Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Seminar for Applied Mathematics SAM

Exponential ReLU DNN expression of holomorphic maps in high dimension

J. A. A. Opschoor and Ch. Schwab and J. Zech

Research Report No. 2019-35
July 2019

Seminar für Angewandte Mathematik
Eidgenössische Technische Hochschule
CH-8092 Zürich
Switzerland

Funding SNF: 184530

Exponential ReLU DNN expression of holomorphic maps in high dimension

Joost A.A. Opschoor* Christoph Schwab* Jakob Zech†

July 9, 2019

Abstract

For a parameter dimension $d \in \mathbb{N}$, we consider the approximation of many-parametric maps $u : [-1,1]^d \to \mathbb{R}$ by deep ReLU neural networks. The input dimension d may possibly be large, and we assume quantitative control of the domain of holomorphy of u: i.e., u admits a holomorphic extension to a Bernstein polyellipse $\mathcal{E}_{\rho_1} \times ... \times \mathcal{E}_{\rho_d} \subset \mathbb{C}^d$ of semiaxis sums $\rho_i > 1$ containing $[-1,1]^d$. We establish the exponential rate $O(\exp(-bN^{1/(d+1)}))$ of expressive power in terms of the total NN size N and of the input dimension d of the ReLU NN in $W^{1,\infty}([-1,1]^d)$. The constant $b > 0$ depends on $(\rho_j)_{j=1}^d$ which characterizes the coordinate-wise sizes of the Bernstein-ellipses for u. We prove exponential convergence in stronger norms for the approximation by DNNs with more regular, so-called "rectified power unit" (RePU) activations.

Key words: Deep ReLU neural networks, approximation rates, exponential convergence
Subject Classification: 41A25, 41A10, 41A46

Contents

*Seminar for Applied Mathematics, ETH Zürich, CH 8092 Zürich, Switzerland
†Department of Aeronautics and Astronautics, MIT, 02139 Cambridge, MA, USA

1 Introduction

In recent years, so-called *deep artificial neural networks* ('DNNs' for short) have seen a dramatic development in applications from data science and machine learning.

Accordingly, after early results in the '90s on genericity and universality of DNNs (see [22] for a survey and references), in recent years the refined mathematical analysis of their approximation properties has received increasing attention. A particular class of many-parametric maps whose DNN approximation needs to be considered in many applications are real-analytic and holomorphic maps. Accordingly, the question of DNN expression rate bounds for such maps has received some attention in the approximation theory literature [16, 17, 7].

It is well-known that multi-variate, holomorphic maps admit *exponential expression rates by multivariate polynomials*. In particular, countably-parametric maps $u : [-1, 1]^\infty \to \mathbb{R}$ can be represented under certain conditions by so-called *generalized polynomial chaos* expansions which, in turn, can be N-term truncated with controlled approximation rate bounds in terms of N. The polynomials which appear in such expansions can, in turn, be represented by DNNs, either exactly for certain activation functions, or approximately for example for the so-called rectified linear unit ("ReLU") activation with exponentially small representation error [13, 26].

The purpose of the present paper is to establish corresponding DNN expression rate bounds in Lipschitz-norm for high-dimensional, analytic maps $u : [-1, 1]^d \to \mathbb{R}$. We focus on ReLU DNNs, but comment in passing also on versions of our results for other DNN activation functions. Next, we briefly discuss the relation of previous results to the present work and also outline the structure of this paper.

1.1 Recent mathematical results on expressive power of DNNs

The survey [22] presented succinct proofs of genericity of shallow NNs in various function classes, as shown originally e.g. in [11, 10, 15] and reviewed the state of mathematical theory of DNNs up to that point. Moreover, exponential expression rate bounds for analytic functions by neural networks had already been achieved in the '90s. We mention in particular [17] where smooth, nonpolynomial activation functions were considered.

More closely related to the present work are the references [7, 16]. In [16], approximation rates for deep NN approximations of multivariate functions which are analytic have been investigated. Exponential rate bounds in terms of the total size of the NN have been obtained, for sigmoidal activation functions. In [7], DNN approximation of certain functions $u : [-1, 1]^d \to \mathbb{R}$ which admit holomorphic extensions to $\mathbb{C}^d$ by deep ReLU NNs has been considered. In particular, it was assumed that u admits a Taylor expansion about the origin of $\mathbb{C}^d$ which converges absolutely and uniformly on $[-1, 1]^d$. It is well-known that not every u which is real-analytic in $[-1, 1]^d$ admits such an expansion. In the present paper, we prove sharper expression rate bounds for both, the ReLU activation σ_1 and RePU activations σ_r, for functions which merely are assumed to be real-analytic in $[-1, 1]^d$, in $L^\infty([-1, 1]^d)$ and in stronger norms thereby generalizing both [7] and [16].

1.2 Contributions of the present paper

We prove exponential expression rate bounds of DNNs for d-variate, real-valued functions which depend analytically on their d inputs. Specifically, for holomorphic mappings $u : [-1,1]^d \to \mathbb{R}$, we prove expression error bounds in $L^\infty([-1,1]^d)$ and in $W^{k,\infty}([-1,1]^d)$, for $k \in \mathbb{N}$ (the precise range of k depending on properties of the NN activation σ). We consider both, ReLU activation $\sigma_1 : \mathbb{R} \to \mathbb{R}_+ : x \mapsto x_+$ and RePU activations $\sigma_r : \mathbb{R} \to \mathbb{R}_+ : x \mapsto (x_+)^r$ for some integer $r \geq 2$. Here, $x_+ = \max\{x, 0\}$. The expression error bounds with σ_1 as activation are in $W^{1,\infty}([-1,1]^d)$ and of the general type $O(\exp(-bN^{1/(d+1)}))$ in terms of the NN size N and with a constant $b > 0$ depending on the domain of analyticity, but independent of N (however, with the constant implied in the Landau symbol $O(\cdot)$ depending exponentially on d, in general). With activation σ_r for $r \geq 2$, the bounds are in $W^{k,\infty}([-1,1]^d)$ for arbitrary fixed $k \in \mathbb{N}$ and of the type $O(\exp(-bN^{1/d}))$ in terms of the NN size N. For all $r \in \mathbb{N}$, the parameters of the σ_r-neural networks approximating u (so-called "weights" and "biases") are continuous functions of u in appropriate norms.

1.3 Outline

The structure of the paper is as follows. In Section 2, we present the definition of the DNN architectures and fix notation and terminology. We also review in Section 2.2 a "ReLU DNN calculus", from recent work [21, 8], which will facilitate the ensuing DNN expression rate analysis. A first set of key results are ReLU DNN expression rates in $W^{1,\infty}([-1,1]^d)$ for multivariate Legendre polynomials, which are proved in Section 2.3. These novel expression rate bounds are explicit in the $W^{1,\infty}$-accuracy and in the polynomial degree. They are of independent interest and remarkable in that the ReLU DNNs which emulate the polynomials at exponential rates, as we prove, realize continuous, piecewise affine functions. They are based on [13, 26]. The proofs, being constructive, shed a rather precise light on the architecture, in particular depth and width of the ReLU DNNs, that is sufficient for polynomial emulation. In Section 2.4, we briefly comment on corresponding results for RePU activations; as a rule, the same exponential rates are achieved for slightly smaller NNs and in norms which are stronger than $W^{1,\infty}$.

Section 3 then contains the main results of this note: exponential ReLU DNN expression rate bounds for d-variate, holomorphic maps. They are based on a) polynomial approximation of these maps and on b) ReLU DNN reapproximation of the approximating polynomials. These are presented in Sections 3.1 and 3.2. Again we comment in Section 3.3 on modifications in the results for RePU activations. Section 4 contains a brief indication of further directions and open problems.

Acknowledgement: SNSF Early Postdoc.Mobility Fellowship 184530 to JZ. Research performed in part during a visit to the CRM Montreal, Canada, of CS and JZ in May 2019.

1.4 Notation

We adopt standard notation consistent with our previous works [29, 30]: $\mathbb{N} = \{1, 2, \dots\}$ and $\mathbb{N}_0 := \mathbb{N} \cup \{0\}$. We write $\mathbb{R}_+ := \{x \in \mathbb{R} : x \geq 0\}$. The symbol C will stand for a generic, positive constant independent of any asymptotic quantities in an estimate, which may change its value even within the same equation.

In statements about polynomial expansions we require multiindices $\boldsymbol{\nu} = (\nu_j)_{j=1,\dots,d} \in \mathbb{N}_0^d$ for $d \in \mathbb{N}$. The *total order* of a multiindex $\boldsymbol{\nu}$ is denoted by $|\boldsymbol{\nu}|_1 := \sum_{j=1}^d \nu_j$. The notation $\operatorname{supp} \boldsymbol{\nu}$ stands for the *support* of the multiindex, i.e. $\operatorname{supp} \boldsymbol{\nu} = \{j \in \{1, \dots, d\} : \nu_j \neq 0\}$. The size of the support of $\boldsymbol{\nu} \in \mathbb{N}_0^d$ is $|\operatorname{supp} \boldsymbol{\nu}|$; it will, subsequently, indicate the number of active coordinates in the multivariate monomial term $\boldsymbol{y}^{\boldsymbol{\nu}} := \prod_{j=1}^d y_j^{\nu_j}$.

A subset $\Lambda \subseteq \mathbb{N}_0^d$ is called *downward closed*[1], if $\boldsymbol{\nu} = (\nu_j)_{j=1}^d \in \Lambda$ implies $\boldsymbol{\mu} = (\mu)_{j=1}^d \in \Lambda$ for all $\boldsymbol{\mu} \leq \boldsymbol{\nu}$. Here, the ordering "$\leq$" on $\mathbb{N}_0^d$ is defined as $\mu_j \leq \nu_j$, for all $j = 1, \ldots, d$. We write $|\Lambda|$ to denote the finite cardinality of a set Λ.

We write $B_\varepsilon := \{z \in \mathbb{C} : |z| < \varepsilon\}$. Elements of $\mathbb{C}^d$ will be denoted by boldface characters such as $\boldsymbol{y} = (y_j)_{j=1}^d \in [-1,1]^d \subset \mathbb{C}^d$. For $\boldsymbol{\nu} \in \mathbb{N}_0^d$, standard notations $\boldsymbol{y}^{\boldsymbol{\nu}} := \prod_{j=1}^d y_j^{\nu_j}$ and $\boldsymbol{\nu}! = \prod_{j=1}^d \nu_j!$ will be employed (with the conventions $0! := 1$ and $0^0 := 1$). For finite index sets $\Lambda \subset \mathbb{N}_0^d$ we denote $\mathbb{P}_\Lambda := \mathrm{span}\{\boldsymbol{y}^{\boldsymbol{\nu}}\}_{\boldsymbol{\nu} \in \Lambda}$.

2 Deep neural network approximations

2.1 DNN architecture

We consider *deep neural networks (DNNs for short)* of feed forward type. Such a NN f can mathematically be described as a repeated composition of linear transformations with a nonlinear *activation function*.

More precisely: For an *activation function* $\sigma : \mathbb{R} \to \mathbb{R}$, a fixed *number of hidden layers* $L \in \mathbb{N}$, numbers $N_\ell \in \mathbb{N}$ of *computation nodes in layer* $\ell \in \{0, \ldots, L\}$, $f : \mathbb{R}^{N_0} \to \mathbb{R}^{N_{L+1}}$ is realized by a feedforward neural network, if for certain *weights* $w_{i,j}^\ell \in \mathbb{R}$, and *biases* $b_j^\ell \in \mathbb{R}$ it holds for all $x = (x_i)_{i=1}^{N_0}$

$$z_j^1 = \sigma\left(\sum_{i=1}^{N_0} w_{i,j}^1 x_i + b_j^1\right), \quad j \in \{1, \ldots, N_1\}, \tag{2.1}$$

and

$$z_j^{\ell+1} = \sigma\left(\sum_{i=1}^{N_\ell} w_{i,j}^{\ell+1} z_i^\ell + b_j^{\ell+1}\right), \quad \ell \in \{1, \ldots, L-1\}, \;\; j \in \{1, \ldots, N_{\ell+1}\}, \tag{2.2}$$

and finally

$$f(x) = (z_j^{L+1})_{j=1}^{N_{L+1}} = \left(\sum_{i=1}^{N_L} w_{i,j}^{L+1} z_i^L + b_j^{L+1}\right)_{j=1}^{N_{L+1}}. \tag{2.3}$$

In this case $n = N_0$ is the dimension of the input, and $m = N_{L+1}$ is the dimension of the output. Furthermore z_j^ℓ denotes the output of unit j in layer ℓ. The weight $w_{i,j}^\ell$ has the interpretation of connecting the ith unit in layer $\ell - 1$ with the jth unit in layer ℓ.

Except when explicitly stated, we will not distinguish between the network (which is defined through σ, the $w_{i,j}^\ell$ and b_j^ℓ) and the function $f : \mathbb{R}^{N_0} \to \mathbb{R}^{N_{L+1}}$ it realizes. We note in passing that this relation is typically not one-to-one, i.e. different NNs may realize the same function as their output. Let us also emphasize that we allow the weights $w_{i,j}^\ell$ and biases b_j^ℓ for $\ell \in \{1, \ldots, L+1\}$, $i \in \{1, \ldots, N_{\ell-1}\}$ and $j \in \{1, \ldots, N_\ell\}$ to take any value in $\mathbb{R}$, i.e. we do not consider quantization as e.g. in [1, 21].

As is customary in the theory of NNs, the number of hidden layers L of a NN is referred to as *depth*[2] and the total number of nonzero weights and biases as the *size* of the NN. Hence, for a DNN f as in (2.1)-(2.3), we define

$$\mathrm{size}(f) := |\{(i,j,\ell) : w_{i,j}^\ell \neq 0\}| + |\{(j,\ell) : b_j^\ell \neq 0\}| \qquad \text{and} \qquad \mathrm{depth}(f) := L.$$

[1] Index sets with the "downward closed" property are also referred to in the literature [18] as *lower sets*.

[2] In other recent references (e.g. [19]), slightly different terminology for the number L of layers in the DNN differing from the convention in the present paper by a constant factor, is used. This difference will be inconsequential for all results that follow.

In addition, $\mathrm{size}_{\mathrm{in}}(f) := |\{(i,j) : w^1_{i,j} \neq 0\}| + |\{j : b^1_j \neq 0\}|$ and $\mathrm{size}_{\mathrm{out}}(f) := |\{(i,j) : w^{L+1}_{i,j} \neq 0\}| + |\{j : b^{L+1}_j \neq 0\}|$, which are the number of nonzero weights and biases in the input layer of f and in the output layer, respectively.

The proofs of our main results Theorem 3.7 and Theorem 3.9 are constructive, in the sense that we will explicitly construct NNs with the desired properties. We construct these NNs by assembling smaller networks, using the operations of concatenation and parallelization, as well as so-called "identity-networks" which realize the identity mapping. Below, we recall the definitions. For these operations, we also provide bounds on the number of nonzero weights in the input layer and the output layer of the corresponding network, which can be derived from the definitions in [21].

2.2 DNN calculus

Throughout, as activation function σ we consider either the ReLU activation function

$$\sigma_1(x) := \max\{0, x\} \qquad x \in \mathbb{R} \tag{2.4}$$

or, as suggested in [16, 14, 12], for $r \in \mathbb{N}$, $r \geq 2$, the RePU activation function

$$\sigma_r(x) := \max\{0, x\}^r \qquad x \in \mathbb{R}. \tag{2.5}$$

If a NN uses σ_r as activation function, we refer to it as σ_r-NN. ReLU NNs are referred to as σ_1-NNs. We assume throughout that all activations in a DNN are of equal type.

Remark 2.1 (Historical note on rectified power units)**.** *"Rectified power unit" (RePU) activation functions are particular cases of so-called* sigmoidal functions of order $k \in \mathbb{N}$ for $k \geq 2$, *i.e.* $\lim_{x\to\infty} \frac{\sigma(x)}{x^k} = 1$*,* $\lim_{x\to-\infty} \frac{\sigma(x)}{x^k} = 0$ *and* $|\sigma(x)| \leq K(1+|x|)^k$ *for* $x \in \mathbb{R}$*. The use of NNs with such activation functions for function approximation dates back to the early 1990's, cf. e.g. [16, 14]. In fact, proofs in [16, Section 3] proceed in three steps. First, a given function* f *was approximated by a polynomial, then this polynomial was expressed as a linear combination of powers of a RePU, and finally it was shown that for* $r \geq 2$ *and arbitrary* $A > 0$ *the RePU* σ_r *can be approximated on* $[-A, A]$ *with arbitrary small* $L^\infty([-A, A])$*-precision* ε *by a NN with a sigmoidal activation function of order* $k \geq r$*, which has depth* 1 *and fixed network size ([16, Lemma 3.6]).*

The exact realization of polynomials by σ_r*-networks for* $r \geq 2$ *was observed in the proof of [16, Theorem 3.3], based on ideas in the proof of [3, Theorem 3.1]. The same result was recently rediscovered in [12, Theorem 6], whose authors were apparently not aware of [3, 16].*

We now indicate several fundamental operations on NNs which will be used in the following. These operations have been frequently used in recent works [21, 19, 8].

2.2.1 Parallelization

We now recall the parallelization of two networks f and g, which in parallel emulates f and g. We first describe the parallelization of networks with the same inputs as in [21], the parallelization of networks with different inputs is similar and introduced directly afterwards.

Let f and g be two NNs with the same depth $L \in \mathbb{N}_0$ and the same input dimension $n \in \mathbb{N}$. Denote by m_f the output dimension of f and by m_g the output dimension of g. Then there exists a neural network (f, g), called *parallelization* of f and g, which in parallel emulates f and g, i.e.

$$(f, g) : \mathbb{R}^n \to \mathbb{R}^{m_f} \times \mathbb{R}^{m_g} : \boldsymbol{x} \mapsto (f(\boldsymbol{x}), g(\boldsymbol{x})).$$

It holds that $\mathrm{depth}((f, g)) = L$ and that $\mathrm{size}((f, g)) = \mathrm{size}(f) + \mathrm{size}(g)$, $\mathrm{size}_{\mathrm{in}}((f, g)) = \mathrm{size}_{\mathrm{in}}(f) + \mathrm{size}_{\mathrm{in}}(g)$ and $\mathrm{size}_{\mathrm{out}}((f, g)) = \mathrm{size}_{\mathrm{out}}(f) + \mathrm{size}_{\mathrm{out}}(g)$.

