Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Seminar for Applied Mathematics SAM

On the approximation of functions by tanh neural networks

T. De Ryck and S. Lanthaler and S. Mishra

Research Report No. 2021-14

April 2021
Latest revision: August 2021

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

On the approximation of functions by tanh neural networks

Tim De Ryck*, Samuel Lanthaler, Siddhartha Mishra

Seminar for Applied Mathematics, ETH Zürich, Rämistrasse 101, 8092 Zürich, Switzerland

Abstract

We derive bounds on the error, in high-order Sobolev norms, incurred in the approximation of Sobolev-regular as well as analytic functions by neural networks with the hyperbolic tangent activation function. These bounds provide explicit estimates on the approximation error with respect to the size of the neural networks. We show that tanh neural networks with only two hidden layers suffice to approximate functions at comparable or better rates than much deeper ReLU neural networks.

Keywords: neural networks, tanh, function approximation, deep learning

1. Introduction

Deep learning, relying on the use of deep artificial neural networks for regression and classification, has been very successful in different contexts in science and engineering in recent years [1]. These include image recognition, natural language understanding, machine translation, game intelligence, robotics, autonomous systems and protein folding.

Deep learning is also being increasingly used in scientific computing, particularly in the numerical solution of partial differential equations (PDEs). A very incomplete list of examples for the successful use of deep learning in this context includes the solution of high-dimensional linear and semi-linear parabolic partial differential equations [2, 3] and references therein, the solution of parametric partial differential equations that arise in many-query problems like uncertainty quantification (UQ), PDE constrained optimization and (Bayesian) inverse problems [4, 5, 6, 7, 8] and in infinite-dimensional operator learning frameworks [9, 10, 11]. Another avenue for the application of deep neural networks in scientific computing is provided by *physics-informed neural networks* (PINNs) [12, 13, 14, 15, 16], which serve as replacements for traditional numerical methods for both forward as well as inverse problems for PDEs.

[*] Corresponding author
Email address: tim.deryck@sam.math.ethz.com (Tim De Ryck)

The question of why deep neural networks are so successful at many diverse tasks in very different fields eludes a definitive answer. A very partial explanation may lie in the fact that artificial neural networks are *universal approximators* i.e., any continuous (even measurable) mapping can be approximated by artificial neural networks to arbitrarily high accuracy [17, 18, 19] and references therein. However, such universality results only imply the existence of a (shallow) neural network and do not provide any quantitative information (bounds) on the width of the underlying neural networks.

The task of quantitatively relating the size and architecture of neural networks to their expressivity i.e., accuracy in approximating functions of a certain hypothesis class, has received considerable attention in the literature in the last few years. A seminal work in the direction is [20], where the author derived explicit estimates on the size (width and depth) of a neural network with a ReLU activation function for approximating Lipschitz functions to any given accuracy in the L^∞-norm. Expressivity results for such ReLU neural networks in Sobolev norms were presented in [21, 22, 4] and references therein, see also [23, 24, 25, 26, 27] and references therein for further approximation results for ReLU and related ReQU and RePU activation functions.

Despite the fact that several quantitative results on the expressivity of neural networks have been obtained in recent years, we highlight some of the lacunae of the current state of the art in this direction,

- Most of the available results are on the expressivity and approximation properties of ReLU neural networks. Although ReLU activations are very common in practical applications of deep learning, there is a large number of areas where other activation functions are employed. One of the most popular activations is the tanh (hyperbolic tangent) activation function and the related sigmoid or logistic function (a scaled and shifted tanh). These activation functions are the basis of heavily used recurrent neural network (RNN) architectures such as LSTM [28] and GRU [29]. Other areas where smooth activation functions such as tanh are preferred over ReLU is in physics-informed neural networks (PINNs) for solving forward and inverse problems for PDEs [13, 14, 15, 16] and references therein, and in the use of quasi-random training points [30, 31]. Although the approximation abilities of general smooth activation functions have been investigated in [32, 33, 34, 35, 36, 37] and references therein, it is fair to say that the level of detail in existing results for the expressivity of ReLU neural networks, is not yet available for tanh neural networks.

- Moreover, most of the approximation results for smooth activation functions, with the exception of the recent paper [34], measure error in L^p-norms. However, it is essential to measure errors in higher-order Sobolev norms for many applications, such as PINNs where the neural network needs to be differentiated in order to evaluate the underlying PDE residual.

- A persistent focus of approximation results for neural networks has been to highlight the role of depth of the neural network, see [38] for a review and further references. In particular, there are several results to the effect that very deep neural networks are, in some sense, more expressive than shallower networks, which in turn might explain the superior performance of deep neural networks in many applications [39, 40, 41, 42]. The empirical superiority of deep networks over their shallower counterparts has indeed been observed in many applications in computer science. However, in the context of scientific computing, empirical experience has revealed that shallower but wider networks result in superior performance over deep and narrow neural networks, see [43] and references therein. A reason for this observation lies in the fact that deeper networks might be harder to train in the relatively data poor regime of scientific computing. Some theoretical understanding of this deterioration of performance for deeper networks, at least in the context of ReLU networks is provided in [44]. However, most of the available approximation theory results trade width for depth and there is little theoretical understanding of why relatively shallow networks can perform well in some contexts.

- Most of the available results on expressivity focus on asymptotic approximation rates i.e., the complexity of the network as the approximation error $\epsilon \to 0$. However, the fundamental question is how large a neural network should be to provide a certain accuracy of this approximation. This requires going beyond asymptotic approximation rates and providing explicit bounds on the underlying constants. Such explicit bounds are available for approximation in Hilbert and L^p spaces with $p < \infty$ [17, 45, 46, 47, 48, 49] and approximation using a non-standard activation function [50, 51], but they remain mostly unavailable for function approximation in $W^{k,\infty}$ spaces with neural networks with widely used activation functions.

- The approximation error is only one component of the total error of neural networks, with optimization and generalization errors being the other components [52, 53]. In particular, standard approaches to estimate the generalization error such as covering number estimates [54] or Rademacher complexity [52] require explicit estimates on the weights of the underlying neural networks, in addition to bounds on their width and depth. Such estimates on weights of the best approximations of functions in the class of neural networks are rarely available in the current literature.

The main objective of this paper is to address some of the afore-mentioned deficiencies in the literature on approximation properties of neural networks. We will focus on the expressivity of neural networks with the very popular tanh activation function and will aim to prove error and complexity bounds in high-order Sobolev norms for such tanh neural networks in approximating functions belonging to Sobolev spaces as well as C^k-spaces. We go beyond the usual practice of proving only asymptotic convergence rates and will provide explicit approximation error bounds for explicit network architectures in order to answer the question of *"How large should a neural network be to approximate a specified function to some chosen accuracy $\epsilon > 0$?"*. All our results will be for tanh neural networks with at most two hidden layers.

A key application of our results is on the approximation of analytic functions by tanh neural networks. We will prove that a two hidden layer tanh neural network suffices to approximate an analytic function at an exponential rate, in terms of the network width, even in Sobolev norms. This result provides an improvement over available results for the approximation of analytic functions by ReLU neural networks [55, 4, 22] and also neural networks with smooth activation functions [56] and further illustrate the powers of rather shallow tanh networks at approximating smooth functions. Finally, we also derive explicit bounds on the width of the tanh neural networks as well as asymptotic bounds on their weights, thus paving the way for bounds on the generalization error for these neural networks.

The rest of the paper is organized as follows: in Section 2, we introduce the notation for the rest of the paper. Our main results, presented in Section 5, rely on the uniform approximation of polynomials by tanh neural networks, discussed in Section 3, and on the approximation of a partition of unity, presented in Section 4. In Section 6, we discuss the contents of this paper and distinguish them from other related papers.

2. Preliminaries

We start by providing an overview of all the notation and the definitions that will be used frequently throughout the paper.

2.1. Multi-index notation

For $d \in \mathbb{N}$, we call a d-tuple of non-negative integers $\alpha \in \mathbb{N}_0^d$ a multi-index. We write $|\alpha| = \sum_{i=1}^d \alpha_i$, $\alpha! = \prod_{i=1}^d \alpha_i!$ and, for $x \in \mathbb{R}^d$, we denote by $x^\alpha = \prod_{i=1}^d x_i^{\alpha_i}$ the corresponding multinomial. Given two multi-indices $\alpha, \beta \in \mathbb{N}_0^d$, we say that $\alpha \leq \beta$ if, and only if, $\alpha_i \leq \beta_i$ for all $i = 1, \ldots, d$. For a multi-index α, we define the following multinomial coefficient

$$\binom{|\alpha|}{\alpha} = \frac{|\alpha|!}{\alpha!}, \tag{1}$$

and, given $\alpha \leq \beta$, we define a corresponding multinomial coefficient by

$$\binom{\beta}{\alpha} = \prod_{i=1}^d \binom{\beta_i}{\alpha_i} = \frac{\beta!}{\alpha!(\beta-\alpha)!}. \tag{2}$$

For $\Omega \subseteq \mathbb{R}^d$ and a function $f : \Omega \to \mathbb{R}$ we denote by

$$D^\alpha f = \frac{\partial^{|\alpha|} f}{\partial x_1^{\alpha_1} \cdots \partial x_d^{\alpha_d}} \tag{3}$$

the classical or distributional (i.e. weak) derivative of f. We will frequently encounter the set $P_{n,d} = \{\alpha \in \mathbb{N}_0^d : |\alpha| = n\}$ (notation as in [57]). In particular, we will need estimates on its cardinality. This is the subject of the following lemma.

Lemma 2.1. *Let $n \in \mathbb{N}$, $d \in \mathbb{N}_{\geq 2}$ and let $P_{n,d} = \{\alpha \in \mathbb{N}_0^d : |\alpha| = n\}$. Then*

$$\left|P_{n,d}\right| = \binom{n+d-1}{n} \leq \sqrt{\pi}\min\{e^{d-1}n^{d-1}, e^n(d-1)^n\} \quad \textit{and} \quad \left|P_{d,d}\right| \leq 5^d. \tag{4}$$

Proof. It is well known that $\left|P_{n,d}\right| = \binom{n+d-1}{n}$. We use Stirling's approximation,

$$\begin{aligned}\left|P_{n,d}\right| = \binom{n+d-1}{n} &\leq \frac{e(n+d-1)^{n+d-1/2}}{2\pi n^{n+1/2}(d-1)^{d-1/2}} \\ &\leq \frac{e}{2\pi}\left(\frac{n+d-1}{d-1}\right)^{d-1}\left(\frac{n+d-1}{n}\right)^{n}\sqrt{\frac{n+d-1}{n(d-1)}} \\ &\leq \frac{e}{\sqrt{2\pi}}\left(1+\frac{n}{d-1}\right)^{d-1}\left(1+\frac{d-1}{n}\right)^{n}.\end{aligned}$$

To estimate the last term, we note that there are two possible approximations: for $a, b \geq 1$ it holds that $(1+a/b)^b \leq ea^b$ and also $(1+a/b)^b \leq e^a$. Using the fact that $e^2/\sqrt{2\pi} \leq \sqrt{\pi}$, we obtain

$$\left|P_{n,d}\right| \leq \sqrt{\pi}e^{d-1}n^{d-1} \qquad \text{and} \qquad \left|P_{n,d}\right| \leq \sqrt{\pi}e^n(d-1)^n. \tag{5}$$

Setting $n = d$ for $\lambda \in \mathbb{N}$, we also find that

$$\left|P_{d,d}\right| \leq \frac{e}{\sqrt{2\pi}}\left(1+\frac{d}{d-1}\right)^{d}\left(1+\frac{d-1}{d}\right)^{d} \leq \left(3+\frac{d}{d-1}\right)^{d} \leq 5^d, \tag{6}$$

since $\frac{e}{\sqrt{2\pi}} \leq 1$ and $\frac{x}{x-1} \leq 2$ for $x \geq 2$.

□

2.2. Sobolev spaces

Let $d \in \mathbb{N}$, $1 \leq p \leq \infty$ and let $\Omega \subseteq \mathbb{R}^d$ be open. We denote by $L^p(\Omega)$ the usual Lebesgue space and for $k \in \mathbb{N}_0$ we define the Sobolev space $W^{k,p}(\Omega)$ as

$$W^{k,p}(\Omega) = \{f \in L^p(\Omega) : D^\alpha f \in L^p(\Omega) \text{ for all } \alpha \in \mathbb{N}_0^d \text{ with } |\alpha| \leq k\}. \tag{7}$$

For $p < \infty$, we define the following seminorms on $W^{k,p}(\Omega)$,

$$|f|_{W^{m,p}(\Omega)} = \left(\sum_{|\alpha|=m}\|D^\alpha f\|_{L^p(\Omega)}^p\right)^{1/p} \qquad \text{for } m = 0, \ldots, k, \tag{8}$$

and for $p = \infty$ we define

$$|f|_{W^{m,\infty}(\Omega)} = \max_{|\alpha|=m} \|D^\alpha f\|_{L^\infty(\Omega)} \qquad \text{for } m = 0, \dots, k. \tag{9}$$

Based on these seminorms, we can define the following norm for $p < \infty$,

$$\|f\|_{W^{k,p}(\Omega)} = \left(\sum_{m=0}^{k} |f|^p_{W^{m,p}(\Omega)} \right)^{1/p}, \tag{10}$$

and for $p = \infty$ we define the norm

$$\|f\|_{W^{k,\infty}(\Omega)} = \max_{0 \le m \le k} |f|_{W^{m,\infty}(\Omega)}. \tag{11}$$

The space $W^{k,p}(\Omega)$ equipped with the norm $\|\cdot\|_{W^{k,p}(\Omega)}$ is a Banach space.

2.3. Neural networks

In this paper, we will consider function approximation using feedforward artificial neural networks where only connections between neighbouring layers are allowed. In the following, we formally introduce our definition of a neural network and the related terminology.

Let $L \in \mathbb{N}$ and $l_0, \dots, l_L \in \mathbb{N}$. Let $\sigma : \mathbb{R} \to \mathbb{R}$ be an *activation function* and define the parameter space

$$\Theta = \bigcup_{L \in \mathbb{N}} \bigcup_{l_0, \dots, l_L \in \mathbb{N}} \bigtimes_{k=1}^{L} \left(\mathbb{R}^{l_k \times l_{k-1}} \times \mathbb{R}^{l_k} \right). \tag{12}$$

For $\theta \in \Theta$, we define $\theta_k := (W_k, b_k)$ and $\mathcal{A}_k : \mathbb{R}^{l_{k-1}} \to \mathbb{R}^{l_k} : x \mapsto W_k x + b_k$ for $1 \le k \le L$ and we denote by $\Psi_\theta : \mathbb{R}^{l_0} \to \mathbb{R}^{l_L}$, $x \mapsto \Psi_\theta(x)$, the function

$$\Psi_\theta(x) = \begin{cases} \mathcal{A}_1(x) & L = 1, \\ (\mathcal{A}_L \circ \sigma \circ \mathcal{A}_{L-1} \circ \sigma \circ \dots \circ \sigma \circ \mathcal{A}_1)(x) & L \ge 2, \end{cases} \tag{13}$$

where σ is applied element-wise. We refer to Ψ_θ as the realization of the *neural network* associated to the parameter θ with L layers and widths $(l_0, l_1, \dots, l_L)$. We refer to the first $L - 1$ layers as *hidden layers*. For $1 \le k \le L$, we say that layer k has width l_k and we refer to W_k and b_k as the *weights and biases* corresponding to layer k. The width of Ψ_θ is defined as $\max(l_0, \dots, l_L)$. If $L = 2$, we say that Ψ_θ is a *shallow neural network*; if $L \ge 3$, we say that Ψ_θ is a *deep neural network*. Hence, a shallow neural network has exactly one hidden layer whereas deep neural networks can have two or more hidden layers.

In this work, we will focus on neural networks which use the hyperbolic tangent as activation function, defined by

$$\sigma(x) := \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad \text{for } x \in \mathbb{R}. \tag{14}$$

We will refer to these networks as *tanh neural networks*. Even though our ideas can be carried over to other smooth activation functions, focusing on a particular activation will allow us to prove precise and explicit bounds without sacrificing the clarity of our arguments. In particular, we note that all results of this work directly apply to the sigmoid or logistic activation function, which is simply a shifted and scaled version of the hyperbolic tangent.

We close this section by recalling some basic properties of neural network calculus which we will use throughout, without explicitly referring to them.

Proposition 2.2 (Parallelization of neural networks)**.** *Let $L \in \mathbb{N}$, $l_0, l_0', \ldots, l_L, l_L' \in \mathbb{N}$ and $\theta, \vartheta \in \Theta$ such that Ψ_θ is a neural network with widths $(l_0, \ldots, l_L)$ and Ψ_ϑ is a neural network with widths $(l_0', \ldots, l_L')$. Then there exists $\eta \in \Theta$ such that Ψ_η is a neural network with widths $(l_0 + l_0', \ldots, l_L + l_L')$ for which it holds that $\Psi_\eta(x) = (\Psi_\theta((x_1, \ldots, x_{l_0})), \Psi_\vartheta((x_{l_0+1}, \ldots, x_{l_0+l_0'})))$ for all $x \in \mathbb{R}^{l_0+l_0'}$.*

Proposition 2.3 (Composition of neural networks)**.** *Let $L, L' \in \mathbb{N}$, $l_0, \ldots, l_L = l_0', \ldots, l_{L'}' \in \mathbb{N}$ and $\theta, \vartheta \in \Theta$ such that Ψ_θ is a neural network with widths $(l_0, \ldots, l_L)$ and Ψ_ϑ is a neural network with widths $(l_0', \ldots, l_{L'}')$. Then there exists $\eta \in \Theta$ such that Ψ_η is a neural network with widths $(l_0, \ldots, l_L = l_0', \ldots, l_{L'}')$ for which it holds that $\Psi_\eta = \Psi_\vartheta \circ \Psi_\theta$.*

3. Uniform approximation of polynomials

The first step in our strategy for deriving bounds on approximation error for tanh neural networks is to provide uniform bounds in Sobolev norms, on the error for approximating polynomials by shallow tanh neural networks. We do so in the current section.

