(\sqrt{c\mathcal{S}L\log(\mathcal{S})\log n} + \sqrt{\log(1/\delta)}\Big) \\
+ 18^2 B_0^2(\lfloor\beta\rfloor + 1)^4 d^{2\lfloor\beta\rfloor+(\beta\vee 1)}(NM)^{-4\beta/d}.
\end{aligned}$$

Let $\mathcal{F}_{\text{CNN}}$ be the class of CNNs defined in (2) with depth $L \leq 378 \cdot 38^2(\lfloor\beta\rfloor + 1)^6 d^{2\lfloor\beta\rfloor+2}\lfloor n^{d/(2d+8\beta)}\rfloor\lceil\log(8n^{d/(2d+8\beta)})\rceil/(s_{\min} - 1)$, filter lengths $2 \leq s_{\min} \leq s_{\max} \leq 9 \times 38^2(\lfloor\beta\rfloor + 1)^4 d^{2\lfloor\beta\rfloor+2}$ and size $\mathcal{S} \leq 8\mathcal{W}L \leq 42 * 8 * 9^2 * 38^4(\lfloor\beta\rfloor + 1)^{10} d^{4\lfloor\beta\rfloor+4}\lfloor n^{d/(2d+8\beta)}\rfloor\lceil\log(8n^{d/(2d+8\beta)})\rceil/(s_{\min} - 1)$, $\mathcal{B} \geq B_0$ and $\delta = \exp\{-n^{d/(d+4\beta)}\}$. By plunging in $\phi_B = (\mathcal{B}+1)^2$ and above values, we have with probability at least $1 - \exp\{-n^{d/(d+4\beta)}\}$,

$$\begin{aligned}
R(\hat{f}_n) - R(f_0) &\leq (\log n)^2 n^{-\beta/(3d/8+2\beta)} \\
&\times \Big\{c(\beta + 1)^8(\mathcal{B} + 1)^2 d^{3\lfloor\beta\rfloor+3}/(s_{\min} - 1) + 18^2\mathcal{B}^2(\lfloor\beta\rfloor + 1)^4 d^{2\lfloor\beta\rfloor+(\beta\vee 1)}\Big\},
\end{aligned}$$

or simply

$$R(\hat{f}_n) - R(f_0) \leq C(d, \beta, \mathcal{B}, s_{\min})(\log n)^2 n^{-\beta/(3d/8+2\beta)},$$

where $C(d, \beta, \mathcal{B}, s_{\min}) = O(\mathcal{B}^2(\lfloor\beta\rfloor + 1)^8 d^{3\lfloor\beta\rfloor+3}/(s_{\min} - 1))$ is a constant independent of n. For the induced classifier $\hat{h}_n = \text{sign}(\hat{f}_n)$, the excess misclassification error satisfies

$$R(\hat{h}_n) - R(h_0) \leq \sqrt{C(d, \beta, B_0, s_{\min})}(\log n) n^{-4\beta/(3d+16\beta)}.$$

The excess risk bound under the approximate low-dimensional manifold assumption can be obtained in a similar way.

C.5 Excess misclassification errors

In this subsection, we summarize the misclassification error bounds obtained in existing results and compare them with ours.

When the hypothesis space is taken to be the class of measurable functions itself, for the excess misclassification error with the 0-1 loss, the convergence rates of order $O(n^{-1/2})$ can be attained using oracle inequalities [41, 59]. Under Tsybakov noise condition of exponent θ and α-Hölder smooth condition on the decision boundaries, [56] proved that the minimax lower bound of the excess misclassification rate is $O(n^{-\alpha(\theta+1)/\{\alpha(\theta+2)+(d-1)\theta\}})$ over all measurable classifiers. With large enough α and θ, the convergence rate can be close to $O(n^{-1})$ arbitrarily.

For classifications using deep neural networks with the SVM hinge loss function, [30] showed that the minimax optimal convergence rate is $O(n^{-\alpha(\theta+1)/\{\alpha(\theta+2)+(d-1)\theta\}})$ under the Tsybakov noise condition of exponent θ and α-Hölder smooth condition on the decision boundaries and $O(n^{-\alpha(\theta+1)/\{\alpha(\theta+2)+d\}})$ under the Tsybakov noise condition of exponent θ and α-Hölder smooth condition on the conditional class probability $\eta(x) = P(Y = 1|X = x)$.

For classifications using deep convolutional networks with the p-norm hinge loss function $\phi(u) = \max\{1 - u, 0\}^p$ with $p \geq 1$ (it is hinge loss when $p = 1$), [18] derived the approximation error rate and the excess misclassification rate for a target function in the Sobolev space $W^{\beta,p}(\mathbb{S}^{d-1})$ with input data supported on the sphere $\mathbb{S}^{d-1}$ in $\mathbb{R}^d$ under the variancing power condition [57]. Two quantities including $\gamma = \max\{1, (d + 3 + \beta)/2(d - 1)\}$ and τ for variancing power condition are involved in the convergence rate. Details can be found in [18]. Based on Table II in [18], we list the excess misclassification rates and compare them in the table below.

Table 5: Excess Misclassification Error

Hypothesis space	Loss	Condition	Rate	Reference
Measurable functions	0-1 loss	θ-noise condition; α-Hölder decision boundary	$O(n^{-\frac{\beta(\theta+1)}{\beta(\theta+2)+(d-1)\theta}})$	Theorem 1 in [56]
DNN	Hinge		$O(n^{-\frac{\beta(\theta+1)}{\beta(\theta+2)+(d-1)(\theta+1)}})$	Theorem 1 in [30]
Deep CNNs	1-norm	$f_0 \in W^{\beta,p}(\mathbb{S}^{d-1})$	$O(n^{-\frac{\beta}{\beta(2-\tau)+2\gamma(d-1)}})$	Theorem 2 in [18]
	p-norm		$O(n^{-\frac{p\beta}{2p\beta(2-\tau)+2p(\gamma+1)(d-1)}})$	
	2-norm	$f_0 \in W^{\beta,p}(\mathbb{S}^{d-1})$; θ-noise condition;	$O(n^{-\frac{2\beta\theta}{(2+\theta)((\gamma+1)(d-1)+2\beta)}})$	Theorem 3 in [18]
	Hinge	θ-noise condition; $f_0 \in W^{\beta,p}([0,1]^d)$	$O(n^{-\frac{\beta(\theta+1)}{d+2\beta(\theta+1)}})$	This paper
	Logistic	$f_0 \in W^{\beta,p}([0,1]^d)$	$O(n^{-\frac{\beta}{2d+4\beta}})$	
	Exponential	$f_0 \in W^{\beta,p}([0,1]^d)$	$O(n^{-\frac{\beta}{2d+4\beta}})$	
	Least square	$f_0 \in W^{\beta,p}([0,1]^d)$	$O(n^{-\frac{4\beta}{3d+16\beta}})$	

D A toy example for CNN approximation

In this subsection, we use a toy example to illustrate the approximation power of CNNs. We examine how the approximation error varies according to the filter size and depth of the CNNs.

We consider a target function f_0 defined as

$$f_0(x) = 2\sin(2\pi x_1) + 4(x_2)^3, \ x \in [0,1]^2$$

where $x = (x_1, x_2) \in [0,1]^2$ is the two-dimensional input. The 2D heatmap and 3D surface visualizations for the target function f_0 are presented in Figure 1.

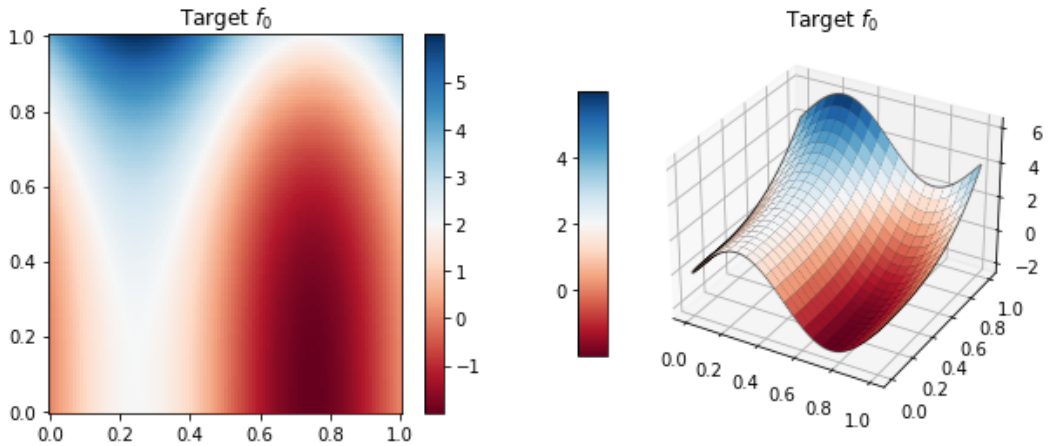

Figure 1: Heatmap and 3D surface plots for the target function f_0 defined on $[0,1]^2$.

To construct CNNs that approximates f_0, we train the CNNs on noiseless data, which is generated by taking values of the target function in $[0,1]^2$. Specifically, we choose 100×100 lattice points $\{X_i\}_{i=1}^{10,000}$ uniformly on the square $[0,1]^2$ and calculate the corresponding target function values $\{Y_i = f_0(X_i)\}_{i=1}^{10,000}$ to get the training data $\{X_i, Y_i\}_{i=1}^{10,000}$ for function approximation.

We implement the training in Python via *Pytorch* and use *Adam* [31] as the optimization algorithm with default learning rate 0.01 and default $\beta = (0.9, 0.99)$ (coefficients used for computing running averages of gradients and their squares). During the training process, we set the batch size as $2,500$, i.e. the $1/4$ of the sample size. The maximum epoch of training is set as 1,000 and an early stopping rule is applied where we stop the iteration if the training losses for the last 200 consecutive epochs do not achieve a new minimum.

We investigate the approximation power of CNNs with different shapes, which are specified through the filter length and the depth. For ease of demonstration, we consider CNNs with L hidden layers with each hidden layer being a convolutional layer with a uniform filter length s. We set the last layer as a fully connected layer with one-dimensional scalar output.

We train 12 CNNs that approximate the target function for $L = 1, 2, 3$ and $s = 20, 50, 100, 200$, then we calculate the empirical L_1 and L_2 distances measuring the approximation error according to

$$L_1(\hat{f}, f_0) = \frac{1}{10,000} \sum_{i=1}^{10,000} |\hat{f}(X_i) - f_0(X_i)|,$$

$$L_2(\hat{f}, f_0) = \sqrt{\frac{1}{10,000} \sum_{i=1}^{10,000} |\hat{f}(X_i) - f_0(X_i)|^2},$$

where $\hat{f}$ denotes the approximate function by CNN. Due to the randomness of the optimization algorithms, we train each CNN for several times and take the best one as the final approximator on the target function.

The summary statistics of empirical L_1 and L_2 distances between the CNNs to the target function on the training data are presented in Table 6. A visualization of CNN approximations on the target function by heatmaps are presented in Figure 2.

Table 6: Approximation errors by CNNs with different filter lengths and depths.