We next recall the parallelization of networks with inputs of possibly different dimension as in [8]. To this end, we let f and g be two NNs with the same depth $L \in \mathbb{N}_0$ whose input dimensions n_f and n_g may be different, and whose output dimensions we will denote by m_f and m_g, respectively.

Then there exists a neural network $(f, g)_{\mathrm{d}}$, called *full parallelization of networks with distinct inputs* of f and g, which in parallel emulates f and g, i.e.

$$(f, g)_{\mathrm{d}} : \mathbb{R}^{n_f} \times \mathbb{R}^{n_g} \to \mathbb{R}^{m_f} \times \mathbb{R}^{m_g} : (\boldsymbol{x}, \tilde{\boldsymbol{x}}) \mapsto (f(\boldsymbol{x}), g(\tilde{\boldsymbol{x}})).$$

It holds that $\mathrm{depth}((f, g)_{\mathrm{d}}) = L$ and that $\mathrm{size}((f, g)_{\mathrm{d}}) = \mathrm{size}(f) + \mathrm{size}(g)$, $\mathrm{size}_{\mathrm{in}}((f, g)_{\mathrm{d}}) = \mathrm{size}_{\mathrm{in}}(f) + \mathrm{size}_{\mathrm{in}}(g)$ and $\mathrm{size}_{\mathrm{out}}((f, g)_{\mathrm{d}}) = \mathrm{size}_{\mathrm{out}}(f) + \mathrm{size}_{\mathrm{out}}(g)$.

Parallelizations of networks with possibly different inputs can be used consecutively to emulate multiple networks in parallel.

2.2.2 Identity networks

We now recall identity networks ([21, Lemma 2.3]), which emulate the identity map.

For all $n \in \mathbb{N}$ and $L \in \mathbb{N}_0$ there exists a σ_1-*identity network* $\mathrm{Id}_{\mathbb{R}^n}$ of depth L which emulates the identity map $\mathrm{Id}_{\mathbb{R}^n} : \mathbb{R}^n \to \mathbb{R}^n : \boldsymbol{x} \mapsto \boldsymbol{x}$. It holds that

$$\mathrm{size}(\mathrm{Id}_{\mathbb{R}^n}) \leq 2n(\mathrm{depth}(\mathrm{Id}_{\mathbb{R}^n}) + 1), \qquad \mathrm{size}_{\mathrm{in}}(\mathrm{Id}_{\mathbb{R}^n}) \leq 2n, \qquad \mathrm{size}_{\mathrm{out}}(\mathrm{Id}_{\mathbb{R}^n}) \leq 2n.$$

Analogously, for $r \geq 2$ there exist σ_r-*identity networks.* To construct them, we use the *concatenation* $f \bullet g$ of two NNs f and g as introduced in [21, Definition 2.2]. As we shall make use of it subsequently in Propositions 2.3 and 2.4, we recall its definition here for convenience of the reader.

Definition 2.2 ([21, Definition 2.2])**.** *Let f, g be such that the output dimension of g equals the input dimension of f, which we denote by k. Denote the weights and biases of f by $\{u^\ell_{i,j}\}_{i,j,\ell}$ and $\{a^\ell_j\}_{j,\ell}$ and those of g by $\{v^\ell_{i,j}\}_{i,j,\ell}$ and $\{c^\ell_j\}_{j,\ell}$. Then, the NN $f \bullet g$ emulates the composition $\boldsymbol{x} \mapsto f(g(\boldsymbol{x}))$ and satisfies $\mathrm{depth}(f \bullet g) = \mathrm{depth}(f) + \mathrm{depth}(g)$. Its weights and biases, for $\ell = 1, \ldots, \mathrm{depth}(f) + \mathrm{depth}(g)$, are given by*

$$w^\ell_{i,j} = \begin{cases} v^\ell_{i,j} & \ell \leq \mathrm{depth}(g), \\ \sum_{q=1}^k v^\ell_{i,q} u^1_{q,j} & \ell = \mathrm{depth}(g) + 1, \\ u^{\ell - \mathrm{depth}(g)}_{i,j} & \ell > \mathrm{depth}(g) + 1, \end{cases} \qquad b^\ell_j = \begin{cases} c^\ell_{i,j} & \ell \leq \mathrm{depth}(g), \\ \sum_{q=1}^k c^\ell_q u^1_{q,j} + a^1_j & \ell = \mathrm{depth}(g) + 1, \\ a^{\ell - \mathrm{depth}(g)}_j & \ell > \mathrm{depth}(g) + 1. \end{cases}$$

Proposition 2.3. *For all $r \geq 2$, $n \in \mathbb{N}$ and $L \in \mathbb{N}_0$ there exists a σ_r-NN $\mathrm{Id}_{\mathbb{R}^n}$ of depth L which emulates the identity function $\mathrm{Id}_{\mathbb{R}^n} : \mathbb{R}^n \to \mathbb{R}^n : \boldsymbol{x} \mapsto \boldsymbol{x}$. It holds that*

$$\mathrm{size}(\mathrm{Id}_{\mathbb{R}^n}) \leq nL(4r^2 + 2r), \qquad \mathrm{size}_{\mathrm{in}}(\mathrm{Id}_{\mathbb{R}^n}) \leq 4nr, \qquad \mathrm{size}_{\mathrm{out}}(\mathrm{Id}_{\mathbb{R}^n}) \leq n(2r + 1).$$

Proof. We proceed in two steps: first we discuss $L = 1$, then $L > 1$.

Step 1. It was shown in [12, Theorem 5] that there exist $(a_k)_{k=0}^r \in \mathbb{R}^{r+1}$ and $(b_k)_{k=1}^r \in \mathbb{R}^r$ such that for all $x \in \mathbb{R}$

$$x = a_0 + \sum_{k=1}^r a_k (x + b_k)^r = a_0 + \sum_{k=1}^r a_k \sigma_r(x + b_k) + \sum_{k=1}^r a_k (-1)^r \sigma_r(-x - b_k).$$

This shows the existence of a network $\mathrm{Id}_{\mathbb{R}^1} : \mathbb{R} \to \mathbb{R}$ of depth 1 realizing the identity on $\mathbb{R}$. The network employs $2r$ weights and $2r$ biases in the first layer, and $2r$ weights and one bias (namely a_0) in the output layer. Its size is thus $6r + 1$.

Step 2. For $L > 1$, we consider the L-fold concatenation $\mathrm{Id}_{\mathbb{R}^1} \bullet \cdots \bullet \mathrm{Id}_{\mathbb{R}^1}$ of the identity network $\mathrm{Id}_{\mathbb{R}^1}$ from Step 1. The resulting network has depth L, input dimension 1 and output dimension 1. The number of weights and the number of biases in the first layer both equal $2r$, the number of weights in the output layer equals $2r$, and the number of biases 1. In each of the $L-1$ other hidden layers, the number of weights is $4r^2$, and the number of biases $2r$. In total, the network has size at most $4r+(L-1)(4r^2+2r)+2r+1 \leq L(4r^2+2r)$, where we used that $r \geq 2$.

Identity networks with input size $n \in \mathbb{N}$ are obtained as the parallelization with distinct inputs of n identity networks with input size 1. □

2.2.3 Sparse concatenation

The *sparse concatenation* of two σ_1-NNs f and g was introduced in [21].

Let f and g be σ_1-NNs, such that the number of nodes in the output layer of g equals the number of nodes in the input layer of f. Denote by n the number of nodes in the input layer of g, and by m the number of nodes in the output layer of f. Then, with "$\bullet$" as in Definition 2.2, the *sparse concatenation of the NNs f and g* is defined as the network

$$f \circ g := f \bullet \mathrm{Id}_{\mathbb{R}^k} \bullet g, \tag{2.6}$$

where $\mathrm{Id}_{\mathbb{R}^k}$ is the σ_1-identity network of depth 1. The network $f \circ g$ realizes the function

$$f \circ g : \mathbb{R}^n \to \mathbb{R}^m : \boldsymbol{x} \mapsto (f(g(\boldsymbol{x})), \tag{2.7}$$

i.e., by abuse of notation, the symbol "$\circ$" has two meanings here, depending on whether we interpret $f \circ g$ as a function or as a network. This will not be the cause of confusion however. It holds $\mathrm{depth}(f \circ g) = \mathrm{depth}(f) + 1 + \mathrm{depth}(g)$,

$$\mathrm{size}(f \circ g) = \mathrm{size}(f) + \mathrm{size}_{\mathrm{in}}(f) + \mathrm{size}_{\mathrm{out}}(g) + \mathrm{size}(g) \leq 2\,\mathrm{size}(f) + 2\,\mathrm{size}(g) \tag{2.8}$$

and

$$\mathrm{size}_{\mathrm{in}}(f \circ g) = \begin{cases} \mathrm{size}_{\mathrm{in}}(g) & \mathrm{depth}(g) \geq 1, \\ 2\,\mathrm{size}_{\mathrm{in}}(g) & \mathrm{depth}(g) = 0, \end{cases} \qquad \mathrm{size}_{\mathrm{out}}(f \circ g) = \begin{cases} \mathrm{size}_{\mathrm{out}}(f) & \mathrm{depth}(f) \geq 1, \\ 2\,\mathrm{size}_{\mathrm{out}}(f) & \mathrm{depth}(f) = 0. \end{cases}$$

For a proof, we refer to [21].

A similar result holds for σ_r-NNs. In this case we define the sparse concatenation $f \circ g$ as in (2.6), but with $\mathrm{Id}_{\mathbb{R}^k}$ now denoting the σ_r-identity network of depth 1 from Proposition 2.3.

Proposition 2.4. *For $r \geq 2$ let f, g be two σ_r-NNs such that the output dimension of g, which we denote by $k \in \mathbb{N}$, equals the input dimension of f, and suppose that $\mathrm{size}_{\mathrm{in}}(f), \mathrm{size}_{\mathrm{out}}(g) \geq k$. Denote by $f \circ g$ the σ_r-network obtained by the σ_r-sparse concatenation.*

Then $\mathrm{depth}(f \circ g) = \mathrm{depth}(f) + 1 + \mathrm{depth}(g)$ *and*

$$\begin{aligned} \mathrm{size}(f \circ g) &\leq \mathrm{size}(f) + (2r-1)\,\mathrm{size}_{\mathrm{in}}(f) + (2r+1)k + (2r-1)\,\mathrm{size}_{\mathrm{out}}(g) + \mathrm{size}(g) \\ &\leq \mathrm{size}(f) + 2r\,\mathrm{size}_{\mathrm{in}}(f) + (4r-1)\,\mathrm{size}_{\mathrm{out}}(g) + \mathrm{size}(g) \\ &\leq (2r+1)\,\mathrm{size}(f) + 4r\,\mathrm{size}(g). \end{aligned} \tag{2.9}$$

Furthermore,

$$\mathrm{size}_{\mathrm{in}}(f \circ g) \leq \begin{cases} \mathrm{size}_{\mathrm{in}}(g) & \mathrm{depth}(g) \geq 1, \\ 2r\,\mathrm{size}_{\mathrm{in}}(g) + 2rk \leq 4r\,\mathrm{size}_{\mathrm{in}}(g) & \mathrm{depth}(g) = 0, \end{cases}$$

$$\mathrm{size}_{\mathrm{out}}(f \circ g) \leq \begin{cases} \mathrm{size}_{\mathrm{out}}(f) & \mathrm{depth}(f) \geq 1, \\ 2r\,\mathrm{size}_{\mathrm{out}}(f) + k \leq (2r+1)\,\mathrm{size}_{\mathrm{out}}(f) & \mathrm{depth}(f) = 0. \end{cases}$$

Proof. It follows directly from Definition 2.2 and Proposition 2.3 that $\mathrm{depth}(f \circ g) = \mathrm{depth}(f) + 1 + \mathrm{depth}(g)$. To bound the size of the network, note that the weights in layers $\ell = 1, \ldots, \mathrm{depth}(g)$ equal those in the first $\mathrm{depth}(g)$ layers of g. Those in layers $\ell = \mathrm{depth}(g) + 2, \ldots, \mathrm{depth}(g) + 2 + \mathrm{depth}(f)$ equal those in the last $\mathrm{depth}(f)$ layers of f. Layer $\ell = \mathrm{depth}(g) + 1$ has $2r\,\mathrm{size}_{\mathrm{out}}(g)$ weights and $2rk$ biases, whereas layer $\ell = \mathrm{depth}(g) + 2$ has $2r\,\mathrm{size}_{\mathrm{in}}(f)$ weights and k biases. This shows Equation (2.9) and the bound on $\mathrm{size}_{\mathrm{in}}(f \circ g)$ and $\mathrm{size}_{\mathrm{out}}(f \circ g)$. □

Identity networks are often used in combination with parallelizations. In order to parallelize two networks f and g with $\mathrm{depth}(f) < \mathrm{depth}(g)$, the network f can be concatenated with an identity network, resulting in a network whose depth equals $\mathrm{depth}(g)$ and which emulates the same function as f.

2.3 ReLU DNN approximation of polynomials

2.3.1 Basic results

In [13] it was shown that deep networks employing both ReL and BiS ("binary step") units are capable of approximating the product of two numbers with a network whose size and depth increase merely logarithmically in the accuracy. In other words, certain neural networks achieve uniform exponential convergence of the operation of multiplication (of two numbers in a bounded interval) w.r.t. the network size. Independently, a similar result for ReLU networks was obtained in [26]. Here, we shall use the latter result in the following slightly more general form shown in [25]. Contrary to [26], it provides a bound of the error in the $W^{1,\infty}([-1,1])$ norm (instead of the $L^\infty([-1,1])$ norm).

Proposition 2.5. *For any $\delta \in (0,1)$ and $M \geq 1$ there exists a σ_1-NN $\tilde{\times}_{\delta,M} : [-M,M]^2 \to \mathbb{R}$ such that*

$$\sup_{|a|,|b|\leq M} |ab - \tilde{\times}_{\delta,M}(a,b)| \leq \delta, \qquad \operatorname*{ess\,sup}_{|a|,|b|\leq M} \max\left\{\left|b - \frac{\partial}{\partial a}\tilde{\times}_{\delta,M}(a,b)\right|, \left|a - \frac{\partial}{\partial b}\tilde{\times}_{\delta,M}(a,b)\right|\right\} \leq \delta, \tag{2.10}$$

where $\frac{\partial}{\partial a}\tilde{\times}_{\delta,M}(a,b)$ and $\frac{\partial}{\partial b}\tilde{\times}_{\delta,M}(a,b)$ denote weak derivatives. There exists a constant $C > 0$ independent of $\delta \in (0,1)$ and $M \geq 1$ such that $\mathrm{size}_{\mathrm{in}}(\tilde{\times}_{\delta,M}) \leq C$, $\mathrm{size}_{\mathrm{out}}(\tilde{\times}_{\delta,M}) \leq C$,

$$\mathrm{depth}(\tilde{\times}_{\delta,M}) \leq C(1 + \log_2(M/\delta)), \qquad \mathrm{size}(\tilde{\times}_{\delta,M}) \leq C(1 + \log_2(M/\delta)).$$

Moreover, for every $a \in [-M,M]$, there exists a finite set $\mathcal{N}_a \subseteq [-M,M]$ such that $b \mapsto \tilde{\times}_{\delta,M}(a,b)$ is strongly differentiable at all $b \in (-M,M)\backslash\mathcal{N}_a$.

It is immediate, that Proposition 2.5 implies the existence of networks approximating the multiplication of n different numbers. We now show such a result, generalizing [25, Proposition 3.3] in that we consider the error again in the $W^{1,\infty}$ norm (instead of the L^∞ norm).

Proposition 2.6. *For any $\delta \in (0,1)$, $n \in \mathbb{N}$ and $M \geq 1$ there exists a σ_1-NN $\tilde{\prod}_{\delta,M} : [-M,M]^n \to \mathbb{R}$ such that*

$$\sup_{(x_i)_{i=1}^n \in [-M,M]^n} \left|\prod_{j=1}^n x_j - \tilde{\prod}_{\delta,M}(x_1,\ldots,x_n)\right| \leq \delta, \tag{2.11a}$$

$$\operatorname*{ess\,sup}_{(x_i)_{i=1}^n \in [-M,M]^n} \sup_{i=1,\ldots,n} \left|\frac{\partial}{\partial x_i}\prod_{j=1}^n x_j - \frac{\partial}{\partial x_i}\tilde{\prod}_{\delta,M}(x_1,\ldots,x_n)\right| \leq \delta, \tag{2.11b}$$

where $\frac{\partial}{\partial x_i}$ denotes a weak derivative.

There exists a constant C independent of $\delta \in (0,1)$, $n \in \mathbb{N}$ and $M \geq 1$ such that

$$\mathrm{size}(\tilde{\prod}_{\delta,M}) \leq C(1+n\log(nM^n/\delta)) \quad \text{and} \quad \mathrm{depth}(\tilde{\prod}_{\delta,M}) \leq C(1+\log(n)\log(nM^n/\delta)). \tag{2.12}$$

Proof. We proceed analogous to the proof of [25, Proposition 3.3], and construct $\tilde{\prod}_{\delta,1}$ as a binary tree of $\tilde{\times}_{\cdot,\cdot}$-networks from Proposition 2.5 with appropriately chosen parameters for the accuracy and the maximum input size.

We define $\tilde{n} := \min\{2^k : k \in \mathbb{N}, 2^k \geq n\}$, and consider the product of $\tilde{n}$ numbers $x_1, \ldots, x_{\tilde{n}} \in [-M, M]$. In case $n < \tilde{n}$, we define $x_{n+1}, \ldots, x_{\tilde{n}} := 1$, which can be implemented by a bias in the first layer. Because $\tilde{n} < 2n$, the bounds on network size and depth in terms of $\tilde{n}$ also hold in terms of n, possibly with a larger constant.

It suffices to show the result for $M = 1$, since for $M > 1$, the network defined through $\tilde{\prod}_{\delta,M}(x_1, \ldots, x_n) := M^n \tilde{\prod}_{\delta/M^n,1}(x_1/M, \ldots, x_n/M)$ for all $(x_i)_{i=1}^n \in [-M, M]^n$ achieves the desired bounds as is easily verified. Therefore, wlog $M = 1$ throughout the rest of this proof.