The observation that a shallow neural network of fixed size can approximate monomials to arbitrary accuracy in the supremum norm was already observed in [36]. A generalization of this approximation result to Sobolev norms was proven in e.g. [34]. In the current section, we present a *novel* generalization that allows us to obtain explicit error estimates for the *uniform* approximation of all polynomials of a certain maximal degree, which will be crucial for the efficient approximation of analytic functions.

3.1. Univariate polynomials

We first describe how to approximate univariate polynomials of any degree with tanh neural networks. We introduce the p-th order central finite difference operator δ_h^p for any $f \in C^{p+2}([a,b])$ for some $p \in \mathbb{N}$ by

$$\delta_h^p[f](x) = \sum_{i=0}^{p} (-1)^i \binom{p}{i} f\left(x + \left(\frac{p}{2} - i\right) h\right). \tag{15}$$

Next we define for any $p \in \mathbb{N}$, $q \in 2\mathbb{N} - 1$ and $M > 0$ the monomials $f_p : [-M, M] \to \mathbb{R}$ and the tanh neural networks $\hat{f}_{q,h} : [-M, M] \to \mathbb{R}$ as

$$f_p(y) := y^p \qquad \text{and} \qquad \hat{f}_{q,h}(y) := \frac{\delta_{hy}^q[\sigma](0)}{\sigma^{(q)}(0) h^q}. \tag{16}$$

We first prove that these neural networks are accurate approximations to monomials with odd degree.

Lemma 3.1. *Let $k \in \mathbb{N}_0$ and $s \in 2\mathbb{N}-1$. Then it holds that for all $\epsilon > 0$ there exists a shallow tanh neural network $\Psi_{s,\epsilon} : [-M, M] \to \mathbb{R}^{\frac{s+1}{2}}$ of width $\frac{s+1}{2}$ such that*

$$\max_{\substack{p\leq s,\\ p \text{ odd}}} \left\| f_p - (\Psi_{s,\epsilon})_{\frac{p+1}{2}} \right\|_{W^{k,\infty}} \leq \epsilon, \tag{17}$$

Moreover, the weights of $\Psi_{s,\epsilon}$ scale as $O\left(\epsilon^{-s/2}(2(s+2)\sqrt{2M})^{s(s+3)}\right)$ for small ϵ and large s.

Proof. Let $p \leq s$ be odd and let $0 < h < 2/pM$. Let $0 \leq m \leq \min\{k, p+1\}$. Then Taylor's theorem guarantees the existence of $\xi_{x,i}$ such that

$$\begin{aligned}
\frac{d^m}{dx^m}\delta^p_{hx}[\sigma](0) &= \sum_{i=0}^{p}(-1)^i\binom{p}{i}\left(\frac{p}{2}-i\right)^m h^m \cdot \sigma^{(m)}\left(\left(\frac{p}{2}-i\right)hx\right)\\
&= \sum_{i=0}^{p}(-1)^i\binom{p}{i}\left(\frac{p}{2}-i\right)^m h^m \left(\sum_{l=m}^{p+1}\frac{\sigma^{(l)}(0)}{(l-m)!}\left(\frac{p}{2}-i\right)^{l-m}(hx)^{l-m}\right)\\
&\quad + \sum_{i=0}^{p}(-1)^i\binom{p}{i}\left(\frac{p}{2}-i\right)^m h^m \frac{\sigma^{(p+2)}(\xi_{x,i})}{(p+2-m)!}\left(\frac{p}{2}-i\right)^{p+2-m}(hx)^{p+2-m}.
\end{aligned}$$

From [58, Theorem 1] it follows that

$$\sum_{i=0}^{p}(-1)^i\binom{p}{i}\left(\frac{p}{2}-i\right)^l = p!\,\delta(l-p) = \begin{cases} p!, & (l=p),\\ 0, & (l\neq p).\end{cases} \tag{18}$$

for $l = 0, \ldots, p$. We observe that (18) remains true also for $l = p+1$, since all summands change sign when i is replaced by $p - i$. Using this fact, we can then rewrite the first term as

$$\begin{aligned}
&\sum_{i=0}^{p}(-1)^i\binom{p}{i}\left(\frac{p}{2}-i\right)^m h^m \left(\sum_{l=m}^{p+1}\frac{\sigma^{(l)}(0)}{(l-m)!}\left(\frac{p}{2}-i\right)^{l-m}(hx)^{l-m}\right)\\
&\quad = h^m \sum_{l=m}^{p+1}\frac{\sigma^{(l)}(0)}{(l-m)!}(hx)^{l-m}\sum_{i=0}^{p}(-1)^i\binom{p}{i}\left(\frac{p}{2}-i\right)^l\\
&\quad = \left\{\begin{matrix} h^m\dfrac{\sigma^{(p)}(0)}{(p-m)!}(hx)^{p-m}p!, & 0\leq m\leq p\\ 0, & m = p+1\end{matrix}\right\} = h^p\sigma^{(p)}(0)f_p^{(m)}(x).
\end{aligned} \tag{19}$$

Combining the previous results, it thus follows that we have

$$\hat{f}^{(m)}_{p,h}(x) - f^{(m)}_p(x) = \sum_{i=0}^{p}(-1)^i\binom{p}{i}\frac{1}{(p+2-m)!}\frac{\sigma^{(p+2)}(\xi_{x,i})}{\sigma^{(p)}(0)}\left(\frac{p}{2}-i\right)^{p+2}h^2x^{p+2-m}.$$

Together with the lower and upper bounds on the derivatives of σ from Lemma Appendix A.1 and Lemma Appendix A.4, this yields for $m \leq \min(k, p+1)$:

$$\begin{aligned} \left| f_p - \hat{f}_{p,h} \right|_{W^{m,\infty}} &\leq \sum_{i=0}^{p} \binom{p}{i} \frac{\left| \sigma^{(p+2)}(\xi_{x,i}) \right|}{\left| \sigma^{(p)}(0) \right|} \left| \frac{p}{2} - i \right|^{p+2} h^2 M^{p+2} \\ &\leq 2^p \frac{(2(p+2))^{p+3}}{1} \left(\frac{p}{2} \right)^{p+2} h^2 M^{p+2} \\ &\leq (2(p+2)pM)^{p+3} h^2. \end{aligned} \tag{20}$$

If $k \leq p+1$, then this shows that

$$\left\| f_p - \hat{f}_{p,h} \right\|_{W^{k,\infty}} \leq (2(p+2)pM)^{p+3} h^2. \tag{21}$$

If $k > p+1$, let $p+2 \leq m \leq k$. In this case, $f_p^{(m)} = 0$, therefore it suffices to bound $\hat{f}_{p,h}^{(m)}$. We see that for $0 < h < 1$,

$$\begin{aligned} \left| \hat{f}_{p,h}^{(m)}(x) \right| &= \left| \frac{1}{h^p \sigma^{(p)}(0)} \sum_{i=0}^{p} (-1)^i \binom{p}{i} \left(\frac{p}{2} - i \right)^m h^m \cdot \sigma^{(m)} \left(\left(\frac{p}{2} - i \right) hx \right) \right| \\ &\leq 2 \sum_{i=0}^{p} \binom{p}{i} \left| \frac{p}{2} - i \right|^m h^2 (2m)^{m+1} \\ &\leq 2^{p+1} \left(\frac{p}{2} \right)^k (2k)^{k+1} h^2 \leq (2pk)^{k+1} h^2. \end{aligned} \tag{22}$$

We thus obtain, for arbitrary $k \in \mathbb{N}$:

$$\left\| f_p - \hat{f}_{p,h} \right\|_{W^{k,\infty}} \leq \left((2(p+2)pM)^{p+3} + (2pk)^{k+1} \right) h^2 =: \epsilon. \tag{23}$$

Furthermore observe that the weights scale as $O\left(\max_i \binom{p}{i} h^{-p} \right)$. For $\epsilon \to 0$ and large p, it holds that $O(h^{-p}) = O\left(\epsilon^{-p/2} ((p+2)\sqrt{2M})^{p(p+3)} \right)$, where the implied constant depends on k. Next, we find using Stirling's approximation that for $0 \leq i \leq p$ it holds that

$$\binom{p}{i} \leq \binom{p}{\frac{p-1}{2}} \leq \frac{e p^{p+1/2}}{2\pi \left(\frac{p-1}{2} \right)^{\frac{p}{2}} \left(\frac{p+1}{2} \right)^{\frac{p}{2}+1}} = O\left(\frac{2^p}{\sqrt{p}} \right). \tag{24}$$

The weights therefore scale as $O\left(\epsilon^{-p/2} (2(p+2)\sqrt{2M})^{p(p+3)} \right)$.

Regarding the network architecture, note that the neurons needed for all $\hat{f}_{p,h}$ are already available in the network $\hat{f}_{s,h}$. This allows us to define the shallow tanh neural network $\Psi_{s,\epsilon}$ by $(\Psi_{s,\epsilon})_p = \hat{f}_{p,h}$ such that it only has $\frac{s+1}{2}$ neurons in its hidden layer. The width follows directly from its definition and the fact that σ is an odd function. □

We would like to state that the above proof is largely inspired by [34, Proposition 4.7] but differs at some crucial points. In particular, we take into account the fact that $\xi_{x,i}$'s are functions of x and derivatives with respect to x have to take this into account.

We now extend the previous result to monomials with even degree. To this end, we rely on the observation that for $n \in \mathbb{N}$ and $\alpha > 0$, it holds that

$$y^{2n} = \frac{1}{2\alpha(2n+1)} \Bigg((y+\alpha)^{2n+1} - (y-\alpha)^{2n+1} - 2\sum_{k=0}^{n-1} \binom{2n+1}{2k} \alpha^{2(n-k)+1} y^{2k} \Bigg). \tag{25}$$

This formula allows us to construct recursively defined tanh neural network approximations of even powers of y. The following lemma quantifies the uniform approximation accuracy of these networks in the Sobolev norm.

Lemma 3.2. *Let $k \in \mathbb{N}_0, s \in 2\mathbb{N}-1$ and $M > 0$. For every $\epsilon > 0$, there exists a shallow tanh neural network $\psi_{s,\epsilon} : [-M, M] \to \mathbb{R}^s$ of width $\frac{3(s+1)}{2}$ such that*

$$\max_{p \le s} \left\| f_p - (\psi_{s,\epsilon})_p \right\|_{W^{k,\infty}} \le \epsilon. \tag{26}$$

Furthermore, the weights scale as $O\left(\epsilon^{-s/2}(\sqrt{M}(s+2))^{3s(s+3)/2}\right)$ for small ϵ and large s.

Proof. For $h, M > 0$, $p \le s$, we define $\hat{f}_{p,h} : [-M-1, M+1] \to \mathbb{R}$ as in (16). For $\epsilon > 0$, $h > 0$ small enough and dependent on ϵ, $\alpha \le 1$ and $y \in [-M, M]$, we define $(\psi_{s,\epsilon}(y))_p = \hat{f}_{p,h}(y)$ for p odd and, for $p = 2n$ even, we define $(\psi_{s,\epsilon}(y))_p = (\psi_{s,\epsilon}(y))_{2n}$ recursively by $(\psi_{s,\epsilon})_0(y) := 1$, and

$$(\psi_{s,\epsilon}(y))_{2n} = \frac{1}{2\alpha(2n+1)} \Bigg(\hat{f}_{2n+1,h}(y+\alpha) - \hat{f}_{2n+1,h}(y-\alpha) - 2\sum_{k=0}^{n-1} \binom{2n+1}{2k} \alpha^{2(n-k)+1} (\psi_{s,\epsilon}(y))_{2k} \Bigg). \tag{27}$$

Moreover, we introduce the notation $E_p = \left\| f_p - (\psi_{s,\epsilon})_p \right\|_{W^{k,\infty}}$. We will prove the statement that for all $\epsilon > 0$, there exists $h > 0$, such that for all $p \le s$, we have

$$E_p \le E_p^* := \frac{2^{p/2}(1+\alpha)^{(p^2+p)/2}}{\alpha^{p/2}} \cdot \epsilon. \tag{28}$$

We first note that choosing h as in Lemma 3.1 implies that

$$\max_{\substack{p \le s, \\ p \text{ odd}}} E_p \le \epsilon, \tag{29}$$

which proves the statement for p odd, since $(1+\alpha)/\alpha \geq 1$. We will now prove (28) for even p using induction. First note that

$$E_2 \leq \frac{1}{6\alpha} \cdot 2\epsilon \leq E_2^*, \tag{30}$$

which proves the base step. To prove the induction step, let $n \in \mathbb{N}$ be such that $2n+1 \leq s$ and $n > 1$, and we assume by the induction hypothesis that $E_{2k} \leq E_{2k}^*$ for all $k < n$. It then follows from (25) and (27), that

$$E_{2n} \leq \frac{1}{2\alpha(2n+1)} \left(E_{2n+1} + E_{2n+1} + 2\sum_{k=1}^{n-1} \binom{2n+1}{2k} \alpha^{2(n-k)+1} E_{2k} \right). \tag{31}$$

Note that by the induction hypothesis and the fact that E_{2k}^* is monotonically increasing in k, we have $E_{2k} \leq E_{2k}^* \leq E_{2(n-1)}^*$. Using also (29), and the fact that $\epsilon \leq E_{2(n-1)}^*$, this allows us to estimate (31), by

$$\begin{aligned} E_{2n} &\leq \frac{1}{\alpha(2n+1)} \left(\max_{\substack{p\leq s,\\ p \text{ odd}}} E_p + \sum_{k=1}^{n-1} \binom{2n+1}{2k} \alpha^{2(n-k)+1} E_{2(n-1)}^* \right) \\ &\leq \frac{1}{\alpha} \left(E_{2(n-1)}^* + (1+\alpha)^{2n+1} E_{2(n-1)}^* \right) \\ &\leq \frac{2}{\alpha} (1+\alpha)^{2n+1} E_{2(n-1)}^*. \end{aligned} \tag{32}$$

Recalling the definition of $E_{2(n-1)}^*$, we obtain

$$E_{2n} \leq \frac{2}{\alpha}(1+\alpha)^{2n+1} E_{2(n-1)}^* \leq \left(\frac{2}{\alpha}(1+\alpha)^{2n+1} \right)^n \cdot \epsilon = E_{2n}^*. \tag{33}$$

This proves the claimed estimate (28) also for the case where $p = 2n$ is even, and therefore concludes the proof of (28).

Next, we optimize (28) by choosing the optimal value of α. Lemma Appendix A.2 proves that the optimal choice is $\alpha = 1/s$. We conclude that for any $\epsilon > 0$, there exists a shallow tanh neural network $\psi_{s,\epsilon}$, with width independent of ϵ, such that

$$\max_{p\leq s} \left\| f_p - (\psi_{s,\epsilon})_p \right\|_{W^{k,\infty}} \leq \sqrt{e}(2es)^{s/2}\epsilon. \tag{34}$$

Replacing $\epsilon \to \epsilon/\sqrt{e}(2es)^{s/2}$ recovers the claimed error bound in the statement of this lemma. To quantify the size of the weights, we observe that equation (27) reveals that the weight bound of Lemma 3.1 needs to be multiplied with a factor

$$\max_k s \binom{s}{2k} \left(\frac{1}{s}\right)^{s-2k} \leq s \sum_{j=0}^{s} \binom{s}{j} \left(\frac{1}{s}\right)^{s-j} \leq s \left(1 + \frac{1}{s}\right)^s = O(s), \tag{35}$$

where we used the binomial theorem. The weight bound can seen to be equal to

$$O\left(\epsilon^{-s/2}s(2\sqrt[4]{2es}\sqrt{2(M+1)}(s+2))^{s(s+3)}\right) = O\left(\epsilon^{-s/2}(\sqrt{M}(s+2))^{3s(s+3)/2}\right) \tag{36}$$

for small ϵ and large s. This proves the weight bound stated in the lemma.

Finally, we note that the constructed approximations indeed correspond to a shallow tanh neural network of the stated size. Indeed, one can see from the fact that σ is odd, equation (15), Lemma 3.1 and equation (27) that a shallow tanh neural network suffices, where the values of the $3(s+1)/2$ neurons in the hidden layer are given by

$$\sigma\left(\left(\frac{s}{2}-i\right)h(y+\beta)\right) \qquad \text{where } i = 0, 1, \dots \frac{s-1}{2} \text{ and } \beta \in \{-\alpha, 0, \alpha\}. \tag{37}$$

$\square$

Remark 3.3. *Combined with the Weierstrass approximation theorem [59], the preceding results show that any continuous function can be uniformly approximated in supremum norm on a compact interval by shallow tanh neural networks to arbitrary accuracy, as was already observed by e.g. [36]. Using the constructive proof of the Weierstrass approximation theorem based on Bernstein polynomials, one can even obtain a rate of convergence in terms of the width of the neural network and the modulus of continuity of the continuous function [60].*

Remark 3.4. *Note that one can also construct monomials with even powers directly, as was done in e.g. [34]. Indeed, there exists an $x \in \mathbb{R}$ such that $\tanh^{(p)}(x) \neq 0$ for all $p \in \mathbb{N}$, allowing us to use a neural network as in (16) for even p as well. However, a key difficulty lies in explicitly finding a function $\gamma : \mathbb{N} \to (0, \infty)$ such that $\left|\tanh^{(p)}(x)\right| \geq \gamma(p)$ for all $p \in \mathbb{N}$. It is unclear if such a function, which is quite essential when proving uniform bounds as in Lemma 3.2, can be constructed directly. Instead, our construction circumvents this issue and can be readily extended to other activation functions.*

3.2. Approximating multivariate polynomials

Next, we consider the approximation of multivariate polynomials using tanh neural networks. As an application, we will also present two different approximations of the multiplication operator.

