Refinement and Universal Approximation via Sparsely Connected ReLU Convolution Nets

Andreas Heinecke, Jinn Ho, and Wen-Liang Hwang, *Senior Member, IEEE*

***Abstract*—We construct a highly regular and simple structured class of sparsely connected convolutional neural networks with rectifier activations that provide universal function approximation in a coarse-to-fine manner with increasing number of layers. The networks are localized in the sense that local changes in the function to be approximated only require local changes in the final layer of weights. At the core of the construction lies the fact that the characteristic function can be derived from a convolution of characteristic functions at the next coarser resolution via a rectifier passing. The latter refinement result holds for all higher order univariate B-splines.**

***Index Terms*—Function approximation, neural networks, splines.**

I. Introduction

DEEP neural networks have brought breakthrough successes in many machine learning applications and much effort is invested towards improving the understanding of the empirically observed advantages of deep over shallow networks. Universality results guarantee that any function in $L^p(\mathbb{R}^n)$ (the space of Lebesgue integrable functions with $\|f\|_{L_q(\mathbb{R}^d)} := (\int |f|^q d\lambda)^{1/q} < \infty$ for $1 \leq q < \infty$ and finite essential supremum in case $q = \infty$) can be approximated arbitrarily well by shallow/deep neural networks provided one allows a sufficiently large number of neurons [1], [2], in particular for ReLU networks [3] in which the activation is the widely used rectifier [4], [5]. For shallow networks there is little connection between network properties and structure as all neurons are aligned in a single hidden layer. For deep networks, in contrast, arguments relating specific network architectures and properties have gained attention from many perspectives (e.g., philosophical [6], empirical [7] or approximation theoretical [8]). Bengio [9] argues that, say, an input image is transformed along the network layers into gradually higher level features, representing increasingly more abstract functions of the raw input. In a similar direction, theoretical results show that additional network layers refine the partition of the input space of the network [10], [11]. These results motivate us to study the property of coarse-to-fine function approximation and its realization via the network layer architecture. Is it possible to construct a universal deep network representation that can approximate any function along the layers with gradually improving accuracy? Answering this question by providing a construction not only improves the understanding of the role played by depth, but also provides hints on how to encode certain properties into the structure of a network. In this article, we demonstrate a step towards understanding the structure of deep networks via the property of the coarse-to-fine function approximation. To be precise, we consider deep feedforward ReLU networks $M_L \circ \rho \circ M_{L-1} \circ \cdots \circ \rho \circ M_1$ consisting of L of layers of affine linear operators $\{M_\ell : \mathbb{R}^{N_{\ell-1}} \to \mathbb{R}^{N_\ell}\}_{\ell=1}^{L}$ (widths $N_0, \ldots, N_L$) with the pointwise acting piecewise linear rectifier activation $\rho(x) = \max\{0, x\}$.

- We present a short and direct construction of a particularly simple structured class of convolutional ReLU networks that provide universal approximation in a coarse-to-fine manner with increasing number of layers.
- The network is sparsely connected (the number of outgoing edges from any node in the hidden layers is two) and localized in the sense that local changes in the function to be approximated only require local changes in the final layer of weights.
- The Fourier and wavelet approach to function approximation and analysis is the use of basis functions. In the classical work on universality of shallow networks, basis functions and function values can be separated in the sense that basis functions are encoded via the activations while function values are specified via weights, providing a uniform way to approximate functions [12]. Our construction follows this spirit, with basis functions encoded in the network structure and function values specified in the final layer weights. The basis functions in our construction are unit interval characteristic functions and their shifts. We generalize our central observation on characteristic functions to all higher order univariate B-splines [13], showing that each can be derived via rectification from linear combinations of its shifts on the next coarser dyadic dilation level.

While there is an extensive literature on universality of deep ReLU networks, their focus is usually not on properties such as coarse-to-fine approximation. For instance, [3] uses max-min representations of continuous piecewise linear functions

Manuscript received February 26, 2020; revised May 29, 2020; accepted June 12, 2020. Date of publication June 25, 2020; date of current version July 17, 2020. The associate editor coordinating the review of this manuscript and approving it for publication was Saurabh Prasad. *(Corresponding author: Wen-Liang Hwang.)*

Andreas Heinecke is with the Yale-NUS College, Singapore 138527 (e-mail: andreas.heinecke@yale-nus.edu.sg).

Jinn Ho and Wen-Liang Hwang are with the Institute of Information Science, Academia Sinica, Taipei 11529, Taiwan (e-mail: hjinn@iis.sinica.edu.tw; whwang@iis.sinica.edu.tw).

