Theory of Deep Convolutional Neural Networks: Downsampling

Ding-Xuan Zhou
School of Data Science and Department of Mathematics
City University of Hong Kong, Kowloon, Hong Kong
Email: mazhou@cityu.edu.hk

Abstract

Establishing a solid theoretical foundation for structured deep neural networks is greatly desired due to the successful applications of deep learning in various practical domains. This paper aims at an approximation theory of deep convolutional neural networks whose structures are induced by convolutions. To overcome the difficulty in theoretical analysis of the networks with linearly increasing widths arising from convolutions, we introduce a downsampling operator to reduce the widths. We prove that the downsampled deep convolutional neural networks can be used to approximate ridge functions nicely, which hints some advantages of these structured networks in terms of approximation or modelling. We also prove that the output of any multi-layer fully-connected neural network can be realized by that of a downsampled deep convolutional neural network with free parameters of the same order, which shows that in general, the approximation ability of deep convolutional neural networks is at least as good as that of fully-connected networks. Finally, a theorem for approximating functions on Riemannian manifolds is presented, which demonstrates that deep convolutional neural networks can be used to learn manifold features of data.

Keywords: deep learning, convolutional neural networks, approximation theory, downsampling, filter masks

1 Introduction and Downsampling

Deep learning has provided powerful applications in many practical domains of science and technology. It is based on structured deep neural networks with structures or network architectures designed according to various purposes. As an important family of structured deep neural networks with convolutional structures, **deep convolutional neural networks** (DCNNs) have been applied successfully to speeches,

images, and many other types of data [14, 9, 13, 7]. Empirical observations have led to a belief that convolutions enable DCNNs to efficiently learn locally shift-invariant features, and thereby to demonstrate their powers in speech and image processing.

Compared with their rapid developments in practical applications and understanding of some computational issues like backpropagation, stochastic gradient descent, and error-correction tuning [7], modelling, approximation, or generalization abilities of structured deep neural networks (structured nets) are not well understood rigorously. In this paper we present an **approximation theory** for downsampled DCNNs in which an operation of **downsampling** is applied to DCNNs and plays a role of pooling in reducing widths of deep neural networks.

Before demonstrating differences between structured deep nets and the classical fully-connected nets (multi-layer neural networks), we recall that a multi-layer neural network for learning functions of input variable vector $x = (x_i)_{i=1}^d \in \mathbb{R}^d$ with ℓ hidden layers of neurons $\{H^{(k)} : \mathbb{R}^d \to \mathbb{R}^{n_k}\}_{k=1}^\ell$ with widths $\{n_k\}$ is defined iteratively by

$$H^{(k)}(x) = \sigma\left(F^{(k)}H^{(k-1)}(x) - \hat{b}^{(k)}\right), \qquad k = 1, 2, \ldots, \ell, \tag{1.1}$$

where $\sigma : \mathbb{R} \to \mathbb{R}$ is an activation function acting componentwise on vectors, $F^{(k)}$ is a $n_k \times n_{k-1}$ matrix, $\hat{b}^{(k)} \in \mathbb{R}^{n_k}$ is a bias vector, and $H^{(0)}(x) = x$ with width $n_0 = d$. The most crucial part in the above fully-connected nets is the **full matrix** $F^{(k)}$ which involves $n_k n_{k-1}$ free parameters to be trained and leads to huge computational complexity in implementing the induced deep learning algorithms. In particular, the classical shallow net corresponding to the 1-layer case $J = 1$ having $N = n_1$ hidden neurons needs to train a $N \times d$ full matrix $F^{(1)} = [t_1\ t_2\ \ldots\ t_N]^T$ with N row vectors $\{t_i \in \mathbb{R}^d\}_{i=1}^N$. These row vectors together with a bias vector $\hat{b} = (\hat{b}_i)_{i=1}^N \in \mathbb{R}^N$ and coefficients $\{c_i\}_{i=1}^N$ form $N(d+2)$ free parameters to be trained in the output function

$$f_N(x) = \sum_{i=1}^N c_i \sigma(t_i \cdot x - \hat{b}_i). \tag{1.2}$$

This number of free parameters is huge when the input data has a large dimension d and/or the number N of hidden neurons is large to achieve good approximation abilities. It leads to technical difficulty in implementing the fully-connected nets. A large literature around the late 1980s [4, 10, 1, 15, 19] on function approximation by fully-connected shallow or multi-layer neural networks compensates for the computational complexity. The most essential component in such an approximation theory is the fully-connected nature of the full matrix $F^{(k)}$ in (1.1) or the complete freedom of the feature vectors $\{t_i\}$ in (1.2).

DCNNs considered in this paper take a special form of multi-layer neural nets (1.1), and their specialty lies in the special sparse **network structures** imposed

by **convolutions**. Instead of full matrices $F^{(k)}$ in (1.1), matrices in our DCNNs of depth J are induced by **convolutional filter masks** $\{w^{(j)} : \mathbb{Z} \to \mathbb{R}\}_{j=1}^{J}$ with the restriction made throughout the paper that each filter mask $w^{(j)}$ is a sequence supported in $\{0, 1, \ldots, s^{(j)}\}$ for some $s^{(j)} \in \mathbb{N}$ called **filter length**, involving only $s^{(j)} + 1$ free parameters. Such a filter mask $w = (w_k)_{k=-\infty}^{\infty}$ supported in $\{0, 1, \ldots, s\}$ for some $s \in \mathbb{N}$ satisfies $w_k = 0$ for $k \notin [0, s]$ and convoluting with it leads to a Toeplitz type $(D + s) \times D$ **convolutional matrix** $T^w := (w_{i-k})_{i=1,\ldots,D+s,k=1,\ldots,D}$ for $D \in \mathbb{N}$ given explicitly by

$$T^w = \begin{bmatrix} w_0 & 0 & 0 & 0 & \cdots & \cdots & 0 \\ w_1 & w_0 & 0 & 0 & \cdots & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ w_s & w_{s-1} & \cdots & w_0 & 0 & \cdots & 0 \\ 0 & w_s & \cdots & w_1 & w_0 & 0 \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & w_s & \cdots & w_1 & w_0 \\ 0 & \cdots & 0 & 0 & w_s & \cdots & w_1 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 0 & w_s \end{bmatrix} \in \mathbb{R}^{(D+s)\times D}. \tag{1.3}$$

In [28, 29], we take convolutional matrices $T^{(j)} := T^{w^{(j)}}$ (with $D = d_{j-1}$ and $s = s^{(j)}$), and study a DCNN $\{h^{(j)} : \mathbb{R}^d \to \mathbb{R}^{d_j}\}_{j=1}^{J}$ with linearly increasing widths $\{d_j = d + js\}$ and uniform filter length $s^{(j)} \equiv s$ as

$$h^{(j)}(x) = \mathcal{A}_{T^{(j)},b^{(j)}} \circ \cdots \circ \mathcal{A}_{T^{(1)},b^{(1)}}(x), \qquad j = 1, 2, \ldots, J. \tag{1.4}$$

Here $\mathcal{A}_{F,b} : \mathbb{R}^{d_{j-1}} \to \mathbb{R}^{d_{j-1}+s^{(j)}}$ is an **activated affine mapping** associated with a $(d_{j-1} + s^{(j)}) \times d_{j-1}$ matrix F and a bias vector $b \in \mathbb{R}^{d_{j-1}+s^{(j)}}$ defined by

$$\mathcal{A}_{F,b}(u) = \sigma\left(Fu - b\right), \qquad u \in \mathbb{R}^{d_{j-1}},$$

and σ is the rectified linear unit (ReLU) activation function given by $\sigma(u) = \max\{u, 0\}$.

Note that the last layer of the fully-connected net (1.1) can be expressed as $H^{(\ell)}(x) = \mathcal{A}_{F^{(\ell)},b^{(\ell)}} \circ \cdots \circ \mathcal{A}_{F^{(1)},b^{(1)}}(x)$ in terms of the mappings $\mathcal{A}_{F,b}$ involving $F^{(k)}$, with "F" standing for "full" matrices in the fully-connected nets, instead of $T^{(j)}$, with "T" standing for "Toeplitz" type matrices in DCNNs. We shall show in Theorem 2 below that $H^{(\ell)}(x)$ produced by a fully-connected net (1.1) with ℓ hidden layers can be realized by the output $h^{(J)}(x)$ of a (downsampled) DCNN with J layers; while the total number of free parameters in the DCNN is at most 8 times of that of (1.1), the number J of layers of the DCNN is much larger than ℓ, the number of layers of (1.1).

In this paper we introduce a downsampling operation into DCNNs to control the widths in (1.4). The ℓ downsamplings are introduced at layers $\mathcal{J} := \{J_k\}_{k=1}^{\ell}$ with $1 < J_1 \leq J_2 \leq \ldots \leq J_\ell = J$. Our idea of applying downsampling operators is motivated by the literature of wavelets [5, 18]. Denote the integer part of $u \in \mathbb{R}_+$ as $[u]$.