First, recall the set $P_{n,q} = \{\alpha \in \mathbb{N}_0^q : |\alpha| = n\}$ from Section 2.1. The multinomial theorem implies that for $\alpha \in P_{n,q}$ and $\omega \in \mathbb{R}^q$, it holds that

$$\sum_{\beta \in P_{n,q}} \binom{n}{\beta} \frac{\alpha^\beta}{n^n} \omega^\beta = \left(\sum_{i=1}^{q} \frac{\alpha_i}{n} \omega_i\right)^n. \tag{38}$$

Now let $x \in \mathbb{R}^d$, set $q = d$ and $\omega = x$. Then the set $\{\omega^\beta : \beta \in P_{n,q}\}$ corresponds to the set of all d-variate monomials of total degree *equal to n*. Similarly, one can set $q = d+1$ and $\omega = (1, x)$, such that $\{\omega^\beta : \beta \in P_{n,q}\}$ corresponds to the set of all d-variate monomials of total degree *at most* n. It is the goal of this section to approximate these monomials ω^β using tanh neural networks. Notice however that the results from the previous section already allow us to approximate the right hand side of (38), as it is merely a composition of a linear map and a univariate monomial. Writing $b_\alpha = \left(\sum_i \alpha_i \omega_i / n\right)^n$ one can interpret (38) for every $\alpha \in P_{n,q}$ as the linear equation $\sum_\beta D_{\alpha,\beta}\omega^\beta = b_\alpha$, where

$$D_{\alpha,\beta} = \binom{n}{\beta}\frac{\alpha^\beta}{n^n}, \tag{39}$$

which leads us to a linear system $\{\sum_\beta D_{\alpha,\beta}\omega^\beta = b_\alpha : \alpha \in P_{n,q}\}$ with as unknowns the monomials ω^β. Since the *Dyson matrix* $D = (D_{\alpha,\beta})_{\alpha,\beta\in P_{n,q}}$, where the order of rows and columns reflects the lexicographic order on $P_{n,q}$, is invertible [57], it is possible to write every monomial as a linear combination of the b_α's. We will exploit this fact to construct approximations of multivariate polynomials and the multiplication $\prod_{i=1}^d x_i$ in particular.

Lemma 3.5. *Let $q, n \in \mathbb{N}$, $k \in \mathbb{N}_0$ and $M > 0$. Then for every $\epsilon > 0$, there exists a shallow tanh neural network $\Psi_{n,q} : [-M, M]^q \to \mathbb{R}^{|P_{n,q}|}$ of width $3\left\lceil\frac{n+1}{2}\right\rceil |P_{n,q}|$ such that*

$$\max_{\beta\in P_{n,q}} \left\|\omega^\beta - (\Psi_{n,q}(\omega))_{\iota(\beta)}\right\|_{W^{k,\infty}} \leq \epsilon, \tag{40}$$

where $\iota : P_{n,q} \to \{1, \dots |P_{n,q}|\}$ is a bijection. Furthermore, the weights of the network scale as $O\left(\epsilon^{-n/2}(n(n+2))^{3(n+2)^2}\right)$ for small ϵ and large n.

Proof. From the previous section, we can see that approximating $b_\alpha = \left(\sum_i \alpha_i\omega_i/n\right)^n$ requires a shallow tanh subnetwork $\widehat{b_\alpha}$ of width $3\left\lceil\frac{n+1}{2}\right\rceil$. As we require $|P_{n,q}|$ such subnetworks, the total network width can be summarized as $3\left\lceil\frac{n+1}{2}\right\rceil |P_{n,q}|$. Now denote by $\widehat{\omega^\beta}$ the neural network approximation one obtains by solving the linear system $\{\sum_\beta D_{\alpha,\beta}\widehat{\omega^\beta} = \widehat{b_\alpha} : \alpha \in P_{n,q}\}$. We then set $(\Psi_{n,q}(\omega))_{\iota(\beta)} := (\widehat{\omega^\beta})_\beta$, where $\iota : P_{n,q} \to \{1, \dots |P_{n,q}|\}$ is a bijection. Then it holds that

$$\left\|(\widehat{\omega^\beta})_\beta - (\omega^\beta)_\beta\right\|_{W^{k,\infty}} \leq \left\|D^{-1}\right\|_\infty \left\|(\widehat{b_\alpha})_\alpha - (b_\alpha)_\alpha\right\|_{W^{k,\infty}}. \tag{41}$$

Now define $h_\alpha(\omega) = \sum_i \alpha_i\omega_i/n$, then it holds that $\widehat{b_\alpha} - b_\alpha = ((\psi_{s,\epsilon})_n - f_n)\circ h_\alpha$, where $\psi_{s,\epsilon}$ is as in Lemma 3.2 and $s = 2\left\lfloor\frac{n}{2}\right\rfloor + 1$. It is easy to check that $\|h_\alpha\|_{W^{k,\infty}} \leq \max\{1, M\}$. Invoking Lemma Appendix A.7 then gives us

$$\left\|\widehat{b_\alpha} - b_\alpha\right\|_{W^{k,\infty}} \leq 16(e^2k^4q^2)^k \left\|(\psi_{s,\epsilon})_n - f_n\right\|_{W^{k,\infty}} \max\{1, M\}^k. \tag{42}$$

In addition, Lemma Appendix A.3 provides us with the bound

$$\left\|D^{-1}\right\|_\infty \leq (n!)^3 |P_{n,q}|^2 2^n \leq \pi e^3 q^{2n} n^{3(n+1/2)}, \tag{43}$$

where we used Stirling's approximation and Lemma 2.1. Now let $\epsilon > 0$. Combining the two obtained inequalities with Lemma 3.2 then proves that

$$\left\| \widehat{(\omega^{\beta})}_{\beta} - (\omega^{\beta})_{\beta} \right\|_{W^{k,\infty}} \leq \left\| D^{-1} \right\|_{\infty} \cdot 16(e^2k^4q^2)^k \max\{1, M\}^k \cdot \epsilon \tag{44}$$

where the weights of $(\psi_{s,\epsilon})_n$ scale as $O\left(\epsilon^{-n/2}(\sqrt{M}(n+2))^{3n(n+3)/2}\right)$ for small ϵ and large n. We can now rescale ϵ such that

$$\left\| \widehat{(\omega^{\beta})}_{\beta} - (\omega^{\beta})_{\beta} \right\|_{W^{k,\infty}} \leq \epsilon. \tag{45}$$

As a consequence, the weights of $\Psi_{n,q}$ will scale as

$$O\left(\epsilon^{-n/2} \left\| D^{-1} \right\|_{\infty}^{n/2+1} \left(4(ek^2q)^k\sqrt{1+M}\right)^n (\sqrt{M}(n+2))^{3n(n+2)/2}\right). \tag{46}$$

Note that $O\left(\left\|D^{-1}\right\|_{\infty}^{n/2+1}\right) = O\left((\pi e^3 qn)^{3(n+2)^2/2}\right)$ and that therefore a (conservative) upper bound of the weights of $\Psi_{n,q}$ is given by

$$O\left(\epsilon^{-n/2}(n(n+2))^{3(n+2)^2}\right) \tag{47}$$

for small ϵ and large n. □

Corollary 3.6 (Approximation of multivariate monomials)**.** *Let $d, s \in \mathbb{N}$, $k \in \mathbb{N}_0$ and $M > 0$. Then for every $\epsilon > 0$, there exists a shallow tanh neural network $\Phi_{s,d} : [-M, M]^d \to \mathbb{R}^{\left|P_{s,d+1}\right|}$ of width $3\left\lceil \frac{s+1}{2} \right\rceil \left|P_{s,d+1}\right|$ such that*

$$\max_{\beta \in P_{s,d+1}} \left\| x^{\beta} - (\Phi_{s,d}(x))_{\iota(\beta)} \right\|_{W^{k,\infty}} \leq \epsilon, \tag{48}$$

where $\iota : P_{s,d+1} \to \{1, \dots \left|P_{s,d+1}\right|\}$ is a bijection. Furthermore, the weights of the network scale as $O\left(\epsilon^{-s/2}(s(s+2))^{3(s+2)^2}\right)$ for small ϵ and large s.

Proof. The statement follows directly from Lemma 3.5 with $n \leftarrow s$, $q \leftarrow d+1$ and $\omega \leftarrow (1, x)$, where $x \in [-M, M]^d$. □

Next, we discuss how the multiplication operator can be approximated. To begin with, Lemma 3.5 shows that the multiplication of d numbers can easily be approximated using a shallow tanh neural network.

Corollary 3.7 (Shallow approximation of multiplication of d numbers)**.** *Let $d \in \mathbb{N}$, $k \in \mathbb{N}_0$ and $M > 0$. Then for every $\epsilon > 0$, there exists a shallow tanh neural network $\widehat{\times}^{\epsilon}_{d} : [-M, M]^d \to \mathbb{R}$ of width $3\left\lceil \frac{d+1}{2} \right\rceil \left|P_{d,d}\right|$ such that*

$$\left\| \widehat{\times}^{\epsilon}_{d}(x) - \prod_{i=1}^{d} x_i \right\|_{W^{k,\infty}} \leq \epsilon. \tag{49}$$

Furthermore, the weights of the network scale as $O(\epsilon^{-d/2})$ for small ϵ.

Proof. The statement follows directly from Lemma 3.5 with $n \leftarrow d$, $q \leftarrow d$ and $\omega \leftarrow x$, where $x \in [-M, M]^d$. □

One issue with this shallow approximation is that the width of the network grows quickly with the dimension. The next lemma shows that the same accuracy can also be obtained using a deep tanh neural network for which both width and depth scale at most linearly with the input dimension.

Lemma 3.8 (Deep approximation of multiplication of d numbers)**.** *Let $d \in \mathbb{N}$, $k \in \mathbb{N}_0$ and $M > 0$. Then for every $\epsilon > 0$, there exists a tanh neural network $\widehat{\times}^{\epsilon}_{d} : [-M, M]^d \to \mathbb{R}$ with $\lceil \log_2(d) \rceil$ hidden layers and of width at most $3d$ such that*

$$\left\| \widehat{\times}^{\epsilon}_{d}(x) - \prod_{i=1}^{d} x_i \right\|_{W^{k,\infty}} \leq \epsilon. \tag{50}$$

Furthermore, the weights of the network scale as $O(\epsilon^{-1/2})$ for small ϵ.

Proof. Using the finite difference approach (15), we can approximate the quadratic function using $\delta_h^2[f](x_0)$ for some $h > 0$ and $x_0 \in [-1, 1]$ such that $\sigma^{(2)}(x_0) \neq 0$. Observing that

$$xy = \frac{1}{4}\left((x+y)^2 - (x-y)^2\right) \tag{51}$$

then provides a recipe to approximate (in Sobolev norm) the multiplication of two numbers using a shallow tanh neural network with 6 neurons in its hidden layer. The proof is similar to that of Lemma 3.1. Moreover, Lemma 3.1 shows as well that the identity can be approximated using a shallow tanh neural network with only one neuron in its hidden layer.

The multiplication of d numbers then follows easily from the multiplication of 2 numbers. In e.g. [4, Proposition 2.36], it is proven that the multiplication of d numbers requires a neural network in the form of a binary tree of depth $\lceil \log_2(d) \rceil$ where each node computes the (approximate) multiplication of two numbers. The proof of our error bound follows from Lemma Appendix A.6 and Appendix A.7. □

Remark 3.9. *For simplicity and motivated by its widespread use, we only focused on the hyperbolic tangent activation function here. Our approach can be generalized to any activation function ϕ for which there exist $\mathcal{P} \subseteq \mathbb{N}$ with $\sup \mathcal{P} = \infty$ and an explicitly known function $\gamma : \mathcal{P} \to (0, \infty)$ with $\left|\phi^{(p)}\right| \geq \gamma(p)$ for all $p \in \mathcal{P}$. Monomials with degree $p \in \mathcal{P}$ can be constructed as in* (16)*, the construction of monomials with degree $p \in \mathbb{N} \setminus \mathcal{P}$ is similar to the one described for multivariate polynomials.*

4. Approximation of partition of unity

Once we have approximated polynomials with shallow tanh neural networks, the next step in our construction is to approximate a suitable partition of unity.

In this section, we show how one can mimic a partition of unity using tanh neural networks. We recall that a partition of unity is a set of functions $f_i : [0,1]^d \to [0,1]$ such that every f_i is non-zero on only a small part of $[0,1]^d$ and such that $\sum_i f_i = 1$. For ReLU and RePU neural networks, such partitions of unity can be constructed exactly [20]. For tanh neural networks, we will prove that an approximate partition of unity can be constructed. A unifying framework for approximating partitions of unity by general neural networks has been proposed in [34].

Let $d, N \in \mathbb{N}$ and $k \in \mathbb{N}_0$. For every $j \in \mathbb{N}^d$ with $\|j\|_\infty \leq N$ we define x_j^N such that $(x_j^N)_i = j_i/N_i$. We also define

$$I_j^N = \bigtimes_{i=1}^{d} \big((j_i - 1)/N, j_i/N\big) . \tag{52}$$

Let $R > 0$ be such that $|\sigma^{(m)}|$ is decreasing on $[R, \infty)$ for every $1 \leq m \leq k$. Given $\epsilon > 0$, we first find an $\alpha = \alpha(N, \epsilon)$ large enough such that

$$\alpha/N \geq R, \quad 1 - \sigma(\alpha/N) \leq \epsilon, \quad \alpha^m \left|\sigma^{(m)}(\alpha/N)\right| \leq \epsilon \text{ for all } 1 \leq m \leq k. \tag{53}$$

This is possible because $\lim_{x\to\infty} \sigma(x) = 1$ and because of Lemma Appendix A.4. In particular, Lemma Appendix A.5 shows that a suitable choice of α is given by

$$\alpha = N \max\left\{R, \ln\left(\frac{(2k)^{k+1}(Nk)^k}{e^k \epsilon}\right)\right\}. \tag{54}$$

For $y \in \mathbb{R}$, we then define

$$\rho_1^N(y) = \frac{1}{2} - \frac{1}{2}\sigma\left(\alpha\left(y - \frac{1}{N}\right)\right), \tag{55}$$

$$\rho_j^N(y) = \frac{1}{2}\sigma\left(\alpha\left(y - \frac{j-1}{N}\right)\right) - \frac{1}{2}\sigma\left(\alpha\left(y - \frac{j}{N}\right)\right) \quad \text{for } 2 \leq j \leq N-1, \tag{56}$$

$$\rho_N^N(y) = \frac{1}{2}\sigma\left(\alpha\left(y - \frac{N-1}{N}\right)\right) + \frac{1}{2}. \tag{57}$$

In the remainder of the paper, we will assume for simplicity that ρ_j^N is always of the second form. The calculations involving ρ_1^N and ρ_N^N can be done entirely similarly and do not change the stated results. Finally, we define for $D \leq d$ the functions

$$\Phi_j^{N,D}(x) = \prod_{i=1}^{D} \rho_{j_i}^{N_i}(x_i) \tag{58}$$

and the sets $\mathcal{V}_D = \{v \in \mathbb{Z}^d : \max_{1\leq i\leq D} |v_i| \leq 1 \text{ and } v_{D+1} = \cdots = v_d = 0\}$. We will prove that the functions $\Phi_j^{N,d}$ approximate a partition of unity in the sense

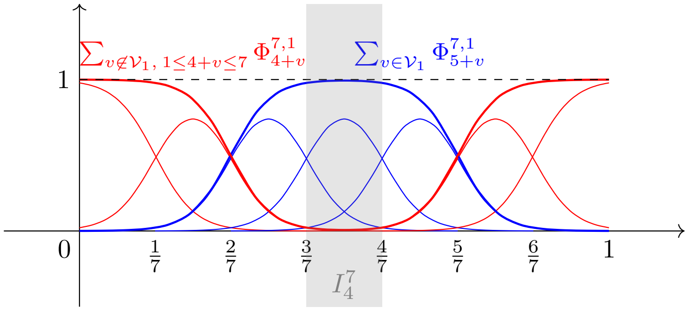

Figure 1: Example of an approximate partition of unity on $[0,1]$ with $N = 7$. The thin lines represent the $\Phi^{7,1}_j = \rho^7_j$, $1 \leq j \leq 7$.

that for every j it holds on I^N_j that,

$$\sum_{v \in \mathcal{V}_d} \Phi^{N,d}_{j+v} \approx 1 \quad \text{and} \quad \sum_{\substack{v \notin \mathcal{V}_d, \\ j+v \in \{1,\ldots,N\}^d}} \Phi^{N,d}_{j+v} \approx 0. \tag{59}$$

An example for $d = 1$ and $N = 7$ is shown in Figure 1. The next two lemmas formalize this approximation. Finally, a tanh neural network approximation of 280 $\Phi^{N,d}_j$ can be constructed by replacing the multiplication operator by the network from e.g. Corollary 3.7 or Lemma 3.8.