This letter has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the authors.

Digital Object Identifier 10.1109/LSP.2020.3005051

(see [14], [15]) to show universality of ReLU networks. Finding these representations is combinatorial in nature, thus computationally expensive. Moreover, the derived networks are not stable in the sense that they can differ significantly for a pair of slightly differing functions. Another example, and perhaps most closely related our construction, is [16] which uses a partition of unity based construction and approximates the multiplication operation via rectifier activation to transition to multivariate functions. In contrast, the network we construct here has convolution structure and yields coarse-to-fine approximation. Other authors focus on the approximation of certain classes of functions, e.g., convex functions [17] or high dimensional bandlimited functions [18]. Universality of convolutional ReLU networks is discussed in [19] via arguments from Fourier analysis, or in [20], which show that convolutional nets with rectifier activation and max-pooling are universal and benefit from depth. The networks constructed are usually fully connected and local changes in the function to be approximated may affect the entire network. The theoretically achievable sparsity of ReLU networks, with in general unbounded width, is for instance studied in [21], [22]. Examples of results arguing specifically for the benefits of depth with respect to complexity are [23]–[25].

II. Deep ReLU Convolution Networks and Scaling

Let $C(\mathbb{R}^d)$ denote the space of continuous real valued functions on $\mathbb{R}^d$ and let $f \in C(\mathbb{R}^d)$ be compactly supported. Without loss of generality, assume that f is supported in $[0,1)^d$ and for $J \in \mathbb{N}$ consider the piecewise constant 2^J-resolution approximation of f defined by

$$f^J := \sum_{k_1,\ldots,k_d=0}^{2^J-1} f\left(\tfrac{k_1}{2^J},\ldots,\tfrac{k_d}{2^J}\right) \chi_{\left[\frac{k_1}{2^J},\frac{k_1+1}{2^J}\right)\times\cdots\times\left[\frac{k_d}{2^J},\frac{k_d+1}{2^J}\right)}, \tag{1}$$

where χ_S denotes the characteristic function of $S \subset \mathbb{R}^d$, given by $\chi_S(\mathbf{x}) = 1$ if $\mathbf{x} \in S$ and $\chi_S(\mathbf{x}) = 0$ otherwise.

A. Univariate Function Approximation

The central observation for the univariate case $d = 1$ is that for each $k \in \{0\ldots, 2^J - 1\}$ each characteristic function in (1) can be constructed from two characteristic functions of doubled support length and one rectifier passing as

$$\chi_{\left[\frac{k}{2^J},\frac{k+1}{2^J}\right)} = \rho \circ \left(\chi_{\left[\frac{k}{2^J},\frac{k+2}{2^J}\right)} - \chi_{\left[\frac{k+1}{2^J},\frac{k+3}{2^J}\right)}\right). \tag{2}$$

This coarse-to-fine refinement process is illustrated in Fig. 1. The refinement process also implies that doubling the approximation precision of a function requires the depth of the network to increase by one and its width to increase exponentially in the number of layers. The number of nodes at resolution 2^1 is roughly doubled to support the refinement of approximating a function at resolution 2^2 using (2). Hence, refining the approximation from 2^2 to 2^3 increases the network depth by one, doubles the layer width at resolutions 2^2, and quadruples it at resolution 2^1.

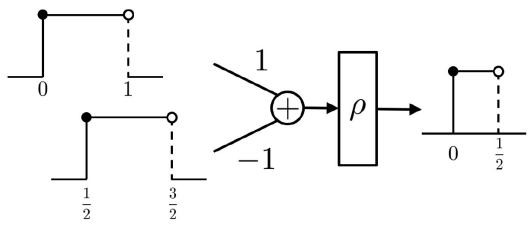

Fig. 1. The rectifier ρ refines the 2^0-resolution characteristic functions $\chi_{[0,1)}$ and $\chi_{[\frac{1}{2},\frac{3}{2})}$ (obtainable by a shift corresponding to bias $-1/2$) to resolution 2^1.

Returning to (2), repeated application yields that $\chi_{\left[\frac{k}{2^J},\frac{k+1}{2^J}\right)}$ is equal to

$$\rho \circ \left[\rho \circ \left(\chi_{\left[\frac{k}{2^J},\frac{k+4}{2^J}\right)} - \chi_{\left[\frac{k+2}{2^J},\frac{k+6}{2^J}\right)}\right) - \rho \circ \left(\chi_{\left[\frac{k+1}{2^J},\frac{k+5}{2^J}\right)} - \chi_{\left[\frac{k+3}{2^J},\frac{k+7}{2^J}\right)}\right)\right], \tag{3}$$

and, iterating this procedure, each characteristic function in (1) can be constructed from shifts of $\chi_{[0,1)}$ via J layers of rectifier activations.