Definition 1. *The downsampling operator $\mathcal{D}_m : \mathbb{R}^D \to \mathbb{R}^{[D/m]}$ with a scaling parameter $m \leq D$ is defined by*

$$\mathcal{D}_m(v) = (v_{im})_{i=1}^{[D/m]}, \qquad v \in \mathbb{R}^D. \tag{1.5}$$

A **downsampled DCNN** *with ℓ downsamplings at layers $\mathcal{J}$ and filter lengths $\{s^{(j)}\}_{j=1}^{J}$ has widths $\{d_j\}_{j=0}^{J}$ defined iteratively by $d_0 = d$ and for $k = 1, \ldots, \ell$,*

$$d_j = \begin{cases} d_{j-1} + s^{(j)}, & \text{if } J_{k-1} < j < J_k, \\ \left[(d_{j-1} + s^{(j)})/d_{J_{k-1}}\right], & \text{if } j = J_k, \end{cases} \tag{1.6}$$

and is a sequence of function vectors $\left\{h^{(j)}(x) : \mathbb{R}^d \to \mathbb{R}^{d_j}\right\}_{j=1}^{J}$ defined iteratively by $h^{(0)}(x) = x$ and for $k = 1, \ldots, \ell$,

$$h^{(j)}(x) = \begin{cases} \mathcal{A}_{T^{(j)},b^{(j)}}\left(h^{(j-1)}(x)\right), & \text{if } J_{k-1} < j < J_k, \\ \mathcal{D}_{d_{J_{k-1}}} \circ \mathcal{A}_{T^{(j)},b^{(j)}}\left(h^{(j-1)}(x)\right), & \text{if } j = J_k. \end{cases} \tag{1.7}$$

Moreover, we require that the bias vectors $b^{(j)} \in \mathbb{R}^{d_{j-1}+s^{(j)}}$ satisfy the restriction

$$b^{(j)}_{s^{(j)}+1} = b^{(j)}_{s^{(j)}+2} = \ldots = b^{(j)}_{d_{j-1}} \tag{1.8}$$

for $j \notin \mathcal{J}$. We call the downsampled DCNN **uniform** *with uniform filter lengths $\mathcal{S} := \{s^{[k]} \in \mathbb{N}\}_{k=1}^{\ell}$ if $s^{(J_{k-1}+1)} = \ldots = s^{(J_k)} = s^{[k]}$ for every $k \in \{1, \ldots, \ell\}$.*

Remark 1. *Acting the activated affine mapping $\mathcal{A}_{T^{(j)},b^{(j)}}$ on $h^{(j-1)}(x)$ produces the vector in (1.7) as*

$$\left(\mathcal{A}_{T^{(j)},b^{(j)}}\left(h^{(j-1)}(x)\right)\right)_i = \sigma\left(\sum_{r=1}^{d_{j-1}} w^{(j)}_{i-r}\left(h^{(j-1)}(x)\right)_r - b^{(j)}_i\right), \quad 1 \leq i \leq d_{j-1} + s^{(j)},$$

where $\sum_{r=1}^{d_{j-1}} w^{(j)}_{\cdot-r}\left(h^{(j-1)}(x)\right)_r$ is exactly the convolution $w^{(j)} * h^{(j-1)}(x)$ of the filter mask $w^{(j)}$ with $h^{(j-1)}(x)$ viewed as a sequence supported in $\{1, \ldots, d_{j-1}\}$. Recall that the convolution of sequence a supported in $\{0, \ldots, s\}$ and sequence b supported in $\{0, \ldots, D-1\}$ is a sequence $a*b$ supported in $\{0, \ldots, D+s-1\}$ given by $(a*b)_i = \sum_{r=-\infty}^{\infty} a_{i-r}b_r$. This illustrates the role of convolution in the definition of DCNNs and the convolutional matrix (1.3).

The restriction (1.8) is satisfied by the vector produced by acting the convolutional matrices $T^{(w)}$ on the constant 1 vector, so we impose this constraint on the bias vector to reduce the number of free parameters.

In this paper we make the following contributions to the approximation theory of DCNNs:

1. To introduce a downsampling operation into the DCNNs (1.4) so that the widths can be reduced from the linearly increasing nature.

2. To present a theorem for approximating ridge functions of the form $g(\xi\cdot x)$ with $\xi\in\mathbb{R}^d$ and $g:\mathbb{R}\to\mathbb{R}$, which demonstrates that for some classes of functions on $\mathbb{R}^d$ with special strcutres, DCNNs may have better approximation ability than fully-connected nets.

3. To prove that the last layer $H^{(\ell)}$ of a multi-layer fully-connected neural network (1.1) can be realized by that of a uniform DCNN, which shows that in general, the approximation ability of DCNNs is at least as good as that of fully-connected nets.

4. To prove a theorem for approximating functions on Riemannian manifolds, which demonstrates that DCNNs can be used to learn manifold features of data.

All the DCNNs constructed in this paper are uniform.

2 Main Results

In terms of the sequences of filter masks $\mathbf{w}=\{w^{(j)}\}_{j=1}^J$, bias vectors $\mathbf{b}=\left\{b^{(j)}\right\}_{j=1}^J$, and filter lengths $\mathbf{s}=\{s^{(j)}\}_{j=1}^J$, we introduce a composed mapping $\mathcal{A}_{\mathbf{w},\mathbf{b}}^{q,p}$ for $p\le q$ as

$$\mathcal{A}_{\mathbf{w},\mathbf{b}}^{q,p}=\mathcal{A}_{T^{(q)},b^{(q)}}\circ\ldots\circ\mathcal{A}_{T^{(p+1)},b^{(p+1)}}\circ\mathcal{A}_{T^{(p)},b^{(p)}}:\mathbb{R}^{d_{p-1}}\to\mathbb{R}^{d_q}.$$

We omit index $\mathbf{s}$ for simplicity. Then the last layer in the downsampled DCNN with ℓ downsamplings at layers $\mathcal{J}$ defined in Definition 1 can be expressed explicitly as

$$h^{(J)}(x)=\mathcal{D}_{d_{J_{\ell-1}}}\circ\mathcal{A}_{\mathbf{w},\mathbf{b}}^{J,J_{\ell-1}+1}\circ\cdots\circ\mathcal{D}_{d_{J_1}}\circ\mathcal{A}_{\mathbf{w},\mathbf{b}}^{J_2,J_1+1}\circ\mathcal{D}_d\circ\mathcal{A}_{\mathbf{w},\mathbf{b}}^{J_1,1}(x). \tag{2.1}$$

The induced hypothesis space of functions on a bounded subset Ω of $\mathbb{R}^d$ is given by

$$\mathcal{H}^{\mathbf{w},\mathbf{b},\mathcal{J},\mathbf{s}}=\left\{c\cdot h^{(J)}(x):\ c\in\mathbb{R}^{d_J}\right\}. \tag{2.2}$$

2.1 Approximating ridge functions

The first purpose of this paper is to show that DCNNs have a nice performance in approximating ridge functions of the form

$$g(\xi\cdot x),\qquad x\in\Omega \tag{2.3}$$

induced by the dot product $\xi \cdot x$ in $\mathbb{R}^d$ with an unknown feature vector $\xi \in \mathbb{R}^d$ and an unknown univariate function $g : \mathbb{R} \to \mathbb{R}$. We denote the norm of $\mathbb{R}^d$ as $|x|$ and the unit ball as $\mathbb{B} := \{x \in \mathbb{R}^d : |x| \le 1\}$. We assume for ridge approximation that $\Omega \subseteq \mathbb{B}$. Denote $\lceil u \rceil$ to be the smallest integer greater than or equal to $u > 0$.

The downsampled DCNN in the following approximation theorem, to be proved in Section 4, has $\ell = 2$ downsamplings at layers $\mathcal{J} = \{J_1 \le \lceil \frac{d-1}{s-1} \rceil, J = J_1 + 1\}$, uniform filter lengths $\mathcal{S} = \{s, 4N + 6\}$ with $s \in [2, d]$, the last filter mask $\{w_i^{(J)} \equiv 1\}_{i=0}^{4N+6}$, and widths

$$d_j = \begin{cases} d + js, & \text{if } j = 0, 1, \ldots, J_1 - 1, \\ 1 \text{ or } 2, & \text{if } j = J_1, \\ 2N + 4, & \text{if } j = J, d_{J_1} = 2, \\ 4N + 7, & \text{if } j = J, d_{J_1} = 1 \end{cases} \tag{2.4}$$

for some parameter $N \in \mathbb{N}$ which determines the approximation accuracy. The last bias vector $b^{(J)}$ is chosen as

$$\left(b^{(J)}\right)_i = \begin{cases} d_{J_1} B^{(J_1)} + t_i, & \text{for } i = 1, 2, \ldots, 2N + 3, \\ B^{(J_1)} + 1, & \text{for } i \ge 2N + 4, \end{cases} \tag{2.5}$$

where $t_i = t_{i,N} := \frac{i-N-2}{N}$ for $i = 1, \ldots, 2N + 3$, and $B^{(J_1)}$ is a parameter depending on $\mathbf{w}$.