Lemma 4.1. *If* $0 < \epsilon < 1/4$, *then*

$$\left\| \sum_{v \in \mathcal{V}_d} \Phi^{N,d}_{j+v} - 1 \right\|_{W^{k,\infty}(I^N_j)} \leq 2^{dk} d\epsilon. \tag{60}$$

Proof. We will prove the statement holds by induction on d. We first note that, for $d = 1$, we have

$$\begin{aligned} \sum_{v \in \mathcal{V}_1} \Phi^{N,1}_{j+v}(x) &= \sum_{l=-1}^{1} \rho^N_{j_1+l}(x_1) \\ &= \frac{1}{2}\sigma\left(\alpha\left(x_1 - \frac{j_1 - 2}{N}\right)\right) - \frac{1}{2}\sigma\left(\alpha\left(x_1 - \frac{j_1+1}{N}\right)\right), \end{aligned} \tag{61}$$

from which easily follows that

$$\sum_{v \in \mathcal{V}_1} \Phi^{N,1}_{j+v}(x) \leq 1. \tag{62}$$

Next, note that for $x \in I_j^N$

$$\begin{aligned}\sum_{v \in \mathcal{V}_1} \Phi_{j+v}^{N,1}(x) &= \frac{1}{2}\sigma\left(\alpha\left(x_1 - \frac{j_1 - 2}{N}\right)\right) - \frac{1}{2}\sigma\left(\alpha\left(x_1 - \frac{j_1 + 1}{N}\right)\right) \\ &\geq \sigma\left(\frac{\alpha}{N}\right) \geq 1 - \epsilon,\end{aligned} \tag{63}$$

where we used the definition of α on the last line. Furthermore, for $1 \leq m \leq k$, we get that

$$\left|\frac{d^m}{dx^m} \sum_{v \in \mathcal{V}_1} \Phi_{j+v}^{N,1}(x)\right| \leq \alpha^m \sigma^{(m)}\left(\frac{\alpha}{N}\right) \leq \epsilon, \tag{64}$$

where we used (61) and the monotonic decay of $\sigma^{(m)}(x)$ for $x \in [\alpha/N, \infty)$ and our choice of α (cf. equation (53)). This allows us to conclude that

$$\left\|\sum_{v \in \mathcal{V}_1} \Phi_{j+v}^{N,1}(x) - 1\right\|_{W^{k,\infty}(I_j^N)} \leq \epsilon. \tag{65}$$

For the induction step, we assume that for some $2 \leq D \leq d$ it holds that

$$\left\|\sum_{v \in \mathcal{V}_{D-1}} \Phi_{j+v}^{N,D-1} - 1\right\|_{W^{k,\infty}(I_j^N)} \leq 2^{(D-1)k}(D-1)\epsilon. \tag{66}$$

Using Lemma Appendix A.6, we find that for $x \in I_j^N$,

$$\begin{aligned}\left\|\sum_{v \in \mathcal{V}_D} \Phi_{j+v}^{N,D}(x) - 1\right\|_{W^{k,\infty}(I_j^N)} &= \left\|\sum_{w \in \mathcal{V}_1} \rho_{j_D+w}^N(x_D) \sum_{v \in \mathcal{V}_{D-1}} \Phi_{j+v}^{N,D-1}(x) - 1\right\|_{W^{k,\infty}(I_j^N)} \\ &\leq \left\|\sum_{w \in \mathcal{V}_1} \rho_{j_D+w}^N(x_D) - 1\right\|_{W^{k,\infty}(I_j^N)} \\ &\quad + 2^k \left\|\sum_{w \in \mathcal{V}_1} \rho_{j_D+w}^N(x_D)\right\|_{W^{k,\infty}(I_j^N)} \left\|\sum_{v \in \mathcal{V}_{D-1}} \Phi_{j+v}^{N,D-1}(x) - 1\right\|_{W^{k,\infty}(I_j^N)} \\ &\leq \epsilon + 2^k 2^{(D-1)k}(D-1)\epsilon \leq 2^{Dk} D\epsilon.\end{aligned} \tag{67}$$

This concludes the proof. □

Lemma 4.2. *Let $k \in \mathbb{N}_0$ and $v \in \mathbb{Z}^d$ with $\|v\|_\infty \geq 2$. Then it holds that*

$$\left\|\Phi_{j+v}^{N,d}\right\|_{W^{k,\infty}(I_j^N)} \leq \max\{1, (2k)^{2k}\alpha^k\}\epsilon. \tag{68}$$

Proof. Let $x \in I_j^N$ and let $1 \leq i \leq d$ be an index such that $|v_i| \geq 2$. Using some basic equalities for the hyperbolic tangent function and the definition of α, we obtain that

$$\begin{aligned}\left|\rho^N_{j_i+v_i}(x_i)\right| &\leq \frac{1}{2}\sigma\left(\frac{2\alpha}{N}\right) - \frac{1}{2}\sigma\left(\frac{\alpha}{N}\right) \\ &= \frac{1}{2}\sigma\left(\frac{\alpha}{N}\right)\left(1 - \sigma\left(\frac{2\alpha}{N}\right)\sigma\left(\frac{\alpha}{N}\right)\right) \\ &\leq \frac{1}{2}\left(1 - \sigma^2\left(\frac{\alpha}{N}\right)\right) \leq \epsilon.\end{aligned} \tag{69}$$

In addition, for every $1 \leq \ell \leq d$, it holds that $\left|\rho^N_{j_\ell+v_\ell}(x_\ell)\right| \leq 1$. This implies that

$$\left\|\Phi^{N,d}_{j+v}\right\|_{L^\infty(I^N_j)} \leq \epsilon. \tag{70}$$

Let $1 \leq m \leq k$, then it holds that (by our choice of the index i),

$$\left|\frac{d^m}{dx_i^m}\rho^N_{j_i+v_i}(x_i)\right| \leq \alpha^m \left|\sigma^{(m)}\left(\frac{\alpha}{N}\right)\right| \leq \epsilon. \tag{71}$$

Now let $\beta \in \mathbb{N}^d$ such that $1 \leq |\beta| \leq k$. Then

$$\left|D^\beta \Phi^{N,d}_{j+v}(x)\right| = \left|\prod_{\ell=1}^{d} \frac{d^{\beta_\ell}}{dx_\ell^{\beta_\ell}} \rho^{N_\ell}_{j_\ell+v_\ell}(x_\ell)\right| \leq \epsilon \prod_{\ell=1,\ell\neq i,\beta_\ell\neq 0}^{d} (2\beta_\ell)^{\beta_\ell+1}\alpha^{\beta_\ell} \leq \epsilon(2k)^{2k}\alpha^k, \tag{72}$$

where we used the fact that $\left|\sigma^{(m)}(x)\right| \leq (2m)^{m+1}$ in the first inequality (cf. Lemma Appendix A.4). Combining (70) and (72) proves the statement. □

5. Main results

5.1. Approximation of functions in Sobolev spaces

We now present the first main result of the paper. It follows from the lemma of Bramble–Hilbert (Lemma Appendix A.8) that localized Taylor polynomials can approximate a function $f \in W^{s,\infty}([0,1]^d)$. For functions $f \in C^s([0,1]^d)$, this approximation follows from Taylor's theorem (Lemma Appendix A.9). We then use the results from the previous two sections to construct tanh neural networks that approximate localized Taylor polynomials in Sobolev norm. We prove that the function f can be approximated by a tanh neural network with two hidden layers and we provide explicit bounds on the width and approximation error.

Theorem 5.1. *Let $d, s \in \mathbb{N}$, $R > 0$ as in (53), $\delta > 0$ and $f \in W^{s,\infty}([0,1]^d)$. There exist constants $\mathcal{C}(d,k,s,f)$, $N_0(d) > 0$, such that for every $N \in \mathbb{N}$ with $N > N_0(d)$ there exists a tanh neural network $\widehat{f}^N$ with two hidden layers, one of width at most $3\left\lceil \frac{s}{2} \right\rceil \left|P_{s-1,d+1}\right| + d(N-1)$ and another of width at most $3\left\lceil \frac{d+2}{2} \right\rceil \left|P_{d+1,d+1}\right| N^d$ (or $3\left\lceil \frac{s}{2} \right\rceil + N - 1$ and $6N$ for $d = 1$), such that,*

$$\left\| f - \widehat{f}^N \right\|_{L^\infty([0,1]^d)} \leq (1+\delta) \frac{\mathcal{C}(d,0,s,f)}{N^s}, \tag{73}$$

and for $k = 1, \dots, s-1$,

$$\left\| f - \widehat{f}^N \right\|_{W^{k,\infty}([0,1]^d)} \leq 3^d \, (1+\delta) \, (2(k+1))^{3k} \max\left\{ R^k, \ln^k\left(\beta N^{s+d+2}\right) \right\} \frac{\mathcal{C}(d,k,s,f)}{N^{s-k}}, \tag{74}$$

where we define

$$\beta = \frac{k^3 2^d \sqrt{d} \max\{1, \|f\|^{1/2}_{W^{k,\infty}([0,1]^d)}\}}{\delta \min\{1, \sqrt{\mathcal{C}(d,k,s,f)}\}}. \tag{75}$$

If $f \in C^s([0,1]^d)$, then it holds that

$$\mathcal{C}(d,k,s,f) = \max_{0 \leq \ell \leq k} \frac{1}{(s-\ell)!} \left(\frac{3d}{2} \right)^{s-\ell} |f|_{W^{s,\infty}([0,1]^d)}, \quad N_0(d) = \frac{3d}{2}, \tag{76}$$

and else it holds that

$$\mathcal{C}(d,k,s,f) = \max_{0 \leq \ell \leq k} \frac{\pi^{1/4}\sqrt{s}}{(s-\ell-1)!} \left(5d^2 \right)^{s-\ell} |f|_{W^{s,\infty}([0,1]^d)}, \quad N_0(d) = 5d^2. \tag{77}$$

In addition, the weights of $\widehat{f}^N$ scale as $O\left(\mathcal{C}^{-s/2} N^{d(d+s^2+k^2)/2} (s(s+2))^{3s(s+2)}\right)$.

Proof. We will prove the theorem in the following manner. We divide the unit cube into N^d cubes of edge length $1/N$. On each of these cubes, f can be approximated in Sobolev norm by a polynomial. The global approximation can then be constructed by multiplying each polynomial with the indicator function of the corresponding cubes and summing over all cubes. We then prove that replacing these polynomials, multiplications and indicator functions with the tanh neural networks from the previous sections results in a new approximation that has approximately the same accuracy. In the last step we will calculate the size of the required neural network.

Step 1: construction of the approximation. Let us denote $J_j^N = \mathop{\times}_{i=1}^d \left((j_i-2)/N, (j_i+1)/N\right)$. We calculate that $\operatorname{diam}(J_j^N) = \frac{3\sqrt{d}}{N}$ and that there exists a ball with diameter $\frac{1}{\sqrt{d}}\operatorname{diam}(J_j^N)$ such that J_j^N is star-shaped with respect to every point in this ball. As a consequence, the Bramble-Hilbert lemma (Lemma Appendix A.8) ensures the existence of a polynomial p_j^N of degree at most $s-1$ such that

$$\left\|f-p_j^N\right\|_{W^{\ell,\infty}(J_j^N)} \leq \frac{\pi^{1/4}\sqrt{s}}{(s-\ell-1)!}\left(\frac{5d^2}{N}\right)^{s-\ell} |f|_{W^{s,\infty}([0,1]^d)} \\ \leq \max_{0\leq m\leq \ell} \frac{\pi^{1/4}\sqrt{s}(5d^2)^{s-m}}{(s-m-1)!}\frac{|f|_{W^{s,\infty}([0,1]^d)}}{N^{s-\ell}} =: \frac{\mathcal{C}(d,\ell,s,f)}{N^{s-\ell}}, \tag{78}$$

for all $0\leq \ell\leq k$, under the assumption that $N>5d^2$, and where we used that $3\sqrt{e}\leq 5$. If moreover $f\in C^s([0,1]^d)$, then Taylor's theorem (Lemma Appendix A.9 with $\delta=\frac{3}{2N}$) ensures the existence of a polynomial p_j^N of degree at most $s-1$ such that

$$\left\|f-p_j^N\right\|_{W^{\ell,\infty}(J_j^N)} \leq \frac{1}{(s-\ell)!}\left(\frac{3d}{2N}\right)^{s-\ell} |f|_{W^{s,\infty}([0,1]^d)} \\ \leq \max_{0\leq m\leq \ell} \frac{1}{(s-m)!}\left(\frac{3d}{2}\right)^{s-m}\frac{|f|_{W^{s,\infty}([0,1]^d)}}{N^{s-\ell}} =: \frac{\mathcal{C}(d,\ell,s,f)}{N^{s-\ell}}, \tag{79}$$

for all $0\leq \ell\leq k$, under the assumption that $N>3d/2$. The remainder of the argument will be independent of which polynomial p_j^N and which definition of $\mathcal{C}(d,\ell,s,f)$ is used. To simplify notation, we also define $p^N=\sum_j p_j^N\chi_j$, where χ_j denotes the indicator function on I_j^N. Next, let q_j^N be a tanh neural network as in Section 1 such that

$$\left\|q_j^N-p_j^N\right\|_{W^{k,\infty}([0,1]^d)} \leq \eta. \tag{80}$$

In addition, we define

$$q_j^N(x)\widehat{\times}\Phi_j^{N,d}(x) := \widehat{\times}_{d+1}^h(q_j^N(x),\phi_{j_1}^{N,d}(x_1),\dots,\phi_{j_d}^{N,d}(x_d)), \tag{81}$$

where $\widehat{\times}_{d+1}^h$ is the network from Corollary 3.7 and $h=h(N)$ will be defined in the remainder of the proof. We then define our approximation as

$$\widehat{f}^N(x) = \sum_{j\in\{1,\dots,N\}^d} q_j^N(x)\widehat{\times}\Phi_j^{N,d}(x). \tag{82}$$

Step 2: estimating the error of the approximation. The triangle inequality gives us

$$\left\|f-\widehat{f}^N\right\|_{W^{k,\infty}([0,1]^d)} \leq \left\|f-\sum_{j\in\{1,\dots,N\}^d} f\cdot\Phi_j^{N,d}\right\|_{W^{k,\infty}([0,1]^d)} + \left\|\sum_{j\in\{1,\dots,N\}^d}(f-q_j^N)\cdot\Phi_j^{N,d}\right\|_{W^{k,\infty}([0,1]^d)} \\ + \left\|\sum_{j\in\{1,\dots,N\}^d}(q_j^N\cdot\Phi_j^{N,d}-q_j^N\widehat{\times}\Phi_j^{N,d})\right\|_{W^{k,\infty}([0,1]^d)} \tag{83}$$

We proceed by bounding each term of the right hand side separately.

Step 2a: First term of (83). Let $i \in \{0, \dots, N\}^d$ be arbitrary. Recalling that $\mathcal{V}_d = \{v \in \mathbb{Z}^d : \|v\|_\infty \leq 1\}$, we observe for $k \geq 1$,

$$\begin{aligned}
\left\| f - \sum_{j \in \{1,\dots,N\}^d} f \cdot \Phi_j^{N,d} \right\|_{W^{k,\infty}(I_i^N)} &\leq 2^k \|f\|_{W^{k,\infty}(I_i^N)} \left\| 1 - \sum_{v \in \mathcal{V}_d} \Phi_{i+v}^{N,d} \right\|_{W^{k,\infty}(I_i^N)} \\
&\quad + 2^k \|f\|_{W^{k,\infty}(I_i^N)} \left\| \sum_{\substack{j \in \{1,\dots,N\}^d \\ j-i \notin \mathcal{V}_d}} \Phi_j^{N,d} \right\|_{W^{k,\infty}(I_i^N)} \\
&\leq 2^k \|f\|_{W^{k,\infty}(I_i^N)} (2^{kd} d\epsilon + N^d (2k)^{2k} \alpha^k \epsilon) \\
&\leq 2^k \|f\|_{W^{k,\infty}(I_i^N)} 2^{kd} d\epsilon \\
&\quad + 2^k \|f\|_{W^{k,\infty}(I_i^N)} N^d (2k)^{2k} N^k (k+1)^k \max\left\{ R^k, \ln^k\left(\frac{2Nk^2}{\epsilon^{\frac{1}{k+1}} e} \right) \right\} \epsilon \\
&\leq \delta (2(k+1))^{3k} \max\left\{ R^k, \ln^k\left(\frac{2Nk^2}{\epsilon^{\frac{1}{k+1}} e} \right) \right\} \frac{\mathcal{C}(d,k,s,f)}{N^{s-k}},
\end{aligned} \tag{84}$$

where we used Lemma Appendix A.6, Lemma 4.1, Lemma 4.2 and Lemma Appendix A.5, as well as a suitable definition of ϵ, satisfying

$$\epsilon \leq \frac{\delta \mathcal{C}(d,k,s,f)}{2^{(k+1)d} d N^{s+d} \|f\|_{W^{k,\infty}([0,1]^d)}}. \tag{85}$$

Analogously, for $k = 0$, one can obtain that

$$\left\| f - \sum_{j \in \{1,\dots,N\}^d} f \cdot \Phi_j^{N,d} \right\|_{L^\infty(I_i^N)} \leq \|f\|_{L^\infty(I_i^N)} (d\epsilon + N^d \epsilon) \leq \frac{\delta}{3} \frac{\mathcal{C}(d,k,s,f)}{N^{s-k}}. \tag{86}$$

Step 2b: Second term of (83) *for* $k = 0$. In order to bound the second term, we first make some auxiliary calculations. To begin with, we consider the case where $k = 0$. We find that

$$\begin{aligned}
\left| \sum_{v \in \mathcal{V}_d} (f - q_{i+v}^N) \Phi_{i+v}^{N,d} \right| &\leq \max_{v \in \mathcal{V}_d} |f - q_{i+v}^N| \left(\sum_{v \in \mathcal{V}_d} |\Phi_{i+v}^{N,d}| \right) \\
&= \max_{v \in \mathcal{V}_d} |f - q_{i+v}^N| \left| \sum_{v \in \mathcal{V}_d} \Phi_{i+v}^{N,d} \right|
\end{aligned} \tag{87}$$

where all functions are evaluated at some $x \in I_i^N$. We can then use the bounds

$$\left| f - q_{i+v}^N \right| \leq \frac{\mathcal{C}(d,0,s,f)}{N^s} + \eta, \tag{88}$$

which follows from (78) and (80), and,

$$\left| \sum_{v \in \mathcal{V}_d} \Phi_{i+v}^{N,d} \right| \leq 1 + d\epsilon, \tag{89}$$

which follows from Lemma 4.1. As a consequence, we find that

$$\left\| \sum_{v \in \mathcal{V}_d} (f - q_{i+v}^N) \Phi_{i+v}^{N,d} \right\|_{L^\infty(I_i^N)} \leq \left(\frac{\mathcal{C}(d,k,s,f)}{N^{s-k}} + \eta \right) (1 + d\epsilon). \tag{90}$$

Combining this result with the triangle inequality, (78), (80) and Lemma 4.2, we find that

$$\begin{aligned}
&\left\| \sum_{j \in \{1,\ldots,N\}^d} (f - q_j^N) \cdot \Phi_j^{N,d} \right\|_{L^\infty(I_i^N)} \\
&\leq \left\| \sum_{v \in \mathcal{V}_d} (f - q_{i+v}^N) \Phi_{i+v}^{N,d} \right\|_{L^\infty(I_i^N)} + \sum_{\substack{j \in \{1,\ldots,N\}^d \\ j-i \notin \mathcal{V}_d}} \left\| (f - q_j^N) \right\|_{L^\infty(I_i^N)} \left\| \Phi_j^{N,d} \right\|_{L^\infty(I_i^N)} \\
&\leq \left(\frac{\mathcal{C}(d,k,s,f)}{N^{s-k}} + \eta \right) (1 + d\epsilon) + N^d \left(\mathcal{C}(d,k,s,f) + \eta \right) \epsilon \\
&\leq \left(1 + \frac{\delta}{3} \right) \frac{\mathcal{C}(d,k,s,f)}{N^{s-k}}.
\end{aligned} \tag{91}$$

where we obtain the last inequality by making a suitable choice of η and ϵ.