Equation (2.11a) follows by the argument given in the proof of [25, Proposition 3.3], we recall it here for completeness. By abuse of notation, for every even $k \in \mathbb{N}$ let a (k-dependent) mapping $R = R^1$ be defined via

$$R(y_1, \ldots, y_k) := \left(\tilde{\times}_{\delta/\tilde{n}^2,2}(y_1, y_2), \ldots, \tilde{\times}_{\delta/\tilde{n}^2,2}(y_{k-1}, y_k)\right) \in \mathbb{R}^{k/2} \tag{2.13}$$

and for $\ell \geq 2$ set $R^\ell := R \circ R^{\ell-1}$. That is, for each product network $\tilde{\times}_{\delta/\tilde{n}^2;2}$ as in Proposition 2.5 we choose maximum input size "$M = 2$" and accuracy "$\delta/\tilde{n}^2$". Hence R^ℓ can be interpreted as a mapping from $\mathbb{R}^{2^\ell} \to \mathbb{R}$. We now define $\tilde{\prod}_{\delta,1} : [-1,1]^n \to \mathbb{R}$ via

$$\tilde{\prod}_{\delta,1}(x_1, \ldots, x_n) := R^{\log_2(\tilde{n})}(x_1, \ldots, x_{\tilde{n}})$$

and next show the error bounds in (2.11) (recall that by definition $x_{n+1} = \cdots = x_{\tilde{n}} = 1$ in case $\tilde{n} > n$).

First, by induction we show that for $\ell \in \{1, \ldots, \log_2(\tilde{n})\}$ and for all $x_1, \ldots, x_{2^\ell} \in [-1,1]$

$$\left| \prod_{j=1}^{2^\ell} x_j - R^\ell(x_1, \ldots, x_{2^\ell}) \right| \leq \delta \frac{2^{2\ell}}{\tilde{n}^2}. \tag{2.14}$$

For $\ell = 1$ it holds that $R(x_1, x_2) = \tilde{\times}_{\delta/\tilde{n}^2,2}(x_1, x_2)$, hence (2.14) follows directly from the choice for the accuracy of $\tilde{\times}_{\delta/\tilde{n}^2,2}$, which is $\delta/\tilde{n}^2$. For $\ell \in \{2, \ldots, \log_2(\tilde{n})\}$, we assume that Equation (2.14) holds for $\ell - 1$. With $|\prod_{j=1}^{2^{(\ell-1)}} x_j| \leq 1$ and $\frac{2^{2(\ell-1)}}{\tilde{n}^2}\delta < 1$, it follows that $\left|R^{\ell-1}(x_1, \ldots, x_{2^{(\ell-1)}})\right| < 2$, hence $R^{\ell-1}(x_1, \ldots, x_{2^{(\ell-1)}})$ may be used as input of $\tilde{\times}_{\delta/\tilde{n}^2,2}$. We find

$$
\begin{aligned}
\left|\prod_{j=1}^{2^\ell} x_j - R^\ell(x_1,\dots,x_{2^\ell})\right| &\le \left|\prod_{j=1}^{2^{\ell-1}} x_j - R^{\ell-1}(x_1,\dots,x_{2^{\ell-1}})\right| \cdot \left|\prod_{j=2^{\ell-1}+1}^{2^\ell} x_j\right| \\
&\quad + \left|R^{\ell-1}(x_1,\dots,x_{2^{\ell-1}})\right| \cdot \left|\prod_{j=2^{\ell-1}+1}^{2^\ell} x_j - R^{\ell-1}(x_{2^{\ell-1}+1},\dots,x_{2^\ell})\right| \\
&\quad + \Big|R^{\ell-1}(x_1,\dots,x_{2^{\ell-1}})R^{\ell-1}(x_{2^{\ell-1}+1},\dots,x_{2^\ell}) \\
&\quad - \tilde{\times}_{\delta/\tilde{n}^2,2}\left(R^{\ell-1}(x_1,\dots,x_{2^{\ell-1}}), R^{\ell-1}(x_{2^{\ell-1}+1},\dots,x_{2^\ell})\right)\Big| \\
&\le \frac{2^{2(\ell-1)}}{\tilde{n}^2}\delta + \frac{2^{2(\ell-1)}}{\tilde{n}^2}\delta\left(1+\frac{2^{2(\ell-1)}}{\tilde{n}^2}\delta\right) + \frac{1}{\tilde{n}^2}\delta \\
&\le \frac{2^{2(\ell-1)} + 2\cdot 2^{2(\ell-1)} + 1}{\tilde{n}^2}\delta \le \frac{2^{2\ell}}{\tilde{n}^2}\delta,
\end{aligned}
$$

where we used $(1+\delta 2^{2(\ell-1)}/\tilde{n}^2) \le 2$. This shows (2.14) for ℓ. Inserting $\ell = \log_2(\tilde{n})$ into (2.14) gives (2.11a).

We next show (2.11b). Without loss of generality, we only consider the derivative with respect to x_1, because each $\tilde{\times}_{\delta/\tilde{n}^2,2}$-network is symmetric under permutations of its arguments. For $\ell \in \{1,\dots,\log_2(\tilde{n})\}$ we show by induction that for almost every $(x_i)_{i=1}^{2^\ell} \in [-1,1]^{2^\ell}$

$$
\left|\frac{\partial}{\partial x_i}\prod_{j=1}^{2^\ell} x_j - \frac{\partial}{\partial x_i} R^\ell(x_1,\dots,x_{2^\ell})\right| \le \delta\frac{2^{2\ell}}{\tilde{n}^2}. \tag{2.15}
$$

Again, $R(x_1,x_2) = \tilde{\times}_{\delta/\tilde{n}^2,2}(x_1,x_2)$ and for $\ell = 1$ Equation (2.15) follows from Proposition 2.5 and the choice for the accuracy of $\tilde{\times}_{\delta/\tilde{n}^2,2}$, which is $\delta/\tilde{n}^2$.

For $\ell > 1$, under the assumption that (2.15) holds for $\ell - 1$, we find

$$
\begin{aligned}
&\left|\frac{\partial}{\partial x_1}\prod_{j=1}^{2^\ell} x_j - \frac{\partial}{\partial x_1} R^\ell(x_1,\dots,x_{2^\ell})\right| \\
&\le \left|\prod_{j=2^{\ell-1}+1}^{2^\ell} x_j\right| \cdot \left|\frac{\partial}{\partial x_1}\prod_{j=1}^{2^{\ell-1}} x_j - \frac{\partial}{\partial x_1} R^{\ell-1}(x_1,\dots,x_{2^{\ell-1}})\right| \\
&\quad + \left|\prod_{j=2^{\ell-1}+1}^{2^\ell} x_j - R^{\ell-1}(x_{2^{\ell-1}+1},\dots,x_{2^\ell})\right| \cdot \left|\frac{\partial}{\partial x_1} R^{\ell-1}(x_1,\dots,x_{2^{\ell-1}})\right| \\
&\quad + \left|R^{l-1}(x_{2^{\ell-1}+1},\dots,x_{2^\ell}) - \left(\frac{\partial}{\partial a}\tilde{\times}_{\delta/\tilde{n}^2,2}\right)\left(R^{l-1}(x_1,\dots,x_{2^{\ell-1}}), R^{l-1}(x_{2^{\ell-1}+1},\dots,x_{2^\ell})\right)\right| \\
&\quad \cdot \left|\frac{\partial}{\partial x_1} R^{\ell-1}(x_1,\dots,x_{2^{\ell-1}})\right| \\
&\le \frac{2^{2(\ell-1)}}{\tilde{n}^2}\delta + \frac{2^{2(\ell-1)}}{\tilde{n}^2}\delta\left(1+\frac{2^{2(\ell-1)}}{\tilde{n}^2}\delta\right) + \frac{1}{\tilde{n}^2}\delta\left(1+\frac{2^{2(\ell-1)}}{\tilde{n}^2}\delta\right) \\
&\le \frac{2^{2(\ell-1)} + 2\cdot 2^{2(\ell-1)} + 2}{\tilde{n}^2}\delta \le \frac{2^{2\ell}}{\tilde{n}^2}\delta,
\end{aligned}
$$

where $\frac{\partial}{\partial a}\tilde{\times}_{\delta/\tilde{n}^2,2}$ denotes the (weak) derivative of $\tilde{\times}_{\delta/\tilde{n}^2,2} : [-2,2]\times[-2,2] \to \mathbb{R}$ w.r.t. its first argument as in Proposition 2.5. This shows (2.15) for $\ell > 1$, as desired. Filling in $\ell = \log_2(\tilde{n})$ gives (2.11b).

The number of binary tree layers (each denoted by R) is bounded by $O(\log_2(\tilde{n}))$. With the bound on the network depth from Proposition 2.5, for $M = 1$ the second part of (2.12) follows.

To estimate the network size, we cannot use the estimate $\operatorname{size}(f \circ g) \le 2\operatorname{size}(f) + 2\operatorname{size}(g)$ from Equation (2.8), because the number of concatenations $\log_2(\tilde{n}) - 1$ depends on n, hence the factors 2 would give an extra n-dependent factor in the estimate on the network size. Instead, from Equation (2.8) we use $\operatorname{size}(f \circ g) \le \operatorname{size}(f) + \operatorname{size}_{\text{in}}(f) + \operatorname{size}_{\text{out}}(g) + \operatorname{size}(g)$ and the bounds from Proposition 2.5. We find ($2^{\log_2(\tilde{n})-\ell}$ being the number of product networks in binary tree layer ℓ)

$$\begin{aligned}
\operatorname{size}(\tilde{\prod}_{\delta,1}) &\le \sum_{\ell=1}^{\log_2(\tilde{n})} 2^{\log_2(\tilde{n})-\ell} \left(\operatorname{size}_{\text{in}}\left(\tilde{\times}_{\delta/\tilde{n}^2,2}\right) + \operatorname{size}\left(\tilde{\times}_{\delta/\tilde{n}^2,2}\right) + \operatorname{size}_{\text{out}}\left(\tilde{\times}_{\delta/\tilde{n}^2,2}\right)\right) \\
&\le \sum_{\ell=1}^{\log_2(\tilde{n})} 2^{\log_2(\tilde{n})-\ell} \left(C + C\big(1 + \log\left(2\tilde{n}^2/\delta\right)\big) + C\right) \\
&\le (\tilde{n}-1)C\big(1 + \log\left(\tilde{n}/\delta\right)\big) \le C(1 + n\log(n/\delta)),
\end{aligned}$$

which finishes the proof of (2.12) for $M = 1$. □

The previous two propositions can be used to deduce bounds on the approximation of univariate polynomials on compact intervals w.r.t. the norm $W^{1,\infty}$. One such result was already proven in [19, Proposition 4.2]:

Proposition 2.7. *There exists a constant $C > 0$ such that the following holds: For every $\delta > 0$, $n \in \mathbb{N}_0$ and every polynomial $p = \sum_{j=0}^{n} c_j y^j \in \mathbb{P}_n$ there exists a σ_1-NN $\tilde{p}_\delta : [-1,1] \to \mathbb{R}$ such that*

$$\|p - \tilde{p}_\delta\|_{W^{1,\infty}([-1,1])} \le \delta$$

and, with $C_0 := \max\{\sum_{j=2}^{n} |c_j|, \delta\}$,

$$\operatorname{size}(\tilde{p}_\delta) \le C(1 + n\log(C_0/\delta) + n\log(n)), \qquad \operatorname{depth}(\tilde{p}_\delta) \le C((1 + \log(n))\log(C_0/\delta) + \log(n)^3).$$

Remark 2.8. *If $y_0 \in \mathbb{R}$ and $p(y) = \sum_{j=0}^{n} c_j (y-y_0)^j$, then Proposition 2.7 can still be applied for the approximation of $p(y)$ for $y \in [y_0-1, y_0+1]$, since the substitution $z = y - y_0$ corresponds to a shift, which can be realized exactly in the first layer of a NN, cp. (2.1). Thus, if $q(z) := \sum_{j=0}^{n} c_j z^j$ and if $\|q - \tilde{q}_\delta\|_{W^{1,\infty}([-1,1])} \le \delta$ as in Proposition 2.7, then $y \mapsto \tilde{p}_\delta(y) := \tilde{q}_\delta(y - y_0)$ is a NN satisfying the accuracy and size bounds of Proposition 2.7 w.r.t. the $W^{1,\infty}([y_0-1, y_0+1])$ norm.*

2.3.2 ReLU DNN approximation of univariate Legendre polynomials

For $j \in \mathbb{N}_0$ we denote by L_j the jth Legendre polynomial, normalized in $L^2([-1,1], \lambda/2)$, where $\lambda/2$ denotes the uniform probability measure on $[-1,1]$. For $j \in \mathbb{N}_0$ it holds that $L_j(x) = \sum_{\ell=0}^{j} c_\ell^j x^\ell$, where, with $m(\ell) := (j-\ell)/2$,

$$c_\ell^j = \begin{cases} 0 & j - \ell \in \{0,\dots,j\} \cap 2\mathbb{Z} + 1, \\ (-1)^m 2^{-j} \binom{j}{m}\binom{j+\ell}{j}\sqrt{2j+1} & j - \ell \in \{0,\dots,j\} \cap 2\mathbb{Z}, \end{cases} \tag{2.16}$$

see e.g. [9, Section 10.10 Equation (16)], (the factor $\sqrt{2j+1}$ is needed to obtain the desired normalization). We define $c_\ell^j := 0$ for $\ell > j$.

Analogous to [19, Equation (4.17)] it holds that $\sum_{\ell=0}^{j} |c_\ell^j| \le 4^j$ for all $j \in \mathbb{N}$ (we use that $\sqrt{2j+1} \le \sqrt{\pi j}$). Inserting this into Proposition 2.7, we find the following result on the approximation of univariate Legendre polynomials by σ_1-NNs.

Proposition 2.9 ([19, Proposition 4.2 and Equation (4.17)]). *For every $j \in \mathbb{N}_0$ and $\delta \in (0,1)$ there exists a σ_1-NN $\tilde{L}_{j,\delta}$ with input dimension one and with output dimension one such that for C independent of j and δ*

$$\begin{aligned} \|L_j - \tilde{L}_{j,\delta}\|_{W^{1,\infty}([-1,1])} &\le \delta, \\ \mathrm{depth}(\tilde{L}_{j,\delta}) &\le C(1+\log_2 j)\big(j+\log_2(1/\delta)\big), \\ \mathrm{size}(\tilde{L}_{j,\delta}) &\le Cj\big(j+\log_2(1/\delta)\big)+1. \end{aligned} \tag{2.17}$$

For future reference we note that by (2.17) and Equation (2.19) below, for all $j \in \mathbb{N}_0$, $\delta \in (0,1)$ and $k \in \{0,1\}$

$$\|\tilde{L}_{j,\delta}\|_{W^{k,\infty}([-1,1])} \le (2j+1)^{1/2+2k} + \delta \le (2j+1)^{1/2+2k} + 1 \le (2j+2)^{2k+1}. \tag{2.18}$$

2.3.3 ReLU DNN approximation of tensor product Legendre polynomials

Let $d \in \mathbb{N}$. Denote the uniform probability measure on $[-1,1]^d$ by μ_d, i.e. $\mu_d := 2^{-d}\lambda$ where λ is the Lebesgue measure on $[-1,1]^d$. Then, for all $\boldsymbol{\nu} \in \mathbb{N}_0^d$ the tensorized Legendre polynomials $L_{\boldsymbol{\nu}}(\boldsymbol{y}) := \prod_{j=1}^d L_{\nu_j}(y_j)$ form a μ_d-orthonormal basis of $L^2([-1,1]^d, \mu_d)$. We shall require the following bound on the norm of the tensorized Legendre polynomials which itself is a consequence of the Markoff inequality, and our normalization of the Legendre polynomials: for any $k \in \mathbb{N}_0$

$$\forall \boldsymbol{\nu} \in \mathbb{N}_0^d: \quad \|L_{\boldsymbol{\nu}}\|_{W^{k,\infty}([-1,1]^d)} \le \prod_{j=1}^d (1+2\nu_j)^{1/2+2k}. \tag{2.19}$$

To provide bounds on the size of the networks approximating the tensor product Legendre polynomials, for finite subsets $\Lambda \subset \mathbb{N}_0^d$ we will make use of the quantity

$$m(\Lambda) := \max_{\boldsymbol{\nu}\in\Lambda} |\boldsymbol{\nu}|_1. \tag{2.20}$$

Proposition 2.10. *For every finite subset $\Lambda \subset \mathbb{N}_0^d$ and every $\delta \in (0,1)$ there exists a σ_1-NN $f_{\Lambda,\delta}$ with input dimension d and output dimension $|\Lambda|$, such that the outputs $\{\tilde{L}_{\boldsymbol{\nu},\delta}\}_{\boldsymbol{\nu}\in\Lambda}$ of $f_{\Lambda,\delta}$ satisfy*

$$\begin{aligned} \forall \boldsymbol{\nu} \in \Lambda: \quad \|L_{\boldsymbol{\nu}} - \tilde{L}_{\boldsymbol{\nu},\delta}\|_{W^{1,\infty}([-1,1]^d)} &\le \delta, \\ \sup_{\boldsymbol{y}\in[-1,1]^d} |\tilde{L}_{\boldsymbol{\nu},\delta}((y_j)_{j\in\operatorname{supp}\boldsymbol{\nu}})| &\le (2m(\Lambda)+2)^d, \end{aligned}$$

and for a constant $C>0$ that is independent of d, Λ and δ it holds

$$\begin{aligned} \mathrm{depth}(f_{\Lambda,\delta}) &\le C(1+d\log d)(1+\log_2 m(\Lambda))\big(m(\Lambda)+\log_2(1/\delta)\big), \\ \mathrm{size}(f_{\Lambda,\delta}) &\le Cd^2 m(\Lambda)^3 + Cd^2 m(\Lambda)^2\log_2(1/\delta) + Cd^2|\Lambda|\big(1+\log_2 m(\Lambda)+\log_2(1/\delta)\big). \end{aligned}$$

Proof. Let $\delta \in (0,1)$ and a finite subset $\Lambda \subset \mathbb{N}_0^d$ be given.