For the regularity of the univariate function $g : \mathbb{R} \to \mathbb{R}$ in (2.3), we assume that for some $0 < \alpha \le 1$, g is Lipschitz-α meaning that for some constant $C_{g,\alpha}$,

$$|g(u) - g(v)| \le C_{g,\alpha} |u - v|^\alpha, \qquad \forall u, v \in \mathbb{R}. \tag{2.6}$$

Theorem 1. *Let $\xi \in \mathbb{B}$, $2 \le s \le d$, and $N \in \mathbb{N}$. If g is Lipschitz-α for some $0 < \alpha \le 1$, then there exists a uniform downsampled DCNN $\left\{h^{(j)}(x)\right\}_{j=1}^{J}$ at layers $\mathcal{J} = \{J_1 \le \lceil \frac{d-1}{s-1} \rceil, J = J_1 + 1\}$, uniform filter lengths $\mathcal{S} = \{s, 4N + 6\}$ with $\{w_i^{(J)} \equiv 1\}_{i=0}^{4N+6}$, and $\mathbf{b} = \left\{b^{(j)}\right\}_{j=1}^{J}$ satisfying (1.8) for $j = 1, \ldots, J - 1$, $b^{(J)}$ given by (2.5) in terms of N and a parameter $B^{(J_1)}$, and coefficients $\{c_i\}_{i=1}^{2N+3}$ such that*

$$\left\| \sum_{i=1}^{2N+3} c_i \left(h^{(J)}(x)\right)_i - g(\xi \cdot x) \right\|_\infty \le \frac{2C_{g,\alpha}}{N^\alpha}. \tag{2.7}$$

To achieve the approximation accuracy $\epsilon \in (0, 1)$, we take $N = \lceil (2C_{g,\alpha}/\epsilon)^{1/\alpha} \rceil$ and require at most $\mathcal{W} \le \frac{3d(d-1)}{s-1} + 2\left(2C_{g,\alpha}/\epsilon\right)^{1/\alpha} + 8$ widths (computation units) and $\mathcal{N} \le 8d + 2\left(2C_{g,\alpha}/\epsilon\right)^{1/\alpha}$ free parameters.

Remark 2. *From the proof of the theorem, we can see that we may take $J_1 = \lceil \frac{d-1}{s-1} \rceil$ to cancel the uncertain parameter J_1 by setting the filter masks $w^{(j)}$ for $j = J_1 + 1, \ldots, \lceil \frac{d-1}{s-1} \rceil$ to be the delta sequence on $\mathbb{Z}$ in the case $J_1 < \lceil \frac{d-1}{s-1} \rceil$. The conclusion of Theorem 1 still holds.*

The dimension-free rates of approximation given by (2.7) demonstrate the nice performance of DCNNs in approximating ridge functions. There has been some evidence in the approximation theory literature that rates of approximation by fully-connected nets might depend on the dimension. As an example, it was shown in [17] that for approximating functions from the unit ball $\mathcal{F} = \{f \in W_\infty^r(\mathbb{B}) : \|f\|_{W_\infty^r} \le 1\}$ *of the Sobolev space* $W_\infty^r(\mathbb{B})$ *with* $r \in \mathbb{N}$ *by the fully-connected shallow net (1.2), the worse-case error depends on the dimension* d *as* $\sup_{f\in\mathcal{F}} \inf_{c_i,t_i,\hat{b}_i} \|f - f_N\|_\infty \ge c_{d,r} N^{-r/(d-1)}$ *with a positive constant* $c_{d,r}$ *independent of* $N \in \mathbb{N}$*. Of course, this worse-case behavior does not imply that the rate in approximating an individual ridge function by the fully-connected net (1.2) must be dimension dependent. It would be interesting to give concrete mathematical statements on advantages of deep CNNs over fully-connected nets. In particular, we conjecture that there are some function classes which can be approximated by DCNNs faster than fully-connected nets.*

2.2 Representing fully-connected nets

Our second purpose of the paper is to show that output functions produced by any deep fully-connected neural net associated with ReLU can be realized by downsampled DCNNs. This extends our earlier work [29, 30] on shallow nets which has recently been established for periodized deep CNNs in [21]. This more general result confirms again that the approximation ability of DCNNs is at least as good as that of fully-connected nets.

Theorem 2. *Let* $\{H^{(k)} : \mathbb{R}^d \to \mathbb{R}^{n_k}\}_{k=1}^\ell$ *be an* ℓ*-layer fully connected neural network satisfying (1.1) with connection matrices* $F^{(k)}$*, bias vector* $\hat{b}^{(k)}$ *such that* $n_k n_{k-1} > 1$ *for each* $k \in \{1, \dots, \ell\}$*. Let* $s^{[k]} \in [2, n_k n_{k-1}]$ *for each* k*. Then there is a uniform downsampled DCNN* $\left\{h^{(j)}(x) : \mathbb{R}^d \to \mathbb{R}^{d_j}\right\}_{j=1}^J$ *with* ℓ *downsamplings at layers* $\{J_k = \sum_{j=1}^k \Delta_j\}$ *with* $\Delta_j \le \lceil \frac{n_j n_{j-1}-1}{s^{[j]}-1} \rceil$ *for each* j*, and uniform filter lengths* $\mathcal{S} = \{s^{[k]}\}_{k=1}^\ell$*, together with bias vectors* $b^{(j)} \in \mathbb{R}^{d_{j-1}+s^{[k]}}$ *satisfying (1.8) for* $j \notin \mathcal{J}$ *such that*

$$h^{(J_k)}(x) = H^{(k)}(x), \qquad \forall k \in \{1, \dots, \ell\}, x \in \Omega. \tag{2.8}$$

The total number of free parameters in the above net is at most $8 \sum_{k=1}^\ell (n_k n_{k-1})$ *and is at most* 8 *times of that of the fully-connected net.*

Remark 3. *The number* 8 *seems too large to support the use of DCNNs. It would be interesting to reduce this number to a much smaller level.*

2.3 Approximating functions on Riemannian manifolds

Our last purpose is to apply Theorem 2 and show that rates of approximating functions on Riemannian manifolds depend on the manifold dimension instead of that of

the ambient Euclidean space.

Theorem 3. *Let Ω be a compact connected m-dimensional C^∞ Riemannian manifold without boundary embedded in $\mathbb{R}^d$ with $d \geq 2$ and $s \in [2, d]$. If f is a twice continuous differentiable function on Ω and has bounded Hessian, then for any $N \in \mathbb{N}$, there is a uniform downsampled DCNN $\left\{h^{(j)}(x) : \mathbb{R}^d \to \mathbb{R}^{d_j}\right\}_{j=1}^3$ with 3 downsamplings and uniform filter lengths $\{s^{(j)} \equiv s\}_{j=1}^J$ of depth*

$$J \leq \frac{C_\Omega^2 (d + 2N)(8mN + 5d)}{s - 1} + 3$$

with a constant $C_\Omega \in \mathbb{N}$ depending on the manifold, together with bias vectors $b^{(j)}$ satisfying (1.8) for $j \notin \mathcal{J}$ such that

$$\inf_{c \in \mathbb{R}^J} \left\| f(x) - c \cdot h^{(J)}(x) \right\|_{C(\Omega)} \leq A_{f,\Omega,m,d} N^{-2/m}, \tag{2.9}$$

where $A_{f,\Omega,m,d}$ is a positive constant independent of N. The total number of free parameters is at most $9C_\Omega^2(d + 2N)(8mN + 5d)$.

To achieve the approximation accuracy $\epsilon \in (0, 1)$, we require a depth of order $O\left((d + m\epsilon^{-m/2})^2/s\right)$ and free parameters of order $O\left((d + m\epsilon^{-m/2})^2\right)$.

2.4 Comparison and Discussion

In this subsection we compare our results with those in the literature and give some theoretical justifications for the success of DCNNs in terms of approximation rates. Comparisons are made by means of the total number of free parameters $\mathcal{N}$ and the total number of computation units $\mathcal{W}$ (widths, or hidden units) required for the same approximation accuracy $\epsilon > 0$.

A classical literature for approximation of functions by shallow or multi-layer fully connected nets was well developed around 1990. A series of results [4, 10, 15] are about universality of this approximation for any non-polynomial locally bounded and piecewise continuous activation function, which was recently developed for DCNNs with ReLU in [29]. Quantitative results about rates of approximation were obtained in [10, 1, 19, 3] and references therein for understanding efficiency of neural networks. When a C^∞ activation function satisfies $\lim_{u \to -\infty} \sigma(u) = 0, \lim_{u \to \infty} \sigma(u) = 1$ (sigmoidal function) and $f = F|_{[-1,1]^d}$ for some $F \in L^2(\mathbb{R}^d)$ with the Fourier transform $\hat{F}$ satisfying $|w|\hat{F}(w) \in L^1(\mathbb{R}^d)$, it was shown in [1] that for the shallow net (1.2) and an arbitrary probability measure μ, there holds $\|f_N - f\|_{L^2_\mu([-1,1]^d)} = O(1/\sqrt{N})$. This result was extended to the case with ReLU recently in [11]. Most results in the classical literature about rates of approximation by fully connected nets were

obtained for continuous activation functions σ with two special assumptions: one is that for some $b \in \mathbb{R}$,

$$\sigma^{(i)}(b) \neq 0, \qquad \forall i \in \mathbb{N} \cup \{0\}, \tag{2.10}$$

and the other is that for some integer $q \neq 1$, there holds

$$\lim_{u \to -\infty} \sigma(u)/|u|^q = 0 \quad \text{and} \quad \lim_{u \to \infty} \sigma(u)/u^q = 1. \tag{2.11}$$

Such a result was presented in [19] for shallow nets (1.2) as

$$\|f_N - f\|_{C([-1,1]^d)} \leq c_{f,d,r} N^{-r/d}, \qquad \forall N \in \mathbb{N} \tag{2.12}$$

with a constant $c_{f,d,r}$, under the condition that the approximated function f lies in the space $C^r([-1,1]^d)$ of r-th continuously differentiable functions on $[-1,1]^d$. For the approximation accuracy $\|f_N - f\|_{C([-1,1]^d)} \leq \epsilon$, one needs

$$\mathcal{W} = N \geq \left(\frac{c_{f,d,r}}{\epsilon}\right)^{d/r}, \quad \mathcal{N} \geq (d+2)\left(\frac{c_{f,d,r}}{\epsilon}\right)^{d/r}. \tag{2.13}$$

