Applied and Computational Harmonic Analysis 68 (2024) 101582

Contents lists available at ScienceDirect

Applied and Computational Harmonic Analysis

journal homepage: www.elsevier.com/locate/acha

Learning ability of interpolating deep convolutional neural networks

Tian-Yi Zhou *, Xiaoming Huo

H. Milton Stewart School of Industrial and Systems Engineering, Georgia Institute of Technology, 765 Ferst Drive, Atlanta, GA 30332-0205, United States of America

ARTICLE INFO

Communicated by Radu Balan

Keywords:
Convolutional neural networks
Deep learning
Benign overfitting
Learning rates
Overparameterized network

ABSTRACT

It is frequently observed that overparameterized neural networks generalize well. Regarding such phenomena, existing theoretical work mainly devotes to linear settings or fully-connected neural networks. This paper studies the learning ability of an important family of deep neural networks, deep convolutional neural networks (DCNNs), under both underparameterized and overparameterized settings. We establish the first learning rates of underparameterized DCNNs without parameter or function variable structure restrictions presented in the literature. We also show that by adding well-defined layers to a non-interpolating DCNN, we can obtain some interpolating DCNNs that maintain the good learning rates of the non-interpolating DCNN. This result is achieved by a novel network deepening scheme designed for DCNNs. Our work provides theoretical verification of how overfitted DCNNs generalize well.

1. Introduction

Neural networks are computing systems with powerful applications in many disciplines, such as data analysis and pattern and sequence recognition. In particular, deep neural networks with well-designed structures, numerous trainable free parameters, and massive-scale input data have outstanding performances in function approximation [32,18], classification [19,11], regression [30], and feature extraction [23]. The success of deep neural networks in practice has motivated research activities intended to rigorously explain their capability and power, in addition to the literature on shallow neural networks [26].

In this paper, we study an important family of neural networks known as convolutional neural networks. Given that neural networks, in general, are powerful and versatile, researchers have been working to improve their computational efficiency further. When the data dimension is large such as the AlexNet [19] of input dimension about $150,000$, fully-connected neural networks are not feasible. Structures are often imposed on neural networks to reduce the number of trainable free parameters and get feasible deep learning algorithms for various practical tasks [20]. The structure we are interested in is induced by one-dimensional convolution (1-D convolution), and the resulting networks are deep convolutional neural networks (DCNNs) [31]. The convolutional structure of DCNNs reduces the computational complexity and is believed to capture local shift-invariance properties of image and speech data. Such features of DCNNs contribute to the massive popularity of DCNNs in image processing and speech recognition.

In recent years, there has been a line of work studying overparameterization in deep learning. It is frequently observed that overparameterized deep neural networks, such as DCNNs, generalize well while achieving zero training error [6]. This phenomenon,

* Corresponding author.
E-mail address: tzhou306@gatech.edu (T.-Y. Zhou).

https://doi.org/10.1016/j.acha.2023.101582
Received 12 May 2022; Received in revised form 29 July 2023; Accepted 8 August 2023
Available online 16 August 2023

known as benign overfitting, seems to confront the classical bias-variance trade-off in statistical theory. Such a mismatch between observations and classical theory sparked avid research attempting to understand how benign overfitting occurs. Theoretical work studying benign overfitting was initiated in [4], where a linear regression setting with Gaussian data and noise was considered. It presented conditions for minimum-norm interpolators to generalize well. In a non-linear setting induced by the ReLU activation function, benign overfitting is verified for deep fully-connected neural networks in [22]. On top of that, a recent work [10] shows that training shallow neural networks with shared weights by gradient descent can achieve an arbitrarily small training error.

In this paper, we study the learning ability of DCNNs under both underparameterized and overparameterized settings. We aim to show that an overparameterized DCNN can be constructed to have the same convergence rate as a given underparameterized one while it perfectly fits the input data. In other words, we intend to prove that interpolating DCNNs generalize well.

The main contributions of the paper are as follows. Our first result rigorously proves that for an arbitrary DCNN with good learning rates, we can add more layers to build overparameterized DCNNs satisfying the interpolation condition while retaining good learning rates. Here, "learning rates" refers to rates of convergence of the output function to the regression function in a regression setting. Our second result establishes the learning rates of DCNNs in general. Previously in [33], convergence rates of approximating functions in some Sobolev spaces by DCNNs were given without generalization analysis. Moreover, learning rates of DCNNs for learning radial functions were given in [24], where the bias vectors and filters are assumed to be bounded, with bounds depending on the sample size and depths. More recently, learning rates for learning additive ridge functions were presented in [14]. Unlike these existing works, the learning rates we derive do not require any restrictions on norms of the filters or bias vectors, or variable structures of the target functions. Without the boundedness of free parameters, the standard covering number arguments do not apply. To overcome such a challenge, we derive a special estimate of the pseudo-dimension of the hypothesis space generated by a DCNN. Previously, a pseudo-dimension estimate was given in [3] for fully-connected neural networks, using the piecewise polynomial property of the activation function. We shall apply our pseudo-dimension estimate to, in turn, bound the empirical covering number of the hypothesis space. In such a way, we can achieve our results without restrictions on free parameters.

Combining our first and second results, we prove that for any input data, there exist some overparameterized DCNNs which interpolate the data and achieve a good learning rate. The third result provides theoretical support for the possible existence of benign overfitting under the DCNN setting.

The rest of this paper is organized as follows. In Section 2, we introduce notations and definitions used throughout the paper, including the definition of DCNNs to be studied. In Section 3, we present our main results that describe how a DCNN achieves benign overfitting. The proof of our first result is given in Section 4, and the proofs of our second and third results are provided in Section 5. In Section 6, we present the results of numerical experiments which corroborate our theoretical findings. Lastly, in Section 7, we present some discussions and compare our work with the existing literature.

2. Problem formulation

In this section, we define the DCNNs to be studied in this paper and the corresponding hypothesis space (Subsection 2.1). Then, we introduce the regression setting with data and the regression function (Subsection 2.2).

2.1. Deep convolutional neural networks and the corresponding hypothesis space

To begin with, we formulate the 1-D convolution. Let $w = \{w_j\}_{j=-\infty}^{+\infty}$ be a filter supported in $\{0, 1, \dots, s\}$ for some filter length $s \in \mathbb{N}$, which means $w_j \neq 0$ only for $0 \leq j \leq s$. Suppose $x = \{x_j\}_{j=-\infty}^{+\infty}$ is another sequence supported in $\{1, 2, \dots, d\}$ for some $d \in \mathbb{N}$ and is denoted as an input vector $x = (x_1, \dots, x_d)^T \in \mathbb{R}^d$ for networks in the following. The 1-D convolution of w with x, denoted by $w*x$, is defined as

$$(w*x)_i = \sum_{k\in\mathbb{Z}} w_{i-k}x_k = \sum_{k=1}^{d} w_{i-k}x_k, \qquad i \in \mathbb{Z}. \tag{2.1}$$

We can see, from (2.1), the convoluted sequence $w*x$ is supported in $\{1, 2, \dots, d+s\}$ and can be expressed in the following matrix form

$$\left[(w*x)_i\right]_{i=1}^{d+s} = T^w x, \tag{2.2}$$

where

$$\begin{aligned}
(T^w)_{i,i} &= w_0, \text{ if } i = 1, 2, \dots, d,\\
(T^w)_{i+1,i} &= w_1, \text{ if } i = 1, 2, \dots, d,\\
&\vdots\\
(T^w)_{i+s,i} &= w_s, \text{ if } i = 1, 2, \dots, d,\\
(T^w)_{i,j} &= 0, \text{ otherwise.}
\end{aligned} \tag{2.3}$$

T^w is a $(d+s)\times d$ sparse Toeplitz-type matrix often referred as the "convolutional matrix." The sparsity of T^w can be attributed to the large number of zero entries. This approach is known as "zero-padding," where we have expanded the vector $x \in \mathbb{R}^d$ to a sequence on $\mathbb{Z}$ by adding zero entries outside the support $\{1, 2, \dots, d\}$.

Now we define DCNNs by means of convolutional matrices. We take the ReLU activation function $\sigma : \mathbb{R} \to \mathbb{R}$ given by $\sigma(u) = \max\{0, u\}$. It acts on vectors componentwise.

Definition 1. A DCNN of depth $J \in \mathbb{N}$ consists of a sequence of function vectors $\{h^{(j)} : \mathbb{R}^d \to \mathbb{R}^{d_j}\}_{j=1}^J$ of widths $\{d_j := d + js\}_{j=0}^J$ defined with a sequence of filters $\mathbf{w} = \{w^{(j)}\}_{j=1}^J$ each of filter length $s \in \mathbb{N}$ and a sequence of bias vectors $\mathbf{b} = \{b^{(j)} \in \mathbb{R}^{d_j}\}$ by $h^{(0)}(x) = x$ and iteratively

$$h^{(j)}(x) = \sigma\left(T^{(j)} h^{(j-1)}(x) - b^{(j)}\right), \qquad j = 1, 2, \dots, J, \tag{2.4}$$

where $T^{(j)} := T^{w^{(j)}} = \left[w^{(j)}_{i-k}\right]_{1\le i\le d_{j-1}+s, 1\le k\le d_{j-1}}$ is a $(d_{j-1}+s)\times d_{j-1}$ convolutional matrix. The hypothesis space generated by this DCNN is given by

$$\mathcal{H}_{J,s} = \operatorname{span}\left\{c \cdot h^{(J)}(x) + a : c \in \mathbb{R}^{d_J}, a \in \mathbb{R}, \mathbf{w}, \mathbf{b}\right\}. \tag{2.5}$$

We often take the bias vector $b^{(j)}$ of the so-called "identical-in-middle" form

$$b^{(j)} = [b_1, \cdots, b_{s-1}, \underbrace{b_s, \cdots, b_s}_{d_j - 2s + 2}, b_{d_j - s + 2}, \cdots, b_{d_j}]^T \tag{2.6}$$

with $d_j - 2(s-1)$ repeated entries in the middle. This special shape of the bias vector $b^{(j)}$, together with the sparsity of the convolutional matrix T^w, tells us that the j-th layer of the DCNN involves only $(s+1) + (2s-1) = 3s$ free parameters.

To reduce data redundancy, we introduce a downsampling operator $\mathcal{D}_m : \mathbb{R}^K \to \mathbb{R}^{\lfloor K/m \rfloor}$ with a scaling parameter $m \in \mathbb{N}$ by

$$\mathcal{D}_m(v) = (v_{im})_{i=1}^{\lfloor K/m \rfloor}, \qquad v \in \mathbb{R}^K, \tag{2.7}$$

where $\lfloor u \rfloor$ denotes the integer part of $u > 0$. In other words, the downsampling operator $\mathcal{D}_m$ only "picks up" the m-th, $2m$-th, …, $\lfloor K/m \rfloor m$-th entries of v.

Definition 2. A downsampled DCNN of depth J with downsampling at layer $J_1 \in \{1, \dots, J-1\}$ has widths $d_0 = d$ and

$$d_j = \begin{cases} d_{j-1} + s, & \text{if } j \neq J_1, \\ \lfloor (d_{j-1} + s)/d \rfloor, & \text{if } j = J_1, \end{cases} \tag{2.8}$$

and function vectors $\{h^{(j)} : \mathbb{R}^d \to \mathbb{R}^{d_j}\}_{j=1}^J$ given by $h^{(0)}(x) = x$ and iteratively

$$h^{(j)}(x) = \begin{cases} \sigma\left(T^{(j)} h^{(j-1)}(x) - b^{(j)}\right), & \text{if } j \neq J_1, \\ \sigma\left(\mathcal{D}_d\left(T^{(j)} h^{(j-1)}(x)\right) - b^{(j)}\right), & \text{if } j = J_1. \end{cases} \tag{2.9}$$

In other words, the downsampling operation aims to reduce the width of a certain layer of DCNN while preserving information on data features. The hypothesis space is defined in the same way as (2.5).

In this paper, we take bias vectors $b^{(j)}$ to satisfy (2.6) for $j = 1, 2, \dots, J-1$. If no additional constraints are imposed, the number of free parameters for an output function from the hypothesis space (including filters and, biases, coefficients) equals

$$\sum_{j=1}^{J-1} (s + 1 + 2s - 1) + (s+1) + d_J + d_J + 1 = 3s(J-1) + s + 2 + 2d_J. \tag{2.10}$$

DCNNs considered in this paper are based on a "zero-padding" approach and have increasing widths. In the literature, DCNNs without zero-padding have also been introduced [15,19], and they have decreasing widths, leading to limited approximation abilities and the necessity of channels for learning. Moreover, DCNNs induced by group convolutions were studied with nice approximation properties presented in [28].

2.2. Data and regression function

Consider a training sample $D := \{z^i = (x^i, y^i)\}_{i=1}^n$ drawn independently and identically distributed from an unknown distribution ρ on $Z := \Omega \times \mathcal{Y}$. Throughout this paper, we assume that Ω is a closed bounded subset of $\mathbb{R}^d$ and $\mathcal{Y} = [-M, M]$ for some $M \ge 1$.

The regression function f_ρ is given by conditional means $f_\rho(x) = \mathbb{E}[y|x] = \int_{\mathcal{Y}} y d\rho(y|x)$ of the conditional distributions $\rho(\cdot|x)$ at $x \in \Omega$. Since $|y| \le M$ almost surely, we have $|f_\rho(x)| = |\mathbb{E}[y|x]| \le M$. So, it is natural to project function values onto the interval $[-M, M]$ and define the truncation operator π_M for any real-valued function f by

$$\pi_M f(x) = \begin{cases} f(x), & \text{if } |f(x)| \le M, \\ M, & \text{if } f(x) > M, \\ -M, & \text{if } f(x) < -M. \end{cases} \tag{2.11}$$

Moreover, we denote by $\mathcal{H}_{int,J,s}$ the set of all output functions $f \in \mathcal{H}_{J,s}$ from the hypothesis space $\mathcal{H}_{J,s}$ (2.5) satisfying the following interpolation condition

$$f(x^i) = y^i, \qquad i = 1, \dots, n. \tag{2.12}$$

When downsampling is used in the DCNN, we use the same notation to denote the set of interpolating output functions generated from the downsampled DCNN.

3. Main results

In this section, we state our main results. The corresponding theorems will be proved in Sections 4 and 5.

Here, we introduce our first result. Our first result shows that a good generalization ability of any given non-interpolating DCNN ("teacher DCNN") can be maintained by some interpolating output function $f \in \mathcal{H}_{int,J,s}$ of a "student DCNN" obtained by adding well-defined layers to the given DCNN. The learning and approximation ability is measured in regression by the L_2 norm $\|f\|_2 := \left\{\int_\Omega |f(x)|^2 d\rho_\Omega\right\}^{1/2}$ for $f \in L^2_{\rho_\Omega}$ with respect to the marginal distribution ρ_Ω of ρ on Ω.

Theorem 1. *Let $2 \le d \in \mathbb{N}$ and $S \le d/2$. Assume that ρ_Ω has no positive mass at any point $x \in \Omega$. Suppose that the hypothesis space $\mathcal{H}_{J_2,S}$ defined by (2.5) generated by a DCNN of depth $J_2 \in \mathbb{N}$ with filter length S can approximate the regression function f_ρ within accuracy $\{E_{J_2} > 0\}_{J_2 \in \mathbb{N}}$ as follows*

$$\inf_{f \in \mathcal{H}_{J_2,S}} \left\|\pi_M f - f_\rho\right\|_2 < E_{J_2}, \tag{3.1}$$

then there exists a downsampled DCNN of depth $J_1 + J_2 + J_3$ with filter length $s = 2S$, downsampling at the J_1-th layer where $J_1 = \lceil \frac{2d^2}{s-1} \rceil$, and $J_3 = \left\lceil \frac{(N-1)d_{J_1+J_2}}{s} \right\rceil$ for some odd $N \ge 3n$, such that

$$\inf_{f \in \mathcal{H}_{int,J_1+J_2+J_3,S}} \left\|\pi_M f - f_\rho\right\|_2 < E_{J_2}. \tag{3.2}$$

The additional number of free parameters of an output function from the deepened DCNN equals

$$d_{J_1+J_2+J_3} + J_1(s+2) + 3.$$

More specifically, suppose we are given a DCNN of depth J_2, which approximates f_ρ sufficiently well. Theorem 1 states that, by adding $J_1 + J_3$ well-defined convolutional layers to this given DCNN, we obtain some $f \in \mathcal{H}_{int,J_1+J_2+J_3,S}$ which interpolates the data and, at the same time, possesses the same generalization error bound as the given DCNN.

The main tool of proving Theorem 1 is a network deepening scheme, which adds well-defined layers to a given DCNN such that the deepened student DCNN interpolates the input data. The detailed proof is given in Section 4.

To verify the benign overfitting of DCNNs, it is necessary to show that an interpolating DCNN can achieve a good learning rate. The phrase "learning rate" here refers to the convergence rate of the excess generalization error (excess error), which will be defined shortly. We now turn our attention to finding learning rates of a DCNN of depth J and filter length s. It will act as a teacher DCNN later in Theorem 3. Such a DCNN generates the hypothesis space $\mathcal{H}_{J,s}$ defined in (2.5). We are interested in a global minimum of the following empirical risk minimization (ERM) problem

$$f_{D,J,s} := \arg\min_{f \in \mathcal{H}_{J,s}} \varepsilon_D(f), \tag{3.3}$$

where $\varepsilon_D(f) = \frac{1}{n}\Sigma_{i=1}^n (f(x^i) - y^i)^2$ is the empirical risk of function $f \in C(\Omega)$, the space of continuous functions on Ω with norm $\|f\|_{C(\Omega)} = \sup_{x \in \Omega} |f(x)|$.

To analyze the performance of learning algorithms for regression, we consider the generalization error defined by

$$\varepsilon(f) = \int_Z (f(x) - y)^2 \, d\rho.$$

It is minimized by the regression function f_ρ, and the excess generalization error (excess error) equals the error norm square

$$\varepsilon(f) - \varepsilon(f_\rho) = \|f - f_\rho\|_2^2 \tag{3.4}$$

and can be used in regression analysis [17].