To construct a ReLU convolutional network from the above observation, let $\mathbf{b}_0 := [1,-1]^\top$ and, for $j = 1,\ldots,J$, derive $\mathbf{b}_j$ from $\mathbf{b}_{j-1}$ by inserting zeros between every two entries, i.e., $\mathbf{b}_j := [1,0,\ldots,0,-1]^\top \in \mathbb{R}^{2^j+1}$. Using $|\cdot|$ to denote the length of a vector (and for consistency of notation letting $|\mathbf{b}_{-1}| := 1$), define the vector-valued characteristic functions

$$\boldsymbol{\chi}_{J-j}^{(J)} := \left[\chi_{\left[\frac{k}{2^J},\frac{k+2^j}{2^J}\right)}\right]_{k=0,\ldots,2^J-1+\sum_{k=0}^{j}(|\mathbf{b}_{k-1}|-1)}^{\top} \tag{4}$$

for $j = 0,\ldots,J$. Each of these vector valued functions contains as components all consecutive 2^{-J}-translates of a characteristic function needed to combine all consecutive 2^{-J}-translates of the characteristic functions of half support length via a rectifier passing as in (2). Defining for $j = 0,\ldots,J-1$ the circulant matrix $\mathbf{C}_{J-j}^{(J)}(\mathbf{b}_j) \in \mathbb{R}^{|\boldsymbol{\chi}_{J-j}^{(J)}|\times|\boldsymbol{\chi}_{J-(j+1)}^{(J)}|}$ by

$$\begin{bmatrix} 1 & 0 & \cdots & 0 & -1 & & & & \\ & 1 & 0 & \cdots & 0 & -1 & & & \\ & & \ddots & \ddots & \cdots & \ddots & \ddots & & \\ & & & \ddots & \ddots & \cdots & \ddots & \ddots & \\ & & & & 1 & 0 & \cdots & 0 & -1 \\ & & & & & 1 & 0 & \cdots & 0 & -1 \end{bmatrix},$$

i.e., with rows given by the (zero extensions of) $\mathbf{b}_j$, the recursion between characteristic vectors at two consecutive resolutions is

$$\boldsymbol{\chi}_{J-j}^{(J)} = \rho \circ \mathbf{C}_{J-j}^{(J)}(\mathbf{b}_j) \circ \boldsymbol{\chi}_{J-(j+1)}^{(J)} \tag{5}$$

for $j = 0,\ldots,J-1$. This motivates the following definition.

Definition 1: Given a depth $J \in \mathbb{N}$ define

$$\mathrm{CNN}_{\rho,J} := \rho \circ \mathbf{C}_J^{(J)}(\mathbf{b}_0) \circ \cdots \circ \rho \circ \mathbf{C}_1^{(J)}(\mathbf{b}_{J-1}), \tag{6}$$

and $\mathrm{CNN}_{\rho,J|_j}$ to be the first j layers of $\mathrm{CNN}_{\rho,J}$.

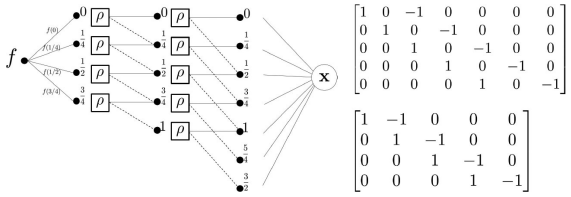

Fig. 2. ReLU network approximating $f(\mathbf{x})$ at resolution 2^2 via $\mathbf{f}_2^\top \circ \rho \circ \mathbf{C}_2^{(2)}(\mathbf{b}_0) \circ \rho \circ \mathbf{C}_1^{(2)}(\mathbf{b}_1)\mathbf{x}$. Weights on solid and dashed edges are 1 and -1, respectively; $\mathbf{C}_1^{(2)}$ and $\mathbf{C}_2^{(2)}$ are the top and bottom matrices on the right, i.e., $\mathbf{b}_1 = [1, 0, -1]^\top$, $\mathbf{b}_0 = [1, -1]^\top$; and $\mathbf{f}_2 = [f(0), f(\frac{1}{4}), f(\frac{1}{2}), f(\frac{3}{4})]^\top$. Numbers at black nodes indicate left end of supports of the respective characteristic functions to be processed by the rectifier.