Step 2c: Second term of (83) *for* $k \geq 1$. Next we consider the case where $0 < k < s$. Let $\beta \in \mathbb{N}_0^d$ be such that $|\beta| \leq k$. Then as a consequence of the general Leibniz rule we find that

$$D^\beta \left(\sum_{v \in \mathcal{V}_d} (f - q_{i+v}^N) \Phi_{i+v}^{N,d} \right) \leq \sum_{\beta' \leq \beta} \binom{\beta}{\beta'} \sum_{v \in \mathcal{V}_d} \left| D^{\beta'} (f - q_{i+v}^N) \right| \left| D^{\beta - \beta'} \Phi_{i+v}^{N,d} \right| \tag{92}$$

where all functions are evaluated at some $x \in I_i^N$. For every $v \in \mathcal{V}_d$ and $\beta' \leq \beta$ with $\ell := \left| \beta - \beta' \right|$, we can then use the bounds

$$\left| D^{\beta'} (f - q_{i+v}^N) \right| \leq \left\| f - q_{i+v}^N \right\|_{W^{k-\ell,\infty}(I_i^N)} \leq \frac{\mathcal{C}(d,k-\ell,s,f)}{N^{s-k+\ell}} + \eta, \tag{93}$$

which follows from (78) and (80), and,

$$\left|D^{\beta-\beta'}\Phi^{N,d}_{i+v}\right| \leq \alpha^{\ell}(2\ell)^{2\ell} = N^{\ell}(2\ell)^{2\ell}\max\left\{R^{\ell}, \ln^{\ell}\left(\frac{(2k)^{k+1}(Nk)^{k}}{e^{k}\epsilon}\right)\right\}, \tag{94}$$

which follows from Lemma Appendix A.4 and Lemma Appendix A.5. As $\sum_{\beta'\leq\beta}\binom{\beta}{\beta'} \leq 2^k$ (as a consequence of the multi-binomial theorem), we find that

$$\left\|\sum_{v\in\mathcal{V}_d}(f-q^N_{i+v})\Phi^{N,d}_{i+v}\right\|_{W^{k,\infty}(I^N_i)} \leq 2^k 3^d\left(\frac{\mathcal{C}(d,k,s,f)}{N^{s-k}}+\eta N^k\right)(2k)^{2k}\max\left\{R^k, \ln^k\left(\frac{(2Nk^2)^{k+1}}{\epsilon e^k}\right)\right\}. \tag{95}$$

Combining this result with the triangle inequality, Lemma Appendix A.5, Lemma Appendix A.6, (78), (80), Lemma 4.2 and the fact that $\ln(x) \leq \sqrt{x}$ for $x > 0$, we find that

$$\begin{aligned}
&\left\|\sum_{j\in\{1,\ldots,N\}^d}(f-q^N_j)\cdot\Phi^{N,d}_j\right\|_{W^{k,\infty}(I^N_i)} \\
&\leq \left\|\sum_{v\in\mathcal{V}_d}(f-q^N_{i+v})\Phi^{N,d}_{i+v}\right\|_{W^{k,\infty}(I^N_i)} + \sum_{\substack{j\in\{1,\ldots,N\}^d\\ j-i\notin\mathcal{V}_d}}\left\|(f-q^N_j)\Phi^{N,d}_j\right\|_{W^{k,\infty}(I^N_i)} \\
&\leq \left\|\sum_{v\in\mathcal{V}_d}(f-q^N_{i+v})\Phi^{N,d}_{i+v}\right\|_{W^{k,\infty}(I^N_i)} + \sum_{\substack{j\in\{1,\ldots,N\}^d\\ j-i\notin\mathcal{V}_d}} 2^k\left\|(f-q^N_j)\right\|_{W^{k,\infty}(I^N_i)}\left\|\Phi^{N,d}_j\right\|_{W^{k,\infty}(I^N_i)} \\
&\leq 2^k 3^d\left(\frac{\mathcal{C}(d,k,s,f)}{N^{s-k}}+\eta N^k\right)(2k)^{2k}\max\left\{R^k, \ln^k\left(\frac{(2Nk^2)^{k+1}}{\epsilon e^k}\right)\right\} \\
&\quad + N^d 2^k\left(\mathcal{C}(d,k,s,f)+\eta\right)(2k)^{2k}N^k(k+1)^k\left(\frac{2Nk^2}{e}\right)^{k/2}\sqrt{\epsilon} \\
&\leq 3^d\left(1+\frac{\delta}{3}\right)(2(k+1))^{3k}\max\left\{R^k, \ln^k\left(\frac{2Nk^2}{\epsilon^{\frac{1}{k+1}}e}\right)\right\}\frac{\mathcal{C}(d,k,s,f)}{N^{s-k}},
\end{aligned} \tag{96}$$

where we obtain the last inequality by making a suitable choice of η and ϵ, satisfying

$$\eta \leq \frac{\delta\mathcal{C}}{6N^s} \quad \text{and} \quad \epsilon \leq \frac{\delta^2}{N^{2s+2d+k}k^k}, \tag{97}$$

where we assumed that $0 < \delta < 5/6$.

Step 2d: Third term of (83). Finally, using the triangle inequality, Lemma Appendix A.7, Corollary 3.6 and Lemma Appendix A.4 we obtain that for some $C_k > 0$ depending only on k,

$$
\begin{aligned}
&\left\| \sum_{j\in\{1,\dots,N\}^d} (q_j^N \cdot \Phi_j^{N,d} - q_j^N \widehat{\times} \Phi_j^{N,d}) \right\|_{W^{k,\infty}(I_i^N)} \\
&\leq N^d C_k (d+1)^d d^{2k} \cdot \left\| \widehat{\times}^h_{d+1} - \prod_{i=1}^{d+1} x_i \right\|_{W^{k,\infty}} \left(\|f\|_{W^{k,\infty}([0,1]^d)} + \left\| f - q_j^N \right\|_{W^{k,\infty}([0,1]^d)} + \left\| \rho_i^N \right\|_{W^{k,\infty}([0,1]^d)} \right)^k \\
&\leq N^d C_k (d+1)^d d^{2k} \cdot h \left(\|f\|_{W^{k,\infty}([0,1]^d)} + \frac{\mathcal{C}(d,k,s,f)}{N^{s-k}} + \eta + (2\alpha k)^{k+1} \right)^k \\
&\leq 2^k \frac{\delta}{3} \frac{\mathcal{C}(d,k,s,f)}{N^{s-k}},
\end{aligned}
\tag{98}
$$

where we obtain the last inequality by making a suitable choice of h and η, satisfying

$$
\eta \leq \|f\|_{W^{k,\infty}([0,1]^d)} \quad \text{and} \quad h \leq \frac{2^k \delta \mathcal{C}}{3N^{d+s-k} C_k (d+1)^d d^{2k} (2\|f\|_{W^{k,\infty}([0,1]^d)} + \mathcal{C} + (2\alpha k)^{k+1})^k}. \tag{99}
$$

Step 2e: Final error bound. As i was chosen arbitrary, combining the contributions from the three terms of (83) then proves that

$$
\left\| f - \widehat{f}^N \right\|_{L^\infty([0,1]^d)} \leq (1+\delta) \frac{\mathcal{C}(d,0,s,f)}{N^s}. \tag{100}
$$

Moreover, from (85) and (97) we find that a suitable definition of ϵ is given by

$$
\epsilon = \frac{\delta^2 \min\{1, \mathcal{C}(d,k,s,f)\}}{N^{2s+2d+k} k^k 2^{(k+1)d} d \max\{1, \|f\|_{W^{k,\infty}([0,1]^d)}\}}, \tag{101}
$$

from which it follows that for $k \geq 1$,

$$
\epsilon^{\frac{1}{k+1}} \geq \frac{\delta \min\{1, \sqrt{\mathcal{C}(d,k,s,f)}\}}{N^{s+d+1} k 2^d \sqrt{d} \max\{1, \|f\|^{1/2}_{W^{k,\infty}([0,1]^d)}\}}. \tag{102}
$$

Combining this observation with all previous steps of the proof then leads to the error bound

$$
\left\| f - \widehat{f}^N \right\|_{W^{k,\infty}([0,1]^d)} \leq 3^d (1+\delta) (2(k+1))^{3k} \max\left\{ R^k, \ln^k\left(\beta N^{s+d+2} \right) \right\} \frac{\mathcal{C}(d,k,s,f)}{N^{s-k}}, \tag{103}
$$

for $k \geq 1$ and where we define

$$
\beta = \frac{k^3 2^d \sqrt{d} \max\{1, \|f\|^{1/2}_{W^{k,\infty}([0,1]^d)}\}}{\delta \min\{1, \sqrt{\mathcal{C}(d,k,s,f)}\}}. \tag{104}
$$

Step 3: Estimating the network and weights sizes. The first hidden layer requires $3\left\lceil\frac{s}{2}\right\rceil\left|P_{s-1,d+1}\right|$ neurons for the computation of all multivariate monomials (cf. Corollary 3.6). For $d = 1$, the result follows from Lemma 3.2 instead of Corollary 3.6. For the computation of all $\rho_j^N(x_i)$ another $d(N-1)$ neurons are needed in the first hidden layer. The second hidden layer needs at most $3\left\lceil\frac{d+2}{2}\right\rceil\left|P_{d+1,d+1}\right|$ neurons for realizing $\widehat{\times}^h_{d+1}$, which needs to be performed N^d times. For $d = 1$, six neurons are sufficient to approximate the multiplication (see (51)).

In the proof we achieved the wanted accuracy by making suitable choices of η, ϵ, h. From equation (100) and Lemma Appendix A.5, it follows that

$$\alpha = O\left(Ns\ln(\mathcal{C}N)\right). \tag{105}$$

For the approximate multiplication, (99) requires that $h^{-1} = O(N^{d+s+k^2}s^{k^2+k})$. Corollary 3.7 then proves that the weights of $\widehat{\times}^h_{d+1}$ grow as $O(N^{d(d+s+k^2)/2}s^{d(k^2+k)/2})$. Finally, the condition $\eta^{-1} = O(\mathcal{C}^{-1}N^s)$ from (97) corresponds to weights growing as

$$O\left(\mathcal{C}^{-s/2}N^{s^2/2}(s(s+2))^{3s(s+2)}\right) \tag{106}$$

as a consequence of Corollary 3.6. Calculating the maximum size of all the weights concludes the proof. □

Remark 5.2. *The result of Theorem 5.1 can be generalized to functions $f \in W^{k,p}(\Omega)$ for $p < \infty$. For this, a slightly more general version of Lemmas Appendix A.6 and Appendix A.8 is needed. The convergence rate will still be as in Theorem 5.1, only the constant $\mathcal{C}$ will be different.*

Remark 5.3. *Recently, it has been shown that the curse of dimensionality can be lessened for functions in so-called Korobov spaces [24, 23, 61]. In particular, in [23, Theorem 4.2], this framework is used to show how ReQU neural networks can approximate a $C^k([-1,1]^d)$-function to an accuracy of $\epsilon > 0$ in supremum norm with at most $O\big(\frac{d}{k}\ln\big(\frac{1}{\epsilon}\big) + d\big)$ hidden layers and at most $O\Big(\epsilon^{-\frac{1+\delta}{k}}\big(\frac{1+\delta}{k}\ln\big(\frac{1}{\epsilon}\big)\big)^{d-1}\Big)$ neurons and non-zero weights. As their proof builds upon the mimicking of polynomials, it is clear from our results that similar approximation rates can be obtained using tanh neural networks.*

One particularly useful consequence of Theorem 5.1 is that it provides an explicit error bound on the approximation of Lipschitz functions using tanh neural networks.

Corollary 5.4. *Let $d \in \mathbb{N}$ and let $f : [0,1]^d \to \mathbb{R}$ be a Lipschitz continuous function with Lipschitz constant $L > 0$. For every $N \in \mathbb{N}$ with $N > 5d^2$ there exists a tanh neural network $\widehat{f}^N$ with two hidden layers of widths at most $d(N-1)$ and $3\left\lceil\frac{d+1}{2}\right\rceil\left|P_{d,d}\right|N^d$ (or $N-1$ and $6N$ for $d = 1$), such that*

$$\left\|f - \widehat{f}^N\right\|_{L^\infty([0,1]^d)} \leq \frac{7d^2L}{N}. \tag{107}$$

Proof. The corollary follows directly from Theorem 5.1 by setting $k = 0$, $s = 1$, choosing $\delta > 0$ in such a way that $5(1+\delta)\pi^{1/4} \leq 7$ and observing that $|f|_{W^{1,\infty}} \leq L$ because of the Lipschitz continuity of f. The constructed network in Theorem 5.1 is based on localized $(s-1)$-th order polynomials. For $s = 1$ this corresponds to constant functions, thereby removing the need to mimick monomials. As a consequence, the network width can be simplified to the widths stated in the corollary. □

5.2. Approximation of analytic functions

We now investigate how we can apply Theorem 5.1 to analytic functions. As the class of analytic functions coincides with the Gevrey class G^1, it follows that for every analytic function there exists a constant $C_f > 0$ such that

$$|f|_{W^{s,\infty}([0,1]^d)} \leq C_f^{s+1} s! \text{ for all } s \in \mathbb{N}_0. \tag{108}$$

A related concept is the class of (Q, R)-analytic functions [62, 63, 64], where $Q, R > 0$, consisting of analytic functions for which the following smoothness condition holds,

$$|f|_{W^{s,\infty}([0,1]^d)} \leq Q R^{-s} s! \text{ for all } s \in \mathbb{N}_0. \tag{109}$$

Note that any analytic function is (C_f, C_f^{-1})-analytic by the previous characterization of analyticity. Hence, a function is analytic on some compact interval if and only if it is (Q, R)-analytic for some $Q, R > 0$ on that interval. The following corollaries discuss multiple ways to approximate analytic functions using tanh neural networks. All constants mentioned in the statements can be easily deduced from the proofs.

We start with the basic consequence of Theorem 5.1 for (Q, R)-analytic functions. It provides explicit estimates on both the approximation error in supremum norm and the network size. It can easily be generalized to Sobolev norm using Theorem 5.1.

Corollary 5.5. *Let $d \in \mathbb{N}$, $\delta, Q, R > 0$, $\Omega \subset \mathbb{R}^d$ open with $[0,1]^d \subset \Omega$ and let f be (Q, R)-analytic on Ω. Then for every $s \in \mathbb{N}_0, N \in \mathbb{N}$ with $N > 3d/2$, there is a tanh neural network $\widehat{f}^{N,s}$ with two hidden layers of widths at most $3\left\lceil\frac{s}{2}\right\rceil\left|P_{s-1,d+1}\right| + d(N-1)$ and $3\left\lceil\frac{d+2}{2}\right\rceil\left|P_{d+1,d+1}\right|N^d$ (or $3\left\lceil\frac{s}{2}\right\rceil + N - 1$ and $6N^d$ for $d = 1$) such that*

$$\left\|f - \widehat{f}^{N,s}\right\|_{L^\infty([0,1]^d)} \leq (1+\delta) Q \left(\frac{3d}{2RN}\right)^s. \tag{110}$$

Moreover, if $R > d/2$ then for every $s \in \mathbb{N}_0$ there is a shallow tanh neural network $\widehat{f}^s$ with at most $\frac{3s}{2}\left|P_{s-1,d+1}\right|$ (or $3\left\lceil\frac{s}{2}\right\rceil$ for d=1) in its hidden layer such that

$$\left\|f - \widehat{f}^s\right\|_{L^\infty([0,1]^d)} \leq (1+\delta) Q \left(\frac{d}{2R}\right)^s. \tag{111}$$

Proof. The first part of the statement follows directly from Theorem 5.1 by taking $\delta = 1/3$. The second part follows from taking $N = 1$ in Theorem 5.1 and observing that the proof can be simplified in this case. Indeed, one can then directly use Taylor's theorem (Lemma Appendix A.9) with $\delta = \frac{1}{2}$ instead of $\delta = \frac{3}{2N}$ and there is no more need for an approximate partition of unity, thereby also removing the need for a second hidden layer. $\square$

The following corollary enables a consistent comparison with the available literature, as it bounds the approximation error in terms of one single parameter. Whereas other papers focus on the number of non-zero weights and biases as complexity measure, we opted for the network width. This is useful in practice as the network width can be directly chosen, whereas it is very challenging to exactly control the sparsity of the neural network (i.e. the number of non-zero weights and biases). Moreover, many bounds on the generalization error require an estimate of the network width [54, 65].