Here, we present our second result. Our second result establishes the learning rates of DCNNs in general. Remarkably, we do not give any restrictions on norms of the filters and bias vectors or variable structures of the regression function f_ρ. The Sobolev space

$W_2^r(\Omega)$ with an integer index r consists of restrictions to Ω of F from the Sobolev space $W_2^r(\mathbb{R}^d)$ on $\mathbb{R}^d$ meaning that F and all its partial derivatives up to order r are squared integrable on $\mathbb{R}^d$.

Theorem 2. *Let $2 \le S \le d$, $M \ge 1$, $n \ge 3$, and $\Omega \subseteq [-1,1]^d$. Assume $|y| \le M$ almost surely and $f_\rho \in W_2^r(\Omega)$ with an integer index $r > 2 + d/2$. Define $f_{D,J,S}$ by the ERM scheme (3.3) with the DCNN of depth J stated in Definition 1 with "identical-in-middle" bias vectors satisfying (2.6) for layers $1, 2, \dots J-1$. The number of free parameters of this DCNN is $P_{D,J,S} \le 5dJ + 2$. For any $0 < \delta < 1$, we have with probability at least $1 - \delta$,*

$$\left\| \pi_M f_{D,J,S} - f_\rho \right\|_2^2 \le C_{M,r,f_\rho} d(\log d) \left(1 + \frac{\log(2/\delta)}{\sqrt{n}} \right) \left(\frac{(\log n) J^2 (\log J)}{n} + \frac{1}{\sqrt{n}} + \frac{\log J}{J} \right) \tag{3.5}$$

where C_{M,r,f_ρ} is a constant independent of n, δ or d.
Specifically, when $J = \lceil n^\alpha \rceil$ with any $0 < \alpha < 1/2$, the number of free parameter is $P_{D,\lceil n^\alpha \rceil,S} \le 5d \lceil n^\alpha \rceil + 2$. With probability at least $1 - \delta$, we have

$$\left\| \pi_M f_{D,J,S} - f_\rho \right\|_2^2 \le \max\left\{ 8M^2, 6C_{M,r,f_\rho} \right\} d(\log d) \left(1 + \frac{\log(2/\delta)}{\sqrt{n}} \right) \frac{(\log n)^2}{n^{\min\{1-2\alpha,\alpha\}}}. \tag{3.6}$$

Theorem 2 establishes learning rates of DCNNs explicitly in terms of the dimension of the input data d and size n of the training sample. It states that when $J = \lceil n^\alpha \rceil$ for any $0 < \alpha < 1/2$, the DCNN output function $\pi_M f_{D,J,S}$ converges to the regression function f_ρ with high probability. In particular, with the choice of $J = \lceil n^{\frac{1}{3}} \rceil$, the rate of convergence is of order $\mathcal{O}((\log n)^2 n^{-1/3})$. To our best knowledge, this is the first result presenting learning rates of DCNNs for learning a general function without any variable structure assumption. It differs from the existing learning rates of DCNNs for learning functions with variable structures such as additive ridge functions [14] and radial functions [24]. This is also the first learning rate of DCNNs without parameter restrictions presented in the literature. The proof of Theorem 2 is given in Section 5.

Next, we present our third result. According to Theorem 2, for any $0 < \alpha < 1/2$, an underparameterized, non-interpolating DCNN with $\mathcal{O}(n^\alpha)$ free parameters converges to the regression function with a learning rate of order $\mathcal{O}((\log n)^2 n^{-1/3})$. In this paper, we refer to "underparameterized neural networks" [5] as networks that have the number of trainable free parameters of order $o(n)$. Next, by applying the results in Theorem 1, we can deepen this underparameterized DCNN to an interpolating, overparameterized one. The following theorem suggests that this overparameterized DCNN with $\mathcal{O}(n^{1+\alpha})$ free parameters can not only interpolate any input data, but also achieve the same learning rate as the underparameterized DCNN.

Theorem 3. *Under the assumption of Theorem 2 and that ρ_Ω has no positive mass at any point $x \in \Omega$, for any $0 < \alpha < 1/2$, there exists a downsampled DCNN of depth $J_1 + J_2 + J_3$ with even filter length s satisfying $4 \le s \le d$, downsampling at the J_1-th layer, where $J_1 = \lceil \frac{2d^2}{s-1} \rceil$, $J_2 = \lceil n^\alpha \rceil$, and $J_3 = \left\lceil \frac{4n\left(d_{J_1} + s\lceil n^\alpha \rceil\right)}{s} \right\rceil$, such that for any $0 < \delta < 1$, with probability $1 - \delta$, we have*

$$\inf_{f \in \mathcal{H}_{int, J_1+J_2+J_3, s}} \left\| \pi_M f - f_\rho \right\|_2^2 < \max\left\{ 8M^2 + 1, 6C_{M,r,f_\rho} + 1 \right\} d(\log d) \left(1 + \frac{\log(2/\delta)}{\sqrt{n}} \right) \frac{(\log n)^2}{n^{\min\{1-2\alpha,\alpha\}}}. \tag{3.7}$$

The number of free parameters of this DCNN is of order $\mathcal{O}(n^{1+\alpha})$.

Theorem 3 tells us that there exists an interpolating DCNN that generalizes well. This result provides theoretical support for the possible existence of benign overfitting in the DCNN setting. At this point, we are unable to derive learning rates of DCNN with $J = \lceil n^\alpha \rceil$ layers for $1/2 \le \alpha < 1$. The challenge is mainly due to an upper bound of pseudo-dimension stated in Theorem 4 below (an extension to a previous result in [3]), which does not cover the case $1/2 \le \alpha < 1$. It would be interesting to develop new approaches to estimate the pseudo-dimension of DCNNs, which in turn estimates the covering number. Furthermore, it is worth noting that the learning rates we obtained (for both underparameterized and overparameterized DCNNs) do not achieve the minimax rate of convergence for least squares regression [30]. Whether one can obtain a minimax rate of convergence of DCNNs remains an open problem.

4. Achieving interpolation condition by deepening DCNNs

In this section, we present a network deepening scheme for proving Theorem 1. Here, we first give an outline of the proof.

Suppose we are given a non-interpolating DCNN that outputs a function f^*. Theorem 1 tells us that we can construct a larger DCNN that interpolates any data while maintaining the generalization ability of f^*. The proof is carried out by means of an interpolator of the form

$$f(x) = f^*(x) + \sum_{\ell=1}^{n} \left(y^\ell - f^*(x^\ell) \right) \phi_\ell(x) \quad \text{with} \quad \phi_\ell(x) = \phi\left(\eta^* \xi \cdot (x - x^\ell) \right), \tag{4.1}$$

where $\eta^* \in \{1, -1\}$ is a sign number, $\xi \in \mathbb{R}^d$ is a nonzero vector, and $\phi = \phi^{(\epsilon)} : \mathbb{R} \to \mathbb{R}$ given with some $\epsilon > 0$ by

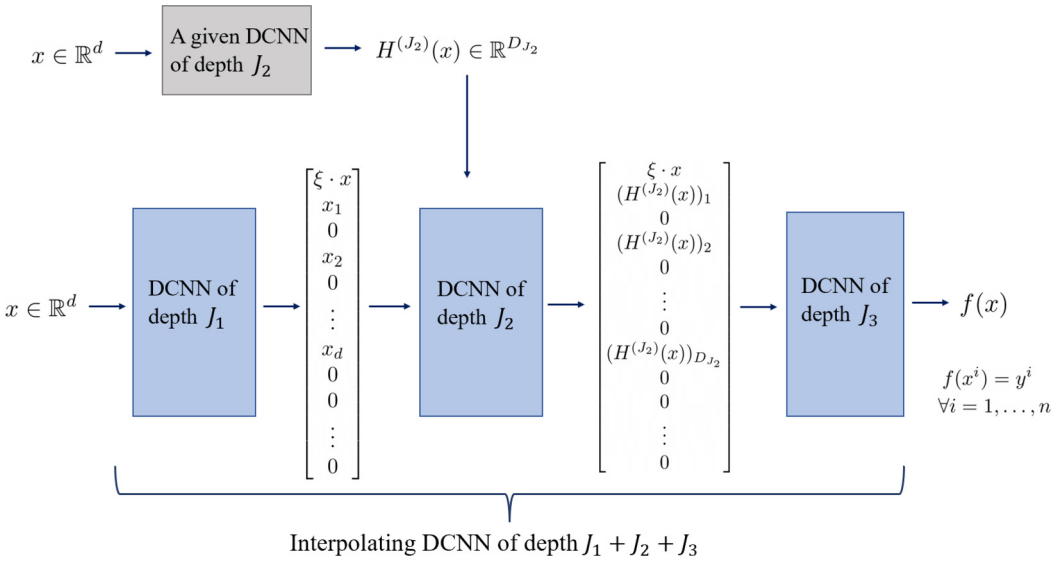

Fig. 1. Given an input $x \in \mathbb{R}^d$ and a given non-interpolating DCNN of depth J_2, the figure illustrates the process of constructing an interpolating DCNN of depth $J_1 + J_2 + J_3$.

$$\phi(u) = \frac{1}{\epsilon}\{\sigma(u+\epsilon) - \sigma(u)) - (\sigma(u) - \sigma(u-\epsilon)\}$$
$$= \begin{cases} 1 + \frac{1}{\epsilon}u & \text{if } -\epsilon \le u < 0, \\ 1 - \frac{1}{\epsilon}u & \text{if } 0 \le u \le \epsilon, \\ 0 & \text{if } |u| > \epsilon. \end{cases}$$

We can see that ϕ is a hat function vanishing outside $[-\epsilon, \epsilon]$, equal to 1 at 0, and linear on $[-\epsilon, 0]$ and on $[0, \epsilon]$. When ϵ is small enough, the basis function ϕ_ℓ in (4.1) takes the form

$$\phi_\ell(x^j) = \begin{cases} 1 & \text{if } j = \ell, \\ 0 & \text{if } j \ne \ell, \end{cases}$$

leading to the desired interpolation property (that is, $f(x^i) = y^i$ for all i) while maintaining the generalization abilities of f^*. Our goal is to construct a DCNN that produces this function f. The construction of such a DCNN can be divided into three steps.

First, we use a group of convolutional layers to realize linear features $\xi \cdot x$ in the basis functions ϕ_ℓ for an arbitrary $\xi \in \mathbb{R}^d$ (Subsection 4.1). Then, we introduce a network deepening scheme for any given DCNN (Subsection 4.2). It doubles the widths of the given DCNN, maintains its learning rate, and at the same time embeds the linear features $\xi \cdot x$. This is the key novel idea in our deepening scheme. Lastly, we use ridge functions to construct DCNN interpolators (Subsections 4.3 and 4.4). The overview of this network construction is illustrated in Fig. 1.

Before we get into these three steps, we first introduce a proposition. Suppose we have any given sequence W and network input $U(x)$, where $U \in (C(\Omega))^K$ with norm $\|U\|_\infty = \sup_{x\in\Omega} \|U(x)\|_{\ell^\infty(\mathbb{R}^K)}$. This proposition suggests that we can construct a DCNN that outputs $T^W U(x)$, where T^W is a convolutional matrix induced by W. Later in constructing linear features (the first step) and ridge functions (the third step), we will apply this result to produce some special vectors truncated from a sequence W.

Denote $\mathbf{1}_K = (1, \ldots, 1) \in \mathbb{R}^K$. Recall the "identical-in-middle" bias vector introduced in (2.6):

$$b^{(j)} = [b_1, \cdots, b_{s-1}, \underbrace{b_s, \cdots, b_s}_{d_j - 2s + 2}, b_{d_j - s + 1}, \cdots, b_{d_j}]^T.$$

Proposition 1. *If $W = (W_k)_{k=-\infty}^{\infty}$ is a sequence supported in $\{0, \ldots, \mathcal{V}\}$, $K \in \mathbb{N}$ is the input dimension, and $2 \le s \le K$, then there exist filters $\{w^{(j)}\}_{j=1}^{J^*}$ each supported in $\{0, \ldots, s\}$ with $J^* = \lceil \frac{\mathcal{V}}{s-1} \rceil$ such that the convolutional factorization $W = w^{(J^*)} * w^{(J^*-1)} * \cdots * w^{(2)} * w^{(1)}$ holds. The filter $w^{(i)}$ induces a $(K + is) \times (K + (i-1)s)$ convolutional matrix $T^{(i)}$ given by (2.3).*

Consider a DCNN $\left\{\hat{h}^{(j)}(x) : \mathbb{R}^K \to \mathbb{R}^{K+js}\right\}_{j=0}^{J^*}$ satisfying $\hat{h}^{(j)}(x) = \sigma\left(T^{(j)}\hat{h}^{(j-1)}(x) - b^{(j)}\right)$. If the input layer $\hat{h}^{(0)}$ is given by $U(x) + \vec{C}$ with $\vec{C} \in \mathbb{R}^K$ and $\|U\|_\infty \le \mathcal{B}^{(0)}$ for some $\mathcal{B}^{(0)} > 0$, and the bias vectors are given by $b^{(1)} = T^{(1)}\vec{C} - \|w^{(1)}\|_1 \mathcal{B}^{(0)} \mathbf{1}_{K+s}$, and

$$b^{(j)} = \left\{\prod_{i=1}^{j-1} \|w^{(i)}\|_1\right\} \mathcal{B}^{(0)} T^{(j)} \mathbf{1}_{K+(j-1)s} - \left\{\prod_{i=1}^{j} \|w^{(i)}\|_1\right\} \mathcal{B}^{(0)} \mathbf{1}_{K+js}, \qquad j = 2, \ldots, J^* - 1, \tag{4.2}$$

then $b^{(j)}$ satisfy the "identical-in-middle" condition (2.6) for $j = 2, \dots, J^* - 1$, the following expressions hold

$$\hat{h}^{(j)}(x) = T^{(j)}T^{(j-1)} \cdots T^{(1)}U(x) + \left\{ \prod_{i=1}^{j} \|w^{(i)}\|_1 \right\} \mathcal{B}^{(0)} \mathbf{1}_{K+js}, \quad j = 1, \dots, J^* - 1, \tag{4.3}$$

and

$$T^{(J^*)}\hat{h}^{(J^*-1)}(x) = T^W U(x) + \left\{ \prod_{i=1}^{J^*-1} \|w^{(i)}\|_1 \right\} \mathcal{B}^{(0)} T^{(J^*)} \left(\mathbf{1}_{K+(J^*-1)s} \right), \tag{4.4}$$

where $T^W = \left[W_{\ell-k} \right]_{\ell,k} \in \mathbb{R}^{(K+J^*s)\times K}$. Moreover, if $\vec{C} = C\mathbf{1}_K$ for some $C \geq 0$, then (2.6) is also satisfied for $j = 1$.

The expression (4.4) will be applied in the first and third steps of our construction, with different choices of W.

Remark 1. The convolutional matrix T^W is a $(K + J^*s) \times K$ Toeplitz-type matrix induced by W and is given by

$$T^W = \begin{bmatrix} W_0 & 0 & \cdots & 0 \\ W_1 & W_0 & \ddots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ W_{K-1} & \cdots & W_1 & W_0 \\ W_K & \ddots & \ddots & W_1 \\ \vdots & \ddots & \ddots & \vdots \\ W_{2K-1} & W_{2K-2} & \cdots & W_K \\ \vdots & \ddots & \ddots & \vdots \\ W_{J^*s} & \cdots & \ddots & \vdots \\ 0 & W_{J^*s} & \cdots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & W_{J^*s} \end{bmatrix}. \tag{4.5}$$

If we apply the downsampling operator $D_K : \mathbb{R}^{K+J^*s} \to \mathbb{R}^{\lfloor (K+J^*s)/K \rfloor}$ in (2.7) to the last layer of DCNN proposed in Proposition 1, we get

$$D_K \left(T^{(J^*)}\hat{h}^{(J^*-1)}(x) \right) = \begin{bmatrix} W_{K-1} & \cdots & W_1 & W_0 \\ W_{2K-1} & W_{2K-2} & \cdots & W_K \\ W_{3K-1} & W_{2K-2} & \cdots & W_{2K} \\ \vdots & \ddots & \ddots & \vdots \\ W_{K+K\lfloor J^*s/K \rfloor - 1} & \cdots & \ddots & W_{K\lfloor J^*s/K \rfloor} \end{bmatrix} U(x) + b, \tag{4.6}$$

where b is some constant vector. We see that all matrix entries in (4.6) are potentially different. Later, we use this important observation to generate specific features using DCNNs.

Before we prove Proposition 1, we first introduce two supporting Lemmas.

Lemma 1 was stated as Theorem 3 in [33]. It suggests that any sequence W can be factorized into convolutions of some smaller sequences. The filters $\{w^{(j)}\}_{j=1}^{J^*}$ in Proposition 1 are generated according to this result.

Lemma 1. *If $W = (W_k)_{k=-\infty}^{\infty}$ is supported in $\{0, \dots, \mathcal{V}\}$, then there exist filters $\{w^{(j)}\}_{j=1}^{p}$ each supported in $\{0, \dots, s\}$ with $p \leq \lceil \frac{\mathcal{V}}{s-1} \rceil$ satisfying a convolutional factorization*

$$W = w^{(p)} * w^{(p-1)} * \cdots * w^{(2)} * w^{(1)}.$$

A nice property stated in the following lemma (Lemma 2) is that products of convolutional matrices $\prod_{i=1}^{j} T^{(i)}$ are still convolutional matrices but induced by the convoluted sequence of larger support. The proof of Lemma 2 is given in Appendix A.1.

Lemma 2. *Let $\{w^{(j)}\}_{j=1}^{J^*}$ be filters supported in $\{0, \dots, s\}$, then for $j = 1, \dots, J^*$, we have*

$$\prod_{i=1}^{j} T^{(i)} = T^{\mathbf{w},j}, \tag{4.7}$$

where $\prod_{i=1}^{j} T^{(i)} = T^{(j)}T^{(j-1)} \cdots T^{(1)}$, and $T^{\mathbf{w},j} := \left[W^{(j)}_{\ell-k} \right]_{\ell=1,\dots,K+js,k=1,\dots,K}$ is the $(K+js)\times K$ convolutional matrix induced by the convoluted sequence $W^{(j)} = w^{(j)} * w^{(j-1)} * \cdots * w^{(1)}$ supported in $\{0, \dots, js\}$.