The proof is divided into three steps. In the first step, we define ReLU NN approximations of tensor product Legendre polynomials $\{\tilde{L}_{\boldsymbol{\nu},\delta}\}_{\boldsymbol{\nu}\in\Lambda}$ and fix the parameters used in the NN approximation. In the second step, we estimate the error of the approximation, and the $L^\infty([-1,1]^d)$-norm of the $\tilde{L}_{\boldsymbol{\nu},\delta}$, $\boldsymbol{\nu} \in \Lambda$. In the third step, we describe the network $f_{\Lambda,\delta}$ and estimate its depth and size.

Step 1. For all $\boldsymbol{\nu} \in \mathbb{N}_0^d$, we define $n_{\boldsymbol{\nu}} := |\operatorname{supp}\boldsymbol{\nu}|$ and $M_{\boldsymbol{\nu}} := 2|\boldsymbol{\nu}|_1 + 2$. We can now define

$$\tilde{L}_{\boldsymbol{\nu},\delta}((y_j)_{j\in\operatorname{supp}\boldsymbol{\nu}}) := \tilde{\prod}_{M_{\boldsymbol{\nu}}^{-3}\delta/2, M_{\boldsymbol{\nu}}}\left(\left\{\tilde{L}_{\nu_j,\delta'}(y_j)\right\}_{j\in\operatorname{supp}\boldsymbol{\nu}}\right), \tag{2.21}$$

where $\tilde{\prod}_{M_{\boldsymbol{\nu}}^{-3}\delta/2, M_{\boldsymbol{\nu}}} : [-M_{\boldsymbol{\nu}}, M_{\boldsymbol{\nu}}]^{|\operatorname{supp}\boldsymbol{\nu}|} \to \mathbb{R}$ is as in Proposition 2.6. For the approximate univariate Legendre polynomials $\{\tilde{L}_{\nu_j,\delta'}\}_{j\in\operatorname{supp}\boldsymbol{\nu},\boldsymbol{\nu}\in\Lambda}$ as in Proposition 2.9, we set the accuracy parameter as $\delta' := \frac{1}{2}d^{-1}(2m(\Lambda)+2)^{-d-1}\delta < 1$. Let us point out that by (2.18) for all $\boldsymbol{\nu} \in \mathbb{N}_0^d$ and all $j \in \operatorname{supp}\boldsymbol{\nu}$

$$\|\tilde{L}_{\nu_j,\delta'}\|_{L^\infty([-1,1])} \le 2\nu_j + 2 \le 2|\boldsymbol{\nu}|_1 + 2 = M_{\boldsymbol{\nu}} \le 2m(\Lambda) + 2,$$

so that, as required by Proposition 2.6, the absolute values of the arguments of $\tilde{\prod}_{M_{\boldsymbol{\nu}}^{-3}\delta/2, M_{\boldsymbol{\nu}}}$ in (2.21) are all bounded by $M_{\boldsymbol{\nu}}$.

Step 2. For the $L^\infty([-1,1])$-error of $\tilde{L}_{\boldsymbol{\nu},\delta}$ we find

$$\begin{aligned}
&\sup_{\boldsymbol{y}\in[-1,1]^d} \left| L_{\boldsymbol{\nu}}(\boldsymbol{y}) - \tilde{L}_{\boldsymbol{\nu},\delta}((y_j)_{j\in\operatorname{supp}\boldsymbol{\nu}}) \right| \\
\le &\sup_{\boldsymbol{y}\in[-1,1]^d} \left| L_{\boldsymbol{\nu}}(\boldsymbol{y}) - \prod_{j\in\operatorname{supp}\boldsymbol{\nu}} \tilde{L}_{\nu_j,\delta'}(y_j) \right| \\
&+ \sup_{\boldsymbol{y}\in[-1,1]^d} \left| \prod_{j\in\operatorname{supp}\boldsymbol{\nu}} \tilde{L}_{\nu_j,\delta'}(y_j) - \tilde{\prod}_{M_{\boldsymbol{\nu}}^{-3}\delta/2, M_{\boldsymbol{\nu}}} \left(\left\{ \tilde{L}_{\nu_j,\delta'}(y_j) \right\}_{j\in\operatorname{supp}\boldsymbol{\nu}} \right) \right| \\
\le &\sup_{\boldsymbol{y}\in[-1,1]^d} \sum_{k\in\operatorname{supp}\boldsymbol{\nu}} \left| \prod_{\substack{j\in\operatorname{supp}\boldsymbol{\nu}:\\ j<k}} \tilde{L}_{\nu_j,\delta'}(y_j) \right| \cdot \left| L_{\nu_k}(y_k) - \tilde{L}_{\nu_k,\delta'}(y_k) \right| \cdot \left| \prod_{\substack{j\in\operatorname{supp}\boldsymbol{\nu}:\\ j>k}} L_{\nu_j}(y_j) \right| + \frac{\delta}{2M_{\boldsymbol{\nu}}^3}.
\end{aligned}$$

Using Proposition 2.10, (2.18), (2.19) and $M_{\boldsymbol{\nu}} = 2|\boldsymbol{\nu}|_1 + 2 \le 2m(\Lambda) + 2$, the last term can be bounded by

$$|\operatorname{supp}\boldsymbol{\nu}| M_{\boldsymbol{\nu}}^{n_{\boldsymbol{\nu}}-1}\delta' + \frac{\delta}{2} \le \frac{|\operatorname{supp}\boldsymbol{\nu}|}{d} \frac{M_{\boldsymbol{\nu}}^{n_{\boldsymbol{\nu}}-1}}{(2m(\Lambda)+2)^{d+1}} \frac{\delta}{2} + \frac{\delta}{2} \le \delta.$$

It follows that for all $\boldsymbol{\nu} \in \Lambda$

$$\begin{aligned}
\sup_{\boldsymbol{y}\in[-1,1]^d} \left| \tilde{L}_{\boldsymbol{\nu},\delta}((y_j)_{j\in\operatorname{supp}\boldsymbol{\nu}}) \right| &\le \sup_{\boldsymbol{y}\in[-1,1]^d} |L_{\boldsymbol{\nu}}(\boldsymbol{y})| + \sup_{\boldsymbol{y}\in[-1,1]^d} \left| L_{\boldsymbol{\nu}}(\boldsymbol{y}) - \tilde{L}_{\boldsymbol{\nu},\delta}((y_j)_{j\in\operatorname{supp}\boldsymbol{\nu}}) \right| \\
&\le \prod_{j=1}^d (1+2\nu_j)^{1/2} + \delta \\
&\le \prod_{j=1}^d (1+2\nu_j)^{1/2} + 1 \le M_{\boldsymbol{\nu}}^d.
\end{aligned}$$

To determine the error of the gradient, without loss of generality we only consider the derivative with respect to y_1. In the case $1 \notin \operatorname{supp}\boldsymbol{\nu}$, we trivially have $\frac{\partial}{\partial y_1}(L_{\boldsymbol{\nu}}(\boldsymbol{y}) - \tilde{L}_{\boldsymbol{\nu},\delta}(\boldsymbol{y})) = 0$ for all

$\boldsymbol{y} \in [-1,1]^d$. Thus let $\nu_1 \neq 0$ in the following. Then, with $\delta' = \frac{1}{2}d^{-1}(2m(\Lambda)+2)^{-d-1}\delta$

$$
\begin{aligned}
&\sup_{\boldsymbol{y}\in[-1,1]^d}\left|\frac{\partial}{\partial y_1}L_{\boldsymbol{\nu}}(\boldsymbol{y}) - \frac{\partial}{\partial y_1}\tilde{L}_{\boldsymbol{\nu},\delta}((y_j)_{j\in\operatorname{supp}\boldsymbol{\nu}})\right| \\
&\le \sup_{\boldsymbol{y}\in[-1,1]^d}\left|\frac{\partial}{\partial y_1}L_{\boldsymbol{\nu}}(\boldsymbol{y}) - \frac{\partial}{\partial y_1}\prod_{j\in\operatorname{supp}\boldsymbol{\nu}}\tilde{L}_{\nu_j,\delta'}(y_j)\right| \\
&\quad + \sup_{\boldsymbol{y}\in[-1,1]^d}\left|\frac{\partial}{\partial y_1}\prod_{j\in\operatorname{supp}\boldsymbol{\nu}}\tilde{L}_{\nu_j,\delta'}(y_j) - \frac{\partial}{\partial y_1}\tilde{\prod}_{M_{\boldsymbol{\nu}}^{-3}\delta/2,M_{\boldsymbol{\nu}}}\left(\left\{\tilde{L}_{\nu_j,\delta'}(y_j)\right\}_{j\in\operatorname{supp}\boldsymbol{\nu}}\right)\right| \\
&\le \sup_{\boldsymbol{y}\in[-1,1]^d}\left|\frac{\partial}{\partial y_1}L_{\nu_1}(y_1) - \frac{\partial}{\partial y_1}\tilde{L}_{\nu_1,\delta'}(y_1)\right|\cdot\left|\prod_{\substack{j\in\operatorname{supp}\boldsymbol{\nu}:\\ j>1}}L_{\nu_j}(y_j)\right| \\
&\quad + \sup_{\boldsymbol{y}\in[-1,1]^d}\sum_{1\neq k\in\operatorname{supp}\boldsymbol{\nu}}\left|\frac{\partial}{\partial y_1}\tilde{L}_{\nu_1,\delta'}(y_1)\right|\cdot\left|\prod_{\substack{1\neq j\in\operatorname{supp}\boldsymbol{\nu}:\\ j<k}}\tilde{L}_{\nu_j,\delta'}(y_j)\right|\cdot\left|L_{\nu_k}(y_k)-\tilde{L}_{\nu_k,\delta'}(y_k)\right|\cdot\left|\prod_{\substack{j\in\operatorname{supp}\boldsymbol{\nu}:\\ j>k}}L_{\nu_j}(y_j)\right| \\
&\quad + \sup_{\boldsymbol{y}\in[-1,1]^d}\left|\prod_{1\neq j\in\operatorname{supp}\boldsymbol{\nu}}\tilde{L}_{\nu_j,\delta'}(y_j) - \left(\frac{\partial}{\partial x_1}\tilde{\prod}_{M_{\boldsymbol{\nu}}^{-3}\delta/2,M_{\boldsymbol{\nu}}}\right)\left(\left\{\tilde{L}_{\nu_j,\delta'}(y_j)\right\}_{j\in\operatorname{supp}\boldsymbol{\nu}}\right)\right|\cdot\left|\frac{\partial}{\partial y_1}\tilde{L}_{\nu_1,\delta'}(y_1)\right|,
\end{aligned}
$$

where $\frac{\partial}{\partial x_1}\tilde{\prod}_{M_{\boldsymbol{\nu}}^{-3}\delta/2,M_{\boldsymbol{\nu}}}$ denotes the (weak) derivative of $\tilde{\prod}_{M_{\boldsymbol{\nu}}^{-3}\delta/2,M_{\boldsymbol{\nu}}} : [-M_{\boldsymbol{\nu}}, M_{\boldsymbol{\nu}}]^{|\operatorname{supp}\boldsymbol{\nu}|} \to \mathbb{R}$ with respect to its first argument, cf. Proposition 2.6.

Using (2.19) and Proposition 2.9 for the first term, Proposition 2.9, (2.18) and (2.19) for the second term and Proposition 2.6 and (2.19) for the third term, we further bound the NN approximation error by

$$
\delta' M_{\boldsymbol{\nu}}^{n_{\boldsymbol{\nu}}-1} + (|\operatorname{supp}\boldsymbol{\nu}|-1)M_{\boldsymbol{\nu}}^3 M_{\boldsymbol{\nu}}^{n_{\boldsymbol{\nu}}-2}\delta' + \frac{\delta}{2M_{\boldsymbol{\nu}}^3}M_{\boldsymbol{\nu}}^3 \le |\operatorname{supp}\boldsymbol{\nu}| M_{\boldsymbol{\nu}}^{n_{\boldsymbol{\nu}}+1}\tfrac{1}{2}d^{-1}(2m(\Lambda)+2)^{-d-1}\delta + \frac{\delta}{2} \le \delta.
$$

Step 3. We now describe the network $f_{\Lambda,\delta}$, which in parallel emulates $\{\tilde{L}_{\boldsymbol{\nu},\delta}\}_{\boldsymbol{\nu}\in\Lambda}$. The network is constructed as the concatenation of two subnetworks, i.e.

$$
f_{\Lambda,\delta} = f^{(1)}_{\Lambda,\delta} \circ f^{(2)}_{\Lambda,\delta}.
$$

The subnetwork $f^{(2)}_{\Lambda,\delta}$ evaluates, in parallel, approximate univariate Legendre polynomials in the input variables $(y_j)_{j\in\operatorname{supp}\boldsymbol{\nu}}$. With $T := \{(j,\nu_j) \in \mathbb{N}^2 : \boldsymbol{\nu} \in \Lambda, j \in \operatorname{supp}\boldsymbol{\nu}\}$ it is defined as

$$
f^{(2)}_{\Lambda,\delta}\left((y_j)_{j\le d}\right) := \left(\left\{\operatorname{Id}_{\mathbb{R}} \circ \tilde{L}_{\nu_j,\delta'}(y_j)\right\}_{(j,\nu_j)\in T}\right),
$$

where the pair of round brackets denotes a parallelization. The depth of the identity networks is chosen such that all components of the parallelization have equal depth.

The subnetwork $f^{(1)}_{\Lambda,\delta}$ takes the output of $f^{(2)}_{\Lambda,\delta}$ as input and computes

$$
\begin{aligned}
f_{\Lambda,\delta}\left((y_j)_{j\le d}\right) &= f^{(1)}_{\Lambda,\delta}\left(f^{(2)}_{\Lambda,\delta}\left((y_j)_{j\le d}\right)\right) \\
&= \left(\left\{\tilde{L}_{\boldsymbol{\nu},\delta}\left((y_j)_{j\le d}\right)\right\}_{\boldsymbol{\nu}\in\Lambda}\right) \\
&= \left(\left\{\operatorname{Id}_{\mathbb{R}} \circ \tilde{\prod}_{M_{\boldsymbol{\nu}}^{-3}\delta/2,M_{\boldsymbol{\nu}}}\left(\left\{\tilde{L}_{\nu_j,\delta'}(y_j)\right\}_{j\in\operatorname{supp}\boldsymbol{\nu}}\right)\right\}_{\boldsymbol{\nu}\in\Lambda}\right),
\end{aligned}
$$

where in the last two lines the outer pair of round brackets denotes a parallelization. Again, the depth of the identity networks is such that all components of the parallelization have equal depth.

We have the following expression for the network depth:

$$\operatorname{depth}(f_{\Lambda,\delta}) = \operatorname{depth}\left(f^{(1)}_{\Lambda,\delta}\right) + 1 + \operatorname{depth}\left(f^{(2)}_{\Lambda,\delta}\right).$$

We can choose the depths of the identity networks in the definition of $f^{(2)}_{\Lambda,\delta}$ such that (denoting here and in the remainder of this proof by $C > 0$ constants independent of d, Λ and $\delta \in (0,1)$)

$$\begin{aligned}
\operatorname{depth}\left(f^{(2)}_{\Lambda,\delta}\right) &= 1 + \max_{\substack{\boldsymbol{\nu}\in\Lambda,\\ j\in\operatorname{supp}\boldsymbol{\nu}}} \operatorname{depth}(\tilde{L}_{\nu_j,\delta'}) \\
&\leq C(1+\log_2 m(\Lambda))\big(m(\Lambda) + \log_2(1/\delta')\big) \\
&\leq C(1+\log_2 m(\Lambda))\big(m(\Lambda) + \log_2(d) + 1 + (d+1)\log_2(4m(\Lambda)) + \log_2(1/\delta)\big) \\
&\leq Cd(1+\log_2 m(\Lambda))\big(m(\Lambda) + \log_2(1/\delta)\big),
\end{aligned}$$

where we used that $2m(\Lambda) + 2 \leq 4m(\Lambda)$ when $\Lambda \neq \{\mathbf{0}\}$.