Approximation error $L_1(L_2)$		Filter length 20	50	100	200
Hidden layers	1	0.807(0.969)	0.450(0.539)	0.139(0.186)	0.062(0.084)
	2	0.112(0.144)	0.055(0.070)	0.047(0.064)	0.025(0.037)
	3	0.078(0.098)	0.051(0.070)	0.037(0.046)	0.032(0.045)

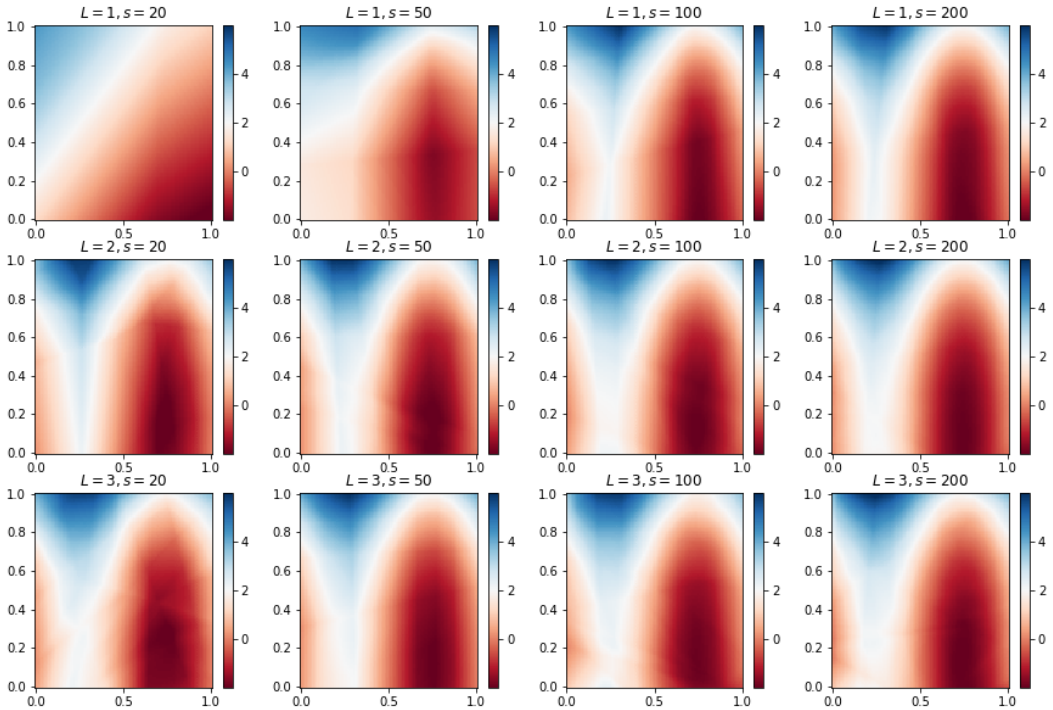

Figure 2: Heatmaps for the CNN approximations on the target function. The CNNs are designated with depth $L = 1, 2, 3$ and filter length $s = 20, 50, 100, 200$.

E Proofs

In this section, we prove Theorems 2.1,2.2, 2.4, 4.4, 4.6, 4.7, as well as the non-asymptotic error bound results for the cross entropy, the SVM, the logistic, the exponential and the least squares losses.

E.1 Proof of Theorem 2.1

As in [22], this approximation result in term of $\|\cdot\|_{W^{\beta-1,p}}$ norm is proved for target function $f \in W^{\beta,p}(\mathcal{X})$, which can be seen as an generalization of Theorem 1 in [60] where L^∞ error bounds for $f \in W^{\beta,\infty}(\mathcal{X})$ is derived. Besides, our results improves the prefactor in d of the network width in Theorem 4.1 in [22]. The main idea of our proof is to approximate the averaged Taylor expansion of function f in Sobolev functions. The definition and properties of averaged Taylor expansion can be found in subsection E.13 of this supplementary material. By Lemma E.6, let B be a ball with proper radius in Ω such that Ω is star-shaped with respect to B, for $1 \le p \le \infty$ we have

$$\|f - T^\beta f\|_{W^{m,p}(\Omega)} \le C_{m,\beta,\gamma} r_\Omega^{\beta-m} \|f\|_{W^{m,p}(\Omega)}, \qquad k = 0, 1, \ldots, \beta$$

where $C_{k,\beta,\gamma} > 0$ is a constant, r_Ω is the diameter of $\Omega \subseteq \mathcal{X}$, γ is the chunkiness parameter of Ω defined in definition E.4 and $T^\beta f$ is the averaged Taylor expansion over B defined as in (33):

$$T^\beta f(x) = \int_B T_y^\beta f(x)\phi(y)dy,$$

where

$$T_y^\beta f(x) = \sum_{\|\alpha\|_1 \le \beta} \frac{1}{\alpha!} D^\alpha f(y)(x-y)^\alpha,$$

and ϕ is an arbitrary cut-off function supported in $\bar{B}$ being infinitely differentiable.

This reminder term could be well controlled when the approximation to Taylor expansion in implemented in a fairly small local region. Then we can focus on the approximation of the Taylor expansion locally. The proof is divided into several parts:

- Partition $\mathcal{X}$ into small cubes $\cup_\theta Q_\theta$, and construct a network ψ that approximately maps each $x \in Q_\theta$ to a fixed point $x_\theta \in Q_\theta$. Hence, ψ approximately discretize $\mathcal{X}$.
- For any multi-index α, construct a network ϕ_α that approximates the coefficient of the averaged Taylor expansion over a ball B in Q_θ, that is $x \in Q_\theta \mapsto c_\alpha(x_\theta)$, where $Q_\theta = \sum_{\alpha \le \beta} c_\alpha(x_\theta)x^\alpha$ by Lemma E.2. Once $\mathcal{X}$ is discretized, the approximation is reduced to a data fitting problem.
- Construct a network $P_\alpha(x)$ to approximate the polynomial $x^\alpha := x_1^{\alpha_1} \dots x_d^{\alpha_d}$ where $x = (x_1, \dots, x_d)^\top \in \mathbb{R}^d$ and $\alpha = (\alpha_1, \dots, \alpha_d)^\top \in \mathbb{N}_0^d$. In particular, we can construct a network $\phi_\times(\cdot,\cdot)$ approximating the product function of two scalar inputs.
- Lastly, our construction of neural network can be written in the form,

$$\phi(x) = \sum_{\|\alpha\|_1 \le \beta} \phi_\times\Big(\phi_\alpha(x), P_\alpha(x)\Big).$$

Proof. Before proving the theorem, we firstly show that each ReLU activated multilayer-perceptron can be computed by a downsampled CNN f_{CNN} defined as in (23). The proof here follows Theorem 2 in [64] where we construct downsampled CNN exactly computing a fully-connected layer. Due to the simplicity of the formulation of ReLU activated multilayer-perceptron, we focus on showing that each fully-connected layer can be reproduced by a down sampled CNN with proper choice of filters. Without loss of generality, let

$$W = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1d_{in}} \\ w_{21} & w_{22} & \cdots & w_{2d_{in}} \\ \vdots & \cdots & \cdots & \vdots \\ w_{d_{out}1} & w_{d_{out}2} & \cdots & w_{d_{out}d_{in}} \end{bmatrix},$$

be the weight matrix and $b = (b_1, \dots, b_{d_{out}})^\top \in \mathbb{R}^{d_{out}}$ be the bias vector of a fully-connect layer $A_F(\cdot) : \mathbb{R}^{d_{in}} \to \mathbb{R}^{d_{out}}$, where $w_{ij}, i = 1, \dots, d_{out}; j = 1, \dots, d_{in}$ are weights. Then the output of the fully-connect layer with respect to $x \in \mathbb{R}^{d_{in}}$ is

$$A_F(x) = \sigma(Wx + b).$$

To construct a downsampled CNN to represent the fully-connected layer, we define a sequence $w = (w_0, \dots, w_{d_{in} \times d_{out}}) \in \mathbb{R}^{d_{in} \times d_{out}}$ with length $d_{in} \times d_{out}$ by stacking the reversed row vectors of the weight matrix W as

$$w_{i+(r-1)d_{in}} = W_{r,d_{in}-i}, \quad r = 1, 2, \dots, d_{out},\ i = 0, 1, \dots, d_{in} - 1,$$

or

$$w = (w_{1d_{in}}, \dots, w_{11}, w_{2d_{in}}, \dots, w_{21}, \dots, w_{d_{out}d_{in}}, \dots, w_{d_{out}1}),$$

where w_{ij} are weights of the matrix W. By Lemma 2 in [64], there exists a sequence of filter $\{w^{(i)}\}_{i=1}^I$ of equal filter length s for some $s \in [2, d_{in} \times d_{out}]$ and $I \le \lceil (d_{in} \times d_{out} - 1)/(s-1) \rceil$ such that w has convolutional factorization $w^{(I)} * w^{(I-1)} * \cdots * w^{(1)}$. We let $\{w^{(i)}\}_{i=1}^I$ be the filters of the I Toeplitz type convolutional matrices $T^{(1)}, \dots, T^{(I)}$ of the convolutional operators

$$A^{(i)}(x) = \sigma(T^{(i)}x + b^{(i)})$$

for $i = 1, \ldots, I$. Next we construct the bias vectors $\{b^{(i)}\}_{i=1}^{I}$ according to Lemma 3 in [64]. Let

$$b^{(1)} = -B\|w^{(1)}\|_1 \mathbb{1}_{d_{in}+s}$$

with B denoting the infinity norm of the input x of the fully-connected layer, and let

$$b^{(i)} = -B(\Pi_{j=1}^{i-1}\|w^{(j)}\|_1)T^{(j)}\mathbb{1}_{d_{in}+s(i-1)} + B(\Pi_{j=1}^{i}\|w^{(j)}\|_1)\mathbb{1}_{d_{in}+is}$$

for $i = 2, \ldots, I-1$. Then we have

$$A^{(I-1)} \circ \cdots \circ A^{(1)}(x) = T^{(i-1)} \cdots T^{(1)}x + B(\Pi_{j=1}^{I-1}\|w^{(j)}\|_1)\mathbb{1}_{d_{in}}.$$

Finally, we set

$$b^{(i)} = \begin{cases} -B(\Pi_{j=1}^{I-1}\|w^{(j)}\|_1)T^{(j)}\mathbb{1}_{d_{in}+s(I-1)} + \theta, & \text{if } I \geq 2, \\ \theta, & \text{if } I = 1, \end{cases}$$

where $\theta \in \mathbb{R}^{d_{in}+s(I-1)}$ s an arbitrary vector satisfying $D_I(\theta) = b$ with $D_I : \mathbb{R}^{d_{in}+sI} \to \mathbb{R}^{d_{out}}$ denoting the downsampling operator and b is the bias vector of the target fully-connected layer. Then we can defined the downsampled CNN by

$$f_{CNN} = D_I \circ A^{(I)} \circ \cdots \circ A^{(1)},$$

and by Lemma 1 in [64] we have

$$f_{CNN}(x) = D_I(\sigma(T^{(I)} \cdots T^{(1)}x + \theta)) = \sigma(Wx + b).$$

And the the total number of free parameters in computing the fully-connected layer A_F is no more than

$$(3s+2)\lceil \frac{(d_{in} \times d_{out} - 1)}{(s-1)} \rceil \leq 8d_{in} \times d_{out},$$

which is 8 times of the total parameters of the fully-connected layer. The depth of f_{CNN} is $I + 1 \leq \lceil (d_{in} \times d_{out} - 1)/(s-1) \rceil + 1$. If we take $s = d_{in} \times d_{out}$, then f_{CNN} is a 2-layer convolutional network. By induction, we know that for any MLP with depth L, width $(d_0, d_1, \ldots, d_{L+1})$ and size $\mathcal{S}$, there exists a downsampled CNN as defined in (23) that exactly computes the MLP. The downsampled CNN has size no more than $8\mathcal{S}$, depth no more than $L + \sum_{i=0}^{L} \lceil d_i \times d_{i+1}/(s^{(i)} - 1) \rceil$ where $s^{(i)}$ is the filter length of the $2i$-th layer of downsampled CNN and the $2i-1$-th layers are downsampling operations. More specifically, if take the maximum possible filter size $s^{(i)} = d_i \times d_{i+1}$ for each construction of the fully-connected layer above, the downsampled CNN that computed the MLP has depth no more than $2L$, size no more than $8\mathcal{S}$ and maximum filter length no more than $\mathcal{W} = \max\{d_0 d_1, \ldots, d_L d_{L+1}\}$. In general, the maximum filter length $s = \mathcal{O}(1)$, and the constructed downsampled CNN has size no more than $8\mathcal{S}$, depth no more than $L + \sum_{i=0}^{L} \lceil d_i \times d_{i+1}/(s^{(i)} - 1) \rceil \leq L(1 + \mathcal{W}^2/(s_{\min} - 1)) \leq 2L\mathcal{W}^2/(s_{\min} - 1)$ where $s_{\min}$ is the minimum of the filter lengths over convolutional layers.