Via this fixed architecture, the approximations f^j of f ($j = 1, \ldots, J$) can be represented as compositions of a j-layer convolutional neural network with the characteristic function vector $\boldsymbol{\chi}_0$. The first layer of $\mathrm{CNN}_{\rho,J}$ computes a discrete convolution of $\mathbf{b}_{J-1}$ with $\boldsymbol{\chi}_0^{(J)}$, followed by componentwise rectification. The result is consecutively passed, i.e., refined, through all layers via convolution and rectification to yield $\boldsymbol{\chi}_1^{(J)}, \boldsymbol{\chi}_2^{(J)}, \ldots$, and eventually $\boldsymbol{\chi}_J^{(J)} = \mathrm{CNN}_{\rho,J} \circ \boldsymbol{\chi}_0^{(J)}$. In the course of this coarse-to-fine procedure only the positions of the two non-zero entries 1 and -1 of the convolution filters $\mathbf{b}_j$ differ in that their distance is (essentially) doubled between consecutive layers by inserting zeros between every two entries. The coarse-to-fine architecture is only dependent on J and consecutively adds convolutional layers in order to increase resolution, meanwhile leaving earlier layers unchanged, before in a final layer the information about the specific function that is to be approximated enters. This is the content of the following result, which is illustrated in Fig. 2.

Theorem 2: Let $f \in C(\mathbb{R})$ be supported in [0,1) and $J \in \mathbb{N}$. Then for all $j = 1, \ldots, J$,

$$f^j = \mathbf{f}_j^\top \circ \mathrm{CNN}_{\rho,J|_j} \circ \boldsymbol{\chi}_0^{(J)}, \tag{7}$$

where

$$\mathbf{f}_J := [f(0), f(1/2^J), \ldots, f((2^J - 1)/2^J)]^\top$$

and $\mathbf{f}_j$ is derived from $\mathbf{f}_{j+1}$ by replacing its k-th entry by zero whenever $k \notin 2^{-(J-j)}\mathbb{N} \cup \{0\}$ and concatenating zeros until $\mathbf{f}_j \in \mathbb{R}^{|\boldsymbol{\chi}_j^{(J)}|}$. The total number of edges in the convolution layers of (7) is $\mathcal{O}(2^{2j})$.

Proof: The component functions in all but the last characteristic vector valued function have overlapping support. Therefore, $f^j = \mathbf{f}_j^\top \circ \boldsymbol{\chi}_j^{(J)}$ for $j = 0, \ldots, J$, and (7) follows recursively in combination with (5). It follows from (2) that the number of edges/nodes needed to obtain a characteristic function at resolution 2^j is recursively given by $N_j = 2N_{j-1}$, where $N_1 = 2$, thus is of order $\mathcal{O}(2^j)$. Since $\boldsymbol{\chi}_j^{(J)}$ has $\mathcal{O}(2^j)$ components, $\mathcal{O}(2^{2j})$ edges are needed for the approximation f^j. ∎

The network in (7) is composed with a vector valued characteristic function. ReLU networks represent continuous piecewise linear functions and therefore can only approximate characteristic functions. We next consider a canonical approximation of $\boldsymbol{\chi}_0^{(J)}$ by a vector valued function $\widetilde{\boldsymbol{\chi}}_0^{(J)}$ of same dimension whose component functions are continuous piecewise linear, and observe how the resulting error propagates through the convolution layers. Given an error margin $\eta < 1/2^{J+1}$, let $\widetilde{\chi}_{0,0} := \widetilde{\chi}_{[0,1)}$ be the continuous piecewise linear function that coincides with $\chi_{0,0} := \chi_{[0,1)}$ outside $[-\frac{\eta}{2}, \frac{\eta}{2}] \cup [1 - \frac{\eta}{2}, 1 + \frac{\eta}{2}]$ and is linear on $[-\frac{\eta}{2}, \frac{\eta}{2}]$ and $[1 - \frac{\eta}{2}, 1 + \frac{\eta}{2}]$. Approximate the remaining components $\chi_{0,k}$ of $\boldsymbol{\chi}_0^{(J)}$ by the appropriate shifts $\widetilde{\chi}_{0,k}$ of $\widetilde{\chi}_{0,0}$. Then $\|\chi_{0,k} - \widetilde{\chi}_{0,k}\|_{L_1(\mathbb{R})} = \eta$ for all component approximation errors.