Similarly, due to $M_{\boldsymbol{\nu}} = 2|\boldsymbol{\nu}|_1 + 2 \leq 4m(\Lambda)$ (if $\Lambda \neq \{\mathbf{0}\}$), we can choose the identity networks in the definition of $f^{(1)}_{\Lambda,\delta}$ such that

$$\begin{aligned}
\operatorname{depth}\left(f^{(1)}_{\Lambda,\delta}\right) &= 1 + \max_{\boldsymbol{\nu}\in\Lambda} \operatorname{depth}\left(\tilde{\prod}_{M_{\boldsymbol{\nu}}^{-3}\delta/2, M_{\boldsymbol{\nu}}}\right) \\
&\leq \max_{\boldsymbol{\nu}\in\Lambda} C\left(1 + \log_2(n_{\boldsymbol{\nu}})\log_2(n_{\boldsymbol{\nu}} M_{\boldsymbol{\nu}}^{n_{\boldsymbol{\nu}}+3} 2/\delta)\right) \\
&\leq C\max_{\boldsymbol{\nu}\in\Lambda}\left(1 + \log_2(n_{\boldsymbol{\nu}})\big(\log_2 n_{\boldsymbol{\nu}} + 1 + (n_{\boldsymbol{\nu}}+3)\log_2(4m(\Lambda)) + \log_2(1/\delta)\big)\right) \\
&\leq C(1 + d\log d)\big(1 + \log_2 m(\Lambda) + \log_2(1/\delta)\big),
\end{aligned}$$

where we used that $n_{\boldsymbol{\nu}} \leq d$. Finally, we find the following bound on the network depth:

$$\operatorname{depth}(f_{\Lambda,\delta}) \leq C(1 + d\log d)(1 + \log_2 m(\Lambda))\big(m(\Lambda) + \log_2(1/\delta)\big).$$

For the network size, we find that

$$\operatorname{size}(f_{\Lambda,\delta}) \leq 2\operatorname{size}\left(f^{(1)}_{\Lambda,\delta}\right) + 2\operatorname{size}\left(f^{(2)}_{\Lambda,\delta}\right).$$

To estimate the size of $f^{(2)}_{\Lambda,\delta}$, we note that the depth of each of the identity networks in the definition of $f^{(2)}_{\Lambda,\delta}$ is at most $\operatorname{depth}(f^{(2)}_{\Lambda,\delta}) \leq Cd(1+\log_2 m(\Lambda))\big(m(\Lambda) + \log_2(1/\delta)\big)$. To estimate the number of evaluations of approximate univariate Legendre polynomials, we use that $|T| =$

$|\{(j,\nu_j)\in\mathbb{N}^2 \,:\, \boldsymbol{\nu}\in\Lambda, j\in\operatorname{supp}\boldsymbol{\nu}\}|\le m(\Lambda)d$. Using this, it follows that

$$\begin{aligned}
\operatorname{size}\left(f^{(2)}_{\Lambda,\delta}\right) &= \sum_{(j,\nu_j)\in T}\operatorname{size}\left(\mathrm{Id}_{\mathbb{R}}\circ\tilde{L}_{\nu_j,\delta'}\right)\\
&\le \sum_{(j,\nu_j)\in T} 2\operatorname{size}\left(\mathrm{Id}_{\mathbb{R}}\right)+2\operatorname{size}\left(\tilde{L}_{\nu_j,\delta'}\right)\\
&\le 4m(\Lambda)d\left(\operatorname{depth}\left(f^{(2)}_{\Lambda,\delta}\right)+1\right)+2d\sum_{k=1}^{m(\Lambda)}\operatorname{size}\left(\tilde{L}_{k,\delta'}\right)\\
&\le Cdm(\Lambda)\cdot Cd(1+\log_2 m(\Lambda))\big(m(\Lambda)+\log_2(1/\delta)\big)\\
&\quad + 2d\sum_{k=1}^{m(\Lambda)} Ck\big(k+1+\log_2 d+(d+1)\log_2(4m(\Lambda))+\log_2(1/\delta)\big)+2dm(\Lambda)\\
&\le Cd^2(1+\log_2 m(\Lambda))\big(m(\Lambda)^2+m(\Lambda)\log_2(1/\delta)\big)\\
&\quad + Cdm(\Lambda)^3+Cd^2m(\Lambda)^2\big(1+\log_2 m(\Lambda)+\log_2(1/\delta)\big)\\
&\le Cd^2m(\Lambda)^3+Cd^2m(\Lambda)^2\log_2(1/\delta).
\end{aligned}$$

The depth of each of the identity networks in the definition of $f^{(1)}_{\Lambda,\delta}$ is bounded by $\operatorname{depth}(f^{(1)}_{\Lambda,\delta})\le C(1+d\log d)\big(1+\log_2 m(\Lambda)+\log_2(1/\delta)\big)$. It follows that

$$\begin{aligned}
\operatorname{size}\left(f^{(1)}_{\Lambda,\delta}\right) &= \sum_{\boldsymbol{\nu}\in\Lambda}\operatorname{size}\left(\mathrm{Id}_{\mathbb{R}}\circ\tilde{\prod}_{M_{\boldsymbol{\nu}}^{-3}\delta/2,M_{\boldsymbol{\nu}}}\right)\\
&\le \sum_{\boldsymbol{\nu}\in\Lambda} 2\operatorname{size}\left(\mathrm{Id}_{\mathbb{R}}\right)+2\operatorname{size}\left(\tilde{\prod}_{M_{\boldsymbol{\nu}}^{-3}\delta/2,M_{\boldsymbol{\nu}}}\right)\\
&\le 4|\Lambda|\left(\operatorname{depth}\left(f^{(1)}_{\Lambda,\delta}\right)+1\right)+C\sum_{\boldsymbol{\nu}\in\Lambda}\left(1+n_{\boldsymbol{\nu}}\log_2(n_{\boldsymbol{\nu}}M_{\boldsymbol{\nu}}^{n_{\boldsymbol{\nu}}+3}2/\delta)\right)\\
&\le C(1+d\log d)|\Lambda|\big(1+\log_2 m(\Lambda)+\log_2(1/\delta)\big)+C(1+d\log d)|\Lambda|\\
&\quad + Cd\sum_{\boldsymbol{\nu}\in\Lambda}\big(1+(n_{\boldsymbol{\nu}}+3)\log_2(4m(\Lambda))+\log_2(1/\delta)\big)\\
&\le Cd^2|\Lambda|\big(1+\log_2 m(\Lambda)+\log_2(1/\delta)\big).
\end{aligned}$$

Hence, we arrive at

$$\begin{aligned}
\operatorname{size}(f_{\Lambda,\delta}) &\le 2\operatorname{size}\left(f^{(1)}_{\Lambda,\delta}\right)+2\operatorname{size}\left(f^{(2)}_{\Lambda,\delta}\right)\\
&\le Cd^2m(\Lambda)^3+Cd^2m(\Lambda)^2\log_2(1/\delta)+Cd^2|\Lambda|\big(1+\log_2 m(\Lambda)+\log_2(1/\delta)\big). \qquad\square
\end{aligned}$$

2.4 RePU DNN emulation of polynomials

The approximation of polynomials by neural networks can be significantly simplified if instead of the ReLU activation σ_1 we consider as activation function the so-called *rectified power unit* ("RePU" for short): recall that for $r\in\mathbb{N}$, $r\ge 2$, the RePU activation is defined by $\sigma_r(x)=\max\{0,x\}^r$, $x\in\mathbb{R}$. In contrast to σ_1-NNs, as shown in [12], for every $r\in\mathbb{N}$, $r\ge 2$ there exist RePU networks of depth 1 realizing the multiplication of two real numbers *without error*. This yields the following result proven in [12, Theorem 9] for $r=2$. With [12, Theorem 5] this extends to all $r\ge 2$.

Proposition 2.11. *Fix $d\in\mathbb{N}$ and $r\in\mathbb{N}$, $r\ge 2$. Then there exists a constant $C>0$ (depending on d) such that for any finite downward closed $\Lambda\subseteq\mathbb{N}_0^d$ and any $p\in\mathbb{P}_\Lambda$ there is a σ_r-network $\tilde{p}:\mathbb{R}^d\to\mathbb{R}$ which realizes p exactly and such that* $\operatorname{size}(\tilde{p})\le C|\Lambda|$ *and* $\operatorname{depth}(\tilde{p})\le C\log_2(|\Lambda|)$.

Remark 2.12. *Let $\psi : \mathbb{R} \to \mathbb{R}$ be an arbitrary C^2 function that is not linear, i.e. it does not hold $\psi''(x) = 0$ for all $x \in \mathbb{R}$. In [23] it is shown that ψ-networks can approximate the multiplication of two numbers a, b in a fixed bounded interval up to* arbitrary accuracy *with a* fixed number of units*. We also refer to [30, Section 3.3] where we explain this observation from [23] in more detail. From this, analogous to [12, Theorem 9], one can obtain a version of Proposition 2.11 for arbitrary C^2 activation functions. To state it, we fix $d \in \mathbb{N}$. Then there exists $C > 0$ (depending on d) such that for every $\delta > 0$, for every downward closed $\Lambda \subseteq \mathbb{N}_0^d$ and every $p \in \mathbb{P}_\Lambda$, there exists a ψ-neural network $q : [-M, M]^d \to \mathbb{R}$ such that* $\sup_{x\in[-M,M]^d} |p(x) - q(x)| \le \delta$, $\mathrm{size}(q) \le C|\Lambda|$ *and* $\mathrm{depth}(q) \le C \log_2(|\Lambda|)$.

3 Exponential expression rate bounds

We now proceed to the statement and proof of the main result of the present note, namely the exponential rate bounds for the DNN expression of d-variate holomorphic maps. First, in Section 3.1 we recall (classical) polynomial approximation results for analytic functions. Subsequently, these are used to deduce DNN approximation results for ReLU and RePU networks.

3.1 Polynomial approximation

Fix $d \in \mathbb{N}$. For $\rho > 1$ define the open Bernstein ellipse

$$\mathcal{E}_\rho := \left\{ \frac{z + z^{-1}}{2} \,:\, z \in \mathbb{C},\ 1 \le |z| < \rho \right\} \subset \mathbb{C},$$

and for $\boldsymbol{\rho} = (\rho_j)_{j=1}^d \subseteq (1, \infty)^d$ set

$$\mathcal{E}_{\boldsymbol{\rho}} := \bigtimes_{j=1}^{d} \mathcal{E}_{\rho_j} \subseteq \mathbb{C}^d. \tag{3.1}$$

Let $u : [-1, 1]^d \to \mathbb{R}$ admit a complex holomorphic extension to the polyellipse $\mathcal{E}_{\boldsymbol{\rho}}$. Such a function can be approximated on $[-1, 1]^d$ by multivariate Legendre expansions, with the error decaying uniformly like $\exp(-\beta N^{1/d})$ for some $\beta > 0$ and in terms of the dimension N of the approximation space. This statement is made precise in the next theorem.

Remark 3.1. *Suppose that $u : [-1, 1]^d \to \mathbb{R}$ is (real) analytic. Then it allows a complex holomorphic extension to some open set $O \subseteq \mathbb{C}^d$ containing $[-1, 1]^d$. Since for $\rho > 1$ close to 1, the maximal distance of a point in $\mathcal{E}_\rho$ to the interval $[-1, 1]$ becomes arbitrarily small, there always exists $\rho > 1$ such that u allows a holomorphic extension to $\bigtimes_{j=1}^d \mathcal{E}_\rho$.*

For the proof of the theorem we shall use the following result mentioned in [27].

Lemma 3.2. *Let $(a_j)_{j=1}^d \in (0, \infty)^d$. Then, with $a := \sum_{j=1}^d 1/a_j$*

$$\left| \left\{ \boldsymbol{\nu} \in \mathbb{N}_0^d \,:\, \sum_{j=1}^{d} \frac{\nu_j}{a_j} \le 1 \right\} \right| \le \frac{1}{d!} (1 + a)^d \prod_{j=1}^{d} a_j. \tag{3.2}$$

The lemma is proved by computing (as an upper bound of the left-hand side in (3.2)) the volume of the set $\{(x_j)_{j=1}^d \in \mathbb{R}_+^d \,:\, \sum_{j=1}^d \frac{(x_j-1)}{a_j} \le 1\}$, which equals the right-hand side in (3.2). The significance of this result is, that it provides an upper bound for multiindex sets of the type

$$\Lambda_\varepsilon := \{\boldsymbol{\nu} \in \mathbb{N}_0^d \,:\, \boldsymbol{\rho}^{-\boldsymbol{\nu}} \ge \varepsilon\}, \qquad \varepsilon \in (0, 1). \tag{3.3}$$

To see this, note that due to $\log(\boldsymbol{\rho}^{-\boldsymbol{\nu}}) = -\sum_{j=1}^d \nu_j \log(\rho_j)$, for any $\varepsilon \in (0,1)$ we have

$$\Lambda_\varepsilon = \left\{ \boldsymbol{\nu} \in \mathbb{N}_0^d \,:\, \sum_{j=1}^d \nu_j \log(\rho_j) \le \log(1/\varepsilon) \right\}.$$

Applying Lemma 3.2 with $a_j = \log(1/\varepsilon)/\log(\rho_j)$ we thus get (also see [2, Lemma 4.2]):

Lemma 3.3. *It holds*

$$|\Lambda_\varepsilon| \le \frac{1}{d!} \left(\log(1/\varepsilon) + \sum_{j=1}^d \log(\rho_j) \right)^d \prod_{j=1}^d \frac{1}{\log(\rho_j)}. \tag{3.4}$$

Remark 3.4. *Note that*

$$\left\{ \boldsymbol{\nu} \in \mathbb{N}_0^d \,:\, 0 \le \nu_j \le \frac{-\log(\varepsilon)}{d \log(\rho_j)} \ \forall j \right\} \subseteq \Lambda_\varepsilon \subseteq \left\{ \boldsymbol{\nu} \in \mathbb{N}_0^d \,:\, 0 \le \nu_j \le \frac{-\log(\varepsilon)}{\log(\rho_j)} \ \forall j \right\}. \tag{3.5}$$

This implies the existence of a constant C (depending on $\boldsymbol{\rho}$) such that for all $\varepsilon \in (0,1)$ with $\rho_{\min} := \min_{j=1,\dots,d} \rho_j$ and $\rho_{\max} := \max_{j=1,\dots,d} \rho_j$ (cp. (2.20))

$$\begin{aligned} m(\Lambda_\varepsilon) &= \max\{|\boldsymbol{\nu}|_1 \,:\, \boldsymbol{\rho}^{-\boldsymbol{\nu}} \ge \varepsilon\} = \max\{n \in \mathbb{N}_0 \,:\, \rho_{\min}^{-n} \ge \varepsilon\} \qquad (3.6)\\ &= \max\left\{ n \in \mathbb{N}_0 \,:\, n \le \frac{-\log(\varepsilon)}{\log(\rho_{\min})} \right\} \le d \frac{\log(\rho_{\max})}{\log(\rho_{\min})} \left(\prod_{j=1}^d \frac{-\log(\varepsilon)}{d \log(\rho_j)} \right)^{1/d} \\ &\le C d |\Lambda_\varepsilon|^{1/d}. \qquad (3.7) \end{aligned}$$

We are now in position to prove the following theorem, variations of which can be considered as classical.

Theorem 3.5. *Let $k \in \mathbb{N}_0$, $d \in \mathbb{N}$ and $\boldsymbol{\rho} = (\rho_j)_{j=1}^d \in (1,\infty)^d$. Let $u : \mathcal{E}_{\boldsymbol{\rho}} \to \mathbb{C}$ be holomorphic. Then, for all $k \in \mathbb{N}_0$ and for any $\beta > 0$ such that*

$$\beta < \left(d! \prod_{j=1}^d \log(\rho_j) \right)^{1/d} \tag{3.8}$$

there exists $C > 0$ (depending on d, $\boldsymbol{\rho}$, k, β and u) such that with

$$l_{\boldsymbol{\nu}} := \int_{[-1,1]^d} u(\boldsymbol{y}) L_{\boldsymbol{\nu}}(\boldsymbol{y}) \mathrm{d}\mu_d(\boldsymbol{y}), \qquad \boldsymbol{\nu} \in \mathbb{N}_0^d \tag{3.9}$$

and Λ_ε in (3.3) it holds for all $\varepsilon \in (0,1)$

$$\left\| u - \sum_{\boldsymbol{\nu} \in \Lambda_\varepsilon} l_{\boldsymbol{\nu}} L_{\boldsymbol{\nu}} \right\|_{W^{k,\infty}([-1,1]^d)} \le C \, \mathrm{e}^{-\beta |\Lambda_\varepsilon|^{1/d}}.$$

Proof. Due to the holomorphy of u on $\mathcal{E}_{\boldsymbol{\rho}}$, $l_{\boldsymbol{\nu}} \in \mathbb{R}$ satisfies the bound

$$|l_{\boldsymbol{\nu}}| \le \|u\|_{L^\infty(\mathcal{E}_{\boldsymbol{\rho}})} \boldsymbol{\rho}^{-\boldsymbol{\nu}} \prod_{j=1}^d (1+2\nu_j)^{1/2}, \qquad \boldsymbol{\nu} \in \mathbb{N}_0^d. \tag{3.10}$$

For $d = 1$ a proof can be found in [6, Chapter 12]. For general $d \in \mathbb{N}$ the bound follows by application of the one dimensional result in each variable. For more details we refer for instance to [4] or [28, Corollary B.2.7].

Since $(L_{\boldsymbol{\nu}})_{\boldsymbol{\nu}\in\mathbb{N}_0^d}$ forms an orthonormal basis of (the Hilbert space) $L^2([-1,1]^d, \mu_d)$ we have

$$u(\boldsymbol{y}) = \sum_{\boldsymbol{\nu}\in\mathbb{N}_0^d} l_{\boldsymbol{\nu}} L_{\boldsymbol{\nu}} \tag{3.11}$$

converging in $L^2([-1,1]^d, \mu_d)$. Furthermore, with (3.10) and (2.19), for $k \in \mathbb{N}_0$ and every $\boldsymbol{\nu} \in \mathbb{N}_0^d$

$$|l_{\boldsymbol{\nu}}| \|L_{\boldsymbol{\nu}}\|_{W^{k,\infty}([-1,1]^d)} \leq \|u\|_{L^\infty(\mathcal{E}_{\boldsymbol{\rho}})} \boldsymbol{\rho}^{-\boldsymbol{\nu}} \prod_{j=1}^d (1+2\nu_j)^{1+2k}. \tag{3.12}$$

Using [30, Lemma 3.13] (which is a variation of [5, Lemma 7.11]) $\sum_{\boldsymbol{\nu}\in\mathbb{N}_0^d} |l_{\boldsymbol{\nu}}| \|L_{\boldsymbol{\nu}}\|_{W^{k,\infty}([-1,1]^d)} < \infty$, and thus (3.11) also converges in $W^{k,\infty}([-1,1]^d)$.