Corollary 5.6. *Let $d \in \mathbb{N}$, $k \in \mathbb{N}_0$, $\delta, Q, R > 0$, $\Omega \subset \mathbb{R}^d$ open with $[0,1]^d \subset \Omega$ and let f be (Q,R)-analytic on Ω. Then there exists a constant $c_{d,k,\alpha,f} > 0$ such that for every $\mathcal{N} \in \mathbb{N}$ there exists a tanh neural network $\widehat{f}^{\mathcal{N}}$ with two hidden layers of width at most $O(\mathcal{N})$ for $\mathcal{N} \to \infty$ such that*

$$\left\| f - \widehat{f}^{\mathcal{N}} \right\|_{W^{k,\infty}([0,1]^d)} \leq c_{d,k,\alpha,f} \mathcal{N}^{\frac{k}{d+1}} \exp\left(-\alpha \mathcal{N}^{\frac{1}{d+1}} \log(\mathcal{N})\right) \leq \frac{c_{d,k,\alpha,f}}{\mathcal{N}^{\alpha - k/(d+1)}}. \tag{112}$$

In particular, for $k = 0$ it holds that

$$\left\| f - \widehat{f}^{\mathcal{N}} \right\|_{L^{\infty}([0,1]^d)} \leq (1+\delta) Q \cdot \exp\left(-\alpha \mathcal{N}^{\frac{1}{d+1}} \log(\mathcal{N})\right) \leq \frac{(1+\delta)Q}{\mathcal{N}^{\alpha}}. \tag{113}$$

Proof. First we observe that for every $\gamma > 0$ it holds that

$$\ln^k\left(\beta N^{s+d+2}\right) = O\left(\left(\frac{2NR}{3d}\right)^{\gamma s}\right) \tag{114}$$

for large s and N. From Theorem 5.1 we then find that for every N and s there is a network $\widehat{f}^{N,s}$ such that

$$\left\| f - \widehat{f}^{N,s} \right\|_{W^{k,\infty}([0,1]^d)} = O\left(\left(\frac{s}{R}\right)^{k} \left(\frac{3d}{2NR}\right)^{s(1-\gamma)-k}\right) \tag{115}$$

From Theorem 5.1 with the choices $s = k + \alpha(1-\gamma)^{-1}(d+1)\mathcal{N}^{\frac{1}{d+1}}$ and $N = \frac{3d}{2R}\mathcal{N}^{\frac{1}{d+1}}$ for some $\mathcal{N} \in \mathbb{N}$, gives that there exists a constant $c_{d,k,\alpha,f} > 0$ such that

$$\begin{aligned}
\left\| f - \widehat{f}^{\mathcal{N}} \right\|_{W^{k,\infty}([0,1]^d)} &\leq c_{d,k,\alpha,f} \mathcal{N}^{\frac{k}{d+1}} \left(\frac{1}{\mathcal{N}^{\frac{1}{d+1}}}\right)^{\alpha(d+1)\mathcal{N}^{\frac{1}{d+1}}} \\
&= c_{d,k,\alpha,f} \mathcal{N}^{\frac{k}{d+1}} \exp\left(-\alpha \mathcal{N}^{\frac{1}{d+1}} \log(\mathcal{N})\right) \\
&\leq \frac{c_{d,k,\alpha,f}}{\mathcal{N}^{\alpha - k/(d+1)}}.
\end{aligned} \tag{116}$$

In particular, for $k = 0$ we find that

$$\left\| f - \widehat{f}^{\mathcal{N}} \right\|_{L^\infty([0,1]^d)} \leq (1+\delta)Q \cdot \exp\left(-\alpha \mathcal{N}^{\frac{1}{d+1}} \log(\mathcal{N})\right) \leq \frac{(1+\delta)Q}{\mathcal{N}^{\alpha}}. \tag{117}$$

Using Lemma 2.1, we find that the network widths are respectively $O((e\alpha d)^{d+1}\mathcal{N})$ and $O(d5^d\mathcal{N}^{\frac{d}{d+1}})$ for large $\mathcal{N}$ and d (the exact sizes can be easily calculated). □

We thus find that tanh neural networks with two hidden layers result in an exponential convergence rate. Moreover, the above corollary shows that a convergence rate that is independent of the dimension can be obtained, thereby lessening the curse of dimensionality. The proof however shows that even though the rate is free of the curse of dimensionality, the constant implied in the Landau notation still depends (super)exponentially on the dimension. Similar papers observe the same phenomena [4], or do not discuss this.

Remark 5.7. *One can also restate the previous corollary by saying that an approximation rate of $O(\mathcal{N}^k \exp(-\mathcal{N}))$ can be obtained using a tanh neural network with two hidden layers of widths $O\left(\mathcal{N}\binom{\mathcal{N}+d}{\mathcal{N}}\right)$ and $O(1)$ for $\mathcal{N} \to \infty$. Since $O\left(\mathcal{N}\binom{\mathcal{N}+d}{\mathcal{N}}\right)$ grows asymptotically slower than $O(\mathcal{N}^{d+1})$, another (very modest) lessening of the curse of dimensionality is revealed.*

Next, we show that, under an additional assumption, shallow tanh neural networks can also approximate analytic functions at an exponential rate. Moreover, in contrast to Corollary 5.6, there are no hidden constants that grow as $O(d^d)$. For simplicity, we restrict ourselves to approximation in supremum norm (i.e. $k = 0$).

Corollary 5.8. *Let $d \in \mathbb{N}$, $\Omega \subset \mathbb{R}^d$ open with $[0,1]^d \subset \Omega$ and let f be analytic on Ω. If f satisfies for some $C > 0$ that $|f|_{W^{s,\infty}([0,1]^d)} \leq C^s$ for all $s \in \mathbb{N}$, then for every $\mathcal{N} \in \mathbb{N}$ there exists a shallow tanh neural network $\widehat{f}^{\mathcal{N}}$ of width $3\left\lceil \frac{\mathcal{N}+5Cd}{2} \right\rceil \binom{\mathcal{N}+(5C+1)d}{\mathcal{N}+5Cd}$ (or $3\left\lceil \frac{\mathcal{N}}{2} \right\rceil$ for $d = 1$) such that*

$$\left\| f - \widehat{f}^{\mathcal{N}} \right\|_{L^\infty([0,1]^d)} \leq \exp(-\mathcal{N}). \tag{118}$$

Proof. Assume that f satisfies for some $C > 0$ that $|f|_{W^{s,\infty}([0,1]^d)} \leq C^s$ for all $s \in \mathbb{N}$. We calculate that for $\rho > 1$,

$$\frac{C^s}{s!}\left(\frac{3d\rho}{2}\right)^s = \frac{1}{s!}\left(\frac{3Cd\rho}{2}\right)^s \leq \frac{1}{\sqrt{2\pi}}\left(\frac{3Cde\rho}{2s}\right)^s \leq \frac{1}{\sqrt{2\pi}}e^{3Cd\rho/2}, \tag{119}$$

where we used Stirling's approximation and maximized over all s. This proves that f is (Q,R)-analytic with $Q = \frac{1}{\sqrt{2\pi}}e^{3Cd\rho/2}$ and $R = \frac{3d\rho}{2}$. Using Corollary 5.5 with $s = \mathcal{N}$ and $N = 1$ gives us that

$$\left\| f - \widehat{f}^{\mathcal{N}} \right\|_{L^\infty([0,1]^d)} \leq 2\frac{1}{\sqrt{2\pi}}e^{3Cd\rho/2}\rho^{-\mathcal{N}}. \tag{120}$$

If we set $\rho = e$, then $e^{3d\rho/2} \leq e^{5d}$. Therefore it holds that

$$\left\| f - \widehat{f}^{\mathcal{N}} \right\|_{L^\infty([0,1]^d)} \leq \exp(-\mathcal{N} + 5Cd). \tag{121}$$

Note that since now $N = 1$, the network architecture is even simpler: there is no need to construct a partition of unity, nor does there need to be a second hidden layer in order to approximately multiply the results of subnetworks. Therefore, a shallow tanh neural network with $3\left\lceil \frac{\mathcal{N}}{2} \right\rceil \left| P_{\mathcal{N}-1,d+1} \right| \leq 3\left\lceil \frac{\mathcal{N}}{2} \right\rceil \binom{\mathcal{N}+d}{\mathcal{N}}$ neurons in its hidden layer suffices. The statement from the theorem is obtained by making the substitution $\mathcal{N} \leftarrow \mathcal{N} + 5Cd$. □

Finally, we discuss how dimension-independent convergence rates can be obtained for a class of countably-parametric, holomorphic maps $f : U := [-1,1]^{\mathbb{N}} \to \mathbb{R}$, which arise in e.g. elliptic PDEs with uncertain coefficients. This was first discussed in [26] for deep ReLU neural networks and we will show that their results can be adapted to hold for shallow tanh neural networks. More precisely, their results hold for functions u that admit a representation as a sparse Taylor generalized polynomial chaos expansion

$$f(y) = \sum_{\nu \in \mathcal{F}} \frac{D^\nu f(0)}{\nu!} y^\nu, \tag{122}$$

which is unconditionally convergent for $y \in U$ and where $\mathcal{F}$ is defined by

$$\mathcal{F} = \{\nu \in \mathbb{N}_0^{\mathbb{N}} \mid \nu_j \neq 0 \text{ for only finitely many } j\}. \tag{123}$$

For a multi-index $\nu \in \mathbb{N}_0^{\mathbb{N}}$, we denote by $\operatorname{supp}(\nu) = \{j \in \mathbb{N} \mid \nu_j \neq 0\}$ the support of ν, and we denote by $|\nu| = \sum_{j\in\mathbb{N}} |\nu_j|$ the ℓ^1-norm of ν.

It is shown in [26, Section 2] that f admits such a representation if f is (b,ϵ)-holomorphic for $b \in \ell^p(\mathbb{N})$, $p \in (0,1]$ and $\epsilon > 0$. The notion of (b,ϵ)-holomorphy is defined as follows.

Definition 5.9 (Def. 2.1 in [26])**.** *Let V be a Banach space. Let $b \in \ell^p(\mathbb{N})$, $p \in (0,1]$ be a monotonically decreasing sequence. A poly-radius $\rho \in [1,\infty)^{\mathbb{N}}$ is called (b,ϵ)-admissible for some $\epsilon > 0$ if*

$$\sum_{j\in\mathbb{N}} b_j(\rho_j - 1) \leq \epsilon. \tag{124}$$

A continuous function $f : U \to V$ is called (b,ϵ)-holomorphic if there exists a constant $C_f < \infty$ such that the following holds: For every (b,ϵ)-admissible ρ, there exists an extension $\tilde{f} : B_\rho \to V_{\mathbb{C}}$ of f, i.e. we have $\tilde{f}(y) = f(y)$ for all $y \in U \subset B_\rho$, $\tilde{f}$ is holomorphic in each component and such that $\sup_{z\in B_\rho} \left\| \tilde{f}(z) \right\|_{V_{\mathbb{C}}} \leq C_f$. Here, $B_\rho \subset \mathbb{C}^{\mathbb{N}}$ denotes the ball of polyradius ρ:

$$B_\rho = \{z \in \mathbb{C}^{\mathbb{N}} \mid |z_j| < \rho_j,\ \forall j \in \mathbb{N}\},$$

and $V_{\mathbb{C}} \simeq V + iV$ is the complexification of V.

In [26], it is shown that for a (b, ϵ)-holomorphic function f, an approximation rate of $O(\mathcal{N}^{1-1/p})$ can be obtained using a ReLU neural network of depth $O(\log(\mathcal{N}) \log\log(\mathcal{N}))$ or using a neural network with a smoother activation function of depth $O(\log\log(\mathcal{N}))$. We show that their construction can also be used to obtain a dimension-independent approximation rate for shallow tanh neural networks.

Corollary 5.10. *Let $f : U = [-1,1]^{\mathbb{N}} \to \mathbb{R}$ be (b,ϵ)-holomorphic for $b \in \ell^p(\mathbb{N})$, $p \in (0,1)$ and $\epsilon > 0$. Then there exists a constant $C > 0$ such that for every $\mathcal{N} \in \mathbb{N}$ there exists a shallow tanh neural network $\widehat{f}^{\mathcal{N}}$ of width at most $O\left(\mathcal{N}(C\log(\mathcal{N}))^{C\log(\mathcal{N})}\right)$ such that*

$$\left\| f - \widehat{f}^{\mathcal{N}} \right\|_{L^\infty(U)} = O\left(\mathcal{N}^{1-1/p}\right) \qquad \text{for } \mathcal{N} \to \infty. \tag{125}$$

Proof. There exist a constant $C > 0$ and a sequence of index sets $(\Lambda_{\mathcal{N}})_{\mathcal{N}\in\mathbb{N}} \subset \mathcal{F}$ for which it holds that (cf. [26, proof of Thm. 3.9])

$$\sup_{y\in U} \left| f(y) - \sum_{\nu\in\Lambda_{\mathcal{N}}} \frac{D^\nu f(0)}{\nu!} y^\nu \right| = O\left(\mathcal{N}^{1-1/p}\right). \tag{126}$$

and such that $|\Lambda_{\mathcal{N}}| = \mathcal{N}$, $\mathrm{supp}(\nu) \subseteq \{1, \ldots, \mathcal{N}\}$ for all $\nu \in \Lambda_{\mathcal{N}}$ and for all $\mathcal{N}$ [26, proof of Thm. 3.9], and $\sup_{\mathcal{N}\in\mathbb{N}} |\nu| \leq C(1+\log(\mathcal{N}))$, where $|\nu| := \sum_{j\in\mathbb{N}} |\nu_j|$ [26, Thm. 2.7]. The latter implies in particular that $\sup_{\mathcal{N}\in\mathbb{N}} \left|\mathrm{supp}(\nu)\right| \leq C(1 + \log(\mathcal{N}))$.

Based on these results from [26], it therefore suffices to show that we can accurately approximate all monomials $y \mapsto y^\nu$ for $\nu \in \Lambda_{\mathcal{N}}$ with shallow tanh neural networks. For a fixed $\nu \in \Lambda_{\mathcal{N}}$, the monomial $y \mapsto y^\nu$ can be approximated (to arbitrary accuracy) using Corollary 3.6 with $d = n \leftarrow C(1 + \log(\mathcal{N}))$, resulting in a shallow tanh neural network of width $O\left((Ce(1 + \log(\mathcal{N})))^{C(1+\log(\mathcal{N}))+1}\right)$. The network $\hat{f}^{\mathcal{N}}$ from the statement can then be constructed by parallelizing all the networks that approximate the individual monomials, yielding an approximation

$$\left\| f - \hat{f}^{\mathcal{N}} \right\|_{L^\infty(U)} = O(\mathcal{N}^{1-1/p}).$$

To be precise, we take the input of this network to be $(y_1, \ldots y_{\mathcal{N}})$ instead of y, which is possible since $\mathrm{supp}(\nu) \subseteq \{1, \ldots, \mathcal{N}\}$ for all $\nu \in \Lambda_{\mathcal{N}}$. As $|\Lambda_{\mathcal{N}}| = \mathcal{N}$, the resulting width of $\hat{f}^{\mathcal{N}}$ is $O\left(\mathcal{N}(Ce(1 + \log(\mathcal{N})))^{C(1+\log(\mathcal{N}))+1}\right)$, which is asymptotically equivalent to the width from the statement for $\mathcal{N} \to \infty$. □

This result implies in particular, that linear functionals of parametric solutions of PDEs can be approximated by shallow tanh neural networks [26]. Following [26, Section 4], the result can also be extended to directly approximate the parametric solution manifold, e.g. to approximate (b, ϵ)-holomorphic operators of the form $f : [-1, 1]^{\mathbb{N}} \to H_0^1([0, 1])$.

5.3. Examples

In this section, we illustrate the bounds derived in Theorem 5.1 with some prototypical examples. In particular, we will investigate the width, weights and sparsity of the networks from the proof of Theorem 5.1.

First, we demonstrate how large the networks of Theorem 5.1 are for a simple function approximation example with $d = 1$ and L^∞-norm. We consider the functions

$$f_a : [0,1] \to [-1,1] : x \mapsto \sin(ax), \quad a > 0. \tag{127}$$

For a given error tolerance $\varepsilon > 0$, we look for a three-layer tanh neural network $\widehat{f}^{N,s}$, as given by Theorem 5.1, such that provably

$$\left\| f_a - \widehat{f}^{N,s} \right\|_\infty \leq \varepsilon. \tag{128}$$

From all the networks that satisfy this condition, we take the one with the minimal width. More rigorously, we select

$$\mathrm{argmin}_{s,N\in\mathbb{N}\,:\,(3a/2N)^s/s!<\varepsilon} \max\left\{3\left\lceil\frac{s}{2}\right\rceil + N - 1, 6N\right\}, \tag{129}$$

where we used that $|f_a|_{W^{s,\infty}} \leq a^s$ for $s \geq 1$. Alternatively, one can also set $N = 1$ in Theorem 5.1, which makes the bound more efficient as no more partition of unity is needed, thereby reducing the need for a second hidden layer. This is similar to the proof of Corollary 5.5. In this case, we select

$$\mathrm{argmin}_{s\in\mathbb{N}\,:\,(a/2)^s/s!<\varepsilon} \left\{3\left\lceil\frac{s}{2}\right\rceil\right\}. \tag{130}$$

We present the result in Figure 2. For the chosen examples, a shallow (i.e. two-layer) tanh neural network achieves a similar level of error as a three-layer network of the same width. This can be explained by the fact that $|f_a|_{W^{s,\infty}}$ grows as $O(a^s)$ and not as $O(a^s s!)$, such that setting $N > 1$ is not required for the bound of Theorem 5.1 to be non-vacuous. Moreover, the networks suggested by Theorem 5.1 are not unreasonably large for this simple example. Yet, they still remain overestimates: we found that e.g. $f_{2\pi}$ can already be approximated to an error of 1% by a shallow tanh network of width four. Finally, the exponential convergence is evident as a small increase in the network width already leads to a very large improvement in the accuracy.

Next, we investigate whether the blow-up of the network weights from the theoretical results is observed in practice. We approximate univariate monomials of odd power in supremum norm on the interval $[0, 1]$ using shallow neural networks whose sizes are determined by Lemma 3.1. We generate a training set using 2000 randomly generated points, based on the uniform distribution on $[0, 1]$ and minimize the training loss for 2000 epochs using the Adam optimizer [66]. The results can be found in Table 1 and show that the weights do not blow up in practice. Rather, the weights remain small for this example. This

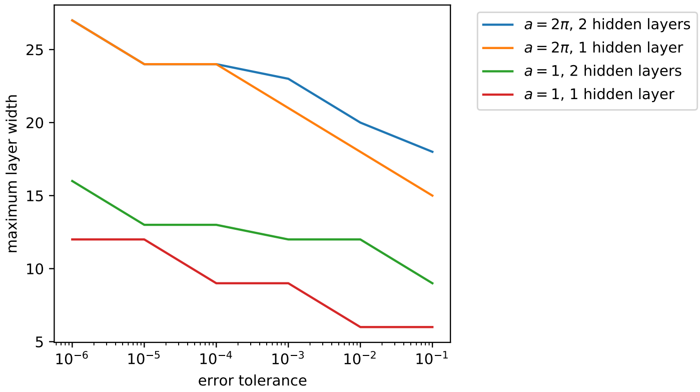

Figure 2: Needed layer width according to Theorem 5.1 to approximate the function f_a to a given error tolerance.

power	1	3	5	7	9
MSE	$4.91 \cdot 10^{-7}$	$1.54 \cdot 10^{-6}$	$2.87 \cdot 10^{-5}$	$1.13 \cdot 10^{-4}$	$6.13 \cdot 10^{-5}$
largest weight	3.13	2.14	2.27	4.55	4.41

Table 1: MSE and largest weight (in absolute value) of shallow tanh neural networks that approximate univariate monomials with odd powers on $[0, 1]$.

is possibly a consequence of the phenomenon of implicit regularization in deep learning, e.g. [67].