Lemma 3: For $j = 1, \ldots, J$, let $\widetilde{\boldsymbol{\chi}}_j^{(J)} \in \mathbb{R}^{|\boldsymbol{\chi}_j^{(J)}|}$ be defined recursively to have component functions $\widetilde{\chi}_{j,k} := \rho \circ (\widetilde{\chi}_{j-1,k} - \widetilde{\chi}_{j-1,k+1})$. Then

$$\left\|\boldsymbol{\chi}_J^{(J)} - \widetilde{\boldsymbol{\chi}}_J^{(J)}\right\|_1 := \sum_{k=0}^{2^J-1} \left\|\chi_{[\frac{k}{2^J}, \frac{k+1}{2^J})} - \widetilde{\chi}_{J,k}\right\|_{L_1(\mathbb{R})} \leq 2^{2J} \left\|\chi_{[0,1)} - \widetilde{\chi}_{[0,1)}\right\|_{L_1(\mathbb{R})}. \tag{8}$$

Proof: The rectifier is non-decreasing and satisfies $\rho(x + |y|) \leq \rho(x) + |y|$ for all $x, y \in \mathbb{R}$. Letting $e_{j,k} := \widetilde{\chi}_{j,k} - \chi_{j,k}$, we thus have the pointwise estimate

$$\begin{aligned}
\widetilde{\chi}_{j,k} &= \rho \circ (\widetilde{\chi}_{j-1,k} - \widetilde{\chi}_{j-1,k+1}) \\
&= \rho \circ (\chi_{j-1,k} - \chi_{j-1,k+1} + e_{j-1,k} - e_{j-1,k+1}) \\
&\leq \rho \circ (\chi_{j-1,k} - \chi_{j-1,k+1} + |e_{j-1,k}| + |e_{j-1,k+1}|) \\
&\leq \rho \circ (\chi_{j-1,k} - \chi_{j-1,k+1}) + |e_{j-1,k}| + |e_{j-1,k+1}| \\
&= \chi_{j,k} + |e_{j-1,k}| + |e_{j-1,k+1}|.
\end{aligned}$$

Thus, $|e_{j,k}| \leq |e_{j-1,k}| + |e_{j-1,k+1}|$ and the support of $e_{j,k}$ is contained in the supports of $e_{j-1,k}$ and $e_{j-1,k+1}$. Therefore,

$$\begin{aligned}
\|e_{j,k}\|_{L_1(\mathbb{R})} &\leq \|e_{j-1,k}\|_{L_1(\mathbb{R})} + \|e_{j-1,k+1}\|_{L_1(\mathbb{R})} \\
&= 2\|e_{j-1,k}\|_{L_1(\mathbb{R})} \\
&\leq 2^j \|\widetilde{\chi}_{[0,1)} - \chi_{[0,1)}\|_{L_1(\mathbb{R})},
\end{aligned}$$

and (8) follows since $\boldsymbol{\chi}_J^{(J)}$ has 2^J components. ∎

The continuous piecewise linear approximation $\widetilde{\boldsymbol{\chi}}_0^{(J)}$ can for instance be realized by parallel combining ReLU nets with one hidden layer and width four [3]. In this case, those nets only differ by the bias terms. Their weights and number of parallel repititions depend on J. We denote by $\mathrm{CNN}_{\rho,J,\chi}$ the composition of the neural net realization of $\widetilde{\boldsymbol{\chi}}_0^{(J)}$ with $\mathrm{CNN}_{\rho,J}$.

Theorem 4: If $f \in C(\mathbb{R})$ is supported in [0,1) and $J \in \mathbb{N}$ then

$$\begin{aligned}
&\|f^J - \mathbf{f}_J^\top \circ \mathrm{CNN}_{\rho,J,\chi}\|_{L_1(\mathbb{R})} \\
&\quad \leq 2^{2J} \|f\|_\infty \|\widetilde{\chi}_{[0,1)} - \chi_{[0,1)}\|_{L_1(\mathbb{R})}.
\end{aligned}$$