Next, for $j \in \{1,\dots,d\}$ let $\mathbf{e}_j := (\delta_{ij})_{i=1}^d$ and introduce

$$A_\varepsilon := \{\boldsymbol{\nu} \in \mathbb{N}_0^d \,:\, \boldsymbol{\rho}^{-\boldsymbol{\nu}} < \varepsilon,\ \exists\, j \in \operatorname{supp}\boldsymbol{\nu} \text{ s.t. } \boldsymbol{\rho}^{-(\boldsymbol{\nu}-\mathbf{e}_j)} \geq \varepsilon\}.$$

Note that for $\varepsilon \in (0,1)$

$$\{\boldsymbol{\nu} \in \mathbb{N}_0^d \,:\, \boldsymbol{\rho}^{-\boldsymbol{\nu}} < \varepsilon\} = \{\boldsymbol{\mu} + \boldsymbol{\eta} \,:\, \boldsymbol{\mu} \in A_\varepsilon,\ \boldsymbol{\eta} \in \mathbb{N}_0^d\}. \tag{3.13}$$

Furthermore, since for every $\boldsymbol{\nu} \in A_\varepsilon$ there exists $j \in \operatorname{supp}\boldsymbol{\nu} \subseteq \{1,\dots,d\}$ such that $\boldsymbol{\rho}^{-(\boldsymbol{\nu}-\mathbf{e}_j)} \geq \varepsilon$ and therefore $\boldsymbol{\nu} - \mathbf{e}_j \in \Lambda_\varepsilon$, we find with (3.4) that there exists a constant C depending on d and $\boldsymbol{\rho}$ but independent of $\varepsilon \in (0,1)$ such that for all $\varepsilon \in (0,1)$

$$|A_\varepsilon| \leq d|\Lambda_\varepsilon| \leq C(1+\log(1/\varepsilon))^d. \tag{3.14}$$

Furthermore, for such $\boldsymbol{\nu} \in A_\varepsilon$ and $j \in \operatorname{supp}\boldsymbol{\nu} \subseteq \{1,\dots,d\}$ with $\rho_{\min} := \min_{i\in\{1,\dots,d\}} \rho_i$ we get

$$\rho_{\min}^{-|\boldsymbol{\nu}|_1+1} = \rho_{\min}^{-|\boldsymbol{\nu}-\mathbf{e}_j|_1} \geq \boldsymbol{\rho}^{-(\boldsymbol{\nu}-\mathbf{e}_j)} \geq \varepsilon$$

and therefore

$$|\boldsymbol{\nu}|_1 - 1 \leq \frac{\log(1/\varepsilon)}{\log(\rho_{\min})}. \tag{3.15}$$

Now

$$\begin{aligned}
\left\| u - \sum_{\boldsymbol{\nu}\in\Lambda_\varepsilon} l_{\boldsymbol{\nu}} L_{\boldsymbol{\nu}} \right\|_{W^{k,\infty}([-1,1]^d)} &\leq \sum_{\{\boldsymbol{\nu}\in\mathbb{N}_0^d \,:\, \boldsymbol{\rho}^{-\boldsymbol{\nu}}<\varepsilon\}} |l_{\boldsymbol{\nu}}| \|L_{\boldsymbol{\nu}}\|_{W^{k,\infty}([-1,1]^d)} \\
&\leq \sum_{\{\boldsymbol{\nu},\boldsymbol{\mu} \,:\, \boldsymbol{\nu}\in A_\varepsilon,\ \boldsymbol{\mu}\in\mathbb{N}_0^d\}} \|u\|_{L^\infty(\mathcal{E}_{\boldsymbol{\rho}})} \boldsymbol{\rho}^{-(\boldsymbol{\nu}+\boldsymbol{\mu})} \prod_{j=1}^d (1+2(\nu_j+\mu_j))^{1+2k} \\
&\leq \|u\|_{L^\infty(\mathcal{E}_{\boldsymbol{\rho}})} \sum_{\{\boldsymbol{\nu},\boldsymbol{\mu} \,:\, \boldsymbol{\nu}\in A_\varepsilon,\ \boldsymbol{\mu}\in\mathbb{N}_0^d\}} \boldsymbol{\rho}^{-\boldsymbol{\nu}} \boldsymbol{\rho}^{-\boldsymbol{\mu}} \prod_{j=1}^d ((1+2\nu_j)(1+2\mu_j))^{1+2k} \\
&\leq \|u\|_{L^\infty(\mathcal{E}_{\boldsymbol{\rho}})} \varepsilon \left(\sum_{\boldsymbol{\nu}\in A_\varepsilon} \prod_{j=1}^d (1+2\nu_j)^{1+2k} \right) \left(\sum_{\boldsymbol{\mu}\in\mathbb{N}_0^d} \boldsymbol{\rho}^{-\boldsymbol{\mu}} \prod_{j=1}^d (1+2\mu_j)^{1+2k} \right).
\end{aligned}$$

The sum in the second brackets is finite independent of ε by [30, Lemma 3.13]. The sum in the first brackets can be bounded using (3.14) and (3.15) to obtain a constant $C > 0$ depending on u, d, $\boldsymbol{\rho}$ and k such that for all $\varepsilon \in (0,1)$

$$\left\| u - \sum_{\boldsymbol{\nu}\in\Lambda_\varepsilon} l_{\boldsymbol{\nu}} L_{\boldsymbol{\nu}} \right\|_{W^{k,\infty}([-1,1]^d)} \le C\varepsilon |A_\varepsilon| \max_{\boldsymbol{\nu}\in A_\varepsilon} \prod_{j=1}^d (1+2\nu_j)^{1+2k} \le C\varepsilon(1+\log(1/\varepsilon))^{2d+2dk}.$$

To finish the proof, note that our above calculation shows that for any $\tau \in (0,1)$ there exists $C_\tau > 0$ depending on u, d, $\boldsymbol{\rho}$ and k such that $\left\|u - \sum_{\boldsymbol{\nu}\in\Lambda_\varepsilon} l_{\boldsymbol{\nu}} L_{\boldsymbol{\nu}}\right\|_{W^{k,\infty}([-1,1]^d)} \le C_\tau \varepsilon^\tau$ for all $\varepsilon \in (0,1)$. Moreover, (3.4) implies

$$\sum_{j=1}^d \log(\rho_j) - \left(|\Lambda_\varepsilon| d! \prod_{j=1}^d \log(\rho_j) \right)^{1/d} \ge \log(\varepsilon). \tag{3.16}$$

Hence for all $\varepsilon \in (0,1)$

$$\begin{aligned} \left\| u - \sum_{\boldsymbol{\nu}\in\Lambda_\varepsilon} l_{\boldsymbol{\nu}} L_{\boldsymbol{\nu}} \right\|_{W^{k,\infty}([-1,1]^d)} &\le C_\tau \varepsilon^\tau \le C_\tau \exp\left(\tau \left(\sum_{j=1}^d \log(\rho_j) - \left(|\Lambda_\varepsilon| d! \prod_{j=1}^d \log(\rho_j) \right)^{1/d} \right) \right) \\ &= C \exp\left(-\beta |\Lambda_\varepsilon|^{1/d} \right) \end{aligned}$$

where $C := C_\tau \exp(\tau \sum_{j=1}^d \log(\rho_j))$, $\beta := \tau(d! \prod_{j=1}^d \log(\rho_j))^{1/d}$ and where $\tau \in (0,1)$ can be arbitrarily close to 1. ☐

We note that by Stirling's inequality, with $\rho_{\min} = \min_{j=1}^d \rho_j$ and $\rho_{\max} = \max_{j=1}^d \rho_j$,

$$(d/e)\log(\rho_{\min}) \le \left(d! \prod_{j=1}^d \log(\rho_j) \right)^{1/d} \le (d/e)(e^2 d)^{1/(2d)} \log(\rho_{\max}). \tag{3.17}$$

We shall also use the following classic result for Taylor expansions of holomorphic functions.

Lemma 3.6. *Let $0 < \gamma < \kappa$, $k \in \mathbb{N}_0$, $x \in \mathbb{C}$ and assume that $u : B_\kappa^{\mathbb{C}}(x) \to \mathbb{C}$ is holomorphic. Then with $C_{\gamma,\kappa,k} := (\sum_{l=0}^k (\kappa/\gamma)^l)/(1-\gamma/\kappa)$ and $C_u := \sup_{y\in B_\kappa^{\mathbb{C}}(x)} |u(y)|$, for all $j \in \mathbb{N}_0$ it holds $|u^{(j)}(x)|/j! \le C_u \kappa^{-j}$ and*

$$\left\| u(\cdot) - \sum_{j=0}^{m-1} \frac{u^{(j)}(x)}{j!} (\cdot - x)^j \right\|_{W^{k,\infty}(B_\gamma^{\mathbb{C}}(x))} \le C_{\gamma,\kappa,k} C_u \left(\frac{\gamma}{\kappa}\right)^m \qquad \forall m \in \mathbb{N}.$$

Proof. By Cauchy's integral formula, for any $\tilde\kappa \in (0,\kappa)$ it holds

$$\frac{u^{(j)}(x)}{j!} = \frac{1}{2\pi \mathrm{i}} \int_{|\zeta - x| = \tilde\kappa} \frac{u(\zeta)}{(\zeta - x)^{j+1}} \mathrm{d}\zeta \qquad \forall j \in \mathbb{N}_0,$$

which implies $|u^{(j)}(x)/j!| \le C_u \tilde\kappa^{-j}$, and since $\tilde\kappa \in (0,\kappa)$ was arbitrary $|u^{(j)}(x)/j!| \le C_u \kappa^{-j}$ for all $j \in \mathbb{N}_0$. The bound on the truncated Taylor expansion can be deduced from the fact that for all $l \in \mathbb{N}_0$ we have $u^{(l)}(y) = \sum_{j\ge l} (y-x)^{j-l} u^{(j)}(x)/(j-l)!$ and for $u_m := \sum_{j=0}^{m-1} (y-x)^j u^{(j)}(x)/j!$, $u_m^{(l)}(y) = \sum_{j=l}^{m-1} (y-x)^{j-l} u^{(j)}(x)/(j-l)!$ so that for all $l \in \mathbb{N}_0$

$$\sup_{y\in B_\gamma^{\mathbb{C}}(x)} \left| u^{(l)}(y) - u_m^{(l)}(y) \right| \le C_u \sum_{j \ge \max\{m,l\}} \left(\frac{\gamma}{\kappa}\right)^{j-l} = C_u \frac{(\gamma/\kappa)^{\max\{m,l\}-l}}{1-\gamma/\kappa} \le C_u \frac{(\kappa/\gamma)^l}{1-\gamma/\kappa} (\gamma/\kappa)^m,$$

which implies the lemma. ☐

3.2 ReLU DNN approximation

We now come to the main result, concerning the approximation of holomorphic functions on bounded intervals by ReLU networks. The following theorem improves upon [7, Theorem 2.6] in two ways: first, we merely assume u to be analytic from $[-1,1]^d \to \mathbb{R}$ (and not analytic from $(B_1^{\mathbb{C}})^d \to \mathbb{C}$, cp. [7, Theorem 2.6] and Remark 3.1), and second, we consider the error in the $W^{1,\infty}([-1,1]^d)$ norm instead of the $L^\infty([-1,1]^d)$ norm.

Theorem 3.7. *Fix $d \in \mathbb{N}$ and let $\boldsymbol{\rho} = (\rho_j)_{j=1}^d \in (1,\infty)^d$. Assume that $u : [-1,1]^d \to \mathbb{R}$ admits a holomorphic extension to $\mathcal{E}_{\boldsymbol{\rho}}$.*

Then, there exist constants $\beta' = \beta'(\boldsymbol{\rho}, d) > 0$ and $C = C(u, \boldsymbol{\rho}, d) > 0$, and for every $\mathcal{N} \in \mathbb{N}$ there exists a σ_1-NN $\tilde{u}_{\mathcal{N}} : [-1,1]^d \to \mathbb{R}$ satisfying

$$\operatorname{size}(\tilde{u}_{\mathcal{N}}) \leq \mathcal{N}, \qquad \operatorname{depth}(\tilde{u}_{\mathcal{N}}) \leq C\mathcal{N}^{\frac{1}{d+1}} \log_2(\mathcal{N}) \tag{3.18}$$

and the error bound

$$\|u(\cdot) - \tilde{u}_{\mathcal{N}}(\cdot)\|_{W^{1,\infty}([-1,1]^d)} \leq C \exp\left(-\beta' \mathcal{N}^{\frac{1}{d+1}}\right) . \tag{3.19}$$

Proof. Throughout this proof, let $\beta > 0$ be fixed such that (3.8) holds. We proceed in five steps: In Steps 1-2 we treat the case $d = 1$. Subsequently, in Steps 3-5 we treat the case $d \geq 2$.

Step 1. We start with $d = 1$. Throughout this step fix $m \in \mathbb{N}$. We now construct a NN $\hat{u}_m$ approximating u with accuracy

$$\delta := \rho^{-m}$$

(up to some constant), where by assumption $u : \mathcal{E}_\rho \to \mathbb{C}$ is holomorphic. First, we assume $C_u := \sup_{y \in \mathcal{E}_\rho} |u(y)| \leq 1$.

Fix $\kappa \in (0,1)$ so small that $B_\kappa^{\mathbb{C}}(x) \subseteq \mathcal{E}_\rho$ for all $x \in [-1,1]$. Let $-1 = x_0 < \cdots < x_n = 1$ be a finite sequence of equidistant points with $n \in \mathbb{N}$ so large that

$$\frac{1}{n} < \frac{\kappa}{\rho}.$$

Then $(x_j)_{j=0}^n$ induces a partition of $[-1,1]$ into n intervals of length $2/n$. For every $0 \leq j \leq n$ and $y \in I_j := [-1,1] \cap [x_j - 1/n, x_j + 1/n] \subseteq B_\kappa^{\mathbb{C}}(x_j)$ we have

$$u(y) = \sum_{k \in \mathbb{N}_0} t_{j,k}(y - x_j)^k, \qquad t_{j,k} = \frac{u^{(k)}(x_j)}{k!},$$

and, because $C_u \leq 1$, by Lemma 3.6 (with $\gamma := 1/n$ and $\gamma/\kappa = 1/(\kappa n) < 1/\rho$) with $u_{m,j}(y) := \sum_{k=0}^{m-1} t_{j,k}(y - x_j)^k$

$$\|u - u_{m,j}\|_{W^{1,\infty}(I_j)} \leq C_{\gamma,\kappa,1} C_u (\kappa n)^{-m} \leq C_{\gamma,\kappa,1} \rho^{-m} = C_{\gamma,\kappa,1} \delta. \tag{3.20}$$

Moreover, Lemma 3.6 implies that it holds for all $0 \leq j \leq n$ (here we use $\kappa \in (0,1)$)

$$\sum_{k=0}^{m-1} |t_{j,k}| \leq C_u \sum_{k=0}^{m-1} \kappa^{-k} \leq \frac{\kappa^{-m} - 1}{\kappa^{-1} - 1} \leq C\kappa^{-m},$$

for C depending on κ, which depends on ρ.

Let $\tilde{u}_{m,j,\delta} : [-1,1] \to \mathbb{R}$ be an approximation to the polynomial $u_{m,j}$ as in Proposition 2.7, i.e.

$$\|u_{m,j} - \tilde{u}_{m,j,\delta}\|_{W^{1,\infty}([-1,1])} \leq \delta,$$

and thus

$$\|u - \tilde{u}_{m,j,\delta}\|_{W^{1,\infty}(I_j)} \leq C\delta \tag{3.21}$$

for C depending on ρ, but independent of u. Therefore, and due to $\delta = \rho^{-m}$, Proposition 2.7 gives

$$\begin{aligned} \mathrm{size}(\tilde{u}_{m,j,\delta}) &\leq C(1 + m\log(C\kappa^{-m}/\delta) + m\log(m)) \leq Cm^2, \\ \mathrm{depth}(\tilde{u}_{m,j,\delta}) &\leq C((1+\log(m))\log(C\kappa^{-m}/\delta) + \log(m)^3) \leq C(1 + m\log(m)), \end{aligned} \tag{3.22}$$

for a constant C depending on ρ, but independent of m and u.

Next, denote by $(\varphi_j)_{j=0}^n$ the continuous, piecewise affine functions on the partition induced by $(x_j)_{j=0}^n$ on $[-1,1]$ such that $\varphi_j(x_i) = \delta_{i,j}$ for all i, j. As is well known, each φ_j can be expressed without error with a ReLU network of depth 1 and size at most 3 (see for example [25]). For $C > 0$ as in (3.21) and $M := C + 1$ we define a network $\hat{u}_m$ approximating u by

$$\hat{u}_m := \sum_{j=0}^{n} \tilde{\times}_{\delta,M}(\varphi_j, \tilde{u}_{m,j,\delta}). \tag{3.23}$$

We observe that for all $0 \leq j \leq n$

$$\sup_{y\in[-1,1]} |\tilde{u}_{m,j,\delta}(y)| \leq \sup_{y\in[-1,1]} |\tilde{u}_{m,j,\delta}(y) - u(y)| + \sup_{y\in[-1,1]} |u(y)| \leq C\delta + C_u \leq M,$$

so that by Proposition 2.5

$$\begin{aligned} \operatorname*{ess\,sup}_{y\in[-1,1]} |\varphi_j \tilde{u}_{m,j,\delta} - \tilde{\times}_{\delta,M}(\varphi_j, \tilde{u}_{m,j,\delta})| &\leq \delta, \\ \operatorname*{ess\,sup}_{y\in[-1,1]} |\tilde{u}_{m,j,\delta} - \frac{\partial}{\partial a}\tilde{\times}_{\delta,M}(\varphi_j, \tilde{u}_{m,j,\delta})| &\leq \delta, \\ \operatorname*{ess\,sup}_{y\in[-1,1]} |\varphi_j - \frac{\partial}{\partial b}\tilde{\times}_{\delta,M}(\varphi_j, \tilde{u}_{m,j,\delta})| &\leq \delta, \end{aligned}$$

where here and in the following all derivatives are interpreted as weak derivatives. These estimates will be used repeatedly in the following.

We now provide an upper bound for $\|u - \hat{u}_m\|_{W^{1,\infty}([-1,1])}$. For every $0 \leq j \leq n$ it holds

$$\begin{aligned} &\|\varphi_j u - \tilde{\times}_{\delta,M}(\varphi_j, \tilde{u}_{m,j,\delta})\|_{W^{1,\infty}([-1,1])} \\ &\quad \leq \|\varphi_j u - \varphi_j \tilde{u}_{m,j,\delta}\|_{L^\infty([-1,1])} + \|\varphi_j \tilde{u}_{m,j,\delta} - \tilde{\times}_{\delta,M}(\varphi_j, \tilde{u}_{m,j,\delta})\|_{L^\infty([-1,1])} \\ &\qquad + \left\|\varphi_j' u - \varphi_j' \tilde{u}_{m,j,\delta}\right\|_{L^\infty([-1,1])} + \left\|\varphi_j' \tilde{u}_{m,j,\delta} - \varphi_j' \frac{\partial}{\partial a}\tilde{\times}_{\delta,M}(\varphi_j, \tilde{u}_{m,j,\delta})\right\|_{L^\infty([-1,1])} \\ &\qquad + \left\|\varphi_j u' - \varphi_j \tilde{u}_{m,j,\delta}'\right\|_{L^\infty([-1,1])} + \left\|\varphi_j \tilde{u}_{m,j,\delta}' - \frac{\partial}{\partial b}\tilde{\times}_{\delta,M}(\varphi_j, \tilde{u}_{m,j,\delta})\tilde{u}_{m,j,\delta}'\right\|_{L^\infty([-1,1])}. \end{aligned}$$

Using (3.21) and $\delta = \rho^{-m}$ as well as $\|\varphi_j'\|_{L^\infty([-1,1])} \leq n$ and $\operatorname{supp}\varphi_j = I_j = [-1,1] \cap [x_j - 1/n, x_j + 1/n]$, these norms can be bounded by $C\delta$ for a constant C depending on n, u and on ρ. Hence

$$\begin{aligned} \|u - \hat{u}_m\|_{W^{1,\infty}([-1,1])} &= \left\|\sum_{j=0}^{n} \varphi_j u - \tilde{\times}_{\delta,M}(\varphi_j, \tilde{u}_{m,j,\delta})\right\|_{W^{1,\infty}([-1,1])} \\ &\leq \sum_{j=0}^{n} \left\|\varphi_j u - \tilde{\times}_{\delta,M}(\varphi_j, \tilde{u}_{m,j,\delta})\right\|_{W^{1,\infty}([-1,1])} \\ &\leq C\delta. \end{aligned} \tag{3.24}$$

The constant C depends on the number of intervals n in the partition induced by $(x_j)_{j=0}^n$, but we emphasize that n is a fixed constant in this computation and does not increase as $\delta \to 0$.