The ReLU activation function σ used in the recent deep learning literature and considered in this paper does not satisfy the two special assumptions (2.10), (2.11). Explicit rates of approximation by fully connected ReLU nets were obtained recently in [11] for shallow nets, in [22] for nets with 3 hidden layers, and in [24, 25, 2, 20] for nets with more layers. As an example, Theorem 1 of [25] asserts that for any $r \in \mathbb{N}$, $f \in W_\infty^r([0,1]^d)$ can be approximated within an accuracy $\epsilon \in (0,1)$ by a ReLU deep net with at most $c(\log(1/\epsilon)+1)$ layers and at most $c\epsilon^{-d/r}(\log(1/\epsilon)+1)$ computation units with a constant $c = c(d,r)$. But as we pointed out in [29], this constant may increase very fast as d becomes large. To be more specific, the approach in [25] is to first approximate f by a localized Taylor polynomial

$$f_1(x) = \sum_{m \in \{0,1,\ldots,N\}^d} \sum_{\|\alpha\|_1 < r} \frac{D^\alpha f(m/N)}{\alpha!} \phi_m(x)(x - m/N)^\alpha, \tag{2.14}$$

where the localization at scale $1/N$ with $N \in \mathbb{N}$ is made by means of trapezoid functions $\phi_m(x) = \Pi_{i=1}^d \varphi(3Nx_i - m_i)$ supported on $m/N + [-2/N, 2/N]^d$ defined with a univariate trapezoid function $\varphi(u) = \sigma(u+2) - \sigma(u+1) - \sigma(u-1) + \sigma(u-2)$. Then for each basis function $\phi_m(x)(x-m/N)^\alpha$ in (2.14), a ReLU net of depth at least $c_1(d + \|\alpha\|_1)\log(1/\delta)$ was constructed in [25] to achieve an approximation accuracy $(d+r)\delta$ for $\delta \in (0,1)$ where $c_1 = c_1(d,r)$ is a constant. Thus, to have an accuracy $\epsilon \in (0,1)$ for approximating f by a ReLU deep net, one takes $N = \lceil \left(\frac{2^{d+1}d^r}{\epsilon r!}\right)^{1/r} \rceil$ and $\delta = \frac{\epsilon}{2^{d+1}d^r(d+r)}$ as in [25] and the depth of the net is at least $C_0 d(\log(1/\epsilon) + d + r\log d)$ with an absolute constant $C_0 > 0$ while the total number of free parameters for the

approximation and the number of computation units are more than the number of coefficients $\frac{D^\alpha f(m/N)}{\alpha!}$:

$$(N+1)^d \left(\begin{array}{c} d+r-1 \\ d \end{array} \right) > \left(\frac{2^{d+1} d^r}{\epsilon r!} \right)^{d/r} \frac{d^{r-1}}{(r-1)!} > \epsilon^{-d/r} \left(\frac{2^{\frac{d+1}{r}} d}{r} \right)^d \frac{d^{r-1}}{(r-1)!}. \quad (2.15)$$

This shows that for a fixed r, the constant $c(d,r)$ in Theorem 1 of [25] increase very fast as d becomes large.

While the rates of approximation by fully-connected deep nets presented in [25] are valid for any smoothness index $r \in \mathbb{N}$ and for general $f \in W_\infty^r([0,1]^d)$, our Theorem 1 shows that for the special case of $r = 1$ and approximating ridge functions, DCNNs may achieve the same accuracy with a much smaller number of free parameters. To see this, take $r = 1$ in Theorem 1 when the Lipschitz parameter α is 1. Then we can see that for achieving the same approximation accuracy $\epsilon \in (0,1)$, the number of free parameters in the DCNN constructed in Theorem 1 is $\mathcal{N} \leq 8d + 4C_{g,\alpha}/\epsilon$ which is much smaller than the lower bound $\epsilon^{-d} \left(2^{d+1} d\right)^d$ stated in (2.15) when d is large.

Using the rates of approximation derived in this paper, we may get generalization error bounds for DCNN-based learning algorithms, as done for kernel-based algorithms in [23, 6, 8, 26, 12] and distributed learning algorithms in [27, 16, 30]. The main expected difficulty arises from the hypothesis space (2.2) which depends on the filter masks $\mathbf{w}$ and bias vectors $\mathbf{b}$, and is different from a reproducing kernel Hilbert space used in kernel methods.

3 Analysis of Convolutions in DCNNs

Before proving our main results, we analyze the role of convolutions in our downsampled DCNNs.

3.1 Convolutions in factorizations of matrices and filter masks

To understand the structure of the composed mapping $\mathcal{A}_{\mathbf{w},\mathbf{b}}^{J_k, J_{k-1}+1}$ in (2.1), we first consider the product $T^{(J_k)} \cdots T^{(J_{k-1}+2)} T^{(J_{k-1}+1)}$ of Toeplitz type matrices in the activated affine mappings. Here $T^{(j)}$ is a $(d_{j-1} + s^{(j)}) \times d_{j-1}$ matrix of the form (1.3) with $D = d_{j-1}$ and $s = s^{(j)}$. Observe that the sequence $W^{(k)} := w^{(J_k)} * \ldots * w^{(J_{k-1}+1)}$ obtained by convoluting the filter masks $\{w^{(j)}\}_{j=J_{k-1}+1}^{J_k}$ is supported on $\{0, 1, \ldots, \Delta_k\}$ where $\Delta_k = \sum_{j=J_{k-1}+1}^{J_k} s^{(j)}$. We denote the Toeplitz type matrix (1.3) with $D = d_{J_{k-1}}$ and $s = \Delta_k$ induced by this sequence as

$$T^{(J_k, J_{k-1}+1)} := \left(W_{i-k}^{(k)} \right)_{i=1,\ldots,d_{J_{k-1}}+\Delta_k, k=1,\ldots,d_{J_{k-1}}}.$$

It turns out that this matrix induced by th convoluted sequence $W^{(k)}$ is exactly equal to the product $T^{(J_k)}\cdots T^{(J_{k-1}+1)}$, which demonstrates the role of convolutions in matrix factorizations and is proved in the appendix by methods from [30].

Lemma 1. *For $k=1,\ldots,\ell$, we have*

$$T^{(J_k,J_{k-1}+1)}=T^{(J_k)}\cdots T^{(J_{k-1}+2)}T^{(J_{k-1}+1)}. \tag{3.1}$$

The role of convolutions in filter mask decompositions can be seen from the following lemma [29] for factorizing an arbitrary pre-assigned sequence W supported in $\mathbb{Z}_+$ into convolutions $w^{(p)}*\ldots*w^{(2)}*w^{(1)}$ of a sequence $\{w^{(j)}\}_{j=1}^{p}$.

Lemma 2. *Let $s\geq 2$ and $W=(W_k)_{k=-\infty}^{\infty}$ be a sequence supported in $\{0,\ldots,\mathcal{M}\}$ with $\mathcal{M}\geq 0$. Then there exists a finite sequence of filter masks $\{w^{(j)}\}_{j=1}^{p}$ each supported in $\{0,\ldots,s^{(j)}=s\}$ with $p\leq\lceil\frac{\mathcal{M}}{s-1}\rceil$ such that the following convolutional factorization holds true*

$$W=w^{(p)}*w^{(p-1)}*\ldots*w^{(2)}*w^{(1)}. \tag{3.2}$$

3.2 Choosing bias vectors

To derive explicit expressions for $h^{(j)}(x)$, we need to choose the bias vectors according to the special form of the convolutional matrices T^{w}.

For a function vector $h:\Omega\to\mathbb{R}^D$, we denote

$$\|h\|_\infty=\max_{i=1,\ldots,D}\sup_{x\in\Omega}\left|(h(x))_i\right|.$$

Denote $\|w\|_1=\sum_{k\in\mathbb{Z}}|w_k|$ to be the ℓ^1-norm of a finitely supported sequence w on $\mathbb{Z}$. Then we see immediately from the definition of the convolutional matrices $T^{(j)}=T^{w^{(j)}}$ that for any $h:\Omega\to\mathbb{R}^{d_{j-1}}$,

$$\left\|T^{(j)}h\right\|_\infty\leq\|w^{(j)}\|_1\,\|h\|_\infty. \tag{3.3}$$

We use some ideas from our previous study [29] on DCNNs without downsampling and choose the biases to be small enough such that the vectors $T^{(j)}h^{(j-1)}(x)-b^{(j)}$ have nonnegative entries. A special feature in our downsampled DCNNs is the matrix representing the downsampling operator $\mathcal{D}_{m^{(k)}}$ at layer J_k with $k\in\{1,\ldots,\ell\}$ given by

$$M^{(k)}=\begin{bmatrix} 0\cdots0 & 1 & 0\cdots0 & 0 & 0\cdots0 & 0\cdots0 & 0 & 0\cdots \\ 0\cdots0 & 0 & 0\cdots0 & 1 & 0\cdots0 & 0\cdots0 & 0 & 0\cdots \\ \vdots\ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0\cdots0 & 0 & 0\cdots0 & 0 & 0\cdots0 & 0\cdots0 & 1 & 0\cdots \end{bmatrix}, \tag{3.4}$$

where the entries 1 appear in columns $d_{J_k-1}, 2d_{J_k-1}, \ldots, [(d_{J_k-1}+s^{(J_k)})/d_{J_k-1}]d_{J_k-1}$. We denote the constant 1 vector in $\mathbb{R}^{d_j}$ as $\mathbf{1}_{d_j}$ and $J_0 = 0$. The proof of the following lemma will be given in the appendix.