Now we prove the theorem. Without loss of generality, we assume the Sobolev norm of f is 1, i.e. $\|f\|_{W^{\beta,p}(\mathcal{X})} \leq 1$. The reason is that we can always approximate f/B_0 firstly by a network ϕ with approximation error ϵ, then the scaled network $B_0\phi$ will approximate f with error no more than ϵB_0. Besides, it is a trivial case when the Sobolev norm of f is 0. Firstly, when $\beta > 1$, we divide the proof into three steps as follows.

Step 1: Discretization.
Given $N, M \in \mathbb{N}^+$, set $K = \lfloor N^{1/d} \rfloor^2 \lfloor M^{2/d} \rfloor$ and $\delta \in (0, 1/(3K)]$, for each $\theta = (\theta_1, \ldots, \theta_d) \in \{0, 1, \ldots, K-1\}^d$, we define

$$Q_\theta := \Big\{ x = (x_1, \ldots, x_d) : x_i \in [\frac{\theta_i}{K}, \frac{\theta_i + 1}{K} - \delta \cdot 1_{\theta_i < K-1}], i = 1, \ldots, d \Big\}.$$

Let $\Omega(\mathcal{X}, K, \delta) = \mathcal{X} \setminus \cup_\theta Q_\theta$ be the trifling region or the complement of $\cup_\theta Q_\theta$. By the definition of Q_θ, the region $\mathcal{X}$ is approximately divided into hypercubes. By Lemma E.7, there exists a ReLU network ψ_1 with width $4\lfloor N^{1/d} \rfloor + 3$ and depth $4M + 5$ such that

$$\psi_1(x) = \frac{k}{K}, \quad \text{if } x \in [\frac{k}{K}, \frac{k+1}{K} - \delta \cdot 1_{\{k < K-1\}}], k = 0, 1, \ldots, K-1,$$

and the Lipschitz constant of ψ_1 is no more than $2MK^{-2}\delta^{-2}$. We define

$$\psi(x) := (\psi_1(x_1), \dots, \psi_1(x_d)), \quad x = (x_1, \dots, x_d) \in \mathbb{R}^d.$$

Then we have $\psi(x) = \theta/K := (\theta_1/K, \dots, \theta_d/K)^\top$ for $x \in Q_\theta$ and ψ is a ReLU network with width $d(4\lfloor N^{1/d} \rfloor + 3)$ and depth $4M + 5$.

Step 2: Local Approximation of averaged Taylor expansion.
We consider the local approximation of $f \in W^{\beta,p}(\mathcal{X})$ on the subset Q_θ for each θ by its averaged Taylor expansion $T^\beta f(x) = \int_{B_\theta} T_y^\beta f(x)\phi(y)dy$ over the ball B_θ where $T_y^\beta f(x) = \sum_{\|\alpha\|_1 \le \beta} \frac{1}{\alpha!} D^\alpha f(y)(x-y)^\alpha$, the ball $B_\theta = \{y \in \mathcal{X} : \|y - (\theta + (1-\delta)/2)/K\|_2 \le (1-\delta)/(2K)\}$ support in Q_θ, and ϕ is an cut-off function supported in $\bar{B}$ defined in Definition E.1. Note that Q_θ has diameter $r_{Q_\theta} = \sqrt{d}(1-\delta)/K$ and it is star-shaped with respect to the ball B_θ with radius $(1-\delta)/(2K)$. By Lemma E.6, we have error bound for the remainder of the averaged Taylor expansion,

$$\|f - T^\beta f\|_{W^{m,p}(Q_\theta)} \le C_{m,\beta,\gamma} \Big(\frac{\sqrt{d}(1-\delta)}{K} \Big)^{\beta - m} \|f\|_{W^{\beta,p}(Q_\theta)} \qquad m = 0, 1, \dots, \beta.$$

Besides, by Lemma E.2, the averaged Taylor expansion is actually a polynomial $T^\beta f(x) = \sum_{\|\alpha\|_1 \le \beta} c_\alpha^\theta x^\alpha$ for $x \in Q_\theta$ with $|c_\alpha^\theta| \le C(\beta, d)$ for all α where

$$C(\beta, d) = (\beta+1)2^{\beta + d/p} d^{3\beta/2} (\pi^{-d/2}\Gamma(d/2+1))^{1/p} (NM)^{2/p - 2\beta/d} \|f\|_{W^{\beta,p}(\Omega)}.$$

Next we seek to approximate the polynomial $T^\beta f(x) = \sum_{\|\alpha\|_1 \le \beta} c_\alpha^\theta x^\alpha$ locally by deep neural networks. We firstly approximate the coefficients of averaged Taylor expansion. Since $\theta \in \{0, 1, \dots, K-1\}^d$ is one-to-one correspondence to $i_\theta := \sum_{j=1}^d \theta_j K^{j-1} \in \{0, 1 \dots, K^d - 1\}$, we define

$$\psi_0(x) := (K, K^2, \dots, K^d) \cdot \psi(x) = \sum_{j=1}^d \psi_1(x_j) K^j, \quad x \in \mathbb{R}^d,$$

then

$$\psi_0(x) = \sum_{j=1}^d \theta_j K^{j-1} = i_\theta, \quad \text{if } x \in Q_\theta, \quad \theta \in \{0, 1, \dots, K-1\}^d,$$

where $\psi_0(x)$ has width $d(4\lfloor N^{1/d} \rfloor + 3)$ and depth $4M + 5$. For any $\alpha \in \mathbb{N}_0^d$ satisfying $\|\alpha\|_1 \le \beta$ and each $i = i_\theta \in \{0, 1, \dots, K^d - 1\}$, we denote $\xi_{\alpha,i} := (c_\alpha^\theta + C(\beta, d))/(2C(\beta, d)) \in [0, 1]$. Since $K^d \le N^2 M^2$, by Lemma E.8, there exists a ReLU network φ_α with width $16(\beta+2)(N+1)\lceil \log_2(8N) \rceil$ and depth $5(M+2)\lceil \log_2(4M) \rceil$ such that

$$|\varphi_\alpha(i) - \xi_{\alpha,i}| \le (NM)^{-2\beta - 4},$$

for all $i \in \{0, 1, \dots, K^d - 1\}$. We define

$$\phi_\alpha(x) := 2C(\beta, d)\varphi_\alpha(\psi_0(x)) - C(\beta, d) \in [-C(\beta, d), C(\beta, d)], \quad x \in \mathbb{R}^d.$$

Then ϕ_α can be implemented by a network with width $16d(\beta+2)(N+1)\lceil \log_2(8N) \rceil \le 32d(\beta+2)N\lceil \log_2(8N) \rceil$ and depth $5(M+2)\lceil \log_2(4M) \rceil + 4M + 5 \le 15M\lceil \log_2(8M) \rceil$. And we have for any $\theta \in \{0, 1, \dots, K-1\}^d$, if $x \in Q_\theta$,

$$|\phi_\alpha(x) - c_\alpha^\theta| = 2C(\beta, d)|\varphi_\alpha(i_\theta) - \xi_{\alpha,i_\theta}| \le 2C(\beta, d)(NM)^{-2\beta-4}. \tag{24}$$

Step 3: Local Approximation of the polynomial.

Let $\varphi(t) = \min\{\max\{t, 0\}, 1\} = \sigma(t) - \sigma(t-1)$ for $t \in \mathbb{R}$ where $\sigma(\cdot)$ is the ReLU activation function. With a slightly abuse of the notation, we extend its definition to $\mathbb{R}^d$ coordinate-wisely, i.e., for $x = (x_1, \dots, x_d)^\top \in \mathcal{X} \subseteq \mathbb{R}^d$, we let $\varphi(x) = \varphi(x_1, \dots, x_d) := (\varphi(x_1), \dots, \varphi(x_d))^\top$ denote the extension of ϕ to $\mathbb{R}^d$. By Lemma E.9, there exists a ReLU network with width $9N + 1$ and depth $4\beta M + 8$ such that for any $t_1, t_2 \in [a, b]$,

$$\|t_1 t_2 - \phi_\times(t_1, t_2)\|_{W^{1,\infty}([a,b]^2)} \le 6(b-a)^2 N^{-2\beta M - 4}, \tag{25}$$

with $\|\phi_\times\|_{W^{1,\infty}((a,b)^2)} \le 12(b-a)^2$. By Lemma E.10, for any $\alpha \in \mathbb{N}_0^d$ with $\|\alpha\|_1 \le \beta \in \mathbb{N}^+$, there exists a ReLU network P_α with width $9(N+1)+\beta$ and depth $7(\beta+1)^2M$ and $\|P_\alpha\|_{W^{1,\infty}(\mathcal{X})} \le 18$ such that for any $x \in \mathcal{X}$

$$\|P_\alpha(x) - x^\alpha\|_{W^{1,\infty}(\mathcal{X})} \le 10(\beta+1)(N+1)^{-7(\beta+1)M}. \tag{26}$$

Step 4: Approximation of f on $\cup_{\theta\in\{0,1,\dots,K-1\}^d} Q_\theta$.

For any $x \in Q_\theta$, $\theta \in \{0,1,\dots,K-1\}^d$, we can now approximate the averaged Taylor expansion of $f(x)$ by combined sub-networks. Motivated by this, we define

$$\phi_0(x) := \sum_{\|\alpha\|_1\le\beta} \phi_\times\Big(\phi_\alpha(x), P_\alpha(\varphi(x))\Big) \in [-d^\beta C(\beta,d), d^\beta C(\beta,d)],$$

where $\|\phi_0\|_{W^{1,\infty}(\mathcal{X})} \le 48C(\beta,d)^2(\beta+1)d^\beta$. Then ϕ_0 is our constructed neural network approximating the target function f on $\mathcal{X}$. Recall that width and depth of φ is $(2d,1)$, width and depth of ψ is $(d(4\lfloor N^{1/d}\rfloor+3), 4M+5)$, width and depth of P_α is $(9(N+1)+\beta, 7(\beta+1)^2M)$, width and depth of ϕ_α is width $(16d(\beta+2)(N+1)\lceil\log_2(8N)\rceil$,$5(M+2)\lceil\log_2(4M)\rceil+4M+5)$ and width and depth of $\phi_\times$ is $(9N+1, 2\beta M+4)$. Hence, by our construction, ϕ_0 can be implemented by a neural network with width $38(\beta+1)^2d^{\beta+1}N\lceil\log_2(8N)\rceil$ and depth $21(\beta+1)^2M\lceil\log_2(8M)\rceil$.