Now we are in a position to prove Proposition 1.

Proof of Proposition 1. We apply Lemma 1 and find filters $\{w^{(j)}\}_{j=1}^{p}$ with $p \le J^* = \lceil \frac{\mathcal{V}}{s-1} \rceil$ satisfying $W = w^{(p)} * w^{(p-1)} * \cdots * w^{(2)} * w^{(1)}$. When $p < J^*$, we choose each filter $w^{(j)}$ with $j = p+1, \ldots, J^*$ to be the delta sequence which takes the value 1 at 0 and the value 0 at any other integers (that is, $w_0^{(j)} = 1$ and $w_i^{(j)} = 0$ for $i \neq 0$). Then the desired convolutional factorization follows.

Note that the Toeplitz-type matrix $T^{(j)}$ induced by the filter $w^{(j)}$ has the form

$$T^{(j)} = \begin{bmatrix} w_0^{(j)} & 0 & 0 & \cdots & \cdots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ w_s^{(j)} & \cdots & w_0^{(j)} & 0 & \cdots & 0 & 0 \\ 0 & w_s^{(j)} & \cdots & w_0^{(j)} & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & 0 & w_s^{(j)} & \cdots & w_0^{(j)} \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & \cdots & 0 & 0 & w_s^{(j)} \end{bmatrix}. \tag{4.8}$$

We made an important observation that the entry sum of each row in the middle of $T^{(j)}$ equals $\sum_{\ell=0}^{s} w_\ell^{(j)}$. Then, we see that the choice (4.2) of the bias vector $b^{(j)}$ is indeed a "identical-in-middle" bias vector satisfying (2.6) for $j = 2, \ldots, J^* - 1$. It is also satisfied for $j = 1$ if $\vec{C} = C\mathbf{1}_K$ for some $C \ge 0$.

Then we prove the expression (4.3) by induction. Since the row sums of the Toeplitz-type matrix T^w is bounded by $\|w\|_1 = \sum_{k=-\infty}^{\infty} |w_k|$, we know that $\|T^w v\|_\infty \le \|w\|_1 \|v\|_\infty$ for any $v \in \mathbb{R}^K$. The case $j = 1$ holds because $T^{(1)}\hat{h}^{(0)}(x) - b^{(1)} = T^{(1)}U(x) + \|w^{(1)}\|_1 B^{(0)}\mathbf{1}_{K+s}$ has nonnegative entries due to the fact that each component of $T^{(1)}U(x)$ is bounded by $\|w^{(1)}\|_1 \|U(x)\|_\infty \le \|w^{(1)}\|_1 B^{(0)}$.

If (4.3) is valid for $j-1$, that is,

$$\hat{h}^{(j-1)}(x) = T^{(j-1)} \cdots T^{(1)}U(x) + \left\{ \prod_{i=1}^{j-1} \|w^{(i)}\|_1 \right\} B^{(0)}\mathbf{1}_{K+(j-1)s},$$

then

$$T^{(j)}\hat{h}^{(j-1)}(x) - b^{(j)} = T^{(j)}T^{(j-1)} \cdots T^{(1)}U(x) + \left\{ \prod_{i=1}^{j} \|w^{(i)}\|_1 \right\} B^{(0)}\mathbf{1}_{K+js}$$

has nonnegative entries since

$$\|T^{(j)}T^{(j-1)} \cdots T^{(1)}U(x)\|_\infty \le \left\{ \prod_{i=1}^{j} \|w^{(i)}\|_1 \right\} B^{(0)}.$$

Hence,

$$\begin{aligned} \hat{h}^{(j)}(x) &= \sigma\left(T^{(j)}\hat{h}^{(j-1)}(x) - b^{(j)}\right) \\ &= \sigma\left(T^{(j)}T^{(j-1)} \cdots T^{(1)}U(x) + \left\{ \prod_{i=1}^{j} \|w^{(i)}\|_1 \right\} B^{(0)}\mathbf{1}_{K+js} \right) \\ &= T^{(j)}T^{(j-1)} \cdots T^{(1)}U(x) + \left\{ \prod_{i=1}^{j} \|w^{(i)}\|_1 \right\} B^{(0)}\mathbf{1}_{K+js} \end{aligned}$$

This completes the induction procedure and verifies (4.3). The expression for $T^{(J^*)}\hat{h}^{(J^*-1)}(x)$ follows easily. The proof of Proposition 1 is complete. □

4.1. The first step: realizing linear functions by DCNNs

The first step described in this subsection shows how to realize a linear feature $\xi \cdot x$, i.e., the inner product of vectors ξ and x, by the first J_1 layers of a DCNN. In this step, we apply Proposition 1 and the downsampling operation. This idea of generating linear features has been introduced for DCNNs in [33] and for periodized DCNNs using circular weight matrices in [28]. We choose a special sequence W such that the output of this step is a function vector containing $\xi \cdot x$ and x_i.

By choosing $K = d$, $[W_{d-1} \ \cdots \ W_1 \ W_0]$ to be the vector ξ and $[W_{id+d-1} \ \cdots \ W_{id+1} \ W_{id}]$ to be the i-th basis vector e_i for $i = 1, \ldots, d$, we can make the function vector on the right-hand side of (4.6) to have components $\xi \cdot x$ and x_i.

Proposition 2. *Let $2 \le s \le d$, $J_1 = \lceil \frac{2d^2-1}{s-1} \rceil$, $\xi \in \mathbb{R}^d$, Ω be a closed bounded subset of $\mathbb{R}^d$. Take a constant $B^{(0)} \ge \max_{x\in\Omega} \|x\|_\infty$. Then there exist filters $\{w^{(j)}\}_{j=1}^{J_1}$ each supported in $\{0, \ldots, s\}$ such that with $b^{(1)} = -\|w^{(1)}\|_1 B^{(0)}\mathbf{1}_{d+s}$, $\{b^{(j)}\}_{j=2}^{J_1-1}$ given by (4.2) with $K = d, J^* = J_1, B^{(0)} = B^{(0)}$ and*

$$b^{(J_1)} = \left\{ \prod_{i=1}^{J_1-1} \|w^{(i)}\|_1 \right\} B^{(0)} T^{(J_1)} \left(\mathbf{1}_{d+(J_1-1)s} \right) + \left\{ \prod_{i=1}^{J_1} \|w^{(i)}\|_1 \right\} B^{(0)} \mathbf{1}_{d+J_1 s},$$

the first J_1 layer of the DCNN $\left\{h^{(j)}(x) : \mathbb{R}^d \to \mathbb{R}^{d_j}\right\}_{j=0}^{J_1}$ defined by (2.9) with $J_0 = J_1$ produces

$$h^{(J_1)}(x) = \left[\xi \cdot x, x_1, 0, x_2, 0, x_3, 0, \cdots, x_d, \underbrace{0, 0, \cdots, 0}_{\lfloor J_1 s/d \rfloor - 2d + 1} \right]^T + B^{(J_1)} \mathbf{1}_{1+\lfloor J_1 s/d \rfloor}, \qquad x \in \Omega.$$

Here $\{w^{(j)}, \{b^{(j)}\}_{j=1}^{J_1}$ and the constant $B^{(J_1)} := \left\{ \prod_{i=1}^{J_1} \|w^{(i)}\|_1 \right\} B^{(0)}$ depend on ξ, d, s, Ω. The bias vectors $\{b^{(j)}\}_{j=1}^{J_1}$ are "identical-in-middle" satisfying (2.6). The number of free parameters of the first J_1 layers equals $J_1(s+2)+1$.

Proof. Denote the standard basis of $\mathbb{R}^d$ by $\{e_i\}_{i=1}^d$ and the zero vector in $\mathbb{R}^d$ by 0_d. Note that $\{e_i\}_{i=1}^d$ and 0_d are column vectors. Let W be a sequence supported on $\{0, 1, \ldots, 2d^2-1\}$ given by

$$\left[W_{2d^2-1}, \cdots, W_0 \right] = \left[e_d^T; 0_d^T; e_{d-1}^T; 0_d^T; \cdots; e_2^T; 0_d^T; e_1^T; \xi^T \right]. \tag{4.9}$$

Then by Proposition 1 with $\mathcal{V} = 2d^2 - 1$, there exist filters $\{w^{(j)}\}_{j=1}^{J_1}$ each supported in $\{0, \ldots, s\}$ with $J_1 = \lceil \frac{2d^2-1}{s-1} \rceil$ satisfying a convolutional factorization $W = w^{(J_1)} * w^{(J_1-1)} * \cdots * w^{(2)} * w^{(1)}$. Each filters $w^{(j)}$ induces a $(K+js) \times (K+(j-1)s)$ convolutional matrix $T^{(j)}$.

Take $K = d, U(x) = x, \vec{C} = 0, J^* = J_1$. Then the conditions of Proposition 1 are satisfied, and the chosen bias vectors satisfy (2.6). By the expression (4.4) in Proposition 1, the resulting DCNN $\left\{h^{(j)}(x) : \mathbb{R}^d \to \mathbb{R}^{d_j}\right\}_{j=0}^{J_1}$ defined by (2.9) satisfies

$$T^{(J_1)} h^{(J_1-1)}(x) = T^W x + \left\{ \prod_{i=1}^{J_1-1} \|w^{(i)}\|_1 \right\} B^{(0)} T^{(J_1)} \left(\mathbf{1}_{d+(J_1-1)s} \right).$$

We apply the downsampling operator $\mathcal{D}_d$ in the J_1-th layer and obtain

$$h^{(J_1)}(x) = \mathcal{D}_d \left(T^{(J_1)} h^{(J_1-1)}(x) - b^{(J_1)} \right) = \mathcal{D}_d \left(T^W x + B^{(J_1)} \mathbf{1}_{d+J_1 s} \right).$$

Note that $T^W = \left[W_{\ell-k} \right]_{\ell=1,\ldots,d+J_1 s, k=1,\ldots,d}$ is the $(d+J_1 s) \times d$ convolutional matrix induced by the sequence W supported on $\{0, 1, \ldots, 2d^2-1\}$ given by (4.9). As we remarked for the downsampled Toeplitz-type matrix (4.6), the first component of $\mathcal{D}_d \left(T^W x \right)$ is the product of the d-th row $\left[W_{d-1} \cdots W_0 \right] = \xi^T$ of T^W with x and identical to the linear feature $\xi^T x = \xi \cdot x$. The $2i$-th component of $\mathcal{D}_d \left(T^W x \right)$ for $i = 1, \ldots, d$ is the product of the $2id$-th row $\left[W_{2id-1}, \cdots, W_{(2i-1)d} \right] = e_i^T$ of T^W with x and is exactly $e_i^T x = x_i$, while the $(2i-1)$-th component is given by the $(2i-1)d$-th row which is a zero row and thereby vanishes. The (id)-th row of T^W with $i \geq 2d+1$ is $\left[W_{id-1}, \cdots, W_{(i-1)d} \right]$ which is also the zero vector. It follows that

$$\begin{aligned} h^{(J_1)}(x) &= \mathcal{D}_d (T^W x) + \mathcal{D}_d (B^{(J_1)} \mathbf{1}_{d+J_1 s}) \\ &= \left[\xi \cdot x, x_1, 0, x_2, 0, \cdots, x_d, \underbrace{0, 0, \cdots, 0}_{\lfloor J_1 s/d \rfloor - 2d + 1} \right]^T + B^{(J_1)} \mathbf{1}_{1+\lfloor J_1 s/d \rfloor}. \end{aligned}$$

This yields our desired expression of $h^{(J_1)}(x)$.

The number of free parameters from filters equals $J_1(s+1)$, whereas the number of free parameters from biases equals $J_1 + 1$. The proof is complete. □

Note that $J_1 s/d \geq \frac{2d^2-1}{s-1} \frac{s}{d} > 2d-1$. Hence the width of $h^{(J_1)}(x)$ equals $d_{J_1} = 1 + \lfloor J_1 s/d \rfloor \geq 2d$. Also, we have $\|\xi\|_1 \leq \|W\|_1 \leq \prod_{i=1}^{J_1} \|w^{(i)}\|_1$. So we have $\|\xi \cdot x\|_\infty \leq B^{(J_1)}$.

4.2. The second step: deepening a given DCNN

Now we continue to the second step of our construction. In this subsection, we introduce a novel network deepening scheme. Applying this scheme to a given DCNN, we can construct a deeper and wider DCNN that embeds the linear feature $\xi \cdot x$ while preserving the output of the given DCNN.

Suppose we are given a DCNN (as a teacher net) $\left\{H^{(j)}(x) : \mathbb{R}^d \to \mathbb{R}^{D_j}\right\}_{j=0}^{J_2}$ with input $H^{(0)}(x) = x \in \Omega$, depth $J_2 \in \mathbb{N}$ and filter length $S \leq d/2$ satisfying $D_j = d + jS$. Each layer of this DCNN has the form

$$H^{(j)}(x) = \sigma \left(\overset{\circ}{T}{}^{(j)} H^{(j-1)}(x) - \overset{\circ}{b}{}^{(j)} \right), \qquad j = 1, 2, \ldots, J_2, \tag{4.10}$$

where $\left\{ \begin{matrix} \overset{\circ}{T}^{(j)} \\ \end{matrix} = T^{\overset{\circ}{w}^{(j)}} \right\}_{j=1}^{J_2}$ are the convolutional matrices generated by the filter sequence $\overset{\circ}{\mathbf{w}} = \left(\overset{\circ}{w}^{(j)} \right)_{j=1}^{J_2}$ and $\mathbf{b} = \left\{ \begin{matrix} \overset{\circ}{b}^{(j)} \\ \end{matrix} \in \mathbb{R}^{D_j} \right\}_{j=1}^{J_2}$ is a bias vector sequence.

Now we construct a student DCNN with filter length $s = 2S$ and depth $J_1 + J_2$, where $J_1 = \lceil \frac{2d^2-1}{s-1} \rceil$, $J_2 \in \mathbb{N}$ to be determined later. The first J_1 layers are exactly the DCNN described in Proposition 2. For $j = J_1 + 1, \dots, J_1 + J_2$, we denote

$$d_j = d_{J_1} + (j - J_1)s, \quad B^{(j)} = \prod_{i=1}^{j-J_1} \left\| \overset{\circ}{w}^{(i)} \right\|_1 B^{(J_1)}.$$

We define the filter $w^{(j)}$ supported in the set of even integers $\{0, 2, 4, \dots, 2S = s\}$ by

$$\left(w_{2i}^{(j)} \right)_{i=0}^{S} = \overset{\circ}{w}^{(j-J_1)}.$$

Taking the filter to be supported on the set of even integers and identical to that of the teacher net is our novelty. The special form of this filter gives a convolutional matrix $T^{(j)} = \left[w_{k-\ell}^{(j)} \right]$ as

$$T^{(j)} = \begin{bmatrix} \overset{\circ}{w}_0^{(j-J_1)} & 0 & 0 & \cdots & \cdots & 0 \\ 0 & \overset{\circ}{w}_0^{(j-J_1)} & 0 & \cdots & \cdots & 0 \\ \overset{\circ}{w}_1^{(j-J_1)} & 0 & \overset{\circ}{w}_0^{(j-J_1)} & 0 \cdots & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \overset{\circ}{w}_i^{(j-J_1)} & 0 & \overset{\circ}{w}_{i-1}^{(j-J_1)} & 0 & \cdots & \cdots \\ 0 & \overset{\circ}{w}_i^{(j-J_1)} & 0 & \overset{\circ}{w}_{i-1}^{(j-J_1)} & 0 & \cdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & \cdots & 0 & w_s^{(j-J_1)} \end{bmatrix}. \tag{4.11}$$

It implies an essential property of our novel construction: the even entries of the student net output capture all the output entries of the teacher net. We can see that when the even entries of $h^{(j-1)}(x)$ and $b^{(j)}$ are identical to those of $H^{(j-J_1-1)}(x)$ and $\overset{\circ}{b}^{(j-J_1)}$ respectively, the even entries of $h^{(j)}(x)$ are kept the same as those in $H^{(j-J_1)}(x)$. The odd entries are made zero by taking large enough biases except for the first entry, which is kept in the form $\left(\Pi_{i=1}^{j-J_1} \overset{\circ}{w}_0^{(i)} \right) \xi \cdot x + B^{(j)}$, by means of the special form of the first row in (4.11). This leads to the following proposition. We denote $\delta_{j,k} = \begin{cases} 1, & \text{if } j = k, \\ 0, & \text{if } j \neq k. \end{cases}$

Proposition 3. *Define the CNN layers* $\left\{ h^{(j)}(x) : \mathbb{R}^d \to \mathbb{R}^{d_j} \right\}_{j=J_1+1}^{J_1+J_2}$ *by (2.4), where*

$$b^{(j)} = \delta_{j,J_1+1} B^{(J_1)} T^{(J_1+1)} \mathbf{1}_{d_{j-1}} + \beta^{(j)}, \qquad j = J_1 + 1, \dots, J_1 + J_2$$

with the vector $\beta^{(j)} \in \mathbb{R}^{d_j}$ *given by*

$$\beta_i^{(j)} = \begin{cases} \left(1 - \delta_{j,J_1} \right) \overset{\circ}{w}_0^{(j-J_1)} B^{(j-1)} - B^{(j)}, & \text{if } i = 1, \\ \overset{\circ}{b}_{i/2}^{(j-J_1)}, & \text{if } i \in \left\{ 2, 4, 6, \dots, 2D_{j-J_1} \right\}, \\ \left\| \overset{\circ}{w}^{(j-J_1)} \right\|_1 \left\| H^{(j-J_1-1)} \right\|_\infty + 2B^{(j)}, & \text{if } i \notin \left\{ 1, 2, 4, 6, \dots, 2D_{j-J_1} \right\}. \end{cases}$$

Then for $j = J_1 + 1, \dots, J_1 + J_2$, *we have*

$$h^{(j)}(x) = \Big[\left(\Pi_{i=1}^{j-J_1} \overset{\circ}{w}_0^{(i)} \right) \xi \cdot x + B^{(j)}, \quad \left(H^{(j-J_1)}(x) \right)_1, 0, \left(H^{(j-J_1)}(x) \right)_2, \\ 0, \left(H^{(j-J_1)}(x) \right)_3, 0, \cdots, 0, \left(H^{(j-J_1)}(x) \right)_{D_{j-J_1}}, 0, \cdots, 0 \Big]^T.$$

No new free parameters are required for constructing $J_1 + 1, \dots, J_1 + J_2$*-th layers.*

Proof. We prove by induction that for $j = J_1, J_1 + 1, \dots, J_1 + J_2$,

$$h^{(j)}(x) = \Big[\left(\Pi_{i=1}^{j-J_1} \overset{\circ}{w}_0^{(i)} \right) \xi \cdot x + \left(1 - \delta_{j,J_1} \right) B^{(j)}, \quad \left(H^{(j-J_1)}(x) \right)_1, 0, \left(H^{(j-J_1)}(x) \right)_2, \\ 0, \left(H^{(j-J_1)}(x) \right)_3, 0, \cdots, 0, \left(H^{(j-J_1)}(x) \right)_{D_{j-J_1}}, 0, \cdots, 0 \Big]^T + \delta_{j,J_1} B^{(j)} \mathbf{1}_{d_j}. \tag{4.12}$$

The case $j = J_1$ is trivial by Proposition 2 and the input layer $H^{(0)}(x) = x$ of the teacher net.