We now describe the network $\hat{u}_m$. The following σ_1-NN realizes (3.23):

$$\hat{u}_m = \mathrm{Sum}_{n+1} \circ \left(\left\{ \tilde{\times}_{\delta,M} \right\}_{j=0}^{n} \right)_{\mathrm{d}} \circ \left(\left\{ \mathrm{Id}_{\mathbb{R}} \circ \varphi_j, \mathrm{Id}_{\mathbb{R}} \circ \tilde{u}_{m,j,\delta} \right\}_{j=0}^{n} \right).$$

Here, Sum_{n+1} is a network with input dimension $n+1$, output dimension 1, depth 0 and size $n+1$ which implements the sum of its inputs. Round brackets denote parallelizations. The depth of the identity networks is chosen such that all components of the parallelization have equal depth, which is $1 + \max_{j=0}^n \mathrm{depth}\left(\tilde{u}_{m,j,\delta}\right)$.

Next, we bound the size and depth of the network $\hat{u}_m$. By (3.22), the identity networks contribute at most $(2n+2)2(1 + \max_{j=0}^n \mathrm{depth}\left(\tilde{u}_{m,j,\delta}\right)) \le C(4n+4)(1 + m\log(m))$ to the network size, for C independent of u, but depending on ρ. Using the bound $\mathrm{size}(\tilde{\times}_{\delta,M}) \le C(1 + \log(M/\delta)) \le Cm$ (by Proposition 2.5), (3.22) and $\mathrm{size}(\varphi_j) = 3$ for all $0 \le j \le n$

$$\begin{aligned}
\mathrm{size}(\hat{u}_m) &\overset{(2.8)}{\le} \sum_{j=0}^{n} 4\,\mathrm{size}(\mathrm{Sum}_{n+1}) + 4\,\mathrm{size}(\varphi_j) + 4\,\mathrm{size}(\tilde{\times}_{\delta,M}) + 4\,\mathrm{size}(\tilde{u}_{m,j,\delta}) \\
&\quad + 4C(4n+4)(1 + m\log(m)) \\
&\le C_1 m^2,
\end{aligned} \tag{3.25a}$$

for a constant C_1 depending on n and on ρ but independent of $m \in \mathbb{N}$ and u. Similarly one obtains

$$\mathrm{depth}(\hat{u}_m) \le C(1 + m\log(m)), \tag{3.25b}$$

for a constant C that does not depend on $m \in \mathbb{N}$ and u.

Now, for u with $C_u = \sup_{y \in \mathcal{E}_\rho} |u(y)| > 1$ we approximate u/C_u as above, and multiply all weights and biases in the output layer of the resulting network by C_u. This does not affect the network's depth and size, and it follows that (3.24) holds for C replaced by $C_u C$.

Step 2. Fix $\mathcal{N}$ in $\mathbb{N}$. Define $m(\mathcal{N}) := \lfloor (\mathcal{N}/C_1)^{1/2} \rfloor$ and $\tilde{u}_{\mathcal{N}} := \hat{u}_{m(\mathcal{N})}$ whenever $m(\mathcal{N}) \ge 1$, i.e. whenever $\mathcal{N} \ge C_1$ (here C_1 is as in (3.25)). From (3.25) we deduce (3.18). A bound on the error (for $\mathcal{N} \ge C_1$) is obtained via (3.24):

$$\begin{aligned}
\|u - \tilde{u}_{\mathcal{N}}\|_{W^{1,\infty}([-1,1])} &= \|u - \hat{u}_{\lfloor (\mathcal{N}/C_1)^{1/2} \rfloor}\|_{W^{1,\infty}([-1,1])} \le C\rho^{-\lfloor (\mathcal{N}/C_1)^{1/2} \rfloor} \\
&= C\exp\left(-\log(\rho) \lfloor (\mathcal{N}/C_1)^{1/2} \rfloor \right) \le C\rho \exp\left(-\frac{\log(\rho)}{C_1^{1/2}} \mathcal{N}^{1/2} \right).
\end{aligned}$$

This implies (3.19) for $d = 1$ with

$$\beta' = \log(\rho) C_1^{-1/2} \tag{3.26}$$

and for all $\mathcal{N} \ge C_1$. With $\tilde{u}_{\mathcal{N}} := 0$ (i.e. a trivial NN giving the constant value 0) for all (finitely many) $\mathcal{N} < C_1$, we obtain (3.19).

Step 3. We now consider the case $d \ge 2$. In this step, for any $\varepsilon \in (0,1)$ we introduce a network $\hat{u}_\varepsilon$ approximating u (with increasing accuracy as $\varepsilon \to 0$).

Fix $\varepsilon \in (0,1)$ arbitrary, let $\Lambda_\varepsilon \subseteq \mathbb{N}_0^d$ be as in (3.3) and set $u_\varepsilon := \sum_{\boldsymbol{\nu} \in \Lambda_\varepsilon} l_{\boldsymbol{\nu}} L_{\boldsymbol{\nu}}$ with the Legendre coefficients $l_{\boldsymbol{\nu}}$ of u as in (3.9).

Let Affine_u be a NN of depth 0, with input dimension $|\Lambda_\varepsilon|$, output dimension 1 and size at most $|\Lambda_\varepsilon|$ which implements the affine transformation $\mathbb{R}^{|\Lambda_\varepsilon|} \to \mathbb{R} : (z_{\boldsymbol{\nu}})_{\boldsymbol{\nu} \in \Lambda} \mapsto \sum_{\boldsymbol{\nu} \in \Lambda_\varepsilon} l_{\boldsymbol{\nu}} z_{\boldsymbol{\nu}}$. Furthermore, let $f_{\Lambda_\varepsilon,\delta}$ be the network from Proposition 2.10, emulating in parallel approximations to all multivariate Legendre polynomials $(L_{\boldsymbol{\nu}})_{\boldsymbol{\nu} \in \Lambda_\varepsilon}$. We define a NN

$$\hat{u}_\varepsilon := \mathrm{Affine}_u \circ f_{\Lambda_\varepsilon,\delta}.$$

Then

$$\hat{u}_\varepsilon(\boldsymbol{y}) = \sum_{\boldsymbol{\nu}\in\Lambda_\varepsilon} l_{\boldsymbol{\nu}} \tilde{L}_{\boldsymbol{\nu},\delta}(\boldsymbol{y}), \qquad \boldsymbol{y} \in [-1,1]^d,$$

where (with $\beta > 0$ as in (3.8)) the accuracy $\delta > 0$ of the σ_1-NN approximations of the tensor product Legendre polynomials is chosen as

$$\delta := \exp\left(-\beta|\Lambda_\varepsilon|^{1/d}\right).$$

Step 4. For the NN $\hat{u}_\varepsilon$ we obtain the error estimate

$$\begin{aligned}\|u_\varepsilon - \hat{u}_\varepsilon\|_{W^{1,\infty}([-1,1]^d)} &\leq \sum_{\boldsymbol{\nu}\in\Lambda_\varepsilon} |l_{\boldsymbol{\nu}}| \, \|L_{\boldsymbol{\nu}} - \tilde{L}_{\boldsymbol{\nu},\delta}\|_{W^{1,\infty}([-1,1]^d)} \\ &\leq \sum_{\boldsymbol{\nu}\in\Lambda_\varepsilon} |l_{\boldsymbol{\nu}}| \, \delta \\ &= \sum_{\boldsymbol{\nu}\in\Lambda_\varepsilon} |l_{\boldsymbol{\nu}}| \exp\left(-\beta|\Lambda_\varepsilon|^{1/d}\right).\end{aligned}$$

With Theorem 3.5 this yields the existence of a constant $C > 0$ (depending on d, $\boldsymbol{\rho}$, β and u) such that

$$\|u - \hat{u}_\varepsilon\|_{W^{1,\infty}([-1,1]^d)} \leq C \exp\left(-\beta|\Lambda_\varepsilon|^{1/d}\right). \tag{3.27}$$

We now bound the depth and the size of $\hat{u}_\varepsilon$. Using (3.6), we obtain

$$\begin{aligned}\mathrm{depth}(\hat{u}_\varepsilon) &\leq \mathrm{depth}(\mathrm{Affine}_u) + 1 + \mathrm{depth}\left(f_{\Lambda_\varepsilon,\delta}\right) \\ &\leq C(1 + d\log d)(1 + \log_2 m(\Lambda_\varepsilon))\big(m(\Lambda_\varepsilon) + \log_2(1/\delta)\big) \\ &\leq C(1 + d\log d)(1 + \log_2(d) + \log_2|\Lambda_\varepsilon|)\big(Cd|\Lambda_\varepsilon|^{1/d} + \beta|\Lambda_\varepsilon|^{1/d}\big) \\ &\leq C(1+\beta)(1 + d^2(\log d)^2)(1 + |\Lambda_\varepsilon|^{1/d}\log_2|\Lambda_\varepsilon|)\end{aligned} \tag{3.28}$$

for $C > 0$ depending on $\boldsymbol{\rho}$. To bound the NN size, Proposition 2.10 and (3.6) give

$$\begin{aligned}\mathrm{size}(\hat{u}_\varepsilon) &\leq 2\,\mathrm{size}(\mathrm{Affine}_u) + 2\,\mathrm{size}(f_{\Lambda_\varepsilon,\delta}) \\ &\leq 2|\Lambda_\varepsilon| + 2Cd^2 m(\Lambda_\varepsilon)^3 + 2Cd^2 m(\Lambda_\varepsilon)^2\log_2(1/\delta) + 2Cd^2|\Lambda_\varepsilon|\big(1 + \log_2 m(\Lambda_\varepsilon) + \log_2(1/\delta)\big) \\ &\leq 2|\Lambda_\varepsilon| + Cd^5(|\Lambda_\varepsilon|^{1/d})^3 + Cd^4(|\Lambda_\varepsilon|^{1/d})^2\beta|\Lambda_\varepsilon|^{1/d} \\ &\quad + Cd^2|\Lambda_\varepsilon|\big(1 + \log(d) + \log_2|\Lambda_\varepsilon| + \beta|\Lambda_\varepsilon|^{1/d}\big) \\ &\leq C(1+\beta)d^5|\Lambda_\varepsilon|^{3/d} + C(1+\beta)(1 + d^2\log d)|\Lambda_\varepsilon|^{1+1/d} \leq C_2(1+\beta)d^5|\Lambda_\varepsilon|^{1+1/d}\end{aligned} \tag{3.29}$$

for a constant $C_2 > 0$ which depends on $\boldsymbol{\rho}$, but is independent of d, β, u and of $\varepsilon \in (0,1)$.

Step 5. Finally, we define $\tilde{u}_{\mathcal{N}}$. Fix $\beta > 0$ satisfying (3.8) and $\mathcal{N} \in \mathbb{N}$ such that $\mathcal{N} > \mathcal{N}_0 := C_2(1+\beta)d^5$, with the constant C_2 as in (3.29). Set

$$\widehat{\mathcal{N}} := \left(\frac{\mathcal{N}}{\mathcal{N}_0}\right)^{d/(d+1)} \in \mathbb{R}. \tag{3.30}$$

Next, let $\varepsilon \in (0,1)$ be such that

$$\widehat{\mathcal{N}} = \prod_{j=1}^{d}\left(\frac{\log(1/\varepsilon)}{\log(\rho_j)} + 1\right), \tag{3.31}$$

which is possible since $\widehat{\mathcal{N}} > 1$ due to the assumption $\mathcal{N} > \mathcal{N}_0 = C_2(1+\beta)d^5$. Define $\tilde{u}_{\mathcal{N}} := \hat{u}_\varepsilon$.

First let us estimate the size of $\tilde{u}_{\mathcal{N}}$. By (3.5)

$$\widehat{\mathcal{N}} \geq \prod_{j=1}^{d} \left(\left\lfloor \frac{\log(1/\varepsilon)}{\log(\rho_j)} \right\rfloor + 1 \right) = \left| \left\{ \boldsymbol{\nu} \in \mathbb{N}_0^d \,:\, 0 \leq \nu_j \leq \frac{\log(1/\varepsilon)}{\log(\rho_j)} \ \forall j \right\} \right| \geq |\Lambda_\varepsilon|.$$

Hence (3.29) and the definition of $\widehat{\mathcal{N}}$ imply

$$\mathrm{size}(\tilde{u}_{\mathcal{N}}) = \mathrm{size}(\hat{u}_\varepsilon) \leq C_2(1+\beta)d^5|\Lambda_\varepsilon|^{1+1/d} \leq C_2(1+\beta)d^5\widehat{\mathcal{N}}^{1+1/d} \leq \mathcal{N}.$$

Similarly one obtains the bound on the depth of $\tilde{u}_{\mathcal{N}}$ by using (3.28). This shows (3.18).

Next we estimate the error $\|u - \tilde{u}_{\mathcal{N}}\|_{W^{1,\infty}([-1,1]^d)}$. By (3.5)

$$\begin{aligned}\widehat{\mathcal{N}} &\leq \prod_{j=1}^{d} \left(d \left\lfloor \frac{\log(1/\varepsilon)}{d\log(\rho_j)} \right\rfloor + d + 1 \right) = \prod_{j=1}^{d} \left(\left\lfloor \frac{\log(1/\varepsilon)}{d\log(\rho_j)} \right\rfloor + 1 \right) \prod_{j=1}^{d} \left(d + \frac{1}{\left\lfloor \frac{\log(1/\varepsilon)}{d\log(\rho_j)} \right\rfloor + 1} \right) \\ &\leq |\Lambda_\varepsilon|(d+1)^d.\end{aligned}$$

Thus (3.27) gives

$$\|u - \tilde{u}_{\mathcal{N}}\|_{W^{1,\infty}([-1,1]^d)} \leq C \exp\left(-\beta|\Lambda_\varepsilon|^{1/d} \right) \leq C \exp\left(-\beta(d+1)^{-1}\widehat{\mathcal{N}}^{1/d} \right).$$

By (3.30) this is (3.19) for any $\mathcal{N} > \mathcal{N}_0$ and with

$$\beta' = \beta(d+1)^{-1}(C_2(1+\beta)d^5)^{-1/(d+1)} \tag{3.32}$$

for C_2 as in (3.29) (independent of d, β and u). Similar as in Step 2 (by increasing $C > 0$ in (3.19) if necessary) we conclude that (3.19) holds for all $\mathcal{N} \in \mathbb{N}$. $\square$

Remark 3.8. *Note that in step 1 of the proof, the network $\hat{u}_m$ only depends on u through the NNs $\{\tilde{u}_{m,j,\delta}\}_{j=0}^n$. The weights and biases of those networks continuously depend on u with respect to the $L^\infty(\mathcal{E}_{\boldsymbol{\rho}})$-norm (because the only u-dependent weights and biases are the Taylor coefficients of u, which are bounded in terms of C_u).*

Similarly, in step 4, the network $\hat{u}_\varepsilon$ depends on u only via the Legendre coefficients $\{l_{\boldsymbol{\nu}}\}_{\boldsymbol{\nu}\in\Lambda_\varepsilon}$, appearing only as weights in the output layer. In particular, the weights and biases of $\hat{u}_\varepsilon$ continuously depend on u with respect to the $L^2([-1,1]^d, \mu_d)$-norm, because the Legendre coefficients do so. Finally, the $L^2([-1,1]^d, \mu_d)$-norm is bounded by the $L^\infty([-1,1]^d)$-norm.

3.3 RePU DNN approximation

For RePU approximations, with activation $\sigma_r(x)$ for integer $r \geq 2$, we may combine Proposition 2.11 (which is [12, Theorem 9]) and Theorem 3.5 to infer the following result. Note that the decay of the provided upper bound of the error in (3.33) in terms of the network size $\mathcal{N}$ is slightly faster than the one we obtained for ReLU approximations in (3.19).

Theorem 3.9. *Fix $d \in \mathbb{N}$, $k \in \mathbb{N}_0$ and $r \in \mathbb{N}$, $r \geq 2$. Let $\boldsymbol{\rho} = (\rho_j)_{j=1}^d \in (1,\infty)^d$. Assume that $u : [-1,1]^d \to \mathbb{R}$ admits a holomorphic extension to $\mathcal{E}_{\boldsymbol{\rho}}$.*

Then, there exists $C > 0$ such that with β as in (3.8)*, for every $\mathcal{N} \in \mathbb{N}$, there exists a σ_r-NN $\tilde{u}_{\mathcal{N}} : [-1,1]^d \to \mathbb{R}$ satisfying*

$$\mathrm{size}(\tilde{u}_{\mathcal{N}}) \leq C\mathcal{N}, \qquad \mathrm{depth}(\tilde{u}_{\mathcal{N}}) \leq C\log_2(\mathcal{N}) \tag{3.33}$$

and

$$\|u(\boldsymbol{y}) - \tilde{u}_{\mathcal{N}}(\boldsymbol{y})\|_{W^{k,\infty}([-1,1]^d)} \leq C \exp\left(-\beta\mathcal{N}^{\frac{1}{d}} \right). \tag{3.34}$$

Proof. For $\varepsilon \in (0,1)$ let Λ_ε be as in (3.3). This set is finite and downward closed. Hence, by Proposition 2.11 there exists a σ_r-NN $\hat{u}_\varepsilon$ such that $\hat{u}_\varepsilon(\boldsymbol{y}) = \sum_{\boldsymbol{\nu}\in\Lambda_\varepsilon} l_{\boldsymbol{\nu}} L_{\boldsymbol{\nu}}(\boldsymbol{y})$ for all $\boldsymbol{y} \in [-1,1]^d$. According to this proposition, the NN $\hat{u}_\varepsilon$ satisfies $\mathrm{size}(\hat{u}_\varepsilon) \le C|\Lambda_\varepsilon|$ and $\mathrm{depth}(\hat{u}_\varepsilon) \le C \log|\Lambda_\varepsilon|$. This is (3.33) for $\mathcal{N} := |\Lambda_\varepsilon|$. By Theorem 3.5, it holds (3.34).