Lemma 3. *Let $k \in \{1, \ldots, \ell\}$. Assume that for some positive number B and another real number $\hat{B} \in [-B, B]$, there holds*

$$\left\| h^{(J_{k-1})} - \hat{B}\mathbf{1}_{d_{J_{k-1}}} \right\|_\infty \le B. \tag{3.5}$$

If we take $b^{(J_{k-1}+1)} = \hat{B}T^{(J_{k-1}+1)}\mathbf{1}_{d_{J_{k-1}}} - B\|w^{(J_{k-1}+1)}\|_1\mathbf{1}_{d_{J_{k-1}+1}}$ and

$$b^{(j)} = B\left(\Pi_{p=J_{k-1}+1}^{j-1}\|w^{(p)}\|_1\right)T^{(j)}\mathbf{1}_{d_{j-1}} - B\left(\Pi_{p=J_{k-1}+1}^{j}\|w^{(p)}\|_1\right)\mathbf{1}_{d_{j-1}+s^{(j)}} \tag{3.6}$$

for $j = J_{k-1}+2, \ldots, J_k - 1$, then for $J_{k-1} < j < J_k$, (1.8) is satisfied and

$$h^{(j)}(x) = T^{(j)}\cdots T^{(J_{k-1}+1)}\left(h^{(J_{k-1})}(x) - \hat{B}\mathbf{1}_{d_{J_{k-1}}}\right) + B\left(\Pi_{p=J_{k-1}+1}^{j}\|w^{(p)}\|_1\right)\mathbf{1}_{d_j}. \tag{3.7}$$

If we choose $b^{(J_k)}$ by (3.6), then (1.8) is also satisfied for $j = J_k$ and

$$h^{(J_k)}(x) = M^{(k)}T^{(J_k, J_{k-1}+1)}\left(h^{(J_{k-1})}(x) - \hat{B}\mathbf{1}_{d_{J_{k-1}}}\right) + B\left(\Pi_{p=J_{k-1}+1}^{J_k}\|w^{(p)}\|_1\right)\mathbf{1}_{d_{J_k}}. \tag{3.8}$$

4 Constructing DCNNs and Proving Theorems

In this section we prove our main results.

4.1 Approximating ridge functions by DCNNs

To prove Theorem 1 for approximating ridge functions, we apply the convolutional factorization stated in Lemma 2 to the sequence supported in $\{0, 1, \ldots, d-1\}$ given by reversing the components of ξ as $[W_{d-1}\ W_{d-2}\ldots W_0] = [\xi_1\ \xi_2\ldots\xi_d] = \xi^T$.

Proof of Theorem 1. We construct the first J_1 layers and find $h^{(J_1)}(x)$. Take W to be the sequence supported in $\{0, 1, \ldots, d-1\}$ given by $W_i = \xi_{d-i}$ for $i = 0, \ldots, d-1$. Applying Lemma 2 with $\mathcal{M} = d-1$, we know that there exists a sequence of filter masks $\mathbf{w} = \{w^{(j)}\}_{j=1}^{J_1}$ supported in $\{0, \ldots, s^{(j)} = s\}$ with $J_1 \le \lceil \frac{d-1}{s-1} \rceil$ satisfying the convolutional factorization $W = w^{(J_1)} * w^{(J_1-1)} * \ldots * w^{(2)} * w^{(1)}$.

Take $\Delta_1 = J_1 s$ and $d_0 = d$. Then by $s \le d$,

$$\begin{aligned} d + J_1 s &< d + \left(\frac{d-1}{s-1} + 1\right)s \\ &= d + \frac{s(d+s-2)}{s-1} = d + \frac{2d(s-1) - (d-s)(s-2)}{s-1} < 3d. \end{aligned}$$

Thus the $(d+\Delta_1)\times d$ matrix $T^{(J_1,1)}$ in (1.3) has its d-th row $[W_{d-1}\ W_{d-2}\dots W_0]=[\xi_1\ \xi_2\dots\xi_d]=\xi^T$ and its $2d$-th row $[W_{2d-1}\ W_{d-2}\dots W_d]$ being the zero row if $J_1 s\geq d$. So by Lemma 1, after the first downsampling of scale d, the width is $d_{J_1}\in\{1,2\}$, the function vector $M^{(1)}T^{(J_1,1)}x$ has one row or two and equals $\xi^T x=\xi\cdot x$ if $d_{J_1}=1$, and $\begin{bmatrix}\xi^T x=\xi\cdot x\\ 0\end{bmatrix}\in\mathbb{R}^2$ if $d_{J_1}=2$.

The input layer $h^{(0)}(x)=x$ satisfies (3.5) with $\hat{B}=0$ and $B=1$ by our assumption $\Omega\subseteq\mathbb{B}$. Take the bias vectors $\{b^{(j)}\}_{j=1}^{J_1}$ as in (3.6), by Lemma 3, we have

$$h^{(J_1)}(x)=M^{(1)}T^{(J_1,1)}x+B^{(J_1)}\mathbf{1}_{d_{J_1}},$$

where $B^{(J_1)}=\Pi_{p=1}^{J_1}\|w^{(p)}\|_1$.

Then we construct the last layer with $J=J_2=J_1+1$ and filter length $s^{(J)}=4N+6$. It follows from $d_{J_1}\in\{1,2\}$ and the definition (1.6) of the downsampled width that

$$d_J=[(d_{J-1}+4N+6)/d_{J_1}]=\begin{cases}4N+7, & \text{if } d_{J_1}=1,\\ 2N+4, & \text{if } d_{J_1}=2.\end{cases}$$

Take the filter mask $w^{(J)}$ to be supported in $\{0,\dots,4N+6\}$ with $w_i^{(J)}=1$ for $i=0,\dots,4N+6$.

When $d_{J_1}=1$, the $(4N+7)\times 1$ matrix $T^{(J)}$ given by (1.3) has all the $4N+7$ rows identical, having only one entry 1, which tells us that all the entries of $T^{(J)}h^{(J_1)}(x)$ equals $\xi\cdot x+B^{(J_1)}$. We choose the bias vector $b^{(J)}$ by (2.5), that is,

$$b_i^{(J)}=\begin{cases}B^{(J_1)}+t_i, & \text{for } i=1,2,\dots,2N+3,\\ B^{(J_1)}+1, & \text{for } i=2N+4,\dots,4N+7.\end{cases}$$

Then $\left(h^{(J)}(x)\right)_i=\left(\mathcal{A}_{T^{(J)},b^{(J)}}\left(h^{(J-1)}(x)\right)\right)_i=\sigma\left(\xi\cdot x-t_i\right)$ for $i=1,2,\dots,2N+3$ and $\left(h^{(J)}(x)\right)_i=0$ for $i\geq 2N+4$.

When $d_{J_1}=2$, the $(4N+8)\times 2$ matrix $T^{(J)}$ given by (1.3) has the first row $[1,0]$, last row $[0,1]$ and all the middle rows $[1,1]$. Hence

$$\left(T^{(J)}h^{(J_1)}(x)\right)_i=\begin{cases}\xi\cdot x+B^{(J_1)}, & \text{if } i=1,\\ \xi\cdot x+2B^{(J_1)}, & \text{if } i=2,\dots,4N+7,\\ B^{(J_1)}, & \text{if } i=4N+8.\end{cases}$$

Choose the even entries of the bias vector $b^{(J)}$ by (2.5), that is,

$$b_{2i}^{(J)}=\begin{cases}2B^{(J_1)}+t_i, & \text{for } i=1,2,\dots,2N+3,\\ B^{(J_1)}+1, & \text{for } i=2N+4.\end{cases}$$

Then by (1.7) and the above identities on $T^{(J)}h^{(J_1)}(x)$ and $b^{(J)}$, we have

$$\left(h^{(J)}(x)\right)_i=\left(\mathcal{A}_{T^{(J)},b^{(J)}}\left(h^{(J-1)}(x)\right)\right)_{2i}=\sigma\left(\xi\cdot x-t_i\right),\qquad i=1,2,\dots,2N+3$$

and $\left(h^{(J)}(x)\right)_{2N+4} = 0$.