Suppose the claim is true for $j-1$ with $j \geq J_1 + 1$. Then we see in either of the two cases $j - 1 = J_1$ and $j - 1 > J_1$ that the first component of $h^{(j-1)}(x)$ is $\left(h^{(j-1)}(x)\right)_1 = \left(\Pi_{i=1}^{j-J_1-1} \overset{\circ}{w}_0^{(i)}\right) \xi \cdot x + B^{(j-1)}$. The matrix form (4.11) tells us that the first row of $T^{(j)}$ has only one nonzero entry, the first entry $\overset{\circ}{w}_0^{(j-J_1)}$, so we know that

$$\left(T^{(j)} h^{(j-1)}(x)\right)_1 = \overset{\circ}{w}_0^{(j-J_1)} \left(\Pi_{i=1}^{j-J_1-1} \overset{\circ}{w}_0^{(i)}\right) \xi \cdot x + \overset{\circ}{w}_0^{(j-J_1)} B^{(j-1)}.$$

From the choice of the bias vector $b^{(j)}$, we find that in both cases $j - 1 = J_1$ and $j - 1 > J_1$, $b_1^{(j)} = \overset{\circ}{w}_0^{(j-J_1)} B^{(j-1)} - B^{(j)}$. It follows that

$$\left(h^{(j)}(x)\right)_1 = \sigma\left(\overset{\circ}{w}_0^{(j-J_1)} \left(\Pi_{i=1}^{j-J_1-1} \overset{\circ}{w}_0^{(i)}\right) \xi \cdot x + B^{(j)}\right) = \left(\Pi_{i=1}^{j-J_1} \overset{\circ}{w}_0^{(i)}\right) \xi \cdot x + B^{(j)},$$

where we have used the fact that $\left|\left(\Pi_{i=1}^{j-J_1} \overset{\circ}{w}_0^{(i)}\right) \xi \cdot x\right| \leq B^{(j)}$. This verifies the first entry of (4.12).

Now we consider the even entries of (4.12). Observe from (4.11) that the $2i$-th row of $T^{(j)}$ has all odd entries to be zero. Then

$$\left(T^{(j)} h^{(j-1)}(x)\right)_{2i} = \sum_{1 \leq k \leq d_{j-1}/2} \overset{\circ}{w}_{i-k}^{(j-J_1)} \left(h^{(j-1)}(x)\right)_{2k}.$$

For $i = 1, 2, \ldots, D_{j-J_1} = d + (j - J_1)S$, we see from $2\left(d + (j - J_1)S\right) = 2d + (j - J_1)s \leq d_{j-1}$ and the hypothesis assumption $\left(h^{(j-1)}(x) - \delta_{j-1,J_1} B^{(j-1)} \mathbf{1}_{d_{j-1}}\right)_{2k} = \left(H^{(j-J_1-1)}\right)_k$ that

$$\begin{aligned}\left(T^{(j)} h^{(j-1)}(x) - b^{(j)}\right)_{2i} &= \left(T^{(j)} \left(h^{(j-1)}(x) - \delta_{j-1,J_1} B^{(j-1)} \mathbf{1}_{d_{j-1}}\right) - \beta^{(j)}\right)_{2i} \\ &= \sum_{k=1}^{i} \overset{\circ}{w}_{i-k}^{(j-J_1)} \left(H^{(j-J_1-1)}\right)_k - \beta_{2i}^{(j)}.\end{aligned}$$

It follows from $\beta_{2i}^{(j)} = \overset{\circ}{b}_i^{(j-J_1)}$ that the $2i$-th entry of $h^{(j)}(x)$ equals

$$\left(h^{(j)}(x)\right)_{2i} = \left(\sigma\left(\overset{\circ}{T}^{(j-J_1)} H^{(j-J_1-1)}(x) - \overset{\circ}{b}^{(j-J_1)}\right)\right)_i$$

which is exactly $\left(H^{(j-J_1)}(x)\right)_i$.

For $i > D_{j-J_1} = d + (j - J_1)S$, we have

$$\left(T^{(j)} h^{(j-1)}(x) - b^{(j)}\right)_{2i} = \sum_{k=1}^{D_{j-J_1}} \overset{\circ}{w}_{i-k}^{(j-J_1)} \left(H^{(j-J_1-1)}\right)_k - \beta_{2i}^{(j)}.$$

The first term on the right-hand side is bounded by $\left\|\overset{\circ}{w}^{(j-J_1)}\right\|_1 \left\|H^{(j-J_1-1)}\right\|_\infty$. Hence by the choice of $\beta_{2i}^{(j)}$ we obtain $\left(h^{(j)}(x)\right)_{2i} = 0$. This verifies (4.12) for the even entries.

As for the other odd entries with indices $2i + 1$ and $i \geq 1$, we notice that the corresponding row of $T^{(j)}$ has all even entries to be zero. But the only nonzero odd entry of $h^{(j-1)} - \delta_{j-1,J_1} B^{(j-1)} \mathbf{1}_{d_{j-1}}$ is the first one. Hence

$$\begin{aligned}\left(T^{(j)} h^{(j-1)}(x) - b^{(j)}\right)_{2i+1} &= \left(T^{(j)} \left(h^{(j-1)}(x) - \delta_{j-1,J_1} B^{(j-1)} \mathbf{1}_{d_{j-1}}\right) - \beta^{(j)}\right)_{2i+1} \\ &= \overset{\circ}{w}_i^{(j-J_1)} \left(\left(\Pi_{i=1}^{j-J_1-1} \overset{\circ}{w}_0^{(i)}\right) \xi \cdot x + \left(1 - \delta_{j-1,J_1}\right) B^{(j-1)}\right) - \beta_{2i+1}^{(j)}.\end{aligned}$$

Note that the first term on the right-hand side is bounded by $2B^{(j)}$. By the choice of $\beta_{2i+1}^{(j)}$, we see that $\left(h^{(j)}(x)\right)_{2i+1} = 0$.

Combining all the above three cases verifies our claim for j, and proves the proposition. □

At the end of the second group of layers, the width is $d_{J_1+J_2} = d_{J_1} + J_2 s$ and

$$h^{(J_1+J_2)}(x) = \left[\left(\Pi_{i=1}^{J_2} \overset{\circ}{w}_0^{(i)}\right) \xi \cdot x + B^{(J_1+J_2)}, \left(H^{(J_2)}(x)\right)_1, 0, \cdots, 0, \left(H^{(J_2)}(x)\right)_{D_{J_2}}, 0, \cdots, 0\right]^T. \tag{4.13}$$

4.3. The third step: expanding DCNN to replicate linear feature

To this end, we have constructed a DCNN of depth $J_1 + J_2$ where the $(J_1 + J_2)$-th layer is given by $h^{(J_1+J_2)}$ in (4.13). For convenience, we denote

$$\widehat{C} = \left[B^{(J_1+J_2)},\ 0, \cdots, 0\right]^T \in \mathbb{R}^{d_{J_1+J_2}}. \tag{4.14}$$

Then the function vector

$$\widehat{U}(x) := h^{(J_1+J_2)}(x) - \widehat{C} \tag{4.15}$$

has the linear feature $\left(\Pi_{i=1}^{J_2} \mathring{w}_0^{(i)}\right) \xi \cdot x$ as the first component and the entries of $H^{(J_2)}(x)$ as some even components. Recall that $\left(\Pi_{i=1}^{J_2} \mathring{w}_0^{(i)}\right) \xi \cdot x \le \left(\Pi_{i=1}^{J_2} \mathring{w}_0^{(i)}\right) B^{(J_1)} = B^{(J_1+J_2)}$. All the components of $\widehat{U}(x)$ are bounded uniformly by

$$\mathcal{B}^{(0)} := \max\left\{B^{(J_1+J_2)}, \left\|H^{(J_2)}\right\|_\infty\right\} > 0. \tag{4.16}$$

In this subsection, we shall apply Proposition 1 to replicate the linear feature $\xi \cdot x$ for N times by expanding our DCNN to have depth $J_1 + J_2 + J_3$, where J_3 is a positive integer to be determined later. To do so, we choose a sequence W so that it induces a convolutional matrix T^W (given in (4.5)) that consists of N blocks of the identity matrix as

$$\begin{bmatrix} I \\ I \\ \vdots \\ I \end{bmatrix}, \tag{4.17}$$

where I denotes the $d_{J_1+J_2} \times d_{J_1+J_2}$ identity matrix. With this matrix, the function vector $T^W \widehat{U}(x)$ contains the linear feature $\xi \cdot x$ in N components and can be later used to generate ridge functions $\sigma\left(\eta^* \xi \cdot x - t_i\right)$ required in our interpolating scheme (4.1). To obtain our desired form of T^W given by (4.17), we can define the sequence W as follows.

Let N be an odd integer to be determined later. Take a sequence W supported on $\{0, \ldots, (N-1)d_{J_1+J_2}\}$ given by

$$W_i = \begin{cases} 1, & \text{if } i \in \{k d_{J_1+J_2}\}_{k=0}^{N-1}, \\ 0, & \text{otherwise.} \end{cases} \tag{4.18}$$

We define its symbol $\widetilde{W}$ as a polynomial on $\mathbb{C}$ given in the wavelet literature [13] by

$$\widetilde{W}(z) = \sum_{j=0}^{\infty} W_j z^j = \sum_{k=0}^{N-1} z^{k d_{J_1+J_2}} = \frac{1 - z^{N d_{J_1+J_2}}}{1 - z^{d_{J_1+J_2}}}, \qquad z \in \mathbb{C}.$$

It has $(N-1)d_{J_1+J_2}$ complex roots

$$e^{i2\pi \frac{\ell + jN}{N d_{J_1+J_2}}}, \qquad \ell = 1, \ldots, N-1,\ j = 0, \ldots, d_{J_1+J_2} - 1$$

appearing in complex conjugate pairs. Applying the procedure for convolutional factorization described in Lemma 1, with an even filter length s, we can find explicit expressions for the filters $\{w^{(j)}\}_{j=J_1+J_2+1}^{J_1+J_2+J_3}$ supported in $\{0, \ldots, s\}$, with $J_3 = \left\lceil \frac{(N-1)d_{J_1+J_2}}{s} \right\rceil$ and $\|w^{(j)}\|_1 \ge 1$ such that

$$W = w^{(J_1+J_2+J_3)} * \cdots * w^{(J_1+J_2+2)} * w^{(J_1+J_2+1)}. \tag{4.19}$$

Recall that $D_{J_2} = d + J_2 S$ and $d_{J_1+J_2} = d_{J_1} + J_2 s \ge 2d + 1 + 2J_2 S > 2D_{J_2}$. For $j > J_1 + J_2$, denote

$$B^{(j)} = \left(\Pi_{p=J_1+J_2+1}^{j} \|w^{(p)}\|_1\right) \mathcal{B}^{(0)}.$$

The following proposition tells us that we can add J_3 well-defined convolutional layers to our previously obtained DCNN to get a DCNN of depth $J_1 + J_2 + J_3$. This DCNN shall output a vector that contains $\xi \cdot x$ in N components.

Proposition 4. *Let $n \in \mathbb{N}$, $N \ge 3n$ be an odd integer, $t_1 < t_2 < \cdots < t_{3n}$, and W be the sequence given by (4.18) with $\mathcal{V} = (N-1)d_{J_1+J_2}$. Take $K = d_{J_1+J_2}$, $J^* = J_3 := \lceil \frac{(N-1)d_{J_1+J_2}}{s-1} \rceil$, the initial layer $\hat{h}^{(0)}(x) = \widehat{U}(x) + \widehat{C}$ (with $\widehat{U}(x)$ given by (4.15), $\widehat{C}$ given by (4.14)) and the constant $\mathcal{B}^{(0)}$ given by (4.16). Construct the CNN layers $\left\{h^{(j)}(x) : \mathbb{R}^d \to \mathbb{R}^{d_j}\right\}_{j=J_1+J_2+1}^{J_1+J_2+J_3}$ as in Proposition 1 with the last bias vector*

$$b^{(J_1+J_2+J_3)} = B^{(J_1+J_2+J_3-1)} T^{(J_1+J_2+J_3)} \mathbf{1}_{d_{J_1+J_2+J_3-1}} + \beta$$

given in terms of the vector $\beta \in \mathbb{R}^{d_{J_1+J_2+J_3}}$ by

$$\left[\beta_{(i-1)d_{J_1+J_2}+1}\right]_{i=1}^{3n} = \left[\overset{*}{w}(t_i)\right]_{i=1}^{3n}$$

and

$$\beta_i = \begin{cases} -B^{(J_1+J_2+J_3)}, & \text{if } i \in \{2k\}_{k=1}^{D_{J_2}}, \\ B^{(J_1+J_2+J_3)}, & \text{if } i \notin \{2k\}_{k=1}^{D_{J_2}} \cup \left\{(i-1)d_{J_1+J_2}+1\right\}_{i=1}^{3n}, \end{cases}$$

where $\overset{*}{w} := \Pi_{i=1}^{J_2} \overset{\circ}{w}_0^{(i)}$. *Then we have*

$$\left(h^{(J_1+J_2+J_3)}(x)\right)_i = \begin{cases} |\overset{*}{w}|\sigma\left(sgn(\overset{*}{w})\xi \cdot x - t_k\right), & \text{if } i = (k-1)d_{J_1+J_2}+1 \text{ with } k \in \{1,\dots,3n\}, \\ \left(H^{(J_2)}(x)\right)_k + B^{(J_1+J_2+J_3)}, & \text{if } i = 2k \text{ with } k \in \{1,\dots,D_{J_2}\}, \\ 0, & \text{otherwise.} \end{cases}$$

There is only one free parameter $B^{(0)} = \max\left\{B^{(J_1+J_2)}, \left\|H^{(J_2)}\right\|_\infty\right\}$ *in this construction.*

Proof. Observe that $B^{(j)} \geq \left(\Pi_{p=J_1+J_2+1}^{j} \|w^{(p)}\|_1\right)\left\|\widehat{h}^{(J_1+J_2)}\right\|_\infty$ for $j = J_1+J_2+1,\dots,J_1+J_2+J_3$.

Consider $\widehat{U}(x) + \widehat{C} = h^{(J_1+J_2)}(x)$ given by (4.13) and the filters $\{w^{(j)}\}_{j=J_1+J_2+1}^{J_1+J_2+J_3}$ explicitly constructed above satisfying (4.18) and (4.19) with N. Define the CNN layers $\left\{h^{(j)}(x) : \mathbb{R}^d \to \mathbb{R}^{d_j}\right\}_{j=J_1+J_2+1}^{J_1+J_2+J_3}$ with these filters and bias vectors are chosen to be

$$b^{(J_1+J_2+1)} = T^{(J_1+J_2+1)}\widehat{C} - B^{(J_1+J_2+1)}\mathbf{1}_{d_{J_1+J_2+1}},$$

$\{b^{(J_1+J_2+j)}\}_{j=2}^{J_3-1}$ given by (4.2) with $D = d_{J_1+J_2}, J^* = J_3, B^{(0)}$ given by (4.16) and $b^{(J_1+J_2+J_3)}$ defined above.

According to Proposition 1, we have

$$T^{(J_1+J_2+J_3)}\left(h^{(J_1+J_2+J_3-1)}(x)\right) = T^W\widehat{U}(x) + B^{(J_1+J_2+J_3-1)}T^{(J_1+J_2+J_3)}\mathbf{1}_{d_{J_1+J_2+J_3-1}}.$$

Putting the choice of $b^{(J_1+J_2+J_3)}$ and the special form (4.17) of the matrix T^W into the above expression, we obtain

$$\begin{aligned} h^{(J_1+J_2+J_3)}(x) &= T^{(J_1+J_2+J_3)}\left(h^{(J_1+J_2+J_3-1)}(x)\right) - b^{(J_1+J_2+J_3)} \\ &= \left[\widehat{U}(x)^T\ \widehat{U}(x)^T\ \cdots\ \widehat{U}(x)^T\right]^T - \beta. \end{aligned}$$

Observe that each component of $\widehat{U}(x)$ is bounded by $B^{(0)} \leq B^{(J_1+J_2+J_3)}$ and $\beta_i = B^{(J_1+J_2+J_3)}$ for $i \notin \{2k\}_{k=1}^{D_{J_2}} \cup \left\{(i-1)d_{J_1+J_2}+1\right\}_{i=1}^{3n}$, we know that $\left(h^{(J_1+J_2+J_3)}(x)\right)_i$ with such an index i is zero.