For general $\mathcal{N} > 1$, it follows as in Step 5 of the proof of Theorem 3.7 (with $\mathcal{N}$ taking the role of $\widehat{\mathcal{N}}$) that there exists $\varepsilon \in (0,1)$ such that $(d+1)^{-d}\mathcal{N} \le |\Lambda_\varepsilon| \le \mathcal{N}$. This implies that (3.34) holds for any $\mathcal{N} \in \mathbb{N}$ with a constant C depending on d. □

Remark 3.10. *It follows from the proof of [12, Theorem 2], which is the basis for the proof of Proposition 2.11, that the weights of $\tilde{u}_{\mathcal{N}}$ depend continuously on the Legendre coefficients of u, which themselves depend continuously on u w.r.t. the $L^2([-1,1]^d, \mu_d)$-norm, which is bounded by the $L^\infty([-1,1]^d)$-norm.*

Remark 3.11. *A similar result as in Theorem 3.9 was obtained in [16, Theorem 3.3]. It assumed a different class of activation functions, termed "sigmoidal functions of order $k \ge 2$" (see Remark 2.1). The $L^\infty([-1,1]^d)$ error bound provided in [16, Theorem 3.3] is, in our notation, of the type $\exp(-b\mathcal{N}^{1/d})$ for a suitable constant $b > 0$ and a DNN of size $\mathcal{N}\log(\mathcal{N})$. This is slightly worse than Theorem 3.9.*

4 Conclusion

We review in Section 4.1 the main results obtained in the previous sections and indicate in Section 4.2 some implications of these.

4.1 Main Results

We have established for analytic maps $u : [-1,1]^d \to \mathbb{R}$ exponential expression rate bounds in $W^{k,\infty}([-1,1]^d)$ in terms of the DNN size for the ReLU activation (for $k = 0,1$) and for the RePU activations σ_r, $r \ge 2$ (for $k = 0,...,r$). The present analysis improves earlier results in that the NN sizes are slightly reduced and we obtain exponential convergence of ReLU and RePU DNNs for general d-variate analytic functions, without assuming the Taylor expansion of u around $0 \in \mathbb{R}^d$ to converge on $[-1,1]^d$. We also point out that by a simple scaling argument our main results in Theorem 3.7 and Theorem 3.9 imply corresponding expression rate results for analytic functions defined on an arbitrary cartesian product of finite intervals $\times_{j=1}^d [a_j, b_j]$, where $-\infty < a_j < b_j < \infty$ for all $j \in \{1,\dots,d\}$.

4.2 Applications and generalizations

4.2.1 Solution manifolds of PDEs

One possible application of our results concerns the approximation of (quantities of interest) of *solution manifolds of parametric PDEs* depending on a d-dimensional parameter $\boldsymbol{y} \in [-1,1]^d$. Such a situation arises in particular in Uncertainty Quantification (UQ). There, a mathematical model is described by a PDE depending on the parameters $\boldsymbol{y}$, which in turn can for instance determine boundary conditions, forcing terms or diffusion coefficients. It is known for a wide range of linear and nonlinear PDE models (see e.g. [4]), that parametric PDE solutions depend analytically on the parameters. In addition, for these models usually one has precise knowledge on the domain of holomorphic extension of the objective function u, i.e. knowledge of the constants $(\rho_j)_{j=1}^d$ in Thm. 3.5. These constants determine the sets of multiindices Λ_ε in (3.3). As our proofs are constructive and based on the sets Λ_ε, such information can be leveraged to a priori guide the identification of suitable network architectures.

4.2.2 Infinite-dimensional ($d = \infty$) case

The expression rate analysis becomes more involved, if the objective function u depends on an infinite dimensional parameter (i.e., a parameter sequence) $\boldsymbol{y} \in [-1,1]^{\mathbb{N}}$. Such functions occur in UQ for instance if the uncertainty is described by a Karhunen-Loeve expansion. Under certain circumstances, u can be expressed by a so-called *generalized polynomial chaos (gpc) expansion.* Reapproximating truncated gpc expansions by NNs leads to expression rate results for the approximation of infinite dimensional functions, as we showed in [25]. One drawback of [25] is however, that the proofs crucially relied on the assumption that u is holomorphic on certain polydiscs containing $[-1,1]^{\mathbb{N}}$. This criterion is not always met in practice [4]. To overcome this restriction, we will generalize the expression rate results of [25] in the forthcoming paper [20], by basing the analysis on the present results for the approximation of d-variate functions which are merely assumed to be analytic in some (possibly small) neighborhood of $[-1,1]^d$.

4.2.3 Extension to non-holomorphic settings

The present results were based on the quantified holomorphy of the map $u : [-1,1]^d \to \mathbb{N}$. While this can be perceived as a strong requirement (and, consequently, limitation) of the present results, let us indicate that, in fact, the present deep ReLU NN emulation rate bounds do cover more general situations. The key observation is that deep ReLU NNs are closed under concatenation (or under composition of realizations) as we explained in Section 2.2.3.

Let us give a specific example from high-dimensional integration, where the task is to evaluate the integral

$$\int_{[-1,1]^d} u(\boldsymbol{y})\pi(\boldsymbol{y})\mathrm{d}\boldsymbol{y} \ . \tag{4.1}$$

Here, $u : [-1,1]^d \to \mathbb{R}$ is a function which is holomorphic in a polyellipse $\mathcal{E}_{\boldsymbol{\rho}}$ as in (3.1) and π denotes an a-priori given probability density on the co-ordinates $y_1, ..., y_d$. Assuming that the co-ordinates are independent, the density π factors, i.e. $\pi = \bigotimes_{j=1}^d \pi_j$ with certain marginal probability densities π_j which we assume to be absolutely continuous w.r. to the Lebesgue measure, i.e. i.e. $\int_{-1}^1 \pi_j(\xi)\mathrm{d}\xi = 2$. In the case that the *marginals* $\pi_j > 0$ *are simple functions* for example on finite partitions $\mathcal{T}_j$ of $[-1,1]$ (as e.g. if π_j is a histogram for the law of y_j estimated from empirical data), the changes of coordinates in (4.1)

$$T_j(y_j) := -1 + \int_{-1}^{y_j} \pi_j(\xi_j)\mathrm{d}\xi_j : [-1,1] \to [-1,1] \ , \quad j = 1, ..., d \tag{4.2}$$

are bijective. Furthermore, in this case each component map $T_j : [-1,1] \to [-1,1]$ is bijective, continuous and piecewise affine, and can therefore *be exactly represented by a σ_1-NN of depth* 1 *and width proportional to $\#(\mathcal{T}_j)$.*

Denote by $T = (T_1, ..., T_d)^\top$ the d-variate diagonal transformation, and let $T^{-1} : [-1,1]^d \to [-1,1]^d$ denote its inverse (which is also continuous, piecewise linear). Denoting by $dT^{-1}(\boldsymbol{x})$ the Jacobian matrix of T^{-1} at $\boldsymbol{x} \in [-1,1]^d$ we may then rewrite (4.1) as

$$\int_{[-1,1]^d} u(\boldsymbol{y})\pi(\boldsymbol{y})\mathrm{d}\boldsymbol{y} = \int_{[-1,1]^d} u(T^{-1}(\boldsymbol{x}))\pi(T^{-1}(\boldsymbol{x}))\det dT^{-1}(\boldsymbol{x})\mathrm{d}\boldsymbol{x} = \int_{[-1,1]^d} g(\boldsymbol{x})\mathrm{d}\boldsymbol{x}, \tag{4.3}$$

where $g = u \circ T^{-1}$ is not continuously differentiable. Here we have used that $dT^{-1}(T(\boldsymbol{y})) = (dT(\boldsymbol{y}))^{-1}$ and $\det(dT(\boldsymbol{y})) = \pi(\boldsymbol{y})$, i.e. $\det dT^{-1}(\boldsymbol{x}) = \pi(T^{-1}(\boldsymbol{x}))^{-1}$.

Now, the function $\tilde{g}_{\mathcal{N}} := \tilde{u}_{\mathcal{N}} \circ T^{-1}$ with the σ_1-NN $\tilde{u}_{\mathcal{N}}$ constructed in Theorem 3.7 is a σ_1-NN which still affords the error bound (3.19): Denote for $n \in \mathbb{N}$ and $f \in W^{1,\infty}([-1,1]^d, \mathbb{R}^n)$

$$|f|_{W^{1,\infty}([-1,1]^d,\mathbb{R}^n)} := \sup_{\boldsymbol{x} \neq \boldsymbol{y} \in [-1,1]^d} \frac{\|f(\boldsymbol{x}) - f(\boldsymbol{y})\|}{\|\boldsymbol{x} - \boldsymbol{y}\|},$$

where $\|\cdot\|$ is the Euclidean norm on $\mathbb{R}^n$ resp. on $\mathbb{R}^d$. As usual, for $n = 1$ we write $|f|_{W^{1,\infty}([-1,1]^d)} := |f|_{W^{1,\infty}([-1,1]^d,\mathbb{R})}$ instead. With these conventions, it holds

$$\begin{aligned}
\|g(\cdot) - \tilde{g}_{\mathcal{N}}(\cdot)\|_{W^{1,\infty}([-1,1]^d)} &= \|u \circ T^{-1}(\cdot) - \tilde{u}_{\mathcal{N}} \circ T^{-1}(\cdot)\|_{W^{1,\infty}([-1,1]^d)} \\
&= \|u \circ T^{-1}(\cdot) - \tilde{u}_{\mathcal{N}} \circ T^{-1}(\cdot)\|_{L^{\infty}([-1,1]^d)} + |u \circ T^{-1}(\cdot) - \tilde{u}_{\mathcal{N}} \circ T^{-1}(\cdot)|_{W^{1,\infty}([-1,1]^d)} \\
&\leq \|u(\cdot) - \tilde{u}_{\mathcal{N}}(\cdot)\|_{L^{\infty}([-1,1]^d)} + |u(\cdot) - \tilde{u}_{\mathcal{N}}(\cdot)|_{W^{1,\infty}([-1,1]^d)} |T^{-1}|_{W^{1,\infty}([-1,1]^d,\mathbb{R}^d)} \\
&\leq C \exp\left(-\beta' \mathcal{N}^{\frac{1}{d+1}}\right) \qquad (4.4)
\end{aligned}$$

for a constant C which now additionally depends on $|T^{-1}(\cdot)|_{W^{1,\infty}([-1,1]^d,\mathbb{R}^d)}$. The approximation of the integral (4.1) can thus be reduced to the problem of approximating the integral of the surrogate $\tilde{g}_{\mathcal{N}}$, which can be efficiently represented by a σ_1-NN.

More generally, if $\pi : [-1,1]^d \to (0,\infty)$ is for example a continuous density function (not necessarily a product of its marginals) there exists a bijective transport $T : [-1,1]^d \to [-1,1]^d$ such that analogous to (4.3) it holds $\int_{[-1,1]^d} u(\boldsymbol{y})\pi(\boldsymbol{y})\mathrm{d}\boldsymbol{y} = \int_{[-1,1]^d} u(T^{-1}(\boldsymbol{x}))\mathrm{d}\boldsymbol{x}$ (contrary to the situation above, this transformation T is not diagonal in general). One explicit representation of such a transport is provided by the Knothe-Rosenblatt transport, see, e.g. [24, Section 2.3]. It has the property that T inherits the smoothness of π, cp. [24, Remark 2.19]. In case T^{-1} can be realized without error by a σ_1 (or σ_r) network, we find again an estimate of the type (3.19). If T^{-1} does not allow an explicit representation by a NN however, we may still approximate T^{-1} by a NN $\tilde{S}_{\mathcal{N}}$ to obtain a NN $\tilde{g}_{\mathcal{N}} := \tilde{u}_{\mathcal{N}} \circ \tilde{S}_{\mathcal{N}}$ approximating $g = u \circ T^{-1}$. This will introduce an additional error in (4.4) due to the approximation of T^{-1} addressed in [20].

References

[1] 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(1):8–45, 2019. ArXiv:1705.01714.

[2] A. Bonito, R. DeVore, D. Guignard, P. Jantsch, and G. Petrova. Polynomial approximation of anisotropic analytic functions of several variables, 2019. ArXiv:1904.12105.

[3] C. K. Chui and X. Li. Approximation by ridge functions and neural networks with one hidden layer. *J. Approx. Theory*, 70(2):131–141, 1992.

[4] A. Cohen, A. Chkifa, and C. Schwab. Breaking the curse of dimensionality in sparse polynomial approximation of parametric PDEs. *Journ. Math. Pures et Appliquees*, 103(2):400–428, 2015.

[5] A. Cohen, R. DeVore, and C. Schwab. Convergence rates of best N-term Galerkin approximations for a class of elliptic sPDEs. *Found. Comput. Math.*, 10(6):615–646, 2010.

[6] P. Davis. *Interpolation and Approximation*. Dover Books on Mathematics. Dover Publications, 1975.

[7] W. E and Q. Wang. Exponential convergence of the deep neural network approximation for analytic functions. *Sci. China Math.*, 61(10):1733–1740, 2018.

[8] D. Elbrächter, P. Grohs, A. Jentzen, and C. Schwab. DNN Expression Rate Analysis of High-dimensional PDEs: Application to Option Pricing. Technical Report 1809.07669, arXiv, 2018.

[9] A. Erdélyi, W. Magnus, F. Oberhettinger, and F. G. Tricomi. *Higher transcendental functions*, volume 2. McGraw-Hill, New York, 1953. Based on notes left by Harry Bateman.

[10] K. Hornik. Approximation capabilities of multilayer feedforward networks. *Neural networks*, 4(2):251–257, 1991.

[11] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

[12] B. Li, S. Tang, and H. Yu. Better approximations of high dimensional smooth functions by deep neural networks with rectified power units. Mar 2019. ArXiv:1903.05858.

[13] S. Liang and R. Srikant. Why deep neural networks for function approximation? In *Proc. of ICLR 2017*, pages 1 – 17, 2017. ArXiv:1610.04161.

[14] H. Mhaskar. Neural networks for localized approximation of real functions. In *Neural Networks for Signal Processing III - Proceedings of the 1993 IEEE-SP Workshop*, pages 190–196. IEEE, 1993.

[15] H. Mhaskar and C. Micchelli. Approximation by superposition of sigmoidal and radial basis functions. *Advances in Applied Mathematics*, 13(3):350–373, 1992.

[16] H. N. Mhaskar. Approximation properties of a multilayered feedforward artificial neural network. *Advances in Computational Mathematics*, 1(1):61–80, Feb 1993.

[17] H. N. Mhaskar. Neural networks for optimal approximation of smooth and analytic functions. *Neural Computation*, 8:164–177, 1996.

[18] F. Nobile, R. Tempone, and C. G. Webster. An anisotropic sparse grid stochastic collocation method for partial differential equations with random input data. *SIAM J. Numer. Anal.*, 46(5):2411–2442, 2008.

[19] J. A. A. Opschoor, P. C. Petersen, and C. Schwab. Deep ReLU Networks and High-Order Finite Element Methods. Technical Report 2019-07, Seminar for Applied Mathematics, ETH Zürich, Switzerland, 2019.

[20] J. A. A. Opschoor, C. Schwab, and J. Zech. DNN expression rates for Bayesian PDE inversion. Technical Report 2019, Seminar for Applied Mathematics, ETH Zürich, Switzerland, 2019. (in preparation).

[21] P. Petersen and F. Voigtlaender. Optimal approximation of piecewise smooth functions using deep ReLU neural networks. *Neural Networks*, 108:296–330, 2018.

[22] A. Pinkus. Approximation theory of the MLP model in neural networks. In *Acta numerica, 1999*, volume 8 of *Acta Numer.*, pages 143–195. Cambridge Univ. Press, Cambridge, 1999.

[23] D. Rolnik and M. Tegmark. The power of deeper networks for expressing natural functions. Technical Report 1705.05502v1, ArXiv, 2017.

[24] F. Santambrogio. *Optimal transport for applied mathematicians*, volume 87 of *Progress in Nonlinear Differential Equations and their Applications*. Birkhäuser/Springer, Cham, 2015. Calculus of variations, PDEs, and modeling.

[25] C. Schwab and J. Zech. Deep learning in high dimension: neural network expression rates for generalized polynomial chaos expansions in UQ. *Anal. Appl. (Singap.)*, 17(1):19–55, 2019.

[26] D. Yarotsky. Error bounds for approximations with deep ReLU networks. *Neural Netw.*, 94:103–114, 2017.

[27] S. T. Yau and L. Zhang. An upper estimate of integral points in real simplices with an application to singularity theory. *Math. Res. Lett.*, 13(5-6):911–921, 2006.

[28] J. Zech. Sparse-Grid Approximation of High-Dimensional Parametric PDEs, 2018. Dissertation 25683, ETH Zürich.

[29] J. Zech, D. Dung, and C. Schwab. Multilevel approximation of parametric and stochastic pdes. Technical Report 2018-05, Seminar for Applied Mathematics, ETH Zürich, 2018. (to appear in M3AS 2019).

[30] J. Zech and C. Schwab. Convergence rates of high dimensional Smolyak quadrature. Technical Report 2017-27, Seminar for Applied Mathematics, ETH Zürich, 2017. Revision 1.