What is left for approximation is to find the coefficients $(c_r)_{r=1}^{2N+3}$ for a function from the hypothesis space (2.2). For this purpose we need a well-known scheme in approximation theory which can be found with a general form in [28]. In our setting, we take $\mathbf{t} = \{t_i\}_{i=1}^{2N+3}$ to be the $2N+3$ nodes and the approximation scheme $L_{\mathbf{t}}$ on $[t_2, t_{2N+2}] = [-1, 1]$ is defined by

$$L_{\mathbf{t}}(g)(u) = \sum_{i=2}^{2N+2} g(t_i)\delta_i(u), \qquad u \in [-1,1], g \in C[-1,1], \tag{4.1}$$

where the function $\delta_i \in C(\mathbb{R})$ with $i \in \{2, \ldots, 2N+2\}$ is given by

$$\delta_i(u) = N\sigma(u - t_{i-1}) - 2N\sigma(u - t_i) + N\sigma(u - t_{i+1}).$$

It can be found from Lemma 6 of [28] that from the Lipschitz-α continuity of g, we have

$$\|L_{\mathbf{t}}(g) - g\|_{C[-1,1]} = \sup_{u\in[-1,1]} |L_{\mathbf{t}}(g)(u) - g(u)| \le \frac{2C_{g,\alpha}}{N^\alpha}. \tag{4.2}$$

Since $|\xi| \le 1$ and $|x| \le 1$ for every $x \in \Omega$, the above approximation estimate yields

$$\sup_{x\in\Omega} \left| g(\xi \cdot x) - \sum_{i=2}^{2N+2} g(t_i)\delta_i(\xi \cdot x) \right| \le \frac{2C_{g,\alpha}}{N^\alpha}.$$

Moreover,

$$\sum_{i=2}^{2N+2} g(t_i)\delta_i(\xi \cdot x) \in \operatorname{span}\left\{\sigma\left(\xi \cdot x - t_i\right)\right\}_{i=1}^{2N+3} = \operatorname{span}\left\{\left(h^{(J)}(x)\right)_i\right\}_{i=1}^{2N+3}.$$

This proves the bound (2.7) for the approximation error.

The total number of required computation units or widths $\mathcal{W}$ is

$$\mathcal{W} = \sum_{j=1}^{J_1-1}(d + js) + d_{J_1} + 2N + 4 \le \frac{3d(d-1)}{s-1} + 2N + 6.$$

The total number of free parameters $\mathcal{N}$ is the sum of $J_1(s+1)$ contributed by $\mathbf{w}$, $J_1(2s+1)+1$ by $\mathbf{b}$, and $2N+3$ by the coefficients $\{c_i\}$ and can be bounded as

$$\mathcal{N} \le J_1(3s+2) + 2N + 4 \le \lceil \frac{d-1}{s-1} \rceil (3s+2) + 2N + 4.$$

Since $\lceil \frac{d-1}{s-1} \rceil < \frac{d-1}{s-1} + 1$, we know that $(s-1)\lceil \frac{d-1}{s-1} \rceil < d - 1 + (s-1)$ which implies $(s-1)\lceil \frac{d-1}{s-1} \rceil \le d + s - 3$ and

$$\mathcal{N} \le \frac{d+s-3}{s-1}(3s+2) + 2N + 4 = 3d + 2N + 3s + \frac{5(d-2)}{s-1}.$$

Observe that the function $3s+\frac{5(d-2)}{s-1}$ of the variable s on the interval $[2,d]$ is convex. So its maximum value is achieved at one of the two endpoints and we have

$$3s+\frac{5(d-2)}{s-1}\leq \max\left\{6+5(d-2), 3d+\frac{5(d-2)}{d-1}\right\}\leq 5d-2.$$

Hence

$$\mathcal{N}\leq 8d+2N-2.$$

To achieve the approximation accuracy $\epsilon\in(0,1)$, we take $N=\lceil(2C_{g,\alpha}/\epsilon)^{1/\alpha}\rceil$ and require the total width of $\mathcal{W}\leq \frac{3d(d-1)}{s-1}+2\left(2C_{g,\alpha}/\epsilon\right)^{1/\alpha}+8$ and the parameter number $\mathcal{N}\leq 8d+2\left(2C_{g,\alpha}/\epsilon\right)^{1/\alpha}$. This completes the proof of Theorem 1. □

Remark 4. *From our proof, we can see that we may take $J_1=\lceil\frac{d-1}{s-1}\rceil$ by taking the additional filter masks $w^{(j)}$ for $j=J_1+1,\ldots,\lceil\frac{d-1}{s-1}\rceil$ to be the delta sequence on $\mathbb{Z}$. With this choice,*

$$d+J_1 s=d+\lceil\frac{d-1}{s-1}\rceil s\geq d+\left(\frac{d-1}{s-1}\right)s\geq 2d,$$

so $d_{J_1}=2$.

The gap between $\lceil\frac{d-1}{s-1}\rceil$ and J_1 can be large when ξ has some sparse properties. It would be interesting to study better performance of approximating ridge functions by DCNNs when this sparsity is used.

4.2 Realizing fully-connected networks by deep CNNs

In this subsection we turn to representing output functions from fully-connected nets by deep CNNs. In the proof of Theorem 1, we take a sequence W supported on $\{0,\ldots,d-1\}$ by reversing the components of the feature vector ξ in the ridge function (2.3) and make a convolutional factorization. If we view the inner product $\xi\cdot x$ as the matrix-vector product $\xi^T x$, then we can stack the reversed row vectors of the $n_k\times n_{k-1}$ full connection matrix $F^{(k)}$ in (1.1) and form a sequence W supported on $\{0,\ldots,n_k n_{k-1}-1\}$ for a convolutional factorization. This is the key idea in the next proof.

Proof of Theorem 2. We present our construction by induction, starting from the input layer $H^{(0)}(x)=x$ with $k=1$ of width $n_0=d$. Suppose that for some $k\in\{1,\ldots,\ell\}$, the filter masks $\{w^{(j)}\}_{j=1}^{J_{k-1}}$ and the bias vectors $\{b^{(j)}\}_{j=1}^{J_{k-1}}$ with (3.7) valid have been constructed such that the J_{k-1}-th layer $h^{(J_{k-1})}(x)$ is equal to the $(k-1)$-th layer $H^{(k-1)}(x)$ of width $d_{J_{k-1}}=n_{k-1}$ of the fully connected net. We now show how to construct the DCNN layers $\left\{h^{(j)}(x):\mathbb{R}^d\to\mathbb{R}^{d_j}\right\}_{j=J_{k-1}+1}^{J_k}$ for realizing the k-th layer $H^{(k)}(x)$ of the fully connected net. To this end, we define a sequence

W supported on $\{0,\dots,n_kn_{k-1}-1\}$ by stacking the reversed row vectors of the $n_k\times n_{k-1}$ matrix $F^{(k)}$ in (1.1) as

$$W_{i+(r-1)n_{k-1}}=\left(F^{(k)}\right)_{r,n_{k-1}-i},\qquad r=1,2,\dots,n_k, i=0,1,\dots,n_{k-1}-1. \tag{4.3}$$

An essential point for the above definition of W is the identity

$$\left[W_{n_{k-1}-1+(r-1)n_{k-1}}\ W_{n_{k-1}-2+(r-1)n_{k-1}}\dots W_{(r-1)n_{k-1}}\right]=\left[\left(F^{(k)}\right)_{r,\cdot}\right]^T \tag{4.4}$$

which is the r-th row of the full matrix $F^{(k)}$ and is exactly the rn_{k-1}-th row of the convolutional matrix (1.3) with $D=n_{k-1}$ and $s=n_kn_{k-1}-1$.

Applying Lemma 2 to the sequence W with $\mathcal{M}=n_kn_{k-1}-1$ and $s^{[k]}\in[2,n_kn_{k-1}]$, we know that there exists a sequence of filter masks $\{w^{(j)}\}_{j=J_{k-1}}^{J_k}$, of equal filter length $s^{[k]}$, with $J_k\le J_{k-1}+\lceil\frac{n_kn_{k-1}-1}{s^{[k]}-1}\rceil$ such that the sequence W has the convolutional factorization $w^{(J_k)}*w^{(J_1-1)}*\dots*w^{(J_{k-1}+1)}$.

Then we construct the bias vectors $\{b^{(j)}\}_{j=J_{k-1}+1}^{J_k-1}$ as in Lemma 3 with $B=\left\|h^{(J_{k-1})}\right\|_\infty=\left\|H^{(k-1)}\right\|_\infty$ and $\hat{B}=0$. Obviously, (3.5) is satisfied and, when $J_k>J_{k-1}+1$, by Lemma 3, for $j=J_{k-1}+1,\dots,J_k-1$, (1.8) is satisfied and

$$h^{(J_k-1)}(x)=T^{(J_k-1)}\dots T^{(J_{k-1}+1)}H^{(k-1)}(x)+\left\|H^{(k-1)}\right\|_\infty\left(\Pi_{p=J_{k-1}+1}^{J_k-1}\|w^{(p)}\|_1\right)\mathbf{1}_{d_{J_k-1}}. \tag{4.5}$$

At the end, we choose $b^{(J_k)}$ as

$$b^{(J_k)}=\begin{cases}\left\|H^{(k-1)}\right\|_\infty\left(\Pi_{p=J_{k-1}+1}^{J_k-1}\|w^{(p)}\|_1\right)T^{(J_k)}\mathbf{1}_{d_{J_k-1}}+\theta^{(k)}, & \text{when } J_k>J_{k-1}+1,\\ \theta^{(k)}, & \text{when } J_k=J_{k-1}+1,\end{cases}$$

where $\theta^{(k)}\in\mathbb{R}^{d_{J_k-1}+s^{[k]}}$ is an arbitrary vector satisfying $\mathcal{D}_{d_{J_{k-1}}}\theta^{(k)}=\hat{b}^{(k)}$, then we have

$$h^{(J_k)}(x)=\mathcal{D}_{d_{J_k-1}}\sigma\left(T^{(J_k)}\dots T^{(J_{k-1}+1)}H^{(k-1)}(x)-\theta^{(k)}\right). \tag{4.6}$$