For $k \in \{1,2,\dots,D_{J_2}\}$, $\left(h^{(J_1+J_2+J_3)}(x)\right)_{2k}$ is

$$\sigma\left(\left(H^{(J_2)}(x)\right)_k + B^{(J_1+J_2+J_3)}\right) = \left(H^{(J_2)}(x)\right)_k + B^{(J_1+J_2+J_3)}.$$

For $i \in \{1,2,\dots,3n\}$, we have

$$\left(h^{(J_1+J_2+J_3)}(x)\right)_{(i-1)d_{J_1+J_2}+1} = |\overset{*}{w}|\sigma\left(\mathrm{sgn}(\overset{*}{w})\xi \cdot x - t_i\right).$$

This verifies the stated expression for $\left(h^{(J_1+J_2+J_3)}(x)\right)_i$. The proof is complete. □

4.4. Achieving interpolations using ridge functions

The hypothesis space induced from the DCNN of depth $J_1+J_2+J_3$ constructed above is

$$\begin{aligned} \mathcal{H}^* &= \mathrm{span}\left\{c \cdot h^{(J_1+J_2+J_3)}(x) + a : c \in \mathbb{R}^{d_{J_1+J_2+J_3}}, a \in \mathbb{R}\right\} \\ &= \left\{\alpha \cdot H^{(J_2)}(x) + \gamma \cdot \left(\sigma\left(\mathrm{sgn}\left(\overset{*}{w}\right)\xi \cdot x - t_k\right)\right)_{k=1}^{3n} + a : \alpha \in \mathbb{R}^{D_{J_2}}, \gamma \in \mathbb{R}^{3n}, a \in \mathbb{R}\right\}. \end{aligned}$$

The number of free parameters of an output function from this hypothesis space equals

$$d_{J_1+J_2+J_3} + 1 + J_1(s+2) + 1 + 1.$$

We are now in the position to prove Theorem 1. Let us first recall what Theorem 1 suggests. Suppose that there is a DCNN $\left\{H^{(j)}(x) : \mathbb{R}^d \to \mathbb{R}^{D_j}\right\}_{j=0}^{J_2}$ of depth J_2 with filter length $S \in \mathbb{N}$ given in Section 4.2 satisfies the following (same as (3.1)):

$$\inf_{f \in H_{J_2,S}} \left\|\pi_M f - f_\rho\right\|_\rho < E_{J_2},$$

where $\{E_{J_2} > 0\}_{J_2 \in \mathbb{N}}$ is an error sequence achieved by this CNN, which may depend on the confidence level. Theorem 1 shows that this bound of excess generalization error can be maintained by some $f \in \mathcal{H}^*$ (DCNN of depth $J_1+J_2+J_3$), while satisfying the interpolation condition (2.12).

Proof of Theorem 1. Since a convolutional neural network depends on the filter sequence continuously, we know that the generalization error bound (3.1) can be achieved by an output function $f^*(x) = c^* \cdot H^{(J_2)}(x) + a^*$ with $c^* \in \mathbb{R}^{D_{J_2}}, a^* \in \mathbb{R}$ produced by a DCNN $\left\{H^{(j)}(x) : \mathbb{R}^d \to \mathbb{R}^{D_j}\right\}_{j=0}^{J_2}$ of depth J_2 satisfying

$$\overset{\circ}{w}{}_0^{(j)} \neq 0, \qquad \forall j = 1, \ldots, J_2.$$

Under this condition, we have $\overset{*}{w} = \Pi_{i=1}^{J_2} \overset{\circ}{w}{}_0^{(i)} \neq 0$.

From the assumption that ρ_Ω has no positive mass at any point $x \in \Omega$, we know that the points in the set $\left\{x^i\right\}_{i=1}^n$ are distinct almost surely.

When $\left\{x^i\right\}_{i=1}^n$ are distinct, we see that for any $i \neq j \in \{1, \ldots, n\}$, $(x^i - x^j)^\perp$ is a subspace of $\mathbb{R}^d$ of co-dimension 1, where for a nonzero vector $\eta \in \mathbb{R}^d$, $\eta^\perp = \{x \in \mathbb{R}^d : \eta \cdot x = 0\}$ denotes the subspace of $\mathbb{R}^d$ perpendicular to η. Also, for each $i \in \{1, \ldots, n\}$, the set $x^i + \eta^\perp$ has ρ_Ω measure 0 for almost every $\eta \in \mathbb{R}^d$ with respect to the Lebesgue measure. Therefore, there exists some nonzero vector $\xi \in \mathbb{R}^d$ such that

$$\operatorname{sgn}(\overset{*}{w})\xi \cdot (x^i - x^j) \neq 0, \qquad \forall i \neq j \in \{1, \ldots, n\}$$

and

$$\rho_\Omega\left(x^i + \left(\operatorname{sgn}(\overset{*}{w})\xi\right)^\perp\right) = 0, \qquad \forall i \in \{1, \ldots, n\}.$$

This implies that the set $\{u_i := \operatorname{sgn}(\overset{*}{w})\xi \cdot x^i\}_{i=1}^n$ contains distinct real numbers.

Now we can construct an interpolator f satisfying the condition $f(x^i) = y^i,\ \forall i = 1, \ldots, n$. Denote

$$\epsilon^* = \min_{i \neq j \in \{1,\ldots,n\}} \left\{\frac{1}{2}\left|\xi \cdot (x^i - x^j)\right|\right\}.$$

For $\epsilon \in (0, \epsilon^*)$, recall the hat function $\phi = \phi^{(\epsilon)} : \mathbb{R} \to \mathbb{R}$ given by

$$\phi(u) = \frac{1}{\epsilon}\left\{\sigma(u + \epsilon) - \sigma(u)) - (\sigma(u) - \sigma(u - \epsilon)\right\}. \qquad u \in \mathbb{R}.$$

It is supported on $[-\epsilon, \epsilon]$ and equal to 1 at 0. The interpolator is given by

$$f(x) = f^*(x) + \sum_{\ell=1}^n \left(y^\ell - f^*(x^\ell)\right)\phi\left(\operatorname{sgn}(\overset{*}{w})\xi \cdot x - u_\ell\right). \tag{4.20}$$

From the definition of the hat function ϕ, we can see that $f \in \mathcal{H}^*$ with $\{t_i : i = 1, \ldots, 3n\} = \left\{u_\ell - \epsilon : \ell = 1, \ldots, n\right\} \cup \left\{u_\ell : \ell = 1, \ldots, n\right\} \cup \left\{u_\ell + \epsilon : \ell = 1, \ldots, n\right\}$.

The function f is an interpolator satisfying $f(x^i) = y^i\ \forall i$ and thereby lies in $\mathcal{H}_{int, J_1+J_2+J_3, S}$ because for each $i \in \{1, \ldots, n\}$, we have $\operatorname{sgn}(\overset{*}{w})\xi \cdot x^i = u_i$ implying that

$$\phi\left(\operatorname{sgn}(\overset{*}{w})\xi \cdot x^i - u_\ell\right) = \phi\left(u_i - u_\ell\right) = \begin{cases} 1 & \text{if } i = \ell, \\ 0 & \text{if } i \neq \ell, \end{cases} \qquad \ell \in \{1, \ldots, n\}.$$

Hence, we have $f(x^i) = f^*(x^i) + y^i - f^*(x^i) = y^i$.

Finally, we estimate the excess generalization error of $\pi_M f$. Notice that the function $\phi\left(\operatorname{sgn}(\overset{*}{w})\xi \cdot x - u_\ell\right)$ is zero unless $\operatorname{sgn}(\overset{*}{w})\xi \cdot x - u_\ell \in (-\epsilon, \epsilon)$. It follows that the function

$$\sum_{\ell=1}^n \left(y^\ell - f^*(x^\ell)\right)\phi\left(\operatorname{sgn}(\overset{*}{w})\xi \cdot x - u_\ell\right)$$

on the right-hand side of (4.20) is supported on the set

$$X_\epsilon := \cup_{\ell=1}^n \left\{x \in \mathbb{R}^d : \left|\operatorname{sgn}(\overset{*}{w})\xi \cdot (x - x^\ell)\right| < \epsilon\right\}.$$

Hence we find from the expression (4.20) of f that $\pi_M f(x) = \pi_M f^*(x)$ for $x \notin X_\epsilon$ while $|\pi_M f(x) - \pi_M f^*(x)| \leq 2M$ for $x \in X_\epsilon$. Thus, we have

$$\begin{aligned} \left\|\pi_M f - \pi_M f^*\right\|_\rho &= \left\{\int_{X_\epsilon} \left|\pi_M f(x) - \pi_M f^*(x)\right|^2 d\rho_\Omega\right\}^{1/2} \leq 2M\left\{\rho_\Omega\left(X_\epsilon\right)\right\}^{1/2} \\ &\to 2M\left\{\rho_\Omega\left(\cup_{\ell=1}^n\left\{x^\ell + \left(\operatorname{sgn}(\overset{*}{w})\xi\right)^\perp\right\}\right)\right\}^{1/2} = 0 \end{aligned}$$

as $\epsilon \to 0^+$. Then, our estimate follows from the triangle inequality $\left\|\pi_M f - f_\rho\right\|_\rho \le \left\|\pi_M f - \pi_M f^*\right\|_\rho + \left\|\pi_M f^* - f_\rho\right\|_\rho$ and (3.1) for f^*. The proof of Theorem 1 is complete. □

Now that we completed the proof of Theorem 1, let us move on to the proof of Theorem 2.

5. Learning rates of DCNNs

In this section, we derive learning rates of DCNNs for proving Theorem 2. In other words, we establish an upper bound of the excess error $\varepsilon(\pi_M f_{D,J,S}) - \varepsilon(f_\rho) = \left\|\pi_M f_{D,J,S} - f_\rho\right\|_2^2$. Our analysis is based on an error decomposition (Subsection 5.1), followed by bounding the approximation error and the sampling error, respectively (Subsection 5.2). Afterward, we bound the number of layers and achieved the desired learning rates (Subsection 5.3). Our method is different from the approaches in the existing literature of generalization analysis of DCNNs [24,14]. By using a pseudo-dimension estimate of the induced hypothesis space, we no longer require the free parameters to be bounded. Lastly, by applying the results in Theorem 2, we are able to derive the proof of Theorem 3 (Subsection 5.4).

Note that the filter length specified in Theorem 2 is $2 \le S \le d$.

5.1. Error decomposition and estimating the approximation error

We aim to find the upper bounds of the excess error:

$$\varepsilon(\pi_M f_{D,J,S}) - \varepsilon(f_\rho) = \left\|\pi_M f_{D,J,S} - f_\rho\right\|_2^2 = \int_Z (\pi_M f_{D,J,S}(x) - f_\rho(x))^2 \, d\rho,$$

where $\pi_M f_{D,J,S}$ is the truncated global minimizer of (3.3) and f_ρ is the regression function. Let $f_{\mathcal{H}_{J,S}}$ be any function in the hypothesis space $\mathcal{H}_{J,S}$ defined by (2.5). To find such an upper bound, we adopt the following error decomposition.

Lemma 3. *Let $f_{D,J,S}$ be defined by (3.3) and $f_{\mathcal{H}_{J,S}}$ be any function in the hypothesis space $\mathcal{H}_{J,S}$ defined by (2.5). Then there holds*

$$\begin{aligned}\varepsilon(\pi_M f_{D,J,S}) - \varepsilon(f_\rho) &\le \left\{(\varepsilon(\pi_M f_{D,J,S}) - \varepsilon(f_\rho)) - (\varepsilon_D(\pi_M f_{D,J,S}) - \varepsilon_D(f_\rho))\right\} \\ &+ \left\{(\varepsilon_D(f_{\mathcal{H}_{J,S}}) - \varepsilon_D(f_\rho)) - (\varepsilon(f_{\mathcal{H}_{J,S}}) - \varepsilon(f_\rho))\right\} + \left\{\varepsilon(f_{\mathcal{H}_{J,S}}) - \varepsilon(f_\rho)\right\} =: \mathcal{A}_1 + \mathcal{A}_2 + \mathcal{A}_3.\end{aligned} \tag{5.1}$$

Proof. We express the excess error by inserting empirical error terms as follows

$$\begin{aligned}\varepsilon(\pi_M f_{D,J,S}) - \varepsilon(f_\rho) &= \{\varepsilon(\pi_M f_{D,J,S}) - \varepsilon(f_\rho)\} - \{\varepsilon_D(\pi_M f_{D,J,S}) - \varepsilon_D(f_\rho)\} \\ &+ \{\varepsilon_D(\pi_M f_{D,J,S}) - \varepsilon_D(f_\rho)\} - \{\varepsilon_D(f_{\mathcal{H}_{J,S}}) - \varepsilon_D(f_\rho)\} \\ &+ \{\varepsilon_D(f_{\mathcal{H}_{J,S}}) - \varepsilon_D(f_\rho)\} - \{\varepsilon(f_{\mathcal{H}_{J,S}}) - \varepsilon(f_\rho)\} + \{\varepsilon(f_{\mathcal{H}_{J,S}}) - \varepsilon(f_\rho)\}.\end{aligned}$$

As both y^i and $\pi_M f_{D,J,S}(x^i)$ are values on the interval $[-M, M]$ for each i, we know that $\left(\pi_M f_{D,J,S}(x^i) - y^i\right)^2 \le \left(f_{D,J,S}(x^i) - y^i\right)^2$ which yields $\varepsilon_D(\pi_M f_{D,J,S}) \le \varepsilon_D(f_{D,J,S})$. Note that both $f_{D,J,S}$ and $f_{\mathcal{H}_{J,S}}$ lie in the space $\mathcal{H}_{J,S}$. By definition, $f_{D,J,S}$ minimizes the empirical risk $\varepsilon_D(f)$ over $\mathcal{H}_{J,S}$. It follows that

$$\varepsilon_D(\pi_M f_{D,J,S}) - \varepsilon_D(f_\rho) - \{\varepsilon_D(f_{\mathcal{H}_{J,S}}) - \varepsilon_D(f_\rho)\} = \varepsilon_D(\pi_M f_{D,J,S}) - \varepsilon_D(f_{\mathcal{H}_{J,S}}) \le 0.$$

The expression (5.1) is verified. □

To achieve the learning rates stated in Theorem 2, we are going to bound the three terms, $\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3$, on the right-hand side of (5.1) in the following.

The last term of (5.1), $\mathcal{A}_3 = \varepsilon(f_{\mathcal{H}_{J,S}}) - \varepsilon(f_\rho) = \left\|f_{\mathcal{H}_{J,S}} - f_\rho\right\|_2^2 \le \left\|f_{\mathcal{H}_{J,S}} - f_\rho\right\|_{C(\Omega)}^2$, is known as the approximation error, where the last inequality holds due to the fact that the marginal distribution ρ_Ω has measure 1. It has been estimated in [33, Theorem 2] as follows.

Lemma 4. *Let $2 \le S \le d$ and $\Omega \subseteq [-1,1]^d$. If $J \ge 2d/(S-1)$ and $f_\rho = F|_\Omega$ with $F \in W_2^r(\mathbb{R}^d)$ and an integer index $r > 2 + d/2$, then there exist $\mathbf{w}$, $\mathbf{b}$ with (2.6) valid for layers $1, 2, \dots J-1$ and $f_{\mathcal{H}_{J,S}} \in \mathcal{H}_{J,S}$ such that*

$$\mathcal{A}_3 \le \|f_{\mathcal{H}_{J,S}} - f_\rho\|_{C(\Omega)} \le c\, \|F\|_{W_2^r} \sqrt{\log J}\, J^{-\frac{1}{2} - \frac{1}{d}}, \tag{5.2}$$

where c is an absolute constant and $\|F\|_{W_2^r}$ denotes the Sobolev space norm of F.

The second term of (5.1), $\mathcal{A}_2 = \varepsilon_D(f_{\mathcal{H}_{J,S}}) - \varepsilon_D(f_\rho) - \{\varepsilon(f_{\mathcal{H}_{J,S}}) - \varepsilon(f_\rho)\}$, can be estimated by the Bernstein inequality [7] (see, e.g., [9, Lemma A.2]).