For $\theta \in \{0,1,\dots,K-1\}^d$, thanks to Lemma E.6, we have error bound for the remainder of the averaged Taylor expansion,

$$\|f - T^\beta f\|_{W^{m,p}(Q_\theta)} \le C_{\beta,d,r}\Big(\frac{\sqrt{d}(1-\delta)}{K}\Big)^{\beta-m}\|f\|_{W^{\beta,p}(Q_\theta)},$$

where $C_{\beta,d,r} = 3\beta d^{\beta-1}\Gamma(d/2+1)\pi^{-d/2}(1+2\sqrt{d})^d$. Then for any $\theta \in \{0,1,\dots,K-1\}^d$, the approximation error of ϕ_0 on f over Q^θ can be upper bounded,

$$\begin{aligned}\|\phi_0 - f\|_{W^{1,p}(Q_\theta)} &\le \|\phi_0 - T^\beta f\|_{W^{1,p}(Q_\theta)} + \|T^\beta f - f\|_{W^{1,p}(Q_\theta)}\\ &\le \|\sum_{\|\alpha\|_1\le\beta}\phi_\times\Big(\phi_\alpha(x), P_\alpha(\varphi(x))\Big) - c_\alpha^\theta x^\alpha\|_{W^{1,\infty}(Q^\theta)}\\ &\qquad + C_{m,\beta,\gamma}\Big(\frac{\sqrt{d}(1-\delta)}{K}\Big)^{\beta-m}\|f\|_{W^{\beta,p}(Q_\theta)}.\end{aligned} \tag{27}$$

We denote $\mathcal{E}_\alpha = \|c_\alpha^\theta x^\alpha - \phi_\times\big(\phi_\alpha(x), P_\alpha(\varphi(x))\big)\|_{W^{1,\infty}(Q_\theta)}$ for each $\alpha \in \mathbb{N}_0^d$ with $\|\alpha\|_1 \le \beta$. Using the inequality $|t_1t_2 - \phi_\times(t_3,t_4)| \le |t_1t_2 - t_3t_2| + |t_3t_2 - t_3t_4| + |t_3t_4 - \phi_\times(t_3,t_4)| \le |t_2||t_1-t_3| + |t_3||t_2-t_4| + |t_3t_4 - \phi_\times(t_3,t_4)|$ for any $t_1,t_2,t_3,t_4 \in \mathbb{R}$, and by (24), (25) and (26), for $\|\alpha\|_1 \le \beta$ we have

$$\begin{aligned}\mathcal{E}_\alpha \le& |x^\alpha|\|c_\alpha^\theta - \phi_\alpha(x)\|_{W^{1,\infty}(Q_\theta)} + |\phi_\alpha(x)|\|x^\alpha - P_\alpha(\varphi(x))\|_{W^{1,\infty}(Q_\theta)}\\ &+ \Big\|\phi_\alpha(x)P_\alpha(\varphi(x)) - \phi_\times\big(\phi_\alpha(x), P_\alpha(\varphi(x))\big)\Big\|_{W^{1,\infty}(Q_\theta)}\\ \le& 2C(\beta,d)(NM)^{-2\beta-4} + 10C(\beta,d)(\beta+1)(N+1)^{-7\beta M-7M} + 24C(\beta,d)^2N^{-2\beta M-4}\\ \le& (12C(\beta,d)(\beta+1) + 24C(\beta,d)^2)(NM)^{-2\beta-4}\\ \le& 36(\beta+1)^2 2^{2\beta+2d/p} d^{3\beta}(\pi^{-d/2}\Gamma(d/2+1))^{2/p}B_0\|f\|_{W^{\beta,p}(\Omega)}(NM)^{-2\beta-4\beta/d},\end{aligned} \tag{28}$$

where the last inequality holds since $p \ge 1$ and $\|f\|_{W^{\beta,p}(\Omega)} \le B_0$. Besides, observe that the number of terms in the summation can be bounded by

$$\sum_{\alpha\in\mathbb{N}_0^d, \|\alpha\|_1\le\beta} 1 = \sum_{j=0}^{\beta}\sum_{\alpha\in\mathbb{N}_0^d, \|\alpha\|_1=j} 1 \le \sum_{j=0}^{\beta} d^\beta \le (\beta+1)d^\beta. \tag{29}$$

Recall that $K = \lfloor N^{1/d} \rfloor^2 \lfloor M^{2/d} \rfloor$, combining (27), (28) and (29) and $\theta \in \{0, 1, \ldots, K-1\}^d$, we have

$$\begin{aligned}
\|\phi_0 - f\|_{W^{1,p}(Q_\theta)} &\leq \|\phi_0 - T^\beta f\|_{W^{1,p}(Q_\theta)} + \|T^\beta f - f\|_{W^{1,p}(Q_\theta)} \\
&\leq 36 \cdot 2^{2\beta+2d/p}(\beta+1)^3 d^{4\beta}(\pi^{-d/2}\Gamma(d/2+1))^{2/p} B_0 \|f\|_{W^{\beta,p}(\Omega)}(NM)^{-2\beta-4\beta/d} \\
&+ \beta d^{3\beta/2}\Gamma(d/2+1)\pi^{-d/2}(1+1/r)^d \|f\|_{W^{\beta,p}(Q_\theta)}(NM)^{-2(\beta-m)/d} \\
&\leq 37 \cdot 2^{2\beta+2d/p}(1+2\sqrt{d})^d(\beta+1)^3 d^{4\beta}(\pi^{-d/2}\Gamma(d/2+1))^{2/p+1} \\
&\qquad \times B_0 \|f\|_{W^{\beta,p}(Q_\theta)}(NM)^{-2(\beta-m)/d} \qquad (30)
\end{aligned}$$

Recall that $\mathcal{X} = \Omega(\mathcal{X}, K, \delta) \cup (\cup_{\theta \in \{0,1,\ldots,K-1\}^d} Q_\theta)$ and the Lebesgue measure of $\Omega(\mathcal{X}, K, \delta)$ can be upper bounded by $\delta d K$ where $\delta \in (0, 1/(3K)]$ can be arbitrarily small. Then the approximation error $\|f(x) - \phi_0(x)\|_{W^{1,p}(\mathcal{X})}$ can be further bounded as follows.

$$\begin{aligned}
\|f(x) - \phi_0(x)\|_{W^{1,p}(\mathcal{X})} &= \Big(\sum_{\theta \in \{0,1,\ldots,K-1\}^d} \|f - \phi_0\|^p_{W^{1,p}(Q_\theta)} + \|f - \phi_0\|^p_{W^{1,p}(\Omega(\mathcal{X},K,\delta))} \Big)^{1/p} \\
&\leq 37 \cdot B_0 2^{2\beta+2d/p}(1+2\sqrt{d})^d(\beta+1)^3 d^{4\beta}(\pi^{-d/2}\Gamma(d/2+1))^{2/p+1}(NM)^{-2(\beta-m)/d} \\
&\qquad \times \Big(\sum_{\theta \in \{0,1,\ldots,K-1\}^d} \|f\|^p_{W^{\beta,p}(Q_\theta)} \Big)^{1/p} \\
&\leq 37 \cdot 2^{2\beta+2d/p}(1+2\sqrt{d})^d(\beta+1)^3 d^{4\beta}(\pi^{-d/2}\Gamma(d/2+1))^{2/p+1}(NM)^{-2(\beta-m)/d} \\
&\qquad \times B_0 \|f\|_{W^{\beta,p}(\mathcal{X})} \\
&= 37 \cdot 2^{2\beta+2d/p} B_0^2 (\beta+1)^3 (\pi^{-d/2}\Gamma(d/2+1))^{2/p+1}(1+2\sqrt{d})^d d^{4\beta}(NM)^{-2(\beta-m)/d},
\end{aligned}$$

where $\|f - \phi_0\|^p_{W^{1,p}(\Omega(\mathcal{X},K,\delta))}$ can be arbitrarily small since $\|f\|^p_{W^{1,\infty}(\mathcal{X})}$ and $\|\phi_0\|^p_{W^{1,\infty}(\mathcal{X})}$ are bounded, and $\delta \in (0, 1(3K)]$ can be arbitrarily small.

When $p = \infty$, the target function $f \in W^{\beta,\infty}(\mathcal{X})$ actually is Hölder smooth with order β and constant $B_0 = \|f\|_{W^{\beta,\infty}(\mathcal{X})}$. Then by Lemma A.8 in [48], for any $x, x_0 \in \mathcal{X}$, we have

$$\Big| f(x) - \sum_{\|\alpha\|_1 \leq s} \frac{\partial^\alpha f(x_0)}{\alpha!}(x - x_0)^\alpha \Big| \leq d^s \|x - x_0\|_2^\beta,$$

where $s = \lfloor \beta \rfloor$. This reminder term could be well controlled when the approximation to Taylor expansion in implemented in a fairly small local region. By similar construction of approximators [28], it is proved that there exists a function ϕ_0 implemented by a neural network with width $38(\lfloor \beta \rfloor + 1)^2 d^{\lfloor \beta \rfloor + 1} N \lceil \log_2(8N) \rceil$ and depth $21(\lfloor \beta \rfloor + 1)^2 M \lceil \log_2(8M) \rceil$ such that

$$|f(x) - \phi_0(x)| \leq 18 B_0 (\lfloor \beta \rfloor + 1)^2 d^{\beta + (\beta \vee 1)/2}(NM)^{-2\beta/d},$$

for any $x \in \cup_{\theta \in \{0,1,\ldots,K-1\}^d} Q_\theta$. If the probability measure of X is absolutely continuous with respect to the Lebesgue measure, we have

$$\mathbb{E}|f(X) - \phi_0(X)| \leq 18 B_0 (\lfloor \beta \rfloor + 1)^2 d^{\beta + (\beta \vee 1)/2}(NM)^{-2\beta/d}.$$

As we have shown at the beginning of the proof, the multilayer-perceptron ϕ_0 here can be computed by a downsampled CNN with filter lengths $2 \leq s_{\min} \leq s_{\max} \leq \mathcal{W} = 38^2(\lfloor \beta \rfloor + 1)^4 d^{2\lfloor \beta \rfloor + 2} N^2 \lceil \log_2(8N) \rceil^2$ and depth $L \leq 42(\lfloor \beta \rfloor + 1)^2 M \lceil \log_2(8M) \rceil \mathcal{W}/(s_{\min} - 1)$ and size $\mathcal{S} \leq 8\mathcal{W}L$. We know that there exists a $f_{CNN} \in \mathcal{F}_{\text{CNN}}$ such that the approximation results hold. This completes the proof.

□

E.2 Proof of Theorem 2.2

Proof. Let $K \in \mathbb{N}^+$ and $\tau \in (0, 1/K)$, define a region $\Omega(\mathcal{X}, K, \tau)$ of $\mathcal{X}$ as

$$\Omega(\mathcal{X}, K, \varepsilon) = \cup_{i=1}^d \{x = [x_1, x_2, ..., x_d]^T : x_i \in \cup_{k=1}^{K-1}(k/K - \tau, k/K)\}.$$

By Theorem 3.3 of [28], for any $M, N \in \mathbb{N}^+$, there exists a function $f_* \in \mathcal{F}_{MLP}$ with $\|f_*\|_\infty \leq \mathcal{B}$, depth $21(\lfloor\beta\rfloor+1)^2 M\lceil\log_2(8M)\rceil$ and width $38(\lfloor\beta\rfloor+1)^2 d^{\lfloor\beta\rfloor+1} N\lceil\log_2(8N)\rceil$ such that

$$|f_*(x) - f_{0,T}(x)| \leq 18B_0(\lfloor\beta\rfloor+1)^2 d^{\lfloor\beta\rfloor+(\beta\vee 1)/2}(NM)^{-2\beta/d}$$

for any $x \in \mathcal{X}\backslash\Omega(\mathcal{X}, K, \tau)$ where $K = \lfloor N^{1/d}\rfloor^2\lfloor M^{1/d}\rfloor^2$ and τ is an arbitrary number in $(0, \frac{1}{3K}]$ and $\lfloor a\rfloor$ denotes the largest integer smaller than a. Note that the Lebesgue measure of $\Omega(\mathcal{X}, K, \tau)$ is no more than $dK\tau$ which can be arbitrarily small if τ is arbitrarily small. Since P is absolutely continuous with respect to Lebesgue measure, then we have

$$\mathbb{E}\|f_*(X) - f_{0,T}(X)\|_2 \leq 18B_0(\lfloor\beta\rfloor+1)^2 d^{\lfloor\beta\rfloor+(\beta\vee 1)/2}(NM)^{-2\beta/d}.$$

By the proof of Theorem 2.1, any MLP can be computed exactly by a downsampled CNN. Thus there exists a function f implemented by CNN with its depth no more than $L = 42(\lfloor\beta\rfloor+1)^2 M\lceil\log_2(8M)\rceil\mathcal{W}/(s_{\min}-1)$ where $\mathcal{W} = 38^2(\lfloor\beta\rfloor+1)^4 d^{2\lfloor\beta\rfloor+2} N^2\lceil\log_2(8N)\rceil^2$ and its filter lengths $2 \leq s_{\min} \leq s_{\max} \leq \mathcal{W} = 38^2(\lfloor\beta\rfloor+1)^4 d^{2\lfloor\beta\rfloor+2} N^2\lceil\log_2(8N)\rceil^2$ such that

$$\mathbb{E}\|f(X) - f_{0,T}(X)\|_2 \leq 18B_0(\lfloor\beta\rfloor+1)^2 d^{\lfloor\beta\rfloor+(\beta\vee 1)/2}(NM)^{-2\beta/d}.$$