By Lemma 1,

$$T^{(J_k)}\dots T^{(J_{k-1}+1)}=T^{(J_k,J_{k-1}+1)}=T^W=\left(W_{q-i}\right)_{q=1,\dots,d_{J_k-1}+n_kn_{k-1}-1,i=1,\dots,d_{J_k-1}}.$$

Recall that $d_{J_{k-1}}=n_{k-1}$ and thereby $\mathcal{D}_{d_{J_{k-1}}}=\mathcal{D}_{n_{k-1}}$. So for $r\in\{1,\dots,n_k\}$, the rn_{k-1}-th row of the matrix $T^{(J_k)}\dots T^{(J_{k-1}+1)}$ equals

$$\left[W_{rn_{k-1}-1}\ W_{rn_{k-1}-2}\ \dots W_{rn_{k-1}-n_{k-1}}\right]$$

which is exactly the r-th row $\left[\left(F^{(k)}\right)_{r,\cdot}\right]^T$ of the full matrix $F^{(k)}$ according to (4.4). Combining this with (4.6) yields

$$h^{(J_k)}(x)=\sigma\left(F^{(k)}H^{(k-1)}(x)-\hat{b}^{(k)}\right),$$

which verifies $h^{(J_k)}(x) = H^{(k)}(x)$. Since the vector $T^{(J_k)}\mathbf{1}_{d_{J_{k-1}}}$ satisfies (1.8), we know that the total number of free parameters in realizing $H^{(k)}(x)$ from $H^{(k-1)}(x)$ is at most

$$(3s^{[k]}+2)\lceil\frac{n_k n_{k-1}-1}{s^{[k]}-1}\rceil \le 8n_k n_{k-1}-6 \le 8n_k n_{k-1},$$

where we have used the argument in the proof of Theorem 1 for bounding the number $(3s^{[k]}+2)\lceil\frac{n_k n_{k-1}-1}{s^{[k]}-1}\rceil$. This completes the induction procedure and the proof of Theorem 2. □

4.3 Approximation on Riemannian manifolds by DCNNs

We are in a position to use Theorem 2 and a result from [22] to prove Theorem 3.

Proof of Theorem 3. According to Theorem 5.1 of [22], with an atlas of size $C_\Omega \in \mathbb{N}$ for the manifold Ω, to approximate the function f, one can construct a 3-layer neural network $\{H^{(k)} : \mathbb{R}^d \to \mathbb{R}^{n_k}\}_{k=1}^3$ of widths $n_1 = dC_\Omega$, $n_2 = 8m\sum_{i=1}^{C_\Omega} N_i + 4C_\Omega(d-m)$, $n_3 = 2\sum_{i=1}^{C_\Omega} N_i$, where N_i is the number of wavelet terms used on the i-th chart. Moreover, it was proved on page 549 there that if one chooses all wavelet terms up to scale $K \in \mathbb{N}$, the integer part of $\frac{\log N}{\log 2} - 1$, then $2^{K+1} \le N$, $\sum_{i=1}^{C_\Omega} N_i \le C_\Omega N$ and the widths satisfy $4d \le n_2 \le 8mC_\Omega N + 4C_\Omega(d-m)$ and $2 \le n_3 \le 2C_\Omega N$, and the approximation error can be bounded as

$$\inf_{c\in\mathbb{R}^{n_3}} \left\|f(x) - c\cdot H^{(3)}(x)\right\|_{C(\Omega)} \le A_{f,\Omega,m,d} N^{-2/m}, \tag{4.7}$$

where $A_{f,\Omega,m,d}$ is a positive constant independent of N.

Now for the given integer $s \in [2,d]$ and the layer number $\ell = 3$ of the above network $\{H^{(k)} : \mathbb{R}^d \to \mathbb{R}^{n_k}\}_{k=1}^3$, we take the uniform filter lengths $\mathcal{S} = \{s^{[k]}\}_{k=1}^3$ to be identical to s, and apply Theorem 2. Then we know that there is a uniform downsampled DCNN $\left\{h^{(j)}(x) : \mathbb{R}^d \to \mathbb{R}^{d_j}\right\}_{j=1}^J$ with 3 downsamplings and uniform filter lengths $\{s^{[k]} \equiv s\}_{k=1}^3$, together with bias vectors $b^{(j)}$ satisfying (1.8) for $j \notin \mathcal{J}$ such that $h^{(J)}(x) = H^{(3)}(x)$ for $x \in \Omega$. Combining this with (4.7), we know that there is some $c \in \mathbb{R}^J$ such that

$$\left\|f(x) - c\cdot h^{(J)}(x)\right\|_{C(\Omega)} \le A_{f,\Omega,m,d} N^{-2/m}.$$

Moreover, the downsampling layers $\mathcal{J} = \{J_k\}_{k=1}^3$ are given by $J_k = \sum_{j=1}^k \Delta_j$ with $\Delta_j \le \lceil\frac{n_j n_{j-1}-1}{s-1}\rceil$ satisfying

$$\begin{aligned}
\Delta_1 &\le \lceil\frac{n_1 n_0 - 1}{s^{[1]}-1}\rceil \le \lceil\frac{d^2 C_\Omega - 1}{s-1}\rceil,\\
\Delta_2 &\le \lceil\frac{dC_\Omega^2(8mN+4d-4m)-1}{s-1}\rceil,\\
\Delta_3 &\le \lceil\frac{2C_\Omega^2 N(8mN+4d-4m)-1}{s-1}\rceil,
\end{aligned}$$

which implies $J = \sum_{j=1}^{3} \Delta_j \leq \frac{C_\Omega^2 (d+2N)(8mN+5d)}{s-1} + 3$. The total number of free parameters can be bounded as

$$\mathcal{N} \leq 8 \sum_{k=1}^{3} (n_k n_{k-1}) + n_3 \leq 9 C_\Omega^2 (d+2N)(8mN+5d).$$

To achieve the approximation accuracy $\epsilon \in (0,1)$, we require the total network width of order $O\left((d + m\epsilon^{-m/2})^2/s\right)$ and the total number of free parameters of order $O\left((d + m\epsilon^{-m/2})^2\right)$. This proves Theorem 3. □

Appendix

In this appendix, we prove two lemmas stated in Section 3.

Proof of Lemma 1. For $1 \leq p \leq J_k - J_{k-1}$, we denote

$$W^{(k,p)} := w^{(J_{k-1}+p)} * \ldots * w^{(J_{k-1}+2)} * w^{(J_{k-1}+1)}.$$

It is a sequence supported on $\{0, 1, \ldots, \Delta_{k,p}\}$ where $\Delta_{k,p} = \sum_{j=J_{k-1}+1}^{J_{k-1}+p} s^{(j)}$. Its associated Toeplitz type matrix (1.3)

$$T^{(J_{k-1}+p, J_{k-1}+1)} := \left(W_{i-k}^{(k,p)}\right)_{i=1,\ldots,d_{J_{k-1}}+\Delta_{k,p}, k=1,\ldots,d_{J_{k-1}}}$$

satisfies $\left(T^{(J_{k-1}+p, J_{k-1}+1)}\right)_{i,t} = W_{i-t}^{(k,p)} = 0$ when $i - t > \Delta_{k,p}$.

We prove by induction that $T^{(J_{k-1}+p, J_{k-1}+1)} = T^{(J_{k-1}+p)} \ldots T^{(J_{k-1}+2)} T^{(J_{k-1}+1)}$ for $1 \leq p \leq J_k - J_{k-1}$. The case $p = 1$ is trivial by definition.

Suppose that the identity holds for $p = q < J_k - J_{k-1}$. That is, $T^{(J_{k-1}+q, J_{k-1}+1)} = T^{(J_{k-1}+q)} \ldots T^{(J_{k-1}+1)} \in \mathbb{R}^{(d_{J_{k-1}}+\Delta_{k,q}) \times d_{J_{k-1}}}$. Consider $T^{(J_{k-1}+q+1)} T^{(J_{k-1}+q, J_{k-1}+1)}$, the product with the $(d_{J_{k-1}} + \Delta_{k,q+1}) \times (d_{J_{k-1}} + \Delta_{k,q})$ matrix $T^{(J_{k-1}+q+1)}$. The entry with $1 \leq i \leq d_{J_{k-1}} + \Delta_{k,q+1}$ and $1 \leq j \leq d_{J_{k-1}}$ is given by

$$\begin{aligned} \left(T^{(J_{k-1}+q+1)} T^{(J_{k-1}+q, J_{k-1}+1)}\right)_{i,j} &= \sum_{r=1}^{d_{J_{k-1}}+\Delta_{k,q}} \left(T^{(J_{k-1}+q+1)}\right)_{i,r} \left(T^{(J_{k-1}+q, J_{k-1}+1)}\right)_{r,j} \\ &= \sum_{r=1}^{d_{J_{k-1}}+\Delta_{k,q}} w_{i-r}^{(J_{k-1}+q+1)} W_{r-j}^{(k,q)}. \end{aligned}$$