Lemma 5. *For any $0 < \delta < 1$, with probability at least $1 - \frac{\delta}{2}$, we have*

$$\begin{aligned}\mathcal{A}_2 &= \{\varepsilon_D(f_{\mathcal{H}_{J,S}}) - \varepsilon_D(f_\rho)\} - \{\varepsilon(f_{\mathcal{H}_{J,S}}) - \varepsilon(f_\rho)\} \\ &\le 4\log\left(\frac{2}{\delta}\right)\left\|f_{\mathcal{H}_{J,S}} - f_\rho\right\|_\infty \left(\left\|f_{\mathcal{H}_{J,S}} - f_\rho\right\|_\infty + 4M\right)/\sqrt{n}.\end{aligned}$$

Proof. Define a random variable ζ on (Z,ρ) by $\zeta(z) = \zeta(x,y) = (f_{\mathcal{H}_{J,S}}(x) - y)^2 - (f_\rho(x) - y)^2$. We have $\varepsilon(f_{\mathcal{H}_{J,S}}) - \varepsilon(f_\rho) = \mathrm{E}[\zeta]$ and $\varepsilon_D(f_{\mathcal{H}_{J,S}}) - \varepsilon_D(f_\rho) = \frac{1}{n}\sum_{i=1}^n \zeta(z^i)$. Hence, the second term of (5.1) can be expressed in terms of ζ as

$$\{\varepsilon_D(f_{\mathcal{H}_{J,S}}) - \varepsilon_D(f_\rho)\} - \{\varepsilon(f_{\mathcal{H}_{J,S}}) - \varepsilon(f_\rho)\} = \frac{1}{n}\sum_{i=1}^n \zeta(z^i) - \mathrm{E}[\zeta]. \tag{5.3}$$

Note that $|f_\rho(x) - y| \le 2M$ because $|y| \le M$ and $|f_\rho(x)| = |\mathrm{E}[y|x]| \le M$. We have almost surely

$$\begin{aligned}|\zeta(z)| &= \left|\left(f_{\mathcal{H}_{J,S}}(x) - f_\rho(x)\right)\left(f_{\mathcal{H}_{J,S}}(x) - y + f_\rho(x) - y\right)\right| \\ &\le \left\|f_{\mathcal{H}_{J,S}} - f_\rho\right\|_\infty \left(\left\|f_{\mathcal{H}_{J,S}} - f_\rho\right\|_\infty + 4M\right) =: B.\end{aligned}$$

The variance σ^2 of ζ can be bounded as $\sigma^2 \le \mathrm{E}[\zeta^2] \le B^2$. Thus, by the one-sided Bernstein inequality, for any $\eta > 0$, there holds

$$\{\varepsilon_D(f_{\mathcal{H}_{J,S}}) - \varepsilon_D(f_\rho)\} - \{\varepsilon(f_{\mathcal{H}_{J,S}}) - \varepsilon(f_\rho)\} \le \eta$$

with probability at least $1 - \exp\left(-\frac{n\eta^2}{2(\sigma^2 + \eta B/3)}\right)$. Setting this confidence bound to be $1 - \frac{\delta}{2}$, we get a quadratic equation $\frac{n\eta^2}{2(\sigma^2+\eta B/3)} = \log\frac{2}{\delta}$ for η. The positive solution η^* to this equation is given by

$$\begin{aligned}\eta^* &= \frac{\frac{2B}{3}\log\left(\frac{2}{\delta}\right) + \sqrt{\left(\frac{2B}{3}\log\left(\frac{2}{\delta}\right)\right)^2 + 8n\sigma^2\log\left(\frac{2}{\delta}\right)}}{2n} \\ &\le \frac{2B\log\left(\frac{2}{\delta}\right)}{3n} + \frac{\sqrt{2\sigma^2\log\left(\frac{2}{\delta}\right)}}{\sqrt{n}} \\ &\le \frac{4B\log\left(\frac{2}{\delta}\right)}{\sqrt{n}}.\end{aligned}$$

This verifies the desired bound and completes the proof. □

5.2. Estimating the sample error

The first term of (5.1), $\mathcal{A}_1 = (\varepsilon(\pi_M f_{D,J,S}) - \varepsilon(f_\rho)) - (\varepsilon_D(\pi_M f_{D,J,S}) - \varepsilon_D(f_\rho))$, is the sample error. In this subsection, we derive an upper bound for this sampling error using a concentration inequality in terms of the empirical covering numbers. Define the empirical L_1 norm with respect to the sample D by

$$\|f\|_{L_1(D)} = \frac{1}{n}\sum_{i=1}^n |f(x^i)|. \tag{5.4}$$

For $\epsilon > 0$, denote by $\mathcal{N}_1(\epsilon, \mathcal{H}, D)$ the ϵ-empirical covering number of a set of functions $\mathcal{H}$ with respect to $\|\cdot\|_{L_1(D)}$. More specifically, $\mathcal{N}_1(\epsilon, \mathcal{H}, D)$ is the minimal $N \in \mathbb{N}$ such that there exist functions $\{f_1, \dots, f_N\} \in \mathcal{H}$ satisfying

$$\min_{1\le j\le N} \|f - f_j\|_{L_1(D)} \le \epsilon, \qquad \forall f \in \mathcal{H}. \tag{5.5}$$

The following concentration inequality with arbitrary parameters $\alpha, \beta > 0$ and $0 < \varepsilon \le 1/2$ can be found in [17] as Theorem 11.4.

Lemma 6. *Assume $|y| \le M$ almost surely and $M \ge 1$. Let $\alpha, \beta > 0$ and $0 < \varepsilon \le 1/2$. If $\mathcal{H}$ is a set of functions $f : \mathbb{R}^d \to [-M, M]$, then with probability at least*

$$1 - 14\sup_D \mathcal{N}_1\left(\frac{\beta\varepsilon}{20M}, \mathcal{H}, D\right)\exp\left(-\frac{\varepsilon^2(1-\varepsilon)\alpha n}{214(1+\varepsilon)M^4}\right), \tag{5.6}$$

there holds

$$\left\{\varepsilon(f) - \varepsilon(f_\rho)\right\} - \left\{\varepsilon_D(f) - \varepsilon_D(f_\rho)\right\} \le \varepsilon\left(\alpha + \beta + \varepsilon(f) - \varepsilon(f_\rho)\right), \qquad \forall f \in \mathcal{H}.$$

We shall apply Lemma 6 to the function set $\mathcal{H} = \{\pi_M f : f \in \mathcal{H}_{J,S}\}$. To this end, we need to derive a bound for its empirical covering number by means of its empirical packing number and pseudo-dimension.

Denote by $\mathcal{M}_1(\epsilon, \mathcal{H}, D)$ the ϵ-empirical packing number of a set of functions $\mathcal{H}$ with respect to $\|\cdot\|_{L_1(D)}$. More specifically, $\mathcal{M}_1(\epsilon, \mathcal{H}, D)$ is the maximal $N \in \mathbb{N}$ such that there exists functions $\{f_1, \ldots, f_N\} \in \mathcal{H}$ satisfying

$$\|f_j - f_k\|_{L_1(D)} \geq \epsilon, \qquad 1 \leq j < k \leq N.$$

A well-known relationship between ϵ-covering numbers and ϵ-packing numbers asserts that

$$\mathcal{M}_1(2\epsilon, \mathcal{H}, D) \leq \mathcal{N}_1(\epsilon, \mathcal{H}, D) \leq \mathcal{M}_1(\epsilon, \mathcal{H}, D), \qquad \forall \epsilon > 0. \tag{5.7}$$

For a class of real-valued functions, such as those generated by neural networks, a natural measure of complexity that implies similar uniform convergence properties is the pseudo-dimension, introduced by [29] (see, e.g., [1, Theorem 19.2]). The pseudo-dimension $Pdim(\mathcal{H})$ of $\mathcal{H}$ is the largest integer ℓ for which there exist $\{\xi_1, \ldots, \xi_\ell\} \subset \Omega$ and $\{\eta_1, \ldots, \eta_\ell\} \subset \mathbb{R}^\ell$ such that for any $(a_1, \ldots, a_\ell) \in \{0,1\}^\ell$ there is some $v \in \mathcal{H}$ satisfying

$$v(\xi_i) > \eta_i \Leftrightarrow a_i = 1, \qquad \forall\, i.$$

For a set $\mathcal{H}$ of functions from Ω to $[-M, M]$, the empirical packing numbers can be bounded in terms of the pseudo-dimension by the following inequality found in [25, Theorem 1]:

$$\mathcal{M}_1(\epsilon, \mathcal{H}, D) \leq 2\left(\frac{2eM}{\epsilon}\log\left(\frac{2eM}{\epsilon}\right)\right)^{Pdim(\mathcal{H})}, \tag{5.8}$$

for any $0 < \epsilon \leq M$. Observe that $Pdim(\{\pi_M f : f \in \mathcal{H}_{J,S}\}) \leq Pdim(\mathcal{H}_{J,S})$.

The last tool we use is an important bound for the pseudo-dimension of a deep ReLU neural network, which can be found in [3, Theorem 7]. This result is valid only for piecewise polynomial activation functions but is crucial in our study with the ReLU activation function, which is piecewise linear. It asserts that for a ReLU neural network architecture with U computation units (neurons) arranged in L layers, the pseudo-dimension of the hypothesis space is bounded by $c\sum_{i=1}^{L} W_i \log U$, where c is an absolute constant and W_i is the total number of parameters (weights and biases) at the inputs to units in *all the layers up to layer* i. We find by the same proof that this bound still holds when we replace the parameters in weights and biases by *free parameters*. This bound may be extended to some other neural networks with special structures. We state it as a theorem below (Theorem 4), of independent interest, and give detailed proof in Appendix A.2.

Motivated by convolutional neural networks, which have many weight entries to be 0 and many repeated weight and bias entries, we allow some of the parameters to be constants. We also allow some parameters within the same layer to take the same free parameter value. Moreover, unlike the result in [3, Theorem 7], which allows neurons to have connections from any earlier layers [16], we assume that each neuron has connections only from neurons in the previous layer. Recall that for a DCNN, a weight matrix is a convolutional one in which each descending diagonal from left to right shares the same value (refer to (2.3) and (4.5)). Also, a bias vector has identical entries in the middle (refer to (2.6)).

Denote by d_j the width, K_j the number of free parameters, $A^{(j)} \in \mathbb{R}^{K_j}$ the vector of free parameters in the j-th layer of the neural network. For the special case of CNN, $d_j = d + js$ and $K_j = 3s$, which consists of $s+1$ free parameters in $T^{(j)}$ and $2s-1$ in $b^{(j)}$. Also, denote by K the total number of free parameters in the neural network. The following theorem considers neural networks equipped with piecewise polynomial activation functions with $p+1$ pieces.

Theorem 4. *Let $\theta, p \in \mathbb{N}$ and $t_1 < t_2 < \cdots < t_p$ be a sequence of break points such that $\sigma : \mathbb{R} \to \mathbb{R}$ is continuous and $\sigma|_{(t_{i-1}, t_i)}$ is a polynomial of degree at most θ for each $i = 1, \ldots, p, p+1$ (where we set $t_0 = -\infty$ and $t_{p+1} = \infty$). Let $J \in \mathbb{N}$, $d_j, K_j \in \mathbb{N}$ for $j \in \{1, \ldots, J\}$, $A^{(j)} \in \mathbb{R}^{K_j}$, and $c \in \mathbb{R}^{d_J}$. Denote $a = [A^{(1)}, \ldots, A^{(J)}, c] \in \mathbb{R}^K$ with $K := K_1 + \ldots + K_J + d_J$. Consider a neural network $\{h^{(j)}(\cdot, a) : \mathbb{R}^d \to \mathbb{R}^{d_j}\}_{j=1}^J$ of depth J and widths $\{d_j\}_{j=1}^J$ defined with $a \in \mathbb{R}^K$ by $h^{(0)}(x, a) = x$ and iteratively*

$$h^{(j)}(x, a) = \sigma\left(T^{(j)} h^{(j-1)}(x, a) - b^{(j)}\right), \qquad j = 1, \ldots, J, \tag{5.9}$$

where $T^{(j)} \in \mathbb{R}^{d_j \times d_{j-1}}, b^{(j)} \in \mathbb{R}^{d_j}$. Here, the entries of $T^{(j)}$ and $b^{(j)}$ can be divided into a collection of K'_j disjoint subsets with $K'_j \geq K_j$. For subset ℓ, $\ell = 1, \ldots, K_j$, all entries in this subset equal to the ℓ-th component $A^{(j)}_\ell$ of $A^{(j)}$. For $\ell > K_j$, all entries in subset ℓ are equal to a fixed constant in $\mathbb{R}$. The hypothesis space generated by this network is

$$\mathcal{H}_{J,\sigma} = span\left\{f(x, a) = c \cdot h^{(J)}(x, a) : a \in \mathbb{R}^K\right\}. \tag{5.10}$$

Then we have

$$Pdim(\mathcal{H}_{J,\sigma}) \leq J + 1 + \left(d_J + \sum_{j=1}^{J}(J - j + 2)K_j\right)\left(\log_2(4eR) + \log_2(\log_2(2eR))\right), \tag{5.11}$$

where

$$R := 1 + \theta + \theta^2 + \cdots + \theta^J + \sum_{i=1}^{J} d_i p(1 + \theta + \theta^2 + \cdots + \theta^{i-1}).$$

Now we apply this result to our DCNN (equipped with ReLU) given in Definition 1 with the bias vectors satisfying (2.6) for layers $1, 2, \ldots J-1$. The proof of Lemma 7 is given in Appendix A.3.

Lemma 7. *For the DCNN of depth $J \geq 2$ in Definition 1 with ReLU activation functions and (2.6) valid for layers $1, 2, \ldots J-1$, we have*

$$Pdim(\mathcal{H}_{J,S}) \leq c_0(J^2S + d)\log(Jd + J^2S), \tag{5.12}$$

where c_0 is an absolute constant.

We aim to find an upper bound on the first term of (5.1) as tightly as possible. In other words, we search for the best α, β, and ε to minimize the upper bound in the concentration inequality presented in Lemma 6.

The next lemma presents an upper bound.

Lemma 8. *Assume $|y| \leq M$ almost surely and $M \geq 1$. For the DCNN of depth $J \geq 2$ in Definition 1 with (2.6) valid for layers $1, 2, \ldots J-1$, and any $0 < \delta < 1$, with probability at least $1 - \frac{\delta}{2}$, we have*

$$\begin{aligned}\mathcal{A}_1 &= \{\varepsilon(\pi_M f_{D,J,S}) - \varepsilon(f_\rho)\} - \{\varepsilon_D(\pi_M f_{D,J,S}) - \varepsilon_D(f_\rho)\} \\ &\leq \frac{1}{2}\Big\{\frac{1}{n} + \{\varepsilon(\pi_M f_{D,J,S}) - \varepsilon(f_\rho)\} + \frac{c_0' M^4}{n}\left\{\log(M^2 n)\left(J^2S + d\right)\log\left(Jd + J^2S\right) + \log(2/\delta)\right\}\Big\},\end{aligned}$$

where c_0' is an absolute constant.

Proof. Take the function set $\mathcal{H} = \left\{\pi_M f : f \in \mathcal{H}_{J,S}\right\}$. Combining (5.7), (5.8) with Lemma 7, for any $\epsilon > 0$, we have

$$\begin{aligned}\mathcal{N}_1(\epsilon, \mathcal{H}, D) &\leq 2\left(\frac{2eM}{\epsilon}\log\frac{2eM}{\epsilon}\right)^{c_0(J^2S+d)\log(Jd+J^2S)} \\ &\leq 2\left(\frac{2eM}{\epsilon}\right)^{2c_0(J^2S+d)\log(Jd+J^2S)}.\end{aligned}$$

Take $\epsilon = \frac{1}{2}$ and $\beta = \frac{1}{n}$ in Lemma 6. Applying the above bound, we see that the confidence bound in Lemma 6 is

$$\begin{aligned}&1 - 14\sup_D \mathcal{N}_1\left(\frac{1}{40Mn}, \mathcal{H}, D\right)\exp\left(-\frac{\alpha n}{2568M^4}\right) \\ &\geq 1 - 28\exp\left(2c_0\log(80eM^2n)\left(J^2S + d\right)\log\left(Jd + J^2S\right) - \frac{\alpha n}{2568M^4}\right).\end{aligned}$$

We choose $\alpha > 0$ such that the above confidence bound is at least $1 - \frac{\delta}{2}$, which means

$$\alpha \geq \frac{2568M^4}{n}\left(2c_0\log(80eM^2n)\left(J^2S + d\right)\log\left(Jd + J^2S\right) + \log\left(\frac{56}{\delta}\right)\right). \tag{5.13}$$

This is satisfied by taking $c_0' = 2^{15}(c_0 + 1)$ and

$$\alpha^* = \frac{c_0' M^4}{n}\left(\log(M^2n)\left(J^2S + d\right)\log\left(Jd + J^2S\right) + \log\left(\frac{2}{\delta}\right)\right), \tag{5.14}$$

because α^* is greater than the right hand side of (5.13).

Substituting α, β and ϵ by the values we took into the concentration inequality in Lemma 6, we know that with probability at least $1 - \delta/2$, there holds

$$\left\{\varepsilon(f) - \varepsilon(f_\rho)\right\} - \left\{\varepsilon_D(f) - \varepsilon_D(f_\rho)\right\} \leq \frac{1}{2}\left(\alpha^* + \frac{1}{n} + \varepsilon(f) - \varepsilon(f_\rho)\right), \qquad \forall f \in \mathcal{H},$$

which implies

$$\{\varepsilon(\pi_M f_{D,J,S}) - \varepsilon(f_\rho)\} - \{\varepsilon_D(\pi_M f_{D,J,S}) - \varepsilon_D(f_\rho)\} \leq \frac{1}{2}\left(\alpha^* + \frac{1}{n} + \varepsilon(\pi_M f_{D,J,S}) - \varepsilon(f_\rho)\right).$$

We verify the bound and complete the proof. □

5.3. Proof of Theorem 2: achieving learning rates by bounding the number of layers

We are now in a position to prove Theorem 2.

Proof of Theorem 2. Recall the number of free parameters given by (2.10). Since $S \le d$, we know that this number can be bounded by

$$3S(J-1)+S+2+2(d+JS) = S(3J-3+1+2J)+2+2d = S(5J-2)+2d+2 \le d(5J-2)+2d+2 = 5dJ+2.$$

Combining Lemmas 3, 4, 5 and 8, when $J \ge 2d/(S-1)$, we obtain the following bound for the excess error in terms of the number of layers J with probability at least $1-\delta$:

$$\begin{aligned}
\varepsilon(\pi_M f_{D,J,S}) - \varepsilon(f_\rho) \le{}& \frac{1}{2}\left\{\varepsilon(\pi_M f_{D,J,S}) - \varepsilon(f_\rho)\right\} \\
&+ \frac{1}{2n} + \frac{c_0' M^4}{2n}\left(\log(M^2 n)\left(J^2 S + d\right)\log\left(Jd + J^2 S\right) + \log(2/\delta)\right) \\
&+ \frac{4}{\sqrt{n}}\log\left(\frac{2}{\delta}\right)(4M+1)c^2\|F\|_{W_2^r}^2 \log J J^{-1-\frac{2}{d}} \\
&+ c^2\|F\|_{W_2^r}^2 \log J J^{-1-\frac{2}{d}},
\end{aligned}$$

which implies

$$\begin{aligned}
\varepsilon(\pi_M f_{D,J,S}) - \varepsilon(f_\rho) &\le \frac{1}{n} + \frac{4c_0' M^4 \log(M+1)d(\log d)}{n}((\log n)J^2(\log J) + \log(2/\delta)) \\
&\quad + \left(\frac{8}{\sqrt{n}}\log(2/\delta)(4M+1)+2\right)c^2\|F\|_{W_2^r}^2 (\log J)J^{-1-\frac{2}{d}} \\
&\le \frac{1}{n} + 4c_0' M^4 \log(M+1)d(\log d)\left(\frac{(\log n)J^2(\log J)}{n} + \frac{1}{\sqrt{n}}\right)\left(1 + \frac{\log(2/\delta)}{\sqrt{n}}\right) \\
&\quad + \left(\frac{\log(2/\delta)}{\sqrt{n}}\right)(32M+10)c^2\|F\|_{W_2^r}^2\left(\frac{\log J}{J}\right) \\
&\le \left(1 + 4c_0' M^4 \log(M+1)d(\log d) + (32M+10)c^2\|F\|_{W_2^r}^2\right)\left(1 + \frac{\log(2/\delta)}{\sqrt{n}}\right) \\
&\quad \left\{\frac{(\log n)J^2(\log J)}{n} + \frac{1}{\sqrt{n}} + \frac{\log J}{J}\right\}.
\end{aligned}$$

This verifies the stated learning rates in terms of J at (3.5) with the constant C_{M,r,f_ρ} given by

$$C_{M,r,f_\rho} = 1 + 4c_0' M^4 \log(M+1)d(\log d) + (32M+10)c^2\|F\|_{W_2^r}^2.$$

We choose $J = \lceil n^\alpha \rceil$. When $n \ge (2d)^{1/\alpha}$, we see that $J \ge 2d/(S-1)$ is satisfied and thereby with probability at least $1-\delta$,

$$\varepsilon(\pi_M f_{D,J,S}) - \varepsilon(f_\rho) \le 6C_{M,r,f_\rho}\left(1 + \frac{\log(2/\delta)}{\sqrt{n}}\right)\frac{(\log n)^2}{n^{\min\{1-2\alpha,\alpha\}}}.$$

This verifies the stated learning rates in (3.6) with the constant $6C_{M,r,f_\rho}$.