Finally, we show that the neural networks constructed in the proof of Theorem 5.1 are not very sparse i.e., the fraction of non-zero weights of the network, compared to the total number of weights, is not small. Figure 3 shows that the fraction of non-zero weights of the network increases with increasing s and decreasing N (for $d = 1$). For analytic functions, it is (asymptotically) more efficient to increase s than N, as the convergence rate is $O(N^{-s})$. This lets us conclude that the constructed networks corresponding to sensible choices of s and N are, in general, quite dense. This is in agreement with what one observes in practice. This is in contrast to the theoretical results for deep ReLU (and other) neural networks, where the sparsity of the constructed networks generally increases with increasing accuracy [20, 4, 34].

6. Summary and discussion

The main aim of this paper was to provide explicit bounds on the error (in high-order Sobolev) norms with which a neural network with a tanh activation function approximates Sobolev-regular functions, C^k functions and analytic functions. To this end, we prove such explicit bounds on the approximation error for Sobolev functions in Theorem 5.1 and for analytic functions in Corollary 5.5. In both cases, we prove these bounds for a tanh neural network with just 2

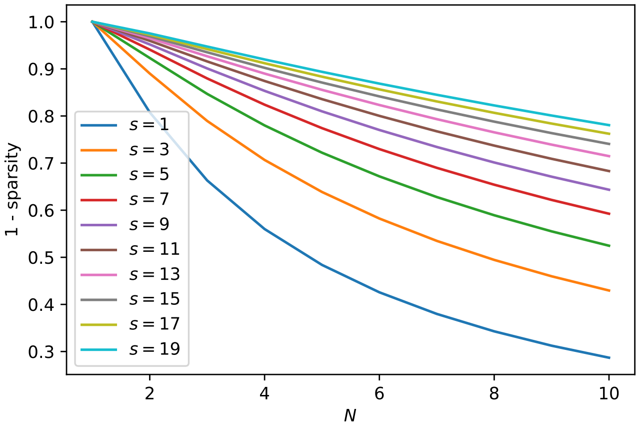

Figure 3: Fraction of non-zero weights (i.e. $1-$ sparsity) of the networks of Theorem 5.1 for different values of s and N in the case where $d = 1$.

hidden layers. Our proofs are constructive and the construction relies on three key ideas: (1) the approximation of monomials by finite differences of a smooth activation function (Lemma 3.5), (2) the approximation of the multiplication operator (Lemma 3.7) and (3) the approximation of a partition of unity (Section 4). In particular, we prove that a neural network with only hidden layers and a tanh activation function yields the same (or better) approximation rates for Sobolev-regular and for analytic functions.

We elaborate this point further by comparing and contrasting our approach and results with the large body of literature on approximation of functions with artificial neural networks.

First, we compare our approach, as stated above, with other related works. The simple, yet very effective trick of approximating monomials by finite differences of smooth activation function has been around for decades [36], but is still a building block in the constructive proofs of many recent papers on neural network expressivity, e.g. [39, 35, 37, 34]. To the best of the authors' knowledge, all available results build upon the observation that there is a $x \in \mathbb{R}$ such that $\sigma^{(n)}(x) \neq 0$ for all $n \in \mathbb{N}$. As such, this construction does not allow for explicit estimates on the approximation error and the network weights (see Remark 3.4). *Our key novelty in this paper is to circumvent this issue by first approximating univariate monomials of odd powers and then expanding to even powers and multivariate monomials.* This allows us to obtain *uniform* explicit bounds for the error in approximating multivariate monomials of a varying degree and paves the way for explicit bounds on the approximation error.

The approximation of the multiplication operator in d dimensions by a shallow neural network (Corollary 3.7) was discussed in [40, Appendix A] for activation functions σ that satisfy that $\sigma(x) = \sum_k \sigma_k x^k$ where $\sigma_k \neq 0$ for $0 \leq k \leq d$.

In particular, they prove that 2^d neurons are both sufficient and necessary. However, this construction does not allow for explicit estimates (again cf. Remark 3.4). Here, we propose a *novel construction* of the multiplication operator with a shallow tanh neural network.

As for the partition of unity, which serves as an essential ingredient in our proofs, an exact partition of unity for ReLU neural networks can be readily constructed [20]. Approximations of partitions of unity with neural networks with sigmoidal activation function can be found in [32, 68, 35] and a general framework for approximations of partitions of unity was proposed in [34]. In this paper, we have constructed approximations of partitions of unity with shallow tanh neural networks, that were motivated by localized polynomials which arise in the Bramble-Hilbert Lemma and the Taylor's theorem. Compared to the other works mentioned above, our results on partitions of unity stand out for the explicit bounds on the approximation error and the weights.

Given the afore-mentioned novel ideas, we were able to obtain explicit bounds on the approximation error. A suitable avenue to compare our results with results obtained in related works lies in the approximation error bounds for analytic functions. We recall that we prove approximation rates to analytic functions in the $W^{k,\infty}$-norm. Although approximation rates in this norm were proved for Sobolev functions in the very recent paper [34], it is unclear if their results can be extended in an efficient way for analytic functions. A key reason for this lies in the fact that the widths of their constructed networks are not explicitly stated and the depth increases with maximal degree of monomials, inhibiting uniform control that is necessary for approximating analytic functions.

Exponential convergence (in terms of network size) of neural networks for analytic functions in the L^∞-norm was first proven in [56] for neural networks with smooth activation functions and in [55] for ReLU neural networks. In [4, 22], the authors prove exponential convergence in $W^{1,\infty}$-norm for ReLU neural networks. We compare our results for approximation of analytic functions with these papers in Table 2. For [56], the parameter ρ is related to the polyradius of the ellipse to which the function needs to be holomorphically extendable. In [55], the additional assumption is made that the analytic function admits a Taylor expansion on $[-1,1]^d$ that converges absolutely and uniformly, which does not hold for general analytic functions. For [4], the parameter β is at least inversely proportional to the dimension d and also depends on the radius of the Bernstein ellipse to which the analytic function can be holomorphically extended. From Table 2, one can clearly observe that Corollary 5.6 yields an asymptotically faster convergence in terms of network width than the other related works. In addition, our results hold in stronger norms and we provide explicit bounds on the approximation error and weights, in contrast to other papers. For instance in [56, 4], the convergence rate even depends on the (unknown) polyradius of the ellipse to which the function can be holomorphically extended. Lastly, note that Corollary 5.8 assumes that $|f|_{W^{s,\infty}([0,1]^d)} \leq C^s$ for some $C > 0$ and all $s \in \mathbb{N}$, which implies that f must be entire. In [69, Theorem 5.4], exponential expressivity of ReLU neural networks for entire functions is proven. Compared

source	norm	activation	depth	width	error bound
[56, Thm 2.3]	$L^\infty([-1,1]^d)$	C^∞	2	$\mathcal{N}$	$O\left(\rho^{-\mathcal{N}^{1/d}}\right)$
[55, Thm. 6]	$L^\infty([-1+\delta,1-\delta]^d)$	ReLU	$O(\mathcal{N})$	$d+4$	$O\left(\exp\left(-d\delta\mathcal{N}^{1/2d}\right)\right)$
[4, Thm. 3.6]	$W^{1,\infty}([-1,1]^d)$	ReLU	$O(\mathcal{N}^{\frac{1}{d+1}}\log(\mathcal{N}))$	$O(\mathcal{N})$	$O\left(\exp\left(-\beta\mathcal{N}^{\frac{1}{d+1}}\right)\right)$
this work	$W^{k,\infty}([0,1]^d)$	tanh	3	$O(\mathcal{N})$	$O\left(\mathcal{N}^{\frac{k}{d+1}}\exp\left(-\mathcal{N}^{\frac{1}{d+1}}\ln(\mathcal{N})\right)\right)$

Table 2: Comparison of upper bounds on the approximation error for analytic functions by neural networks.

to Corollary 5.8, this result is more efficient in terms of non-zero weights, but requires $O\left(\log\big(1/\epsilon\big)\log\log\big(1/\epsilon\big)\right)$ layers to obtain an approximation with accuracy $\epsilon > 0$.

All of our approximation results, including the approximation bounds on analytic functions hold for a tanh neural network with only two hidden layers. This result, see also [56], runs contrary to the prevailing view that depth of neural networks is essential for function approximation and establishes that shallow but wide neural networks can be very expressive when it comes to function approximation and might provide some justification for the use of very shallow and wide neural networks in scientific computing [43, 9, 70].

Finally, it is essential to mention that although we highlight our contribution in terms of the tanh activation function as it is the most commonly used of the smooth activation functions. Our results apply verbatim to the logistic or sigmoid activation function as it is a shifted and scaled tanh. However, our constructions also apply to a much larger class of smooth activation functions as elaborated in Section 3.

We conclude by pointing out some limitations of the presented results. The most important limitation is the fact that the amplitude of the weights in our constructive network can grow very fast (Theorem 5.1). In practice, implicit and explicit regularization mechanisms during training will ensure that such growth of weights will not happen. In fact, we present examples to empirically show that the gradient-descent based training procedure manages to find rather small weights and biases that still provide a very high accuracy. We therefore believe that our bounds are useful in practice, more as upper bounds for setting the network size.

Another limitation, which we share with other published results on approximation with neural networks, is that our results suffer from the curse of dimensionality. We could however prove that it is possible to obtain a dimension-independent convergence rate to analytic functions in Corollary 5.6. Another possible mitigation of the curse of dimensionality for the approximation rate is when the underlying map is (b, ϵ)-holomorphic, see Corollary 5.10 for the precise result. However in these cases, the constants (and hence the network size) can still grow exponentially in the input dimension. Fortunately, one can argue that a large number of high-dimensional functions are in fact compositions of low-dimensional functions, which might explain the success of deep learning in high dimensions [38]. For instance, the Kolmogorov-Arnold superposition theorem [71] even states that all d-variate functions are in fact compositions of univariate functions and the sum of d numbers, which also can be used to lessen the curse of dimensionality [72].

Finally, the weights in our constructed networks are continuous with respect to the function of interest f. It has been proven that the best neural approximation cannot be achieved using continuous weight selection [73]. An example of how discontinuous weight selection can improve the approximation rate can be found in [27].

Acknowledgements

SL and SM received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 770880)

Appendix A. Auxiliary results

Lemma Appendix A.1. *It holds for every $n \in \mathbb{N}$ that $\left|\tanh^{(2n-1)}(0)\right| \geq 1$.*

Proof. For $|x| < \frac{\pi}{2}$, the power series expansion of tanh at x is given by

$$\tanh(x) = \sum_{n=1}^{\infty} \frac{2^{2n}(2^{2n}-1)B_{2n}}{(2n)!} x^{2n-1}, \tag{A.1}$$

where B_n is the n-th Bernoulli number. One can then calculate that for every $n \in \mathbb{N}$,

$$\left|\tanh^{(2n-1)}(0)\right| = \left|\frac{2^{2n}(2^{2n}-1)B_{2n}}{2n}\right| \geq 1. \tag{A.2}$$

This concludes the proof of the statement. □

Lemma Appendix A.2. *Let $s \in 2\mathbb{N}-1$. It holds that*

$$\inf_{\alpha>0} \frac{2^{s/2}(1+\alpha)^{(s^2+s)/2}}{\alpha^{s/2}} \leq \sqrt{e}(2es)^{s/2}, \tag{A.3}$$

where the infimum is reached at $\alpha = 1/s$.

Proof. Some elementary calculations show that

$$\inf_{\alpha>0} \frac{2^{s/2}(1+\alpha)^{(s^2+s)/2}}{\alpha^{s/2}} = 2^{s/2}\left(1+\frac{1}{s}\right)^{(s^2+s)/2} s^{s/2}. \tag{A.4}$$

Moreover, it holds that

$$\left(1+\frac{1}{s}\right)^{(s^2+s)/2} = \left(\left(1+\frac{1}{s}\right)^{s}\right)^{(s+1)/2} \leq e^{(s+1)/2}. \tag{A.5}$$

The statement follows immediately from these inequalities. □

Lemma Appendix A.3. *Let $n, q \in \mathbb{N}$ and $P_{n,q} = \{\alpha \in \mathbb{N}_0^q : |\alpha| = n\}$. If $D = (D_{\alpha,\beta})_{\alpha,\beta \in P_{n,q}}$ is defined as in* (39)*, then D is invertible and*

$$\left\|D^{-1}\right\|_\infty \leq (n!)^3 \left|P_{n,q}\right|^2 2^n. \tag{A.6}$$

Proof. Following [57], let $P_{n,q} = \{\alpha \in \mathbb{N}_0^q : |\alpha| = n\}$ and $I_{n,q} = \{\alpha' \in \mathbb{N}_0^{q-1} : |\alpha'| \leq n\}$. Let $s(k,m)$ be Stirling numbers of the first kind, for $k \leq m$, defined by

$$x(x-1)\cdots(x-m+1) = \sum_{k=0}^{m} s(k,m) x^k. \tag{A.7}$$

For $\alpha', \beta' \in I_{n,q}$, define $S_{\alpha',\beta'} = \prod_{i=1}^{q-1} s(\alpha'_i, \beta'_i)$, where $S(\alpha', \beta') = 0$ unless $\alpha'_i \leq \beta'_i$ for all i. Denote by S the corresponding matrix, where the order of rows and columns reflects the lexicographic order on $I_{n,q}$. Next, define B by

$$B_{\alpha',\beta'} = \binom{\alpha'}{\beta'}(-1)^{|\beta'|} := \prod_{i=1}^{q-1} \binom{\alpha'_i}{\beta'_i}(-1)^{\beta'_i} \quad \text{for } \alpha', \beta' \in I_{n,q}. \tag{A.8}$$

Finally, let L and Λ be diagonal matrices defined by $L_{\alpha',\alpha'} = (-1)^{|\alpha'|}/\alpha'!$ and $\Lambda_{\alpha',\alpha'} = n(n-1)\cdots(n-|\alpha'|+1)/\alpha'!$ for $\alpha' \in I_{n,q}$. It then holds that [57, Corollary 2],

$$D^{-1} = BL^{-1}\Lambda^{-1}SLB. \tag{A.9}$$

To prove an upper bound on the supremum norm of D^{-1}, we first note that $|s(k,m)| \leq \sum_{k=0}^{m} |s(k,m)| = m!$ (which can be seen by setting $x = -1$ in (A.7)) and thus $S(\alpha', \beta') \leq \beta'!$. In addition, it holds for any $\alpha' \in I_{n,q}$ that

$$1 \leq \frac{n!}{\alpha'!(n-|\alpha'|)!} \leq \frac{n!}{\alpha'!}, \tag{A.10}$$

which gives us that $\max_{\alpha' \in I_{n,q}}(\alpha'!) \leq n!$. This gives us consequently

$$\left|L^{-1}\Lambda^{-1}S\right|_{\alpha',\beta'} \leq (\alpha'!)^2 \beta'! \leq (n!)^2 \beta'!, \tag{A.11}$$

$$\left|L^{-1}\Lambda^{-1}SL\right|_{\alpha',\beta'} \leq (n!)^2, \tag{A.12}$$

$$\left|L^{-1}\Lambda^{-1}SLB\right|_{\alpha',\beta'} \leq (n!)^2 \sum_{\gamma' \in I_{n,q}} \binom{\gamma'}{\beta'} \leq (n!)^3 |I_{n,q}|, \tag{A.13}$$

$$\left|BL^{-1}\Lambda^{-1}SLB\right|_{\alpha',\beta'} \leq (n!)^3 |I_{n,q}| \sum_{\gamma' \in I_{n,q}} \binom{\alpha'}{\gamma'} = (n!)^3 |I_{n,q}| 2^{|\gamma'|} \leq (n!)^3 |I_{n,q}| 2^n. \tag{A.14}$$

This and (A.9) let us conclude that $\left\|D^{-1}\right\|_\infty \leq (n!)^3 \left|I_{n,q}\right|^2 2^n$. The lemma then follows from the existence of a one-to-one correspondence of elements in $I_{n,q}$ and $P_{n,q}$. □

Lemma Appendix A.4. *Let $m \in \mathbb{N}$. Then it holds that*

$$\left|\sigma^{(m)}(x)\right| \leq (2m)^{m+1} \min\{\exp(-2x), \exp(2x)\} \quad \text{for all } x \in \mathbb{R}. \tag{A.15}$$

Proof. In [74], the following formula for the derivative of the hyperbolic tangent is proven,

$$\sigma^{(m)}(x) = (-2)^m (\sigma(x)+1) \sum_{k=0}^{m} \frac{k!}{2^k} \left\{ {m \atop k} \right\} (\sigma(x)-1)^k, \tag{A.16}$$

where $\left\{ {m \atop k} \right\}$ denote Stirling numbers of the second kind, for which it holds that $\left\{ {m \atop k} \right\} \leq \frac{k^m}{k!}$. This then gives us

$$\left|\sigma^{(m)}(x)\right| \leq 2^m \left|1+\sigma(x)\right| \sum_{k=0}^{m} k^m \leq 2^m m^{m+1} \left|1+\sigma(x)\right| \leq (2m)^{m+1} \exp(2x), \tag{A.17}$$

as $\sum_{k=0}^{m} k^m \leq m \cdot m^m \leq m^{m+1}$. Furthermore one can note that $\sigma^{(m)}(-x) = -\sigma^{(m)}(x)$, which gives us

$$\left|\sigma^{(m)}(x)\right| \leq 2^m m^{m+1} \left|1-\sigma(x)\right| \leq (2m)^{m+1} \exp(-2x). \tag{A.18}$$