Proof: The estimate follows combining

$$\begin{aligned}
\|f^J - \mathbf{f}_J^\top \circ \mathrm{CNN}_{\rho,J,\chi}\|_{L_1(\mathbb{R})} &= \|\mathbf{f}_J^\top \circ (\boldsymbol{\chi}_J^{(J)} - \widetilde{\boldsymbol{\chi}}_J^{(J)})\|_{L_1(\mathbb{R})} \\
&\leq \|f\|_\infty \|\boldsymbol{\chi}_J^{(J)} - \widetilde{\boldsymbol{\chi}}_J^{(J)}\|_1.
\end{aligned}$$

and (8). ∎

Theorem 4 and a density argument imply that any compactly supported $f \in L_q(\mathbb{R})$ can be arbitrarily well approximated by a ReLU convolutional network in a localized and coarse-to-fine

manner. The network separates the resolution geometry and function values. The input layer is constructed to approximate shifts of characteristic functions, while the output layer carries the actual functional values as weights. The intermediate hidden layers have a scaled convolution structure in which the 2-tap convolution filters are consecutively downsampled by a factor of two. We summarize this in the following universality theorem.

Corollary 5: Let $1 \leq q \leq \infty$ and $f \in L_q(\mathbb{R})$ supported in [0,1). Then for every $\epsilon > 0$ there exists a depth $J \in \mathbb{N}$ and a vector $\boldsymbol{\omega}$ of real weights such that

$$\|f - \boldsymbol{\omega}^\top \circ \mathrm{CNN}_{\rho,J,\chi}\|_{L_1(\mathbb{R})} < \epsilon.$$

Proof: By compactness of the support, $f \in L_1(\mathbb{R})$. Given $\epsilon > 0$, choose f^J with $\|f - f^J\|_{L_1(\mathbb{R})} \leq \epsilon/2$. The result follows choosing $\boldsymbol{\omega} = \mathbf{f}_J$ and $\eta = \|\widetilde{\chi}_{[0,1)} - \chi_{[0,1)}\|_{L_1(\mathbb{R})} < \min\{2^{-(J+1)}, \epsilon 2^{-2J-1}\|f^J\|_\infty^{-1}\}$. ■

Remarks: (i) A coarse-to-fine relations like (2) hold for several other activations. For instance, soft-thresholding, defined by $\theta(t) = t - 1$ if $t > 1$, $\theta(t) = t + 1$ if $t < -1$, and $\theta(t) = 0$ otherwise, satisfies

$$\chi_{[\frac{k}{2^J}, \frac{k+1}{2^J})} = \theta\left(2\chi_{[\frac{k}{2^J}, \frac{k+2}{2^J})} - \chi_{[\frac{k+1}{2^J}, \frac{k+3}{2^J})}\right).$$

(ii) The rectifier ρ facilitates refinement in the sense of (2) for univariate B-splines of any order $m \in \{2, 3, \ldots\}$, given by

$$\phi(x) := \tfrac{1}{(m-1)!} \sum_{k=0}^{m} (-1)^k \binom{m}{k} (x-k)_+^{m-1} \text{ for } x \in \mathbb{R}, \quad (9)$$

where $x_+^n := (\max\{0, x\})^n$. B-splines are important in approximation theory [13] and as refinable functions in wavelet analysis [26]. The B-spline of order m is $(m-2)$-times continuously differentiable, piecewise polynomial of degree at most $m-1$ and a partition of unity. Denoting the integer shifts of its next coarser dyadic dilation by

$$\phi_i = \tfrac{1}{2}\phi\left(\tfrac{\cdot - i}{2}\right) \quad \text{for } i \in \mathbb{Z}, \quad (10)$$

the rectifier provides the following coarse-to-fine relationship for the B-spline ϕ of order m. (Proof provided in Supplementary Material.)

Proposition 6: There exist weights $w_0, \ldots, w_m \in \mathbb{R}$, such that

$$\phi = \rho \circ \left(\sum_{i=0}^{m} w_i \phi_i\right). \quad (11)$$

B. Multivariate Function Approximation

Equation (2), underlying the univariate construction of the previous Section, can be generalized to the multivariate case. E.g., in the bivariate case $d = 2$ the characteristic function $\chi_{[0,\frac{1}{2^J})\times[0,\frac{1}{2^J})}$ is equal to

$$\rho\Big(\chi_{[0,\frac{2}{2^J})\times[0,\frac{2}{2^J})} - \chi_{[0,\frac{2}{2^J})\times[\frac{1}{2^J},\frac{3}{2^J})} - \chi_{[\frac{1}{2^J},\frac{3}{2^J})\times[0,\frac{2}{2^J})} - \chi_{[\frac{1}{2^J},\frac{3}{2^J})\times[\frac{1}{2^J},\frac{3}{2^J})}\Big), \quad (12)$$

which, as in (3), may be repeatedly applied to yield a multilayer convolutional coarse-to-fine construction, see Fig. 3.