When $n < (2d)^{1/\alpha}$, we have $\frac{(\log n)^2}{n^\alpha} \ge \frac{1}{2d}$. Then we can use a simple bound $|\pi_M f_{D,J,S}(x) - y| \le 2M$ and its consequence

$$\varepsilon(\pi_M f_{D,J,S}) - \varepsilon(f_\rho) \le \varepsilon(\pi_M f_{D,J,S}) \le 4M^2,$$

and find that with probability at least $1-\delta$,

$$\varepsilon(\pi_M f_{D,J,S}) - \varepsilon(f_\rho) \le 8M^2 d(\log d)\frac{(\log n)^2}{n^{\min\{1-2\alpha,\alpha\}}}.$$

This also verifies the stated learning rates at (3.6) with the constant $8M^2$. The proof of Theorem 2 is complete. □

5.4. *Proof of Theorem 3: learning rate of interpolating DCNN*

Combining Theorem 1 and Theorem 2, we can derive the learning rates of overparameterized DCNNs that interpolate any input data.

Proof of Theorem 3. Take $S = s/2$, and the DCNN defined in Theorem 2 as a teacher DCNN here. It has $J_2 = \lceil n^\alpha \rceil$ layers for any $0 < \alpha < 1$, with filter length $S \in \{2, 3, \dots, d\}$. Theorem 2 states that, with probability $1-\delta$ for any $0 < \delta < 1$, a truncated output function $\pi_M f_{D,J,S}$ can approximate the regression function f_ρ with accuracy

$$\left\|\pi_M f_{D,J,S} - f_\rho\right\|_2^2 \le \max\left\{8M^2, 6C_{M,r,f_\rho}\right\} d(\log d)\left(1+\frac{\log(2/\delta)}{\sqrt{n}}\right)\frac{(\log n)^2}{n^{\min\{1-2\alpha,\alpha\}}}.$$

We choose $N = 4n+1$ as an odd number satisfying $N \ge 3n$. Given the learning ability of $\pi_M f_{D,J,S}$, Theorem 1 suggests that there exists a downsampled DCNN (student net) of depth $J_1 + J_2 + J_3$ with filter length $2S = s$, downsampled at the J_1-th layer, where $J_1 = \lceil \frac{2d^2}{s-1} \rceil$, $J_3 = \left\lceil \frac{(N-1)d_{J_1+J_2}}{s} \right\rceil = \left\lceil \frac{4nd_{J_1+J_2}}{s} \right\rceil = \left\lceil \frac{4n\left(d_{J_1}+s\lceil n^\alpha \rceil\right)}{s} \right\rceil$, such that with probability $1-\delta$,

$$\inf_{f\in\mathcal{H}_{int}} \left\|\pi_M f - f_\rho\right\|_2^2 < \max\left\{8M^2+1, 6C_{M,r,f_\rho}+1\right\} d(\log d)\left(1+\frac{\log(2/\delta)}{\sqrt{n}}\right)\frac{(\log n)^2}{n^{\min\{1-2\alpha,\alpha\}}}. \tag{5.15}$$

The proof of Theorem 3 is completed. □

6. Numerical experiment

In this section, we present our results of numerical experiments on simulated data to corroborate our theoretical findings in Theorem 2. Our purpose is to demonstrate our theoretical findings. We do not intend to perform numerical experiments with real data.

We conduct numerical experiments for four different input data dimensions $d \in \{10, 30, 50, 100\}$. For each d, we simulate n training data sets $\{(x^i, y^i)\}_{i=1}^n$ with n varying in $\{100, 300, 500, 700, 1000, 1500, 2000, 3000, 4000, 5000, 6000\}$, $x^i \in \mathbb{R}^d$ being a random vector with entries uniformly distributed in $[-10, 10]$, and $y^i \in \mathbb{R}$ generated from the following regression model:

$$y^i = \sin(\|x^i\|_2^4) + \cos(\|x^i\|_2^4) + \epsilon^i, \tag{6.1}$$

where ϵ^i is random noise following $\mathcal{N}(0, 0.01)$. Corresponding to each training data set, we simulate a test data set $\{(x^i_{\text{test}}, y^i_{\text{test}})\}_{i=1}^{2000}$ where $x^i_{\text{test}} \in \mathbb{R}^d$ has entries uniformly distributed in $[-10, 10]$ and $y^i_{\text{test}} = \sin(\|x^i_{\text{test}}\|_2^4) + \cos(\|x^i_{\text{test}}\|_2^4)$. This choice of the regression function is bounded in $[-2, 2]$.

To run the experiments, we adopt the DCNNs following the structure defined in Definition 1. We fix the filter length and depth of the network to be $S = 2$ and $J = \lceil n^{\frac{1}{3}} \rceil$ respectively. For convenience, we neglect the special structure of the bias vector stated in equation (2.6). We train our network using the famous Adam optimization algorithm with constant step sizes, and free parameters (weights and bias) initialized by the default TensorFlow values.

The generalization performance of our trained DCNN is evaluated in terms of the test RMSE (Root Mean Square Error). The experimental results are presented in Fig. 2. From Fig. 2, we observe that for each d, the test RMSE gradually decreases and converges to some constant as the number of training samples n increases. Also, the variance of the test RMSE, as indicated by the blue error bars, reduces significantly as n grows. This verifies our results in Theorem 2 that $\varepsilon(\pi_M f_{D,J,S}) \to \varepsilon(f_\rho)$ with high probability.

7. Related work, discussions, and conclusions

In this section, we discuss prior work in overparameterization in regression and the generalization ability of DCNNs.

As mentioned before, it is frequently observed that overparameterized neural networks generalize well with zero training error for regression loss. Such a phenomenon is known as benign overfitting. Benign overfitting is characterized in [4] in linear least squares regression with Gaussian data and noise. Sufficient and necessary conditions are presented for the input covariance matrix's eigenvalue patterns for the minimum-norm interpolator to generalize well. The methodology there depends heavily on the linearity of the algorithm and the nice properties of the Gaussian random matrices induced by the Gaussian model. This methodology was extended to a setting of two-layer linear neural networks in [12]. Benign overfitting is also verified for stochastic gradient descent in an overparameterized linear regression setting in [34]. This study was further extended in [10] to train a shallow neural network with shared weights by gradient descent with respect to cross-entropy loss. Such a network is equipped with polynomial ReLU activation function $\sigma^q(u) = \max\{0, u\}^q$ with $q > 2$:

$$\frac{1}{m}\mathbf{1}_m^T\left[\sigma^q\left(\mathbf{W}_{+1}y\boldsymbol{\mu}\right) + \sigma^q\left(\mathbf{W}_{+1}\boldsymbol{\xi}\right) - \sigma^q\left(\mathbf{W}_{-1}y\boldsymbol{\mu}\right) - \sigma^q\left(\mathbf{W}_{-1}\boldsymbol{\xi}\right)\right].$$

Here, $\mathbf{W}_{+1}$ and $\mathbf{W}_{-1}$ are $m \times d$ fully-connected weight matrices shared by a signal input $y\boldsymbol{\mu}$, with $y \in \{1, -1\}$ and a fixed vector $\boldsymbol{\mu} \in \mathbb{R}^d$, and a noise vector $\boldsymbol{\xi} \in \mathbb{R}^d$. When $\boldsymbol{\xi}$ follows a standard normal distribution $N(0, \sigma_p^2 I_{d-1})$ on the orthogonal complement of $\boldsymbol{\mu}$, it was shown in [10] that the gradient descent algorithm may achieve an arbitrarily small training error $\epsilon > 0$ when the number of iterations is large enough, depending on the accuracy ϵ.

In a non-linear setting induced by ReLU, benign overfitting is verified for deep fully-connected neural networks in [22]. This result was achieved by a network deepening scheme where great capacity is provided by the fully-connected structure of the networks to induce interpolations. In our setting, the convolutional structure of DCNNs gives rigid constraints, making it infeasible to apply the network deepening scheme illustrated in [22]. We introduce a novel network deepening scheme designated for DCNNs. We double the width of a given teacher DCNN so that the linear features and the induced hat functions are built for attaining interpolation while the learning ability of the teacher DCNN is preserved. This is our first novelty. We would like to point out that our results

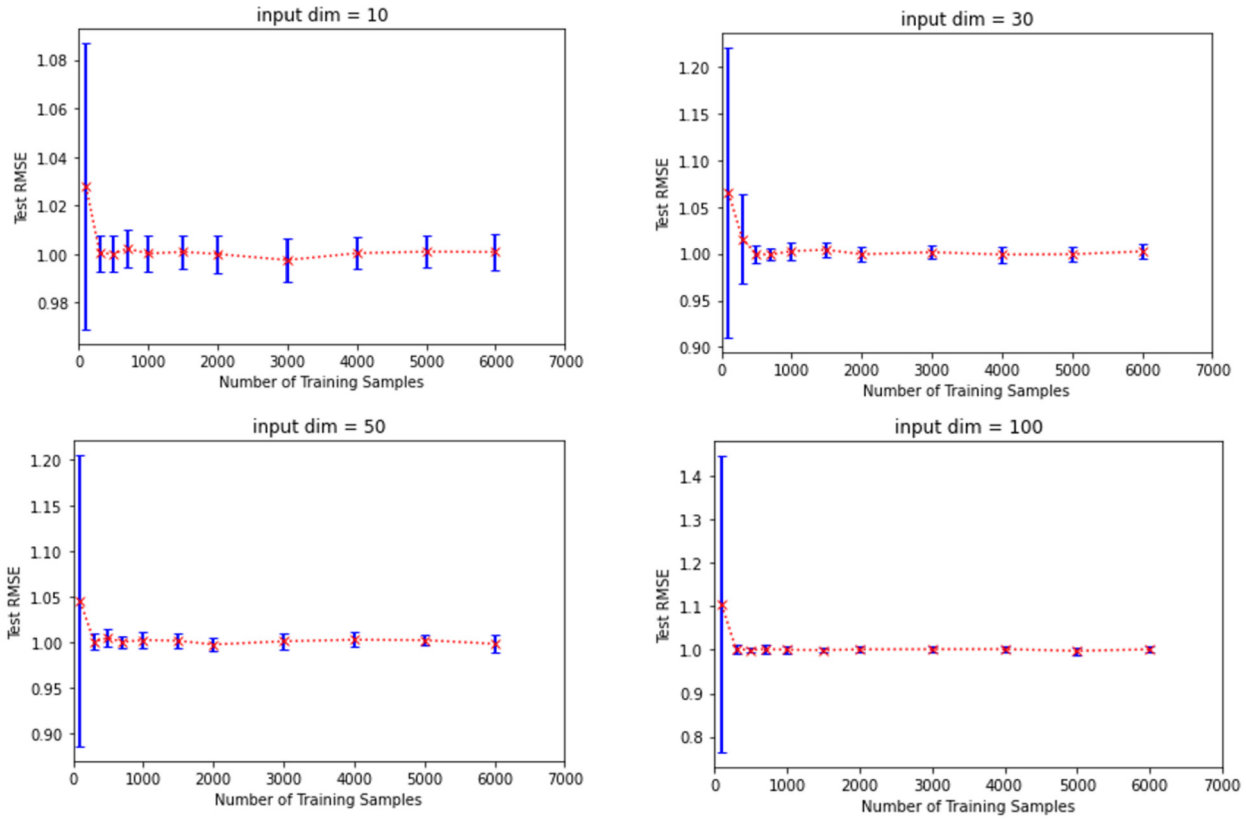

Fig. 2. Generalization errors of simulated data sets with $d = 10$ (top left), $d = 30$ (top right), $d = 50$ (bottom left), $d = 100$ (bottom right) respectively.

theoretically verify that overparameterized DCNNs can generalize well, but we are not able to characterize benign overfitting of DCNNs. In other words, we are not able to identify a sufficient and necessary condition for DCNNs to achieve benign overfitting.

Despite the wide applications of DCNNs in modern learning tasks, theoretical understanding of their generalization ability has been challenging. Promising progress has been made recently. An estimate of the approximation error of DCNNs is given in [33]. It shows that the approximation error decreases as the network depth increases. This result is an important tool we use to prove Theorem 2. In [21], it is shown that implementing ERM on DCNNs yields universally consistent estimators without any restrictions on free parameters. However, an explicit rate of convergence is not provided there. Moreover, for learning a radial regression function by a DCNN followed by one fully-connected layer, a learning rate of order $\mathcal{O}(n^{-1/2})$ is obtained in [24]. This learning rate is derived using a covering number estimate, which requires the filters and biases to be bounded, with bounds depending on the layers where the parameters lie. More recently, for learning an additive ridge function by a DCNN followed by one fully-connected layer, a learning rate of order $\mathcal{O}\left(n^{\frac{-2\alpha}{1+\alpha}}\log n\right)$ is obtained in [14], where $0 < \alpha \leq 1$. This learning rate is considered to be optimal up to a log factor. This result also requires the filters and bias to be bounded.

Some neural networks with similar structures are studied in the literature. For example, periodized CNNs are studied in [28] where convolutions on sequences on the group $\mathbb{Z}_d$ are used to induce circular weight matrices instead of Toeplitz-type ones induced by regular convolutions on sequences on $\mathbb{Z}$. Properties of approximating translation equivariant functions by such neural networks are discussed. Other work studying CNN-variant neural networks include ResNet-type CNNs in [27], and fully-connected networks inducing sparsity in [8].

In this paper, we derived, in our belief, the first learning rate of DCNNs that has been presented so far without any restrictions on free parameters or additional fully-connected layers. Such a breakthrough can be attributed to a novel sample error estimate, which relies on a tight pseudo-dimension estimate of DCNNs with piecewise linear activation functions inspired by that in [3]. Our second novelty is making use of this upper bound of pseudo-dimension, which no longer requires restrictions on filters and biases and can be used to bound the empirical covering number in turn.

Data availability

No data was used for the research described in the article.

Acknowledgments

The authors would like to thank the referees for their detailed comments and constructive suggestions that led to an improved presentation of this paper. The authors are partially sponsored by NSF grants CCF-1740776, DMS 2015363, and IIS-2229876. They are also partially supported by the A. Russell Chandler III Professorship at Georgia Tech.

Appendix A

A.1. Proof of Lemma 2

Proof. The case $j = 1$ is trivial. Suppose that (4.7) holds for $j = m$ with $m \in \{1, \dots, J^* - 1\}$. That is, the $(D + ms) \times D$ matrix $\prod_{i=1}^{m} T^{(i)} = T^{\mathbf{w},m}$ satisfies

$$\left(T^{(m)} \dots T^{(1)}\right)_{\ell,k} = \left(T^{\mathbf{w},m}\right)_{\ell,k} = W^{(m)}_{\ell-k}, \qquad 1 \le \ell \le D + ms,\ 1 \le k \le D.$$

Now we consider the product $T^{(m+1)}\left(\prod_{i=1}^{m} T^{(i)}\right)$, where $T^{(m+1)}$ is a $(D + (m+1)s) \times (D + ms)$ matrix. The (i,k)-entry of this matrix product (with $1 \le i \le D + (m+1)s$, $1 \le k \le D$) equals

$$\begin{aligned}\left(T^{(m+1)}\left(T^{(m)} \dots T^{(1)}\right)\right)_{i,k} &= \sum_{\ell=1}^{D+ms} \left(T^{(m+1)}\right)_{i,\ell} \left(T^{(m)} \dots T^{(1)}\right)_{\ell,k} \\ &= \sum_{\ell=1}^{D+ms} \left(w^{(m+1)}\right)_{i-\ell} W^{(m)}_{\ell-k}.\end{aligned}$$

Note that the sequence $W^{(m)}$ is supported in $\{0, \dots, ms\}$. For $\ell \le 0$, we have $\ell - k < 0$ and thereby $W^{(m)}_{\ell-k} = 0$. For $\ell \ge d_m + 1$, since $k \le d$, we have $\ell - k \ge ms + 1$ implying $W^{(m)}_{\ell-k} = 0$. So there holds

$$\left(T^{(m+1)}\left(T^{(m)} \dots T^{(1)}\right)\right)_{i,k} = \sum_{\ell=-\infty}^{\infty} \left(w^{(m+1)}\right)_{i-\ell} W^{(m)}_{\ell-k}.$$

This is exactly the $(i-k)$-th entry of the convoluted sequence $w^{(m+1)} * W^{(m)}$. This is also the (i,k)-entry of the $(D + (m+1)s) \times D$ matrix $T^{\mathbf{w},m+1}$. This proves (4.7) for $j = m + 1$, which completes the induction procedure and the proof of the lemma. □

A.2. Proof of Theorem 4: pseudo-dimension of feed-forward neural networks

To prove Theorem 4, we need the following technical tool on the number of possible sign vectors attained by polynomial vectors. It is introduced in [2, Lemma 2.1] (the proof can be found in [1, Theorem 8.3]).