□

E.3 Proof of Theorem 2.4

Proof. The data can be projected to a low-dimensional space and then use CNN to approximate the low-dimensional function where the idea is similar to that of Theorem 1.2 in [55]. Based on Theorem 3.1 in [3], there exists a linear projector $A \in \mathbb{R}^{d_\delta \times d}$ that maps a low-dimensional manifold in a high-dimensional space to a low-dimensional space nearly preserving the distance. Specifically, there exists a matrix $A \in \mathbb{R}^{d_\delta \times d}$ such that $AA^T = (d/d_\delta)I_{d_\delta}$ where I_{d_δ} is an identity matrix of size $d_\delta \times d_\delta$, and

$$(1-\delta)\|x_1 - x_2\|_2 \leq \|Ax_1 - Ax_2\|_2 \leq (1+\delta)\|x_1 - x_2\|_2,$$

for any $x_1, x_2 \in \mathcal{M}$. And it is easy to check

$$A(\mathcal{M}_\rho) \subseteq A(\mathcal{X}) \subseteq [-\sqrt{\frac{d}{d_\delta}}, \sqrt{\frac{d}{d_\delta}}]^{d_\delta}.$$

Note that for any $z \in A(\mathcal{M})$, there exists a unique $x \in \mathcal{M}$ such that $Ax = z$. To prove this, let $x' \in \mathcal{M}$ be another point on $\mathcal{M}$ satisfying $Ax' = z$, then $(1-\delta)\|x - x'\|_2 \leq \|Ax - Ax'\|_2 \leq (1+\delta)\|x - x'\|_2$ implies that $\|x - x'\|_2 = 0$. Then for any $z \in A(\mathcal{M})$, define $x_z = \mathcal{SL}(\{x \in \mathcal{M} : Ax = z\})$ where $\mathcal{SL}(\cdot)$ is a set function which returns a unique element of a set. Note that if $Ax = z$ where $x \in \mathcal{M}$ and $z \in A(\mathcal{M})$, then $x = x_z$ by our argument since $\{x \in \mathcal{M} : Ax = z\}$ is a set with only one element when $z \in A(\mathcal{M})$. And we can see that $\mathcal{SL} : A(\mathcal{M}) \to \mathcal{M}$ is a differentiable function with the norm of its derivative locates in $[1/(1+\delta), 1/(1-\delta)]$, since

$$\frac{1}{1+\delta}\|z_1 - z_2\|_2 \leq \|x_{z_1} - x_{z_2}\|_2 \leq \frac{1}{1-\delta}\|z_1 - z_2\|_2,$$

for any $z_1, z_2 \in A(\mathcal{M}) \subseteq E$ where $E := [-\sqrt{d/d_\delta}, \sqrt{d/d_\delta}]^{d_\delta}$. For the high-dimensional function $f_0 : [0,1]^d \to \mathbb{R}^1$, we define its low-dimensional representation $\tilde{f}_0 : \mathbb{R}^{d_\delta} \to \mathbb{R}^1$ by

$$\tilde{f}_0(z) = f_0(x_z), \quad \text{for any } z \in A(\mathcal{M}) \subseteq \mathbb{R}^{d_\delta}.$$

Recall that $f_0 \in \mathcal{H}^\beta(\mathcal{X}, B_0)$, then $\tilde{f}_0 \in \mathcal{H}^\beta(A(\mathcal{M}), B_0/(1-\delta)^\beta)$. Note that $\mathcal{M}$ is compact and A is a linear mapping, then by the extended version of Whitney' extension theorem in [17], there exists a function $\tilde{F}_0 \in \mathcal{H}^\beta(E, B_0/(1-\delta)^\beta)$ such that $\tilde{F}_0(z) = \tilde{f}_0(z)$ for any $z \in A(\mathcal{M})$. With $E = [-\sqrt{d/d_\delta}, \sqrt{d/d_\delta}]^{d_\delta}$, by Theorem 2.1, for any $N, M \in \mathbb{N}^+$, there exists a function $\tilde{f}_n : \mathbb{R}^{d_\delta} \to \mathbb{R}^1$ implemented by a downsampled CNN with filter lengths $2 \leq s_{\min} \leq s_{\max} \leq \mathcal{W} = 38^2(\lfloor\beta\rfloor+1)^4 d_\delta^{2\lfloor\beta\rfloor+2} N^2\lceil\log_2(8N)\rceil^2$ and depth $L = 42(\lfloor\beta\rfloor+1)^2 M\lceil\log_2(8M)\rceil\mathcal{W}/(s_{\min}-1)$ such that

$$|\tilde{f}_n(z) - \tilde{F}_0(z)| \leq 18\frac{B_0}{(1-\delta)^\beta}(\lfloor\beta\rfloor+1)^2 d^{1/2} d_\delta^{(3\beta+1)/2}(NM)^{-2\beta/d_\delta},$$

for all $z \in E \backslash \Omega(E)$ where $\Omega(E)$ is a subset of E with an arbitrarily small Lebesgue measure as well as $\Omega := \{x \in \mathcal{M}_\rho : Ax \in \Omega(E)\}$ does.

If we define $f_n^* = \tilde{f}_n \circ A$ which is $f_n^*(x) = \tilde{f}_n(Ax)$ for any $x \in \mathcal{X}$, then f_n^* is also a downsampled CNN with the same parameter as $\tilde{f}_n$. For any $x \in \mathcal{M}_\rho \backslash \Omega$ and $z = Ax$, there exists a $\tilde{x} \in \mathcal{M}$ such that $\|x - \tilde{x}\|_2 \leq \rho$, then

$$
\begin{aligned}
&|f_n^*(x) - f_0(x)| = |\tilde{f}_n(Ax) - \tilde{F}_0(Ax) + \tilde{F}_0(Ax) - \tilde{F}_0(A\tilde{x}) + \tilde{F}_0(A\tilde{x}) - f_0(x)| \\
&\leq |\tilde{f}_n(Ax) - \tilde{F}_0(Ax)| + |\tilde{F}_0(Ax) - \tilde{F}_0(A\tilde{x})| + |\tilde{F}_0(A\tilde{x}) - f_0(x)| \\
&\leq 18 \frac{B_0}{(1-\delta)^\beta} (\lfloor \beta \rfloor + 1)^2 d^{1/2} d_\delta^{(3\beta+1)/2} (NM)^{-2\beta/d_\delta} + \frac{B_0}{1-\delta} \|Ax - A\tilde{x}\|_2 + |f_0(\tilde{x}) - f_0(x)| \\
&\leq 18 \frac{B_0}{(1-\delta)^\beta} (\lfloor \beta \rfloor + 1)^2 d^{1/2} d_\delta^{(3\beta+1)/2} (NM)^{-2\beta/d_\delta} + \frac{\rho B_0}{1-\delta} \sqrt{\frac{d}{d_\delta}} + \rho B_0 \\
&= 18 \frac{B_0}{(1-\delta)^\beta} (\lfloor \beta \rfloor + 1)^2 d^{1/2} d_\delta^{(3\beta+1)/2} (NM)^{-2\beta/d_\delta} + \rho B_0 \{(1-\delta)^{-1} \sqrt{d/d_\delta} + 1\} \\
&\leq (18 + C_2) \frac{B_0}{(1-\delta)^\beta} (\lfloor \beta \rfloor + 1)^2 d^{1/2} d_\delta^{(3\beta+1)/2} (NM)^{-2\beta/d_\delta},
\end{aligned}
$$

where $C_2 > 0$ is a constant not depending on any parameter. The last inequality follows from $\rho \leq C_2 (NM)^{-2\beta/d_\delta} (\lfloor \beta \rfloor + 1)^2 d^{1/2} d_\delta^{(3\beta+1)/2} \{\sqrt{d/d_\delta} + 1 - \delta\}^{-1} (1-\delta)^{1-\beta}$. Since the probability measure ν of X is absolutely continuous with respect to the Lebesgue measure, we have

$$
\mathbb{E}|f_n^*(X) - f_0(X)| \leq (18 + C_2) \frac{B_0}{(1-\delta)^\beta} (\lfloor \beta \rfloor + 1)^2 d^{1/2} d_\delta^{(3\beta+1)/2} (NM)^{-2\beta/d_\delta}, \tag{31}
$$

where $d_\delta = O(d_{\mathcal{M}} \log(d/\delta)/\delta^2)$ is assumed to satisfy $d_\delta \ll d$. This completes the proof of Theorem 2.4. □

E.4 Proof of Theorem 4.2

The proof follows Theorem 11.8 and Corollary 3.19 in [43]. Recall that (X, Y) has joint distribution $\mathbb{P}$, we let $\mathbb{P}_n$ denote the empirical distribution of the sample $S = \{(X_i, Y_i)\}_{i=1}^n$ and let $(\tilde{X}, \tilde{Y}) \sim \mathbb{P}_n$. Note that for any $f \in \mathcal{F}_{\text{CNN}}$ we have $\|f\|_\infty \leq \mathcal{B}$, then $0 \leq \phi(yf(x)) \leq \phi_B$ for any $(x, y) \sim \mathbb{P}$. For any $f \in \mathcal{F}_{\text{CNN}}$ and $t \geq 0$, we denote by $c(f, t)$ the classifier defined by $c(f, t) : (x, y) \mapsto \mathbb{1}(\phi(Yf(X)) > t)$. Correspondingly, we define the risk of $c(f, t)$ by

$$
R(c(f,t)) = \mathbb{P}[c(f,t)(X,Y) = 1] = \mathbb{P}[\phi(Yf(X)) > t],
$$

and similarly define its empirical risk by $R_n(c(f, t)) = \mathbb{P}_n[\phi(\tilde{Y} f(\tilde{X})) > t]$. Then we write

$$
\begin{aligned}
|R(f) - R_n(f)| &= \left| \mathbb{E}_{(X,Y)} \phi(Yf(X)) - \mathbb{E}_{(\tilde{X},\tilde{Y})} \phi(\tilde{Y} f(\tilde{X})) \right| \\
&= \left| \int_0^{\phi_B} \Big(P[\phi(Yf(X)) > t] - P[\phi(\tilde{Y} f(\tilde{X})) > t] \Big) dt \right| \\
&\leq \phi_B \sup_{t \in [0, \phi_B]} \left| P[\phi(Yf(X)) > t] - P[\phi(\tilde{Y} f(\tilde{X})) > t] \right| \\
&= \phi_B \sup_{t \in [0, \phi_B]} \Big| R(c(f,t)) - R_n(c(f,t)) \Big|.
\end{aligned}
$$

And this implies that

$$
P\Big[\sup_{f \in \mathcal{F}_{\text{CNN}}} |R(f) - R_n(f)| > \epsilon \Big] \leq P\left[\sup_{f \in \mathcal{F}_{\text{CNN}}, t \in [0, \phi_B]} |R(c(f,t)) - R_n(c(f,t))| > \frac{\epsilon}{\phi_B} \right].
$$

The right-hand side can be bounded using a standard generalization bound for classification (Corollary 3.19 in [43]) in terms of the VC-dimension of the class of functions $\{c(f, t) : f \in \mathcal{F}_{\text{CNN}}, t \in [0, \phi_B]\}$, which, by definition of the pseudo-dimension, is $\text{Pdim}(\mathcal{F}_{\text{CNN}})$ the pseudo-dimension of function class

$\mathcal{F}_{\text{CNN}} = \{(x, y) \mapsto \phi(yf(x)) : f \in \mathcal{F}_{\text{CNN}}\}$. Now we have with probability at least $1 - \delta$ over the choice of an i.i.d. sample S of size n that

$$\sup_{f \in \mathcal{F}_{\text{CNN}}} |R(f) - R_n(f)| \leq \phi_B \Big(\sqrt{\frac{2\text{Pdim}(\mathcal{F}_{\text{CNN}}) \log(en)}{n}} + \sqrt{\frac{\log(1/\delta)}{2n}} \Big).$$