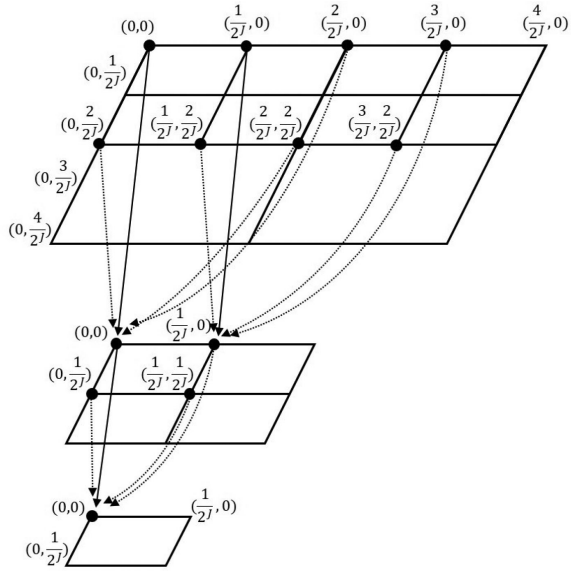

Fig. 3. Coarse-to-fine architecture for the bivariate case. Each characteristic function, here indicated by the upper-left corner of its support, can be derived from four characteristic functions at the next coarser level via a rectifier passing. E.g., the characteristic function of $[0, 1/2^J) \times [0, 1/2^J)$ is indicated at the lowest level by the point (0,0). It can be derived from four characteristic functions at the next coarser level and a rectifier passing as in (12). The point (0,0) at the second level represents $[0, 2/2^J) \times [0, 2/2^J)$, which can be derived from the difference of the characteristic function of $[0, 4/2^J) \times [0, 4/2^J)$ and its three indicated shifts via a rectifier passing.

Considering the appropriate hypercube shifts, (12) generalizes to arbitrary dimension d.

Another possible way to transition to higher dimension is by decomposing characteristic functions of d-dimensional cartesian products into sums of one-dimensional characteristic functions via a rectifier passing. E.g., for $\mathbf{x} = [x_1, x_2]^\top \in \mathbb{R}^2$, $\chi_{[0,\frac{1}{2^J})\times[0,\frac{1}{2^J})}(\mathbf{x})$ is equal to

$$2\rho\left(\tfrac{1}{2}\left(\chi_{[0,\frac{1}{2^J})}(x_1) + \chi_{[0,\frac{1}{2^J})}(x_2)\right) - \tfrac{1}{2}\right),$$

which for $\mathbf{x} = [x_1, \ldots, x_d]^\top \in \mathbb{R}^d$ generalizes to

$$\chi_{[\frac{k_1}{2^J},\frac{k_1+1}{2^J})\times\cdots\times[\frac{k_d}{2^J},\frac{k_d+1}{2^J})}(\mathbf{x}) = d\rho\left(\tfrac{1}{d}\sum_{i=1}^{d} \chi_{[\frac{k_i}{2^J},\frac{k_i+1}{2^J})}(x_i) - \tfrac{d-1}{d}\right)$$

for the characteristic functions of all hypercubes indexed by $k_1, \ldots, k_d \in \{0, \ldots, 2^J - 1\}$. Therefore, $f^J(\mathbf{x})$ is equal to

$$\sum_{k_1,\ldots,k_d=0}^{2^J-1} f\left(\tfrac{k_1}{2^J}, \ldots, \tfrac{k_d}{2^J}\right) d\rho\left(\tfrac{1}{d}\sum_{i=1}^{d} \chi_{[\frac{k_i}{2^J},\frac{k_i+1}{2^J})}(x_i) - \tfrac{d-1}{d}\right).$$

The 2^J-resolution approximation of a compactly supported $f \in C(\mathbb{R}^d)$ can therefore be approximated by combining convolutional neural networks of the form $\mathrm{CNN}_{\rho,J,\chi}$, and a final convolutional layer of weights incorporating the function values. Generalizating Corollary 5 to arbitrary dimension, thus any compact supported $f \in L_q(\mathbb{R}^d)$ $(1 \leq q \leq \infty)$ can be approximated in $L_1(\mathbb{R}^d)$ to arbitrary precision by a deep ReLU network of convolution structure.

REFERENCES

[1] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Math. Control, Signals Syst.*, vol. 2, no. 4, pp. 303–314, Dec. 1989.