This equals $\sum_{r \in \mathbb{Z}} w_{i-r}^{(J_{k-1}+q+1)} W_{r-j}^{(k,q)}$, because for $r \in (-\infty, 0] \cup [d_{J_{k-1}} + \Delta_{k,q} + 1, \infty)$, we have $r - j \in (-\infty, -1] \cup [\Delta_{k,q} + 1, \infty)$ which implies $W_{r-j}^{(k,q)} = 0$ from the support of $W^{(k,q)}$. Thus,

$$\left(T^{(J_{k-1}+q+1)} T^{(J_{k-1}+q, J_{k-1}+1)}\right)_{i,j} = \sum_{r \in \mathbb{Z}} w_{i-r}^{(J_{k-1}+q+1)} W_{r-j}^{(k,q)} = \left(w^{(J_{k-1}+q+1)} * \ldots * w^{(J_{k-1}+1)}\right)_{i-j}$$

which is exactly $\left(T^{(J_{k-1}+q+1,J_{k-1}+1)}\right)_{i,j}$. This together with our induction hypothesis verifies the desired identity for $p = q + 1$, and completes the induction procedure. The last identity with $p = J_k - J_{k-1}$ is the equality (3.1). □

Proof of Lemma 3. We first verify (1.8) from (3.6) for $j = J_{k-1} + 1, \ldots, J_k$. For $i = s^{(j)} + 1, \ldots, d_{j-1}$, we have

$$\left(T^{(j)}\mathbf{1}_{d_{j-1}}\right)_i = \sum_{p=1}^{d_{j-1}} \left(T^{(j)}\right)_{i,p} = \sum_{p=1}^{d_{j-1}} w_{i-p}^{(j)}.$$

Observe that $w^{(j)}$ is supported in $\{0, \ldots, s^{(j)}\}$. So for $p \in (-\infty, 0] \cup [d_{j-1} + 1, \infty)$, we have $i - p \in [s^{(j)} + 1, \infty) \cup (\infty, -1]$ which implies $w_{i-p}^{(j)} = 0$. Thus,

$$\left(T^{(j)}\mathbf{1}_{d_{j-1}}\right)_i = \sum_{p=-\infty}^{\infty} w_{i-p}^{(j)} = \sum_{p=-\infty}^{\infty} w_p^{(j)}, \qquad \forall i = s^{(j)} + 1, \ldots, d_{j-1}.$$

This verifies (1.8).

Then we prove (3.7) by induction. For $j = J_{k-1} + 1$, we have

$$T^{(J_{k-1}+1)}h^{(J_{k-1})}(x) - b^{(J_{k-1}+1)} = T^{(J_{k-1}+1)}\left(h^{(J_{k-1})}(x) - \hat{B}\mathbf{1}_{d_{J_{k-1}}}\right) + B\|w^{(J_{k-1}+1)}\|_1\mathbf{1}_{d_{J_{k-1}+1}}.$$

By (3.3) and (3.5), each component of the above function vector takes nonnegative values. But the ReLU σ is the same as the identity function on $[0, \infty)$, hence

$$h^{(J_{k-1}+1)}(x) = T^{(J_{k-1}+1)}\left(h^{(J_{k-1})}(x) - \hat{B}\mathbf{1}_{d_{J_{k-1}}}\right) + B\|w^{(J_{k-1}+1)}\|_1\mathbf{1}_{d_{J_{k-1}+1}},$$

which verifies (3.7) for $j = 1$.

Suppose that (3.7) holds for $j \geq J_{k-1} + 1$ with $j \leq J_k - 1$. Then $d_{j-1} + s^{(j)} = d_j$. By the induction hypothesis and the choice (3.6) of the bias vector, we have

$$h^{(j)}(x) = \sigma\left(T^{(j)}T^{(j-1)}\cdots T^{(J_k+1)}\left(h^{(J_{k-1})}(x) - \hat{B}\mathbf{1}_{d_{J_{k-1}}}\right) + B\left(\Pi_{p=J_{k-1}+1}^{j}\|w^{(p)}\|_1\right)\mathbf{1}_{d_j}\right).$$

By (3.3) and (3.5) again, each component of the above function vector takes nonnegative values, so (3.7) holds true for j. This completes the induction procedure and verifies (3.7).

What is left is to prove (3.8) when $b^{(J_k)}$ is given by by (3.6). Here

$$\begin{aligned} T^{(J_k)}h^{(J_k-1)}(x) - b^{(J_k)} &= T^{(J_k)}\cdots T^{(J_{k-1}+1)}\left(h^{(J_{k-1})}(x) - \hat{B}\mathbf{1}_{d_{J_{k-1}}}\right) \\ &\quad + B\left(\Pi_{p=J_{k-1}+1}^{J_k}\|w^{(p)}\|_1\right)\mathbf{1}_{d_{J_{k-1}+s^{(J_k)}}}. \end{aligned}$$

Once again, we apply (3.3) and (3.5), and find that each component of the above function vector takes nonnegative values. So $\sigma\left(T^{(J_k)}h^{(J_k-1)}(x) - b^{(J_k)}\right)$ equals the above expression, which implies (3.8) by the linearity of the downsampling operator. The proof of the lemma is complete. □

Acknowledgments

The work described in this paper is supported partially by the Research Grants Council of Hong Kong [Project No. CityU 11306617].

References

[1] A. R. Barron, Universal approximation bounds for superpositions of a sigmoidal function, IEEE Trans. Inform. Theory **39** (1993), 930–945.

[2] H. Bölcskei, P. Grohs, G. Kutyniok, and P. Petersen, Optimal approximation with sparsely connected deep neural networks, SIAM Journal on Mathematics of Data Science **1** (2019), 8–45.

[3] C. K. Chui, X. Li, H. N. Mhaskar, Limitations of the approximation capabilities of neural networks with one hidden layer, Adv. Comput. Math. **5** (1996), 233-243.

[4] G. Cybenko, Approximations by superpositions of sigmoidal functions, Math. Control, Signals, and Systems **2** (1989), 303-314.

[5] I. Daubechies, Ten Lectures on Wavelets, SIAM, 1992.

[6] J. Fan, T. Hu, Q. Wu and D. X. Zhou, Consistency analysis of an empirical minimum error entropy algorithm, Appl. Comput. Harmonic Anal. **41** (2016), 164-189.

[7] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning, MIT Press, 2016.

[8] Z. C. Guo, D. H. Xiang, X. Guo, and D. X. Zhou, Thresholded spectral algorithms for sparse approximations, Anal. Appl. **15** (2017), 433–455.

[9] G. E. Hinton, S. Osindero, Y. W. Teh, A fast learning algorithm for deep belief nets, Neural Comput. **18** (2006), 1527-1554.

[10] K. Hornik, M. Stinchcombe, and H. White, Multilayer feedforward networks are universal approximators, Neural networks **2** (1989), 359366.

[11] J. Klusowski and A. Barron, Approximation by combinations of ReLU and squared ReLU ridge functions with ℓ^1 and ℓ^0 controls, IEEE Transactions on Information Theory **64** (2018), 7649–7656.

[12] M. Kohler, A. Krzyżak, Adaptive regression estimation with multilayer feedforward neural networks, J. Nonparametric Statis. **17** (2005), 891-913.

[13] A. Krizhevsky, I. Sutskever, and G. Hinton G, Imagenet classification with deep convolutional neural networks, *NIPS* (2012): 2097-1105.

[14] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, Gradient-based learning applied to document recognition, Proceedings of the IEEE **86** (1998), 2278-2324.

[15] M. Leshno, Y. V. Lin, A. Pinkus, and S. Schocken, Multilayer feedforward networks with a non-polynomial activation function can approximate any function, Neural Networks **6** (1993), 861-867.

[16] S. B. Lin and D. X. Zhou, Distributed kernel gradient descent algorithms, Constructive Approximation **47** (2018), 249-276.

[17] V. Maiorov, On best approximation by ridge functions in the uniform norm, Constructive Approximation **18** (2002), 61-85.

[18] S. Mallat, Understanding deep convolutional networks, Phil. Trans. Royal Soc. A **374**:20150203.

[19] H. N. Mhaskar, Approximation properties of a multilayered feedforward artificial neural network, Adv. Comput. Math. **1** (1993), 61-80.

[20] P. Petersen and V. Voigtlaender, Optimal approximation of piecewise smooth functions using deep ReLU neural networks, Neural Networks **108** (2018), 296–330.

[21] P. Petersen and F. Voigtlaender, Equivalence of approximation by convolutional neural networks and fully-connected networks, Proc. Amer. Math. Soc., in press. arXiv:1809.00973, 2018.

[22] U. Shaham, A. Cloninger, and R. Coifman, Provable approximation properties for deep neural networks, Applied and Computational Harmonic Analysis **44** (2018), 537–557.

[23] I. Steinwart and A. Christmann, Support Vector Machines, Springer, New York, 2008.

[24] M. Telgarsky, Benefits of depth in neural networks. *29th Annual Conference on Learning Theory* PMLR 49 (2016): 1517–1539.

[25] D. Yarotsky, Error bounds for approximations with deep ReLU networks, Neural Networks **94** (2017), 103–114.

[26] Y. Ying and D. X. Zhou, Unregularized online learning algorithms with general loss functions, Appl. Comput. Harmonic Anal. **42**(2017), 224–244.

[27] Y. C. Zhang, J. Duchi, and M. Wainwright, Divide and conquer kernel ridge regression: A distributed algorithm with minimax optimal rates, J. Mach. Learn. Res. **16** (2015), 3299-3340.

[28] D. X. Zhou, Deep distributed convolutional neural networks: universality, Anal. Appl. **16** (2018), 895–919.

[29] D. X. Zhou, Universality of deep convolutional neural networks, Appl. Comput. Harmonic Anal., in press. arXiv:1805.10769v2

[30] D. X. Zhou, Distributed approximation with deep convolutional neural networks, submitted, 2018.