Next, we bound the pseudo-dimension of $\mathcal{F}_{\text{CNN}}$ by that of $\mathcal{F}_{\text{CNN}}$, and this leads to the final stochastic error bound in terms of the parameters of $\mathcal{F}_{\text{CNN}}$. For a sample $S = \{(X_i, Y_i)\}_{i=1}^n$ with size n, if there exist $t_1, \ldots, t_n \in [0, \phi_B]$ such that for each $(b_1, \ldots, b_n) \in \{0, 1\}^n$ there exists a $f \in \mathcal{F}$ such that $\mathbb{1}(\phi(y_i f(x_i)) > t_i) = b_i$ for $i = 1, \ldots, n$, then since ϕ is convexity and non-increasing, we also have $\mathbb{1}(f(x_i) > \phi^{-1}(t_i)) = 1 - b_i$ if $y_i = 1$ and $\mathbb{1}(f(x_i) > -\phi^{-1}(t_i)) = b_i$ if $y_i = -1$. Thus $\{x_i\}_{i=1}^n$ is shattered by $\mathcal{F}_{\text{CNN}}$ and the threshold values $y_1\phi^{-1}(t_1), \ldots, y_n\phi^{-1}(t_n)$ witness the shattering. This implies $\text{Pdim}(\mathcal{F}_{\text{CNN}}) \leq \text{Pdim}(\mathcal{F}_{\text{CNN}})$. Besides, for piece-wise linear neural networks $\mathcal{F}_{\text{CNN}}$, it is proved in Theorem 6 of [4] that

$$\text{Pdim}(\mathcal{F}_{\text{CNN}}) \leq C \cdot \mathcal{S}L \log \mathcal{S},$$

for some universal constant $C > 0$ where $\mathcal{S}$ and L are the size and depth of $\mathcal{F}_{\text{CNN}}$ respectively. Combine above inequities, we have with probability at least $1 - \delta$ over the choice of an i.i.d. sample S of size n that

$$\sup_{f \in \mathcal{F}_{\text{CNN}}} |R(f) - R_n(f)| \leq \phi_B \Big(\sqrt{\frac{C\mathcal{S}L \log \mathcal{S} \log(n)}{n}} + \sqrt{\frac{\log(1/\delta)}{n}} \Big),$$

for some universal constant $C > 0$.

E.5 Proof of Lemma 4.4

Proof. To deal with the approximation of unbounded target function, we consider truncating the target function f_0 by a constant T, where T may depend on n. Let $f_{0,T}$ be the truncated version of f_0 defined as

$$f_{0,T}(x) = \begin{cases} f_0(x), & \text{if } |f_0(x)| \leq T, \\ T\text{sign}(f_0(x)), & \text{if } |f_0(x)| > T. \end{cases}$$

Then, the approximation error can be decomposed into two terms that are easier to deal with. Denote the error of the loss function ϕ due to truncation by

$$\Delta_\phi(T) := \phi(T) - \inf_{a \in \text{Ran}(f_0)} \phi(a), \tag{32}$$

where $\text{Ran}(f_0)$ is the range of the target function f_0. This error decreases as the threshold T increases, however, a bigger T leads to bigger $\mathcal{B}$ and B_ϕ, thus a bigger stochastic error.

Firstly, we prove that under Assumption 4.3(a) and (b), $f_{0,T}$, the truncated version of the target function f_0, is bounded by T and continuous on $\mathcal{X}$. Recall that for any $a \in \mathbb{R}$ and $\eta \in [0, 1]$, the conditional risk function $H(\eta, a) = \eta\phi(a) + (1 - \eta)\phi(-a)$. Given $\eta \in [0, 1]$, the function $dH(\eta, a)/da = \eta\phi'(a) - (1 - \eta)\phi'(-a)$ is continuous in both η and a. Thus under Assumption 4.3 (b) the solution $a(\eta) = \arg\min_{a \in \bar{\mathbb{R}}} H(\eta, \alpha)$ is continuous with respect to η, and $|a(\eta)| < \infty$ when $\eta \in (0, 1)$. By Assumption 4.3 (a), $\eta(x)$ is continuous on $\{x \in \mathcal{X} : \eta(x) \leq 1 - \delta\}$ for any $\delta \in (0, 1)$, thus $f_0(x) = f_0(\eta(x))$ is continuous and $\eta(x)\phi'(f_0(x)) - (1 - \eta(x))\phi'(-f_0(x)) = 0$ on for any $x \in \mathcal{X}$. And $f_0(x)$ is continuous on $\{x \in \mathcal{X} : |f_0(\eta(x))| \leq T\}$ for any $T \geq 0$. For any $T > 0$, the set $\{x \in \mathcal{X} : |f_0(x)| \leq T\}$ is a compact set. Thus $f_{0,T}$, the truncated version of target function is also continuous on $\mathcal{X}$.

Recall that for any f, the ϕ-risk is defined by $R(f) = \mathbb{E}\phi(Yf(X))$, then for $f_{0,T}$, the truncated version of target function satisfies

$$\begin{aligned} R(f_{0,T}) - R(f_0) &= \mathbb{E}\{\phi(Yf_{0,T}(X)) - \phi(Yf_0(X))\} \\ &= \mathbb{E}\Big[\mathbb{E}\{\phi(Yf_{0,T}(X)) - \phi(Yf_0(X)) \mid X\}\Big] \\ &= \mathbb{E}\Big[\eta(X)\{\phi(f_{0,T}(X)) - \phi(f_0(X))\} + (1 - \eta(X))\{\phi(-f_{0,T}(X)) - \phi(-f_0(X))\}\Big]. \end{aligned}$$

For $x \in \mathcal{X}$, we define $g(x) = \eta(x)\{\phi(f_{0,T}(x)) - \phi(f_0(x))\} + (1 - \eta(x))\{\phi(-f_{0,T}(x)) - \phi(-f_0(x))\}$. Recall that ϕ is non-increasing and we have:

(a) If $-T \le f_0(x) \le T$, then $f_0(x) = f_{0,T}(x)$ and $g(x) = 0$;

(b) If $f_0(x) > T$, then $f_{0,T}(x) = T$ and $\phi(-f_{0,T}(x)) - \phi(-f_0(x)) \le 0$, and

$$\begin{aligned} g(x) =& \eta(x)\{\phi(f_{0,T}(x)) - \phi(f_0(x))\} + (1-\eta(x))\{\phi(-f_{0,T}(x)) - \phi(-f_0(x))\} \\ \le& \eta(x)\{\phi(f_{0,T}(x)) - \phi(f_0(x))\} \\ =& \eta(x)\{\phi(T) - \phi(f_0(x))\} \\ \le& \phi(T) - \phi(f_0(x)); \end{aligned}$$

(c) If $f_0(x) < -T$, then $f_{0,T}(x) = -T$ and $\phi(f_{0,T}(x)) - \phi(f_0(x)) \le 0$, and

$$\begin{aligned} g(x) =& \eta(x)\{\phi(f_{0,T}(x)) - \phi(f_0(x))\} + (1-\eta(x))\{\phi(-f_{0,T}(x)) - \phi(-f_0(x))\} \\ \le& (1-\eta(x))\{\phi(-f_{0,T}(x)) - \phi(-f_0(x))\} \\ =& (1-\eta(x))\{\phi(T) - \phi(-f_0(x))\} \\ \le& \phi(T) - \phi(-f_0(x)). \end{aligned}$$

Thus for any $x \in \mathcal{X}$, we have $g(x) \le \phi(T) - \inf_{a \in \text{Ran}(f_0)} \phi(a)$ and $R(f_{0,T}) - R(f_0) = \mathbb{E}\{g(X)\} \le \phi(T) - \inf_{a \in \text{Ran}(f_0)} \phi(a)$, where $\text{Ran}(f_0)$ is the range of the target function f_0.

$$\inf_{f \in \mathcal{F}_{\text{CNN}}} R(f) - R(f_0) = \inf_{f \in \mathcal{F}_{\text{CNN}}} R(f) - R(f_{0,T}) + R(f_{0,T}) - R(f_0),$$

then left thing to do is to prove

$$\inf_{f \in \mathcal{F}_{\text{CNN}}} R(f) - R(f_{0,T}) \le B_\phi \inf_{f \in \mathcal{F}_{\text{CNN}}} \mathbb{E}|f(X) - f_{0,T}(X)|.$$

With $f_0 \in \mathcal{H}^\beta([0,1]^d, B_0)$, ϕ is $\mathcal{B}_\phi$-Lipschitz, and it is a straightforward proof that

$$\begin{aligned} \inf_{f \in \mathcal{F}_{\text{CNN}}} R(f) - R(f_{0,T}) =& \inf_{f \in \mathcal{F}_{\text{CNN}}} \mathbb{E}\{\phi(Yf(X)) - \phi(Yf_{0,T}(X))\} \\ \le& \inf_{f \in \mathcal{F}_{\text{CNN}}} \mathbb{E}\{\mathcal{B}_\phi |Yf(X) - Yf_{0,T}(X)|\} \\ =& B_\phi \inf_{f \in \mathcal{F}_{\text{CNN}}} \mathbb{E}|f(X) - f_{0,T}(X)|. \end{aligned}$$

□

E.6 Proof of Theorem 4.5

The main focus of the proof is to deal with the truncation. If $f_0 \in \mathcal{H}^\beta([0,1]^d, B_0)$ is uniformly bounded by $T \le \mathcal{B}$, by Theorem 2.2, there exists a ReLU neural network ϕ such that

$$\mathbb{E}|\phi(X) - f_0(X)| \le 18B_0(\beta+1)^2 d^{\beta+(\beta\vee 1)/2}(NM)^{-2\beta/d}.$$

We let $\phi_0(x) = \sigma(\phi(x) + \mathcal{B}) - \sigma(\phi(x) - \mathcal{B}) - \mathcal{B}$ be the truncated network where σ denotes the ReLU activation function. Then

$$\mathbb{E}|\phi_0(X) - f_{0,T}(X)| \le \mathbb{E}|\phi(X) - f_0(X)| \le 18B_0(\beta+1)^2 d^{\beta+(\beta\vee 1)/2}(NM)^{-2\beta/d}.$$

If $f_0 \in \mathcal{H}^\beta([0,1]^d, B_0)$ is not uniformly bounded by by T, then the set $\{x : |f_0(x)| \le T\}$ does not equal to the whole domain $\mathcal{X}$. Actually $\{x : |f_0(x)| \le T\}$ can be written as a union of connected subsets since f_0 is Hölder smooth on its domain $\mathcal{X} = [0,1]^d$, and on interior of each connected subset, the function f_0 keeps its Hölder smoothness β. By [17], the truncated target function $f_{0,T}$ on $\{x \in \mathcal{X} : |f_0(x)| \le T\}$ can be extended to $\mathcal{X}$ to be β-Hölder and bound $T + c$ for some small constant $c > 0$. Let $\tilde{f}_{0,T}$ be the extended function on $\mathcal{X}$, then by Theorem 2.2, there exists a ReLU neural network ϕ such that

$$\mathbb{E}|\phi(X) - \tilde{f}_{0,T}(X)| \le 18B_0(\beta+1)^2 d^{\beta+(\beta\vee 1)/2}(NM)^{-2\beta/d}.$$

We let $\phi_0(x) = \sigma(\phi(x) + \mathcal{B}) - \sigma(\phi(x) - \mathcal{B}) - \mathcal{B}$ be the truncated network where σ denotes the ReLU activation function. Then

$$\mathbb{E}|\phi_0(X) - f_{0,T}(X)| \le \mathbb{E}|\phi(X) - \tilde{f}_{0,T}(X)| \le 18B_0(\beta+1)^2 d^{\beta+(\beta\vee 1)/2}(NM)^{-2\beta/d}.$$

E.7 Proof of Theorem 4.6

Combining Lemma 4.1, Theorem 4.2 and 4.5, Theorem 4.6 is easily proved.

For the proof of Theorem 4.7, we need the following lemma, which can be proved using the technique in [55]. We refer to [55] for the detailed description of the technique.