Lemma 9. *Let $f_1, \dots, f_\ell$ be polynomials of degree at most L in $\kappa \le \ell$ variable. Then the number of distinct sign vectors $\{sgn(f_1(a)), \dots, sgn(f_\ell(a))\} \in \{1,0\}^\ell$ that can be generated by varying $a \in \mathbb{R}^\kappa$ is at most $2(2e\ell L/\kappa)^\kappa$. Here $sgn(u) = 1$ if $u \ge 0$ and 0 otherwise.*

Proof of Theorem 4. For an arbitrary choice of ℓ points $x_1, \dots, x_\ell \in \mathbb{R}^d$, we wish to bound

$$\mathcal{K} := \left| \left\{ (\operatorname{sgn}(f(x_1, a)), \dots \operatorname{sgn}(f(x_\ell, a))) : a \in \mathbb{R}^K \right\} \right|. \tag{A.1}$$

In other words, $\mathcal{K}$ is the number of sign patterns that the neural network can output for the sequence of inputs $(x_1, \dots, x_\ell)$. We will derive upper bounds for $\mathcal{K}$, leading us to the pseudo-dimension estimate for $\mathcal{H}$. To do so, we partition the free parameter domain $\mathbb{R}^K$ into some non-overlapping subsets in such a way that within each subset, the functions $f(x_1, \cdot), \dots, f(x_\ell, \cdot)$ are fixed polynomials of degree at most $1 + \theta + \cdots + \theta^J$. Fix $\{x_1, \dots, x_\ell\} \subset \mathbb{R}^d$. Let $K \le \ell$.

We will construct the desired partition by defining a sequence of partitions $\{S^{(i)}\}_{i=0}^{J}$ of $\mathbb{R}^K$ iteratively layer by layer with $S^{(0)} = \{\mathbb{R}^K\}$ such that:

1. For each $i = 1, \dots, J$,
$$\frac{\left|S^{(i)}\right|}{\left|S^{(i-1)}\right|} \le N_i := 2\left(\frac{2ed_i\ell p(1 + \theta + \theta^2 + \cdots + \theta^{i-1})}{K_1 + \cdots + K_i}\right)^{K_1 + \cdots + K_i}. \tag{A.2}$$
2. For each $i = 1, \dots, J+1$, on each element S of $S^{(i-1)}$, for each $j \in \{1, \dots, \ell\}$ and $k \in \{1, \dots, d_i\}$, the net input component
$$\left(T^{(i)} h^{(i-1)}(x_j, a) - b^{(i)}\right)_k$$
is a polynomial of degree at most $1 + \theta + \theta^2 + \cdots + \theta^{i-1}$ of the variables $A^{(1)}, \dots, A^{(i)}$. Here for $i = J + 1$ with $d_{J+1} = 1, T^{(J+1)} = c^T, A^{(J+1)} = c$, the only input component is the output function value $f(x_j, a) = c \cdot h^{(J)}(x_j, a)$ at x_j for each $j \in \{1, \dots, \ell\}$.

Note that $|S^{(0)}| = 1$. For $i = 1$, the above properties for $S^{(1)}$ hold true because each input component

$$\left(T^{(1)}h^{(0)}(x_j, a) - b^{(1)}\right)_k = \left(T^{(1)}x_j - b^{(1)}\right)_k$$

is an affine function of the variables $A^{(1)}$.

Suppose that the partitions $\{S^{(i)}\}_{i=0}^{q-1}$ have been found. Let us construct $S^{(q)}$. By our induction hypothesis, for each $j \in \{1, \dots, \ell\}$ and $k \in \{1, \dots, d_q\}$, the input component $\left(T^{(q)}h^{(q-1)}(x_j, a) - b^{(q-1)}\right)_k$ restricted onto each element S of $S^{(q-1)}$ equals to a polynomial, denoted as $P_{k,j,S}(a)$, of degree at most $1 + \theta + \theta^2 + \cdots + \theta^{q-1}$ of the variables $A^{(1)}, \dots, A^{(q)}$. For $S \in S^{(q-1)}$, we consider a collection of polynomials

$$\left\{P_{k,j,S}(a) - t_s : k \in \{1, \dots, d_q\}, j \in \{1, \dots, \ell\}, s \in \{1, \dots, p\}\right\}.$$

Since the number of variables $K_1 + \dots + K_q$ in $A^{(1)}, \dots, A^{(q)}$ is bounded by the number of functions $d_q \ell p$ in the collection, applying Lemma 9, we know that the number of distinct sign vectors achieved by this collection is at most N_q. Hence we can partition $\mathbb{R}^K$ into at most N_q non-overlapping subsets such that all these polynomials keep the signs unchanged on each subset. Intersecting these subsets with S gives a partition of S into at most N_q non-overlapping subsets. These partitions with S running over $S^{(q-1)}$ form a partition of $\mathbb{R}^K$ which is a refinement of $S^{(q-1)}$ and is the desired partition $S^{(q)}$. Obviously, (A.2) is satisfied for $i = q$. Moreover, on each $S' \in S^{(q)}$ in the partition of $S \in S^{(q-1)}$, for each $j \in \{1, \dots, \ell\}$ and $k \in \{1, \dots, d_q\}$, the polynomials $P_{k,j,S}(a) - t_s$ keep the signs unchanged and thereby for all $a \in S'$, the value $P_{k,j,S}(a)$ lies on the same interval $[t_\alpha, t_{\alpha+1})$ for some $\alpha \in \{0, \dots, p\}$. In fact, α is either the maximum $s \in \{1, \dots, p\}$ such that $\operatorname{sgn}(P_{k,j,S}(a) - t_s) = 1$ or 0 if $\operatorname{sgn}(P_{k,j,S}(a) - t_s) = 0$ for all $s \in \{1, \dots, p+1\}$.

As $\sigma|_{[t_\alpha, t_{\alpha+1})}$ is a polynomial of degree at most θ, we know that on $S' \subseteq S$, each component of $h^{(q)}(x_j, a)$, with $k \in \{1, \dots, d_q\}$,

$$\left(h^{(q)}(x_j, a)\right)_k = \sigma\left(\left(T^{(q)}h^{(q-1)}(x_j, a) - b^{(q-1)}\right)_k\right) = \sigma\left(P_{k,j,S}(a)\right)$$

is a polynomial of degree at most $\theta(1 + \theta + \theta^2 + \cdots + \theta^{q-1})$. This implies that each input component $\left(T^{(q+1)}h^{(q)}(x_j, a) - b^{(q+1)}\right)_k$ is a polynomial of degree at most $1 + \theta + \theta^2 + \cdots + \theta^q$ of the variables $A^{(1)}, \dots, A^{(q+1)}$ on $S' \in S^{(q)}$. This verifies the desired property for $S^{(q)}$ and completes the induction procedure.

Thus, we have constructed a partition $S^{(J)}$ of $\mathbb{R}^K$ such that on each element S of $S^{(J)}$, for each $j \in \{1, \dots, \ell\}$, the function $f(x_j, a) = c \cdot h^{(J)}(x_j, a)$ is a polynomial of degree at most $1 + \theta + \theta^2 + \cdots + \theta^J$ of the variables $a = [A^{(1)}, \dots, A^{(J)}, c]$. Since $\ell \geq K$, applying Lemma 9, we know that

$$\left|\left\{(\operatorname{sgn}(f(x_1, a)), \dots \operatorname{sgn}(f(x_\ell, a))) : a \in S\right\}\right| \leq 2\left(\frac{2e\ell(1 + \theta + \theta^2 + \cdots + \theta^J)}{K}\right)^K.$$

Since $S^{(J)}$ is a partition of $\mathbb{R}^K$, we see that the number of sign patterns $\mathcal{K}$ defined by (A.1) can be bounded as

$$\begin{aligned}\mathcal{K} &\leq \left|S^{(J)}\right| 2\left(2e\ell(1 + \theta + \theta^2 + \cdots + \theta^J)/K\right)^K \\ &\leq \left(\Pi_{i=1}^J N_i\right) 2\left(2e\ell(1 + \theta + \theta^2 + \cdots + \theta^J)/K\right)^K.\end{aligned}$$

Bounding the geometric mean by the arithmetic one, with $W_i := K_1 + \cdots + K_i$, we find

$$\mathcal{K} \leq 2^{J+1}\left(\frac{2e\ell\left(1 + \theta + \theta^2 + \cdots + \theta^J + \sum_{i=1}^J d_i p(1 + \theta + \theta^2 + \cdots + \theta^{i-1})\right)}{K + \sum_{i=1}^J W_i}\right)^{K + \sum_{i=1}^J W_i}.$$

Note that the points $\{x_1, \dots, x_\ell\}$ are arbitrarily chosen. An upper bound for the VC-dimension is then obtained by computing the largest value of ℓ for which the above number is at least 2^ℓ, yielding

$$2^\ell \leq 2^{J+1}\left(\frac{2e\ell\left(1 + \beta + \beta^2 + \cdots + \beta^J + \sum_{i=1}^J d_i p(1 + \beta + \beta^2 + \cdots + \beta^{i-1})\right)}{K + \sum_{i=1}^J W_i}\right)^{K + \sum_{i=1}^J W_i}.$$

Then the desired bound follows by applying Lemma 10 below and noticing that

$$K + \sum_{i=1}^J W_i = d_J + \sum_{j=1}^J K_j + \sum_{i=1}^J \sum_{j=1}^i K_j = d_J + \sum_{j=1}^J (J - j + 2)K_j \geq K.$$

This completes the proof of Theorem 4. □

Lemma 10 *(Lemma 18 in [3]). Suppose that $2^m \leq 2^t(mr/w)^w$ for some $r \geq 16$ and $m \geq w \geq t \geq 0$. Then, $m \leq t + w\log_2(2r\log_2 r)$.*

A.3. Proof of Lemma 7

Proof. For neural network consider in Lemma 7, we have $S+1$ free parameters in each $w^{(j)}$, and $2S-1$ free parameters from $b^{(j)}$ in the j-th layers for $j=1,\dots,J-1$ and $S+1+d+JS$ free parameters in the J-th layer. In other words, we have

$$K_j = 3S, \qquad \text{for } j=1,\dots,J-1$$

and

$$K_J = S+1+d+JS.$$

Then, we have

$$\begin{aligned} d_J + \sum_{j=1}^{J}(J-j+2)K_j &= d+JS+\sum_{j=1}^{J-1}(J-j+2)K_j + 2K_J \\ &= 2S+2+3(d+JS)+\frac{3}{2}S(J+4)(J-1) \\ &= 2+3d-4S+\frac{15}{2}JS+\frac{3}{2}J^2S. \end{aligned}$$

For ReLU, we have $\theta=1$ and $p=1$. Then,

$$\begin{aligned} R &= 1+J+\sum_{i=1}^{J}(d+iS)i \\ &= 1+J+\frac{J(J+1)}{2}d+\frac{J(J+1)(2J+1)}{6}S \\ &= \frac{1}{3}J^3S+\frac{1}{2}J^2(d+S)+J\left(\frac{d}{2}+\frac{S}{6}+1\right)+1. \end{aligned}$$

Observe that

$$2+3d-4S+\frac{15}{2}JS+\frac{3}{2}J^2S \le 3d+9J^2S \qquad (\because S\ge 2, J^2\ge J)$$

and

$$R \le \frac{1}{3}J^3S+J^2(d+S+1)+1 \le 3J^3S+J^2d \le 3(J^2S+Jd)^2.$$

Hence

$$\begin{aligned} \operatorname{Pdim}(\mathcal{H}_{J,S}) &\le J+1+(3d+9J^2S)\left(\log_2(12e(J^2S+Jd)^2)+\log_2\log_2(6e(J^2S+Jd)^2)\right) \\ &\le J+1+(3d+9J^2S)2\left(\log_2(12e(J^2S+Jd)^2)\right) \\ &\le c_0(J^2S+d)\log(Jd+J^2S), \end{aligned}$$

where c_0 is an absolute constant. The proof is complete. □

References

[1] Martin Anthony, Peter L. Bartlett, Peter L. Bartlett, et al., Neural Network Learning: Theoretical Foundations, vol. 9, Cambridge University Press, Cambridge, 1999.
[2] Peter Bartlett, Vitaly Maiorov, Ron Meir, Almost linear VC dimension bounds for piecewise polynomial networks, Adv. Neural Inf. Process. Syst. 11 (1998).
[3] Peter L. Bartlett, Nick Harvey, Christopher Liaw, Abbas Mehrabian, Nearly-tight VC-dimension and pseudodimension bounds for piecewise linear neural networks, J. Mach. Learn. Res. 20 (1) (2019) 2285–2301.
[4] Peter L. Bartlett, Philip M. Long, Gábor Lugosi, Alexander Tsigler, Benign overfitting in linear regression, Proc. Natl. Acad. Sci. 117 (48) (2020) 30063–30070.
[5] Mikhail Belkin, Daniel Hsu, Siyuan Ma, Soumik Mandal, Reconciling modern machine-learning practice and the classical bias–variance trade-off, Proc. Natl. Acad. Sci. 116 (32) (2019) 15849–15854.
[6] Mikhail Belkin, Siyuan Ma, Soumik Mandal, To understand deep learning we need to understand kernel learning, in: International Conference on Machine Learning, PMLR, 2018, pp. 541–549.
[7] Serge Bernstein, Sur l'extension du théorème limite du calcul des probabilités aux sommes de quantités dépendantes, Math. Ann. 97 (1) (1927) 1–59.
[8] Helmut Bolcskei, Philipp Grohs, Gitta Kutyniok, Philipp Petersen, Optimal approximation with sparsely connected deep neural networks, SIAM J. Math. Data Sci. 1 (1) (2019) 8–45.
[9] Denis Bosq, Linear Processes in Function Spaces: Theory and Applications, vol. 149, Springer Science & Business Media, 2000.
[10] Yuan Cao, Zixiang Chen, Mikhail Belkin, Quanquan Gu, Benign overfitting in two-layer convolutional neural networks, in: Proc. of Advances in Neural Information Processing Systems (NeurIPS) 35, 2022.
[11] Andrei Caragea, Philipp Petersen, Felix Voigtlaender, Neural network approximation and estimation of classifiers with classification boundary in a Barron class, arXiv preprint, arXiv:2011.09363, 2020.
[12] Niladri S. Chatterji, Philip M. Long, Peter L. Bartlett, The interplay between implicit bias and benign overfitting in two-layer linear networks, J. Mach. Learn. Res. 23 (263) (2022) 1–48.
[13] Ingrid Daubechies, Ten Lectures on Wavelets, SIAM, 1992.

[14] Zhiying Fang, Guang Cheng, Optimal learning rates of deep convolutional neural networks: additive ridge functions, Trans. Mach. Learn. Res. (2023).
[15] Ian Goodfellow, Yoshua Bengio, Aaron Courville, Deep Learning, MIT Press, 2016.
[16] Rémi Gribonval, Gitta Kutyniok, Morten Nielsen, Felix Voigtlaender, Approximation spaces of deep neural networks, Constr. Approx. 55 (1) (2022) 259–367.
[17] László Györfi, Michael Kohler, Adam Krzyzak, Harro Walk, et al., A Distribution-Free Theory of Nonparametric Regression, vol. 1, Springer, 2002.
[18] Jason M. Klusowski, Andrew R. Barron, Approximation by combinations of ReLU and squared ReLU ridge functions with ℓ^1 and ℓ^0 controls, IEEE Trans. Inf. Theory 64 (12) (2018) 7649–7656.
[19] Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton, Imagenet classification with deep convolutional neural networks, Adv. Neural Inf. Process. Syst. 25 (2012).
[20] Yann LeCun, Léon Bottou, Yoshua Bengio, Patrick Haffner, Gradient-based learning applied to document recognition, Proc. IEEE 86 (11) (1998) 2278–2324.
[21] Shao-Bo Lin, Kaidong Wang, Yao Wang, Ding-Xuan Zhou, Universal consistency of deep convolutional neural networks, IEEE Trans. Inf. Theory 68 (7) (2022) 4610–4617.
[22] Shao-Bo Lin, Yao Wang, Ding-Xuan Zhou, Generalization performance of empirical risk minimization on over-parameterized deep ReLU nets, arXiv preprint, arXiv:2111.14039, 2021.
[23] Jianchang Mao, Anil K. Jain, Artificial neural networks for feature extraction and multivariate data projection, IEEE Trans. Neural Netw. 6 (2) (1995) 296–317.
[24] Tong Mao, Zhongjie Shi, Ding-Xuan Zhou, Theory of deep convolutional neural networks III: approximating radial functions, Neural Netw. 144 (2021) 778–790.
[25] Shahar Mendelson, Roman Vershynin, Entropy and the combinatorial dimension, Invent. Math. 152 (1) (2003) 37–55.
[26] Hrushikesh Narhar Mhaskar, Approximation properties of a multilayered feedforward artificial neural network, Adv. Comput. Math. 1 (1) (1993) 61–80.
[27] Kenta Oono, Taiji Suzuki, Approximation and non-parametric estimation of ResNet-type convolutional neural networks, in: International Conference on Machine Learning, PMLR, 2019, pp. 4922–4931.
[28] Philipp Petersen, Felix Voigtlaender, Equivalence of approximation by convolutional neural networks and fully-connected networks, Proc. Am. Math. Soc. 148 (4) (2020) 1567–1581.
[29] David Pollard, Empirical Processes: Theory and Applications, IMS, 1990.
[30] Johannes Schmidt-Hieber, Nonparametric regression using deep neural networks with ReLU activation function, Ann. Stat. 48 (4) (2020) 1875–1897.
[31] Alex Waibel, Toshiyuki Hanazawa, Geoffrey Hinton, Kiyohiro Shikano, Kevin J. Lang, Phoneme recognition using time-delay neural networks, IEEE Trans. Acoust. Speech Signal Process. 37 (3) (1989) 328–339.
[32] Yarotsky Dmitry, Error bounds for approximations with deep ReLU networks, Neural Netw. 94 (2017) 103–114.
[33] Ding-Xuan Zhou, Universality of deep convolutional neural networks, Appl. Comput. Harmon. Anal. 48 (2) (2020) 787–794.
[34] Difan Zou, Jingfeng Wu, Vladimir Braverman, Quanquan Gu, Sham Kakade, Benign overfitting of constant-stepsize SGD for linear regression, in: Conference on Learning Theory, PMLR, 2021, pp. 4633–4635.