[2] K. Hornik, "Approximation capabilities of multilayer feedforward networks," *Neural Netw.*, vol. 4, no. 2, pp. 251–257, 1991.

[3] R. Arora, A. Basu, P. Mianjy, and A. Mukherjee, "Understanding deep neural networks with rectified linear units," 2016, [Online]. Available: https://arxiv. org/abs/1611.01491

[4] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. 27th Int. Conf. Mach. Learn.*, 2010, pp. 807–814.

[5] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proc. 14th Int. Conf. Artif. Intell. Statis.*, 2011, pp. 315–323.

[6] L. N. Hoang and R. Guerraoui, "Deep learning works in practice. But does it work in theory?" 2018, *arXiv:1801.10437*.

[7] C. Zhang, S. Bengio, and Y. Singer, "Are all layers created equal?" 2019, *arXiv:1902.01996*.

[8] T. Poggio, H. Mhaskar, L. Rosasco, B. Miranda, and Q. Liao, "Why and when can deep-but not shallow-networks avoid the curse of dimensionality: A review," *Int. J. Autom. Comput.*, vol. 14, no. 5, pp. 503–519, Oct. 2017.

[9] Y. Bengio, "Learning deep architectures for AI," *Found. Trends Mach. Learn.*, vol. 2, no. 1, pp. 1–127, Jan. 2009.

[10] W.-L. Hwang and A. Heinecke, "Un-rectifying non-linear networks for signal representation," *IEEE Trans. Signal Process.*, vol. 68, pp. 196–210, 2020.

[11] R. Balestriero, R. Cosentino, B. Aazhang, and R. Baraniuk, "The geometry of deep networks: Power diagram subdivision," *Advances Neural Inf. Process. Syst.*, vol. 32, pp. 15 806–15 815, 2019.

[12] M. A. Nielsen, *Neural Networks and Deep Learning*, vol. 25, San Francisco, CA, USA: Determination Press, 2015.

[13] C. de Boor, *A Practical Guide to Splines*. New York, NY, USA: Springer-Verlag, 1978.

[14] S. Ovchinnikov, "Max-min representation of piecewise linear functions," *Contr. Algebra Geometry*, vol. 43, no. 1, pp. 297–302, 2002.

[15] S. Wang and X. Sun, "Generalization of hinging hyperplanes," *IEEE Trans. Inf. Theory*, vol. 51, no. 12, pp. 4425–4431, Dec. 2005.

[16] D. Yarotsky, "Error bounds for approximations with deep ReLU networks," *Neural Netw.*, vol. 94, pp. 103–114, Oct. 2017.

[17] B. Hanin, "Universal function approximation by deep neural nets with bounded width and ReLU activations," *Mathematics*, vol. 7, no. 10, pp. 992.1–992.9, 2019.

[18] H. Montanelli, H. Yang, and Q. Du, "Deep ReLU networks overcome the curse of dimensionality for bandlimited functions," 2019, *arXiv:1903.00735*.

[19] D.-X. Zhou, " Deep distributed convolutional neural networks: Universality," *Anal. Appl.*, vol. 16, no. 6, pp. 895–919, Mar. 2018.

[20] N. Cohen and A. Shashua, "Convolutional rectifier networks as generalized tensor decompositions," in *Proc. 33rd Int. Conf. Mach. Learn.*, 2016, pp. 955–963.

[21] H. Bölcskei, P. Grohs, G. Kutyniok, and P. Petersen, "Optimal approximation with sparsely connected deep neural networks," *SIAM J. Math. Data Sci.*, vol. 1, no. 1, pp. 8–45, May 2019.

[22] P. Petersen and F. Voigtländer, "Optimal approximation of piecewise smooth functions using deep ReLU neural networks," *Neural Netw.*, vol. 108, pp. 296–330, Sep. 2018.

[23] M. Telgarsky, "Benefits of depth in neural networks," in *Proc. 29th Annu. Conf. Learn. Theory*, 2016, pp. 1517–1539.

[24] R. Eldan and O. Shamir, "The power of depth for feedforward neural networks," in *Proc. 29th Annu. Conf. Learn. Theory*, 2016, pp. 907–940.

[25] S. Liang and R. Srikant, "Why deep neural networks for function approximation?" in *Proc. 5th Int. Conf. Learn. Representations*, Toulon, France, 2017, pp. 1–17.

[26] B. Dong and Z. Shen, "MRA-based wavelet frames and applications," *IAS Lecture Notes Series*, Summer Program on "The Mathematics of Image Processing," Park City Mathematics Institute, vol. 19, 2010.