E.8 Proof of Theorem 4.7

Proof. The proof is similar to that of Theorem 4.6, except how we deal with $\inf_{f\in\mathcal{F}_{\text{CNN}}}\mathbb{E}\|f(X)-f_{0,T}(X)\|_2$. Combining Lemma 4.4 and using Theorem 2.4, we have

$$R(\hat{f}_n)-R(f_0)\leq\frac{16\sqrt{2}B_\phi\mathcal{B}\mathcal{S}^{1/2}L^{1/4}}{\sqrt{n}}+2B_\phi\mathcal{B}\sqrt{\frac{2\log(1/\delta)}{n}}$$
$$+(18+C_2)\frac{B_0}{(1-\varepsilon)^\beta}(\lfloor\beta\rfloor+1)^2d^{1/2}d_\varepsilon^{3\lfloor\beta\rfloor/2}(NM)^{-2\beta/d_\varepsilon}+\inf_{|a|\leq T}\phi(a)-\inf_{a\in E_\phi}\phi(a).$$

This completes the proof of Theorem 4.7. □

E.9 Proof of the cross entropy loss example in Section 5

For the cross entropy loss function $\phi(a)=-\log\{0.5+a\}$, with the choice $T=\mathcal{B}=0.5-n^{\beta/(2d+4\beta)}$, we have $\mathcal{B}\leq 0.5$, $\Delta_\phi(T)\leq n^{-\beta/(2d+4\beta)}$ and $B_\phi=n^{\beta/(2d+4\beta)}$. Suppose Assumptions 2.3, 4.3 hold and $f_0\in\mathcal{H}^\beta([0,1]^d,B_0)$.

Take $N=1$ in (7) and $M=\lfloor\xi\rfloor$ in (4) and let $\mathcal{F}_{\text{CNN}}$ be the class of CNNs in (2) with depth $L\leq 378\cdot 38^2(\lfloor\beta\rfloor+1)^6d^{2\lfloor\beta\rfloor+2}\lfloor n^{d/\{2d+4\beta\}}\rfloor\lceil\log(8n)\rceil/(s_{\min}-1)$, filter lengths $2\leq s_{\min}\leq s_{\max}\leq 9\times 38^2(\lfloor\beta\rfloor+1)^4d^{2\lfloor\beta\rfloor+2}$ and size $\mathcal{S}\leq 8\mathcal{W}L\leq 42*8*9^2*38^4(\lfloor\beta\rfloor+1)^{10}d^{4\lfloor\beta\rfloor+4}\lfloor n^{d/\{2d+4\beta\}}\rfloor\lceil\log(8n)\rceil/(s_{\min}-1)$.

Theorem 4.6 implies that, with probability at least $1-\exp\{-n^{d/(d+2\beta)}\}$, the excess ϕ-risk of the ERM $\hat{f}_n$ defined in (13) satisfies

$$R(\hat{f}_n)-R(f_0)\leq n^{-\beta/(2d+4\beta)}$$
$$\times\{c(\lfloor\beta\rfloor+1)^8d^{3\lfloor\beta\rfloor+3}(\log n)^2/(s_{\min}-1)+18B_0(\lfloor\beta\rfloor+1)^2d^{\lfloor\beta\rfloor+(\lfloor\beta\rfloor\vee 1)/2}\},$$

or simply

$$R(\hat{f}_n)-R(f_0)\leq C(d,\beta,B_0,s_{\min})(\log n)^2n^{-\beta/(2d+4\beta)},$$

where $C(d,\beta,B_0,s_{\min})=O(B_0(\beta+1)^8d^{3\beta+3}/(s_{\min}-1))$ is a constant independent of n. The excess risk bound under the approximate low-dimensional manifold assumption can be obtained in a similar way.

Under the modified cross entropy loss $\phi(a)=\max\{-\log\{0.5+a\},\tau\}$ with $\tau=-\log\{0.5+T\}$, the corresponding measurable minimizer defined in (14) is $f_{0,T}$ and $\Delta_\phi(T)=0$ by definition since the infimum of the modified cross entropy loss can be achieved within $[-T,T]\subset[-0.5,0.5]$. By choosing $T=\mathcal{B}=0.5-(\log n)^{-1}$ and $\tau=-\log\{1-(\log n)^{-1}\}$, with the modified cross entropy loss, Theorem 4.6 implies that with probability at least $1-\exp\{-n^{d/(d+2\beta)}\}$, the excess ϕ-risk of the ERM $\hat{f}_n$ defined in (13) satisfies

$$R(\hat{f}_n)-R(f_0)\leq C(d,\beta,B_0,s_{\min})(\log n)^2n^{-\beta/(d+2\beta)},$$

where $C(d,\beta,B_0,s_{\min})=O(B_0(\lfloor\beta\rfloor+1)^8d^{3\lfloor\beta\rfloor+3}/(s_{\min}-1))$ is a constant independent of n.

E.10 Proof of the SVM example

Proof. Based on Lemma 4.1, our proof focus on controlling the approximation error $\inf_{f\in\mathcal{F}_{\text{CNN}}}R(f)-R(f_0)$. Recall that $f_0(x)=\text{sign}(2\eta(x)-1)$, we would firstly approximate Hölder smooth function η by CNNs. Similar to the proof in Theorem 4.6, for

any $N, M \in \mathbb{N}^+$, there exists a function $\eta^* \in \mathcal{F}_{CNN}$ with bound $\mathcal{B} = 1$, depth $L \leq 378 \cdot 38^2(\lfloor\beta\rfloor + 1)^6 d^{2\lfloor\beta\rfloor+2} M\lceil\log(8M)\rceil/(s_{\min} - 1)$, filter lengths $2 \leq s_{\min} \leq s_{\max} \leq 38^2(\lfloor\beta\rfloor + 1)^4 d^{2\lfloor\beta\rfloor+2} N\lfloor\log(8N)\rfloor$ and size $\mathcal{S} \leq 8\mathcal{W}L \leq 42 * 8 * 38^4(\lfloor\beta\rfloor + 1)^{10} d^{4\lfloor\beta\rfloor+4} M\lceil\log(8M)\rceil N^2\lceil\log(8N)\rceil^2/(s_{\min} - 1)$ such that

$$|\eta^*(x) - \eta(x)| \leq 18B_0(\lfloor\beta\rfloor + 1)^2 d^{\lfloor\beta\rfloor+(\beta\vee 1)/2}(NM)^{-2\beta/d},$$

for any $x \in \mathcal{X}\backslash\Omega$ where Ω is a set with arbitrarily small Lebesgue measure. Let $\varepsilon_n = 18B_0(\lfloor\beta\rfloor + 1)^2 d^{\lfloor\beta\rfloor+(\beta\vee 1)/2}(NM)^{-2\beta/d}$.

We construct the neural network f_{CNN} by adding one layer to η^* as

$$f_{CNN}(x) = 2\sigma\Big[\frac{1}{\varepsilon_n}\{\eta^*(x) - \frac{1}{2}\}\Big] - 2\sigma\Big[\frac{1}{\varepsilon_n}\{\eta^*(x) - \frac{1}{2}\} - 1\Big] - 1,$$

where σ is the ReLU activation function. Then we can see that $\|f_{CNN}\|_\infty \leq 1$ and $f_{CNN}(x) = 1$ if $\eta^*(x) \geq \varepsilon_n + 1/2$, $f_{CNN}(x) = 2(\eta^*(x) - 1/2)/\varepsilon_n - 1$ if $1/2 + \varepsilon_n > \eta^*(x) \geq 1/2$ and $f_{CNN}(x) = -1$ if $\eta^*(x) < 1/2$. Let $\Omega_{\eta,\varepsilon} = \{x \in \mathcal{X} : |2\eta(x) - 1| > \varepsilon\}$ for $\epsilon > 0$. Then for $x \in \Omega_{\eta,4\varepsilon_n}$, we have $|f_{CNN}(x) - f_0(x)| = 0$, since $\eta^*(x) - 1/2 = (\eta(x) - 1/2) - (\eta^*(x) - \eta(x)) > \varepsilon_n$ when $\eta(x) - 1/2 > 2\varepsilon_n$ and $\eta^*(x) - 1/2 < -\varepsilon_n$ when $\eta(x) - 1/2 < -2\varepsilon_n$. Then

$$\begin{aligned}
R(f_{CNN}) - R(f_0) =& \mathbb{E}\Big[\phi(Yf_{CNN}(X)) - \phi(Yf_0(X))\Big] \\
=& \mathbb{E}\Big[|f_{CNN}(X) - f_0(X)||2\eta(X) - 1|\Big] \\
=& \mathbb{E}_{X\in\Omega_{\eta,4\varepsilon_n}}\Big[|f_{CNN}(X) - f_0(X)||2\eta(X) - 1|\Big] \\
\leq& 8\varepsilon_n P(|2\eta(X) - 1| \leq 4\varepsilon_n) \\
\leq& 8 \times 4^q \times c_{noise} \times \varepsilon_n^{q+1}.
\end{aligned}$$

Setting the parameters in η^* by $N = 1$ and $M = \lfloor n^{d/\{d+4\beta(q+1)\}}\rfloor$ and size (number of parameters) $\mathcal{S} \leq 8\mathcal{W}L \leq 42*8*9^2*38^4(\lfloor\beta\rfloor+1)^{10} d^{4\lfloor\beta\rfloor+4}\lfloor n^{d/\{d+4\beta(q+1)\}}\rfloor\lceil\log(8n)\rceil/(s_{\min}-1)$, and let $\delta = \exp(-n^{d/(d+2\beta(q+1))})$, then by Theorem 4.6, with probability at least $1 - \exp(-n^{d/(d+2\beta(q+1))})$,

$$\begin{aligned}
R(\hat{f}_n) - R(f_0) \leq \quad & (\log n)^2 \times n^{-\beta(q+1)/\{d+2\beta(q+1)\}} \\
\times\Big\{c(\lfloor\beta\rfloor + 1)^8 d^{3\lfloor\beta\rfloor+3} + 8 \times 4^q c_{noise}[18B_0(\lfloor\beta\rfloor + 1)^2 d^{\lfloor\beta\rfloor+(\beta\vee 1)/2}]^{q+1}\Big\}.
\end{aligned}$$

The excess risk bound under the approximate low-dimensional manifold assumption can be obtained in a similar way. □

E.11 Proof of the logistic loss example

The proof is similar to that of the cross entropy loss example. We omitted the proof here.

E.12 Proof of the exponential loss example

The proof is similar to that of the cross entropy loss example. We omitted the proof here.

E.13 Supporting definitions and Lemmas

For ease of reference, we collect several definitions and existing results that we used in our proofs. Firstly, we present the definition and properties of the averaged Taylor expansion in chapter 4 of [10].

Definition E.1 (Averaged Taylor expansion)**.** Let $\beta \in \mathbb{N}, 1 \leq p \leq \infty$ and $f \in W^{\beta-1,p}$, and let $\Omega \in \mathbb{R}^d$, $x_0 \in \Omega$, $r > 0$, $B := B_r(x_0) = \{x \in \mathbb{R}^d : \|x - x_0\|_2 < r\}$ with its closure $\bar{B}$ compact in Ω. The corresponding Taylor polynomial of order β of f averaged over B is defined as

$$T^\beta f(x) = \int_B T_y^\beta f(x)\phi(y)dy, \tag{33}$$

where

$$T_y^\beta f(x) = \sum_{\|\alpha\|_1 \le \beta-1} \frac{1}{\alpha!} D^\alpha f(y)(x-y)^\alpha,$$

and ϕ is an arbitrary cut-off function supported in $\bar{B}$ being infinitely differentiable, i.e., $\phi \in C^\infty(\mathbb{R}^d)$ with $\operatorname{supp} \phi = \bar{B}$ and $\int_{\mathbb{R}^d} \phi(x)dx = 1$. For example, let

$$\psi(x) = \begin{cases} \exp\{-(1-(|x-x_0|/r)^2)^{-1}\}, \text{ if } |x-x_0| < r, \\ 0 \qquad\qquad\qquad\qquad , \text{ else,} \end{cases}$$

and let $c = \int_{\mathbb{R}^d} \psi(x)dx (c > 0)$, then $\phi(x) = \psi(x)/c$ is a cut-off function and $\|\phi\|_{L^\infty(B)} := \max|\phi(x)| \le c(d) \cdot r^{-d}$ where $c(d) > 0$ is a constant depending only on d. Besides,

$$\phi(x) = \begin{cases} \pi^{-d/2}\Gamma(d/2+1)r^{-d}, \text{ if } |x-x_0| < r, \\ 0 \qquad\qquad\qquad\qquad , \text{ else,} \end{cases}$$

is a another example of cut-off function, in which case ϕ puts constant weight on the ball d-dimensional B with radius r.

The averaged Taylor polynomial defined above is an integral of the traditional Taylor expansion weighted by the cut-off function over a region. Following the equations (4.1.5)-(4.1.8) in [10], we show the below lemma that the averaged Taylor polynomial of order β is indeed a polynomial of degree less than β in x.

Lemma E.2. *Let $\beta \in \mathbb{N}, 1 \le p \le \infty$ and $f \in W^{\beta-1,p}(\Omega)$, and let $x_0 \in \Omega, r > 0, R \ge 1$ such that for the ball $B := B_r(x_0)$ with its closure $\bar{B}$ compact in Ω and $\|y\|_\infty \le R$ for all $y \in B$. Then the Taylor polynomial of order β of f averaged over B can be written as*

$$T^\beta f(x) = \sum_{\|\alpha\|_1 \le \beta-1} c_\alpha x^\alpha \qquad \text{for } x \in \Omega,$$

where

$$|c_\alpha| \le \beta(2d)^{\beta-1}(\pi^{-d/2}\Gamma(d/2+1))^{1/p} R^{\beta-1} \|f\|_{W^{\beta-1,p}(\Omega)} r^{-d/p}$$

for all α satisfying $\|\alpha\|_1 \le \beta - 1$.

Proof. Recall that for $x, y \in \mathbb{R}^d$ and $\alpha = (\alpha_1, \dots, \alpha_d) \in \mathbb{N}_0^d$, if we write

$$(x-y)^\alpha = \Pi_{i=1}^d (x_i - y_i)^{\alpha_i} = \sum_{\gamma,\theta \in \mathbb{N}_0^d, \gamma+\theta=\alpha} a(\gamma,\theta) x^\gamma y^\theta,$$

where $\gamma = (\gamma_1, \dots, \gamma_d)$ and $\theta = (\theta_1, \dots, \theta_d)$ are d-tuples of nonnegative integers, and $a(\gamma,\theta) \in \mathbb{R}$ are the constants satisfying $|a(\gamma,\theta)| \le \binom{\alpha}{\gamma} = \alpha!/(\gamma!\theta!)$ in multi-index notation. Then

$$\begin{aligned} T^\beta f(x) = \int_B T_y^\beta f(x)\phi(y)dy &= \sum_{\|\alpha\|_1 \le \beta-1} \sum_{\gamma+\theta=\alpha} \frac{1}{\alpha!} a(\gamma,\theta) x^\gamma \int_B D^\alpha f(y) y^\theta \phi(y) dy \\ &= \sum_{\|\gamma\|_1 \le \beta-1} x^\gamma \Big\{ \sum_{\|\gamma+\theta\|_1 \le \beta-1} \frac{1}{(\gamma+\theta)!} a(\gamma,\theta) \int_B D^{\gamma+\theta} f(y) y^\theta \phi(y) dy \Big\}. \end{aligned}$$

Then we consider bounding the coefficient of x^γ. Note that

$$\begin{aligned} \Big| \int_B D^{\gamma+\theta} f(y) y^\theta \phi(y) dy \Big| &\le \int_B |D^{\gamma+\theta} f(y)||y^\theta||\phi(y)| dy \\ &\le R^{\|\theta\|_1} \|f\|_{W^{\beta-1,p}(B)} \|\phi\|_{L^q(B)} \\ &\le R^{\|\theta\|_1} \|f\|_{W^{\beta-1,p}(B)} \|\phi\|_{L^1(B)}^{1/q} \|\phi\|_{L^\infty(B)}^{1-1/q} \\ &\le R^{\beta-1} \|f\|_{W^{\beta-1,p}(\Omega)} \|\phi\|_{L^\infty(B)}^{1-1/q} \\ &\le R^{\beta-1} \|f\|_{W^{\beta-1,p}(\Omega)} (\pi^{-d/2}\Gamma(d/2+1))^{1/p} r^{-d/p}, \end{aligned}$$

where the second inequality holds since $\|y\|_\infty \leq R,\ \forall y \in B$ and the use of Hölder's inequality with $1/p + 1/q = 1$, the third inequality holds again by Hölder's inequality, and the last two lines hold since $\|\phi\|_{L^1(B)} = 1$ and $\|\phi(x)\|_{L^\infty(B)} \leq \pi^{-d/2}\Gamma(d/2+1) \cdot r^{-d}$. Lastly, the coefficient c_γ of x^γ can be upper bounded by

$$
\begin{aligned}
|c_\gamma| =& \Big| \sum_{\|\gamma+\theta\|_1 \leq \beta-1} \frac{1}{(\gamma+\theta)!} a(\gamma,\theta) \int_B D^{\gamma+\theta} f(y) y^\theta \phi(y) dy \Big| \\
&\leq R^{\beta-1} \|f\|_{W^{\beta-1,p}(\Omega)} \|\phi\|_{L^\infty(B)} \sum_{\|\gamma+\theta\|_1 \leq \beta-1} \frac{1}{(\gamma+\theta)!} |a(\gamma,\theta)| \\
&\leq (\pi^{-d/2}\Gamma(d/2+1))^{1/p} R^{\beta-1} \|f\|_{W^{\beta-1,p}(\Omega)} r^{-d/p} \sum_{\|\gamma+\theta\|_1 \leq \beta-1} \frac{1}{\gamma!\theta!}.
\end{aligned}
$$

Note that

$$
\begin{aligned}
\sum_{\|\gamma+\theta\|_1 \leq \beta-1} \frac{1}{\gamma!\theta!} &= \sum_{s=0}^{\beta-1} \sum_{\|\alpha\|_1 = s} \sum_{\gamma+\theta=\alpha} \frac{1}{\gamma!\theta!} \\
&\leq \sum_{s=0}^{\beta-1} \sum_{\|\alpha\|_1 = s} 2^s \leq \sum_{s=0}^{\beta-1} d^s 2^s \leq \beta(2d)^{\beta-1}.
\end{aligned}
$$

Thus we have

$$
|c_\gamma| \leq \beta(2d)^{\beta-1} (\pi^{-d/2}\Gamma(d/2+1))^{1/p} R^{\beta-1} \|f\|_{W^{\beta-1,p}(\Omega)} r^{-d/p}.
$$

□

Next, we present the main approximation result of averaged Taylor expansion on Sobolev functions by Bramble-Hilbert (Lemma 4.3.8 in [10]) after introduce some prerequisite definitions and conceptions.

Definition E.3 (Star-shaped set)**.** Let $\Omega, B \subset \mathbb{R}^d$. Ω is star-shaped with respect to B if for all $x \in \Omega$, the closed convex hull of $\{x\} \cup B$ is a subset of Ω.

Definition E.4 (Chunkiness parameter)**.** Suppose $\Omega \subset \mathbb{R}^d$ has diameter r_Ω and is star-shaped with respect to a ball B. Let $r_{\max} = \sup\{r \in \mathbb{R}^+ : \Omega \text{ is star-shaped with respect to a ball of radius } r\}$. Then the chunkiness parameter of Ω is defined by $\gamma = r_\Omega / r_{\max}$.

Lemma E.5 (Proposition 4.2.8 in [10])**.** *Let $f \in W^{\beta,p}(\Omega)$ and T^β be its averaged Taylor expansion defined in (33) over the ball B with radius r, then the remainder $R^\beta f := f - T^\beta f$ satisfies*

$$
R^\beta f(x) = \beta \sum_{|\alpha|=\beta} \int_{C_x} k_\alpha(x,z) D^\alpha f(x) dz,
$$

where C_x is the convex hull of $\{x\} \cup B$, $z = x + s(y-x)$, $k_\alpha(x,z) = (1/\alpha!)(x-z)^\alpha k(x,z)$ and

$$
|k(x,z)| \leq \frac{1}{d} \|\phi\|_{L^\infty(B)} (r + |x - x_0|)^d |z - x|^{-d}.
$$

Lemma E.6 (Bramble-Hilbert)**.** *Let B be a ball in Ω such that Ω is star-shaped with respect to B and such that its radius $r > r_{\max}/2$, where $r_{\max}$ is defined in Definition E.4. Let $T^\beta f$ be the Taylor polynomial of order β of f averaged over B where $f \in W^{\beta,p}(\Omega)$ and $p \geq 1$. Then*

$$
\|f - T^\beta f\|_{W^{k,p}(\Omega)} \leq \beta d^{\beta-1} \Gamma(d/2+1) \pi^{-d/2} (1 + r_\Omega/r)^d r_\Omega^{\beta-k} \|f\|_{W^{k,p}(\Omega)}, \qquad k = 0, 1, \ldots, \beta
$$

where r_Ω is the diameter of Ω.

The proof can be found in Lemma 4.3.8 in [10].

Lemma E.7 (Proposition 4.3 in [39])**.** *For any $N, M, d \in \mathbb{N}^+$ and $\delta \in (0, 3K]$ with $K = \lfloor N^{1/d} \rfloor^2 \lfloor M^{2/d} \rfloor$, there exists a one-dimensional function ϕ implemented by a ReLU FNN with width $4\lfloor N^{1/d} \rfloor + 3$ and depth $4M + 5$ such that*

$$
\phi(x) = k, \quad \text{if } x \in [\frac{k}{K}, \frac{k+1}{K} - \delta \cdot 1_{k<K-1}], \text{ for } k = 0, 1, \ldots, K-1.
$$

Lemma E.8 (Proposition 4.4 in [39]). *Given any $N, M, s \in \mathbb{N}^+$ and $\xi_i \in [0,1]$ for $i = 0, 1, \ldots, N^2L^2 - 1$, there exists a function ϕ implemented by a ReLU FNN with width $16s(N+1)\lceil \log_2(8N)\rceil$ and depth $(5M+2)\lceil \log_2(4M)\rceil$ such that*

$$|\phi(i) - \xi_i| \leq N^{-2s}M^{-2s}, \text{ for } i = 0, 1, \ldots, N^2M^2 - 1,$$

and $0 \leq \phi(x) \leq 1$ for any $x \in \mathbb{R}$.

The next lemma demonstrate that the production function and polynomials can be approximated by ReLU neural networks. The basic idea is firstly to approximate the square function using "sawtooth" functions then the production function, which is firstly raised in [60]. A general polynomial can be further approximated combining the approximated square function and production function. The following two lemmas are more general results than those in [60].

Lemma E.9 (Lemma 3.4 in [24]). *For any $N, M \in \mathbb{N}^+$, and $a, b \in \mathbb{R}$ with $a < b$, there exists a function ϕ implemented by a ReLU FNN with width $9N + 1$ and depth $2M$ such that $\|\phi\|_{W^{1,\infty}((a,b)^2)} \leq 12(b-a)^2$ and*

$$\|\phi(x,y) - xy\|_{W^{1,\infty}((a,b)^2)} \leq 6(b-a)^2N^{-M}$$

for any $x, y \in (a, b)$.

Lemma E.10 (Theorem 4.1 in [39], Proposition 3.6. in [24]). *Assume $P(x) = x^\alpha = x_1^{\alpha_1}x_2^{\alpha_2}\cdots x_d^{\alpha_d}$ for $\alpha \in \mathbb{N}^d$ with $\|\alpha\|_1 \leq k \in \mathbb{N}^+$. For any $N, M \in \mathbb{N}^+$, there exists a function ϕ implemented by a ReLU FNN with width $9(N+1) + k - 1$ and depth $14k^2M$ such that $\|\phi\|_{W^{1,\infty}(\mathcal{X})} \leq 18$ and*

$$\|\phi(x) - P(x)\|_{W^{1,\infty}(\mathcal{X})} \leq 10k(N+1)^{-7kM}.$$