The statement follows easily. □

Lemma Appendix A.5. *The conditions stated in* (53) *for $k > 0$ are satisfied if*

$$\alpha = N \max \left\{ R, \ln\left(\frac{(2k)^{k+1}(Nk)^k}{e^k \epsilon} \right) \right\}. \tag{A.19}$$

Proof. The first condition of (53) is trivially satisfied when α is chosen as in the statement. From Lemma Appendix A.4, it follows that

$$\alpha^k (2k)^{k+1} \exp\bigl(-2\alpha/N\bigr) \leq \epsilon \tag{A.20}$$

is a sufficient condition that implies the other conditions of (53). Using $\max_{\alpha>0} \alpha^k \exp\bigl(-\alpha/N\bigr) \leq (Nk)^k \exp(-k)$ we find that

$$\alpha^k \exp\bigl(-2\alpha/N\bigr) = \alpha^k \exp\bigl(-\alpha/N\bigr) \exp\bigl(-\alpha/N\bigr) \leq (Nk)^k \exp(-k) \exp\bigl(-\alpha/N\bigr). \tag{A.21}$$

The statement follows directly. □

Lemma Appendix A.6. *Let $d \in \mathbb{N}$, $k \in \mathbb{N}_0$, $\Omega \subset \mathbb{R}^d$ and $f, g \in W^{k,\infty}(\Omega)$. Then it holds that*

$$\|fg\|_{W^{k,\infty}} \leq 2^k \|f\|_{W^{k,\infty}} \|g\|_{W^{k,\infty}}. \tag{A.22}$$

Proof. The statement follows directly from the general Leibniz rule. □

Lemma Appendix A.7. *Let $d, m, n \in \mathbb{N}$, $\Omega_1 \subset \mathbb{R}^d$, $\Omega_2 \subset \mathbb{R}^m$, $f \in C^n(\Omega_1; \Omega_2)$ and $g \in C^n(\Omega_2; \mathbb{R})$. Then it holds that*

$$\|g \circ f\|_{W^{n,\infty}} \leq 16(e^2 n^4 m d^2)^n \|g\|_{W^{n,\infty}} \max_{1 \leq i \leq m} \left\|(f)_i\right\|^n_{W^{n,\infty}}. \tag{A.23}$$

Proof. Let $\nu \in \mathbb{N}^d$ with $|\nu| = n$. We use the multivariate Faà di Bruno formula [33],

$$D^\nu (g \circ f) = \sum_{1 \leq |\lambda| \leq n} D^\lambda g \sum_{p(\nu, \lambda)} (\nu!) \prod_{j=1}^{n} \frac{(f_{l_j})^{k_j}}{k_j!(l_j!)^{|k_j|}}, \tag{A.24}$$

where $(f_\mu)_i = D^\mu (f)_i$ for $1 \leq i \leq m$ and the set $p(\nu, \lambda)$ is defined as

$$\begin{aligned} p(\nu, \lambda) = \{(\kappa, \ell) := (k_1, \dots, k_n; l_1, \dots, l_n) :\ &\text{for some } 1 \leq s \leq n, \\ &k_i = 0 \text{ and } l_i = 0 \text{ for } 1 \leq i \leq n - s; |k_i| > 0 \text{ for } n - s + 1 \leq i \leq n; \\ &\text{and } 0 \prec l_{n-s+1} \prec \cdots \prec l_n \text{ are such that} \\ &\sum_{i=1}^{n} k_i = \lambda, \sum_{i=1}^{n} |k_i| l_i = \nu\}, \end{aligned} \tag{A.25}$$

where $a \prec b$ either means that $|a| < |b|$ or $a < b$ according to lexicographic ordering; furthermore the vectors k_i are m-dimensional and the l_i are d-dimensional. From the stated conditions, it follows directly that $\sum_{i=1}^n |k_i| \leq n$ and $\sum_{i=1}^n |l_i| \leq n$. Next, we bound the complexity of $p(\nu, \lambda)$. From $\sum_{i=1}^n |k_i| \leq n$, it follows that the number of κ is bounded above by $\left|P_{n,(m+1)n}\right|$, which can in turn be bounded by $\sqrt{\pi} e^n (mn)^n$ by Lemma 2.1. Similarly, it follows that the number of ℓ is bounded above by $\left|P_{n,(d+1)n}\right|$, which can in turn be bounded by $\sqrt{\pi} e^n (dn)^n$ by Lemma 2.1. Therefore, $\left|p(\nu, \lambda)\right| \leq \pi(e^2 n^2 md)^n$. Finally, we can make the estimates that $\left|\{\lambda : 1 \leq |\lambda| \leq n\}\right| \leq \left|P_{n,d+1}\right| \leq \sqrt{\pi} e^n d^n$, $D^\lambda g \leq \|g\|_{W^{n,\infty}}$, $\nu! \leq n!$ and $\prod_{j=1}^n (f_{l_j})^{k_j} \leq \max_{1 \leq i \leq m} \left\|(f)_i\right\|^n_{W^{n,\infty}}$. Together with Stirling's approximation, this yields

$$\begin{aligned} \left\|D^\nu (g \circ f)\right\|_\infty &\leq \sqrt{\pi} e^n d^n \|g\|_{W^{n,\infty}} \cdot \pi(e^2 n^2 md)^n \cdot n! \cdot \max_{1 \leq i \leq m} \left\|(f)_i\right\|^n_{W^{n,\infty}} \\ &\leq 16(e^2 n^4 m d^2)^n \|g\|_{W^{n,\infty}} \max_{1 \leq i \leq m} \left\|(f)_i\right\|^n_{W^{n,\infty}}. \end{aligned} \tag{A.26}$$

□

Lemma Appendix A.8 (Bramble-Hilbert)**.** *Let $\Omega \subset \mathbb{R}^d$ be an open and bounded set of diameter $0 < h < e^{-1/2} d^{-3/2}$ which is star-shaped with respect to every point in an open ball $B \subset \Omega$ with diameter ρh. Then for every $f \in W^{s,\infty}(\Omega)$ there exists a polynomial $\hat{f}$ of degree at most $s - 1$ such that for any $k \in \mathbb{N}_0$ with $k < s$ it holds that,*

$$\left\|f - \hat{f}\right\|_{W^{k,\infty}(\Omega)} \leq \frac{\sqrt{s}\pi^{1/4}(d\sqrt{de}h)^{s-k}}{(s-k-1)!} |f|_{W^{s,\infty}(\Omega)}. \tag{A.27}$$

Proof. By setting $p = q = \infty$ in the penultimate equation in the proof of the main theorem in [75] (note that this reference uses a different definition of Sobolev norm), it follows that there exists a polynomial $\hat{f}$ of degree at most $s-1$ such that for $0 \le m < s$ it holds that,

$$\left| f - \hat{f} \right|_{W^{m,\infty}(\Omega)} \le (s-m) \left(\sum_{\beta \in P_{s-m,d}} (\beta!)^{-2} \right)^{1/2} h^{s-m} \sqrt{\left| P_{s-m,d} \right|} |f|_{W^{s,\infty}(\Omega)}. \tag{A.28}$$

Using Lemma 2.1, we find that $\sqrt{\left| P_{s-m,d} \right|} \le \pi^{1/4} (ed)^{(s-m)/2}$ and from the multinomial theorem it follows that

$$\sum_{\beta \in P_{s-m,d}} (\beta!)^{-2} \le \sum_{\beta' \in P_{2(s-m),2d}} (\beta'!)^{-1} = \frac{(2d)^{2(s-m)}}{(2(s-m))!}. \tag{A.29}$$

One can also calculate that $(2(s-m))! \ge 4^{s-m}(s-m)((s-m-1)!)^2$. Combining the previous observations, we find

$$\left| f - \hat{f} \right|_{W^{m,\infty}(\Omega)} \le \frac{\sqrt{s-m}(d\sqrt{de}h)^{s-m}}{(s-m-1)!} |f|_{W^{s,\infty}(\Omega)}. \tag{A.30}$$

Majorizing over $0 \le m \le k$ then gives the upper bound from the statement. □

Lemma Appendix A.9 (Taylor's theorem)**.** *Let $d, s \in \mathbb{N}$, $0 < \delta < 1/d$. Then for every $f \in C^s([-\delta,\delta]^d)$ there exists a polynomial $\hat{f}$ of degree at most $s-1$ such that for any $k \in \mathbb{N}_0$ with $k < s$ it holds that,*

$$\left\| f - \hat{f} \right\|_{W^{k,\infty}([-\delta,\delta]^d)} \le \frac{(d\delta)^{s-k}}{(s-k)!} |f|_{W^{s,\infty}([-\delta,\delta]^d)} \tag{A.31}$$

Proof. We give a constructive proof. For $f \in C^s([-\delta,\delta]^d)$, we define the polynomial $\hat{f}$ as

$$\hat{f}(x) = \sum_{|\alpha| \le s-1} \frac{D^\alpha f(0)}{\alpha!} x^\alpha. \tag{A.32}$$

Then take $\beta \in \mathbb{N}_0^d$ with $|\beta| \le k$. It then holds that

$$D^\beta \hat{f}(x) = \sum_{\substack{|\alpha| \le s-1 \\ \alpha \ge \beta}} \frac{D^\alpha f(0)}{\alpha!} \frac{\alpha!}{(\alpha-\beta)!} x^{\alpha-\beta} = \sum_{|\gamma| \le s-1-|\beta|} \frac{D^\gamma D^\beta f(0)}{\gamma!} x^\gamma. \tag{A.33}$$

For $x \in [-\delta,\delta]^d$, Taylor's theorem guarantees the existence of a constant $c \in (0,1)$ such that

$$D^\beta f(x) = \sum_{|\gamma| \le s-1-|\beta|} \frac{D^\gamma D^\beta f(0)}{\gamma!} x^\gamma + \sum_{|\gamma| = s-|\beta|} \frac{D^\gamma D^\beta f(cx)}{\gamma!} x^\gamma. \tag{A.34}$$

The previous equalities, together with the multinomial theorem, then prove that

$$\left\| D^{\beta} f - D^{\beta} \hat{f} \right\|_{L^{\infty}([-\delta,\delta]^d)} \leq C_s \sum_{|\gamma|=s-|\beta|} \frac{\delta^{|\gamma|}}{\gamma!} = \frac{C_s (d\delta)^{s-|\beta|}}{(s-|\beta|)!}, \tag{A.35}$$

where $C_s := |f|_{W^{s,\infty}([-\delta,\delta]^d)}$. Under the assumption that $\delta < 1/d$ we then can conclude that

$$\left\| f - \hat{f} \right\|_{W^{k,\infty}([-\delta,\delta]^d)} \leq \frac{C_s (d\delta)^{s-k}}{(s-k)!}. \tag{A.36}$$

□

References

[1] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436–444.

[2] W. E, J. Han, A. Jentzen, Deep learning-based numerical methods for high-dimensional parabolic partial differential equations and backward stochastic differential equations, Communications in Mathematics and Statistics 5 (4) (2017) 349–380.

[3] J. Han, A. Jentzen, W. E, Solving high-dimensional partial differential equations using deep learning, Proceedings of the National Academy of Sciences 115 (34) (2018) 8505–8510.

[4] J. A. Opschoor, C. Schwab, J. Zech, Exponential ReLU DNN expression of holomorphic maps in high dimension, SAM Research Report 2019 (2019).

[5] G. Kutyniok, P. Petersen, M. Raslan, R. Schneider, A theoretical analysis of deep neural networks and parametric PDEs, arXiv preprint arXiv:1904.00377 (2019).

[6] F. Laakmann, P. Petersen, Efficient approximation of solutions of parametric linear transport equations by ReLU DNNs, arXiv preprint arXiv:2001.11441 (2020).

[7] K. O. Lye, S. Mishra, D. Ray, P. Chandrashekar, Iterative surrogate model optimization (ISMO): An active learning algorithm for PDE constrained optimization with deep neural networks, Computer Methods in Applied Mechanics and Engineering 374 (2021) 113575.

[8] K. O. Lye, S. Mishra, D. Ray, Deep learning observables in computational fluid dynamics, Journal of Computational Physics 410 (2020) 109339.

[9] L. Lu, P. Jin, G. E. Karniadakis, DeepOnet: Learning nonlinear operators for identifying differential equations based on the universal approximation theorem of operators, arXiv preprint arXiv:1910.03193 (2019).

[10] S. Lanthaler, S. Mishra, G. E. Karniadakis, Error estimates for DeepOnets: A deep learning framework in infinite dimensions, arXiv preprint arXiv:2102.09618 (2021).

[11] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, A. Anandkumar, Fourier neural operator for parametric partial differential equations, arXiv preprint arXiv:2010.08895v1 (2020).

[12] I. E. Lagaris, A. Likas, D. I. Fotiadis, Artificial neural networks for solving ordinary and partial differential equations, IEEE Transactions on Neural Networks 9(5) (2000) 987–1000.

[13] M. Raissi, G. E. Karniadakis, Hidden physics models: Machine learning of nonlinear partial differential equations, Journal of Computational Physics 357 (2018) 125–141.

[14] M. Raissi, P. Perdikaris, G. E. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, Journal of Computational Physics 378 (2019) 686–707.

[15] S. Mishra, R. Molinaro, Estimates on the generalization error of physics informed neural networks (PINNs) for approximating PDEs, arXiv preprint https://arxiv.org/pdf/2006.16144.pdf (2020).

[16] S. Mishra, R. Molinaro, Estimates on the generalization error of physics-informed neural networks (PINNs) for approximating PDEs II: A class of inverse problems, arXiv preprint arXiv:2007.01138 (2020).

[17] A. R. Barron, Universal approximation bounds for superpositions of a sigmoidal function, IEEE Transactions on Information theory 39 (3) (1993) 930–945.

[18] G. Cybenko, Approximation by superpositions of a sigmoidal function, Mathematics of Control, Signals and Systems 2 (4) (1989) 303–314. `doi:10.1007/BF02551274`.
URL `https://doi.org/10.1007/BF02551274`

[19] K. Hornik, M. Stinchcombe, H. White, Multilayer feedforward networks are universal approximators, Neural Networks 2 (5) (1989) 359 – 366. `doi:https://doi.org/10.1016/0893-6080(89)90020-8`.
URL `http://www.sciencedirect.com/science/article/pii/0893608089900208`

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

[21] I. Gühring, G. Kutyniok, P. Petersen, Error bounds for approximations with deep ReLU neural networks in $W^{s,p}$ norms, Analysis and Applications 18 (05) (2020) 803–859.

[22] L. Herrmann, J. Opschoor, C. Schwab, Constructive deep ReLU neural network approximation, SAM research report 2021-04, ETH Zürich (2021).

[23] B. Li, S. Tang, H. Yu, Better approximations of high dimensional smooth functions by deep neural networks with rectified power units, arXiv preprint arXiv:1903.05858 (2019).

[24] H. Montanelli, Q. Du, New error bounds for deep ReLU networks using sparse grids, SIAM Journal on Mathematics of Data Science 1 (1) (2019) 78–92.

[25] J. A. Opschoor, P. C. Petersen, C. Schwab, Deep ReLU networks and high-order finite element methods, Analysis and Applications 18 (05) (2020) 715–770.

[26] C. Schwab, J. Zech, Deep learning in high dimension: Neural network expression rates for generalized polynomial chaos expansions in UQ, Analysis and Applications 17 (01) (2019) 19–55. `arXiv:https://doi.org/10.1142/S0219530518500203`, `doi:10.1142/S0219530518500203`.
URL `https://doi.org/10.1142/S0219530518500203`

[27] D. Yarotsky, Optimal approximation of continuous functions by very deep ReLU networks, in: Conference on Learning Theory, PMLR, 2018, pp. 639–649.

[28] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural computation 9 (8) (1997) 1735–1780.

[29] K. Cho, B. van Merrienboer, C. Gulcehre, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using RNN encoder-decoder for statistical machine translation, in: Conference on Empirical Methods in Natural Language Processing (EMNLP 2014), 2014.

[30] S. Mishra, T. K. Rusch, Enhancing accuracy of deep learning algorithms by training on low-discrepancy sequences, arXiv preprint arXiv:2005.12564 (2021).

[31] M. Longo, S. Mishra, C. Schwab, T. K. Rusch, Higher-order Quasi-Monte Carlo training of deep neural networks, arXiv preprint arXiv:2009.02713 (2021).

[32] D. Costarelli, R. Spigler, Approximation results for neural network operators activated by sigmoidal functions, Neural Networks 44 (2013) 101–106.

[33] G. Constantine, T. Savits, A multivariate Faà di Bruno formula with applications, Transactions of the American Mathematical Society 348 (2) (1996) 503–520.

[34] I. Gühring, M. Raslan, Approximation rates for neural networks with encodable weights in smoothness spaces, Neural Networks 134 (2021) 107–130.

[35] I. Ohn, Y. Kim, Smooth function approximation by deep neural networks with general activation functions, Entropy 21 (7) (2019) 627.

[36] A. Pinkus, Approximation theory of the MLP model in neural networks, Acta numerica 8 (1) (1999) 143–195.

[37] J. W. Siegel, J. Xu, Approximation rates for neural networks with general activation functions, Neural Networks 128 (2020) 313–321.

[38] T. Poggio, H. Mhaskar, L. Rosasco, B. Miranda, Q. Liao, Why and when can deep-but not shallow-networks avoid the curse of dimensionality: a review, International Journal of Automation and Computing 14 (5) (2017) 503–519.

[39] D. Rolnick, M. Tegmark, The power of deeper networks for expressing natural functions, in: International Conference on Learning Representations, 2018.

[40] H. W. Lin, M. Tegmark, D. Rolnick, Why does deep and cheap learning work so well?, Journal of Statistical Physics 168 (6) (2017) 1223–1247.

[41] Y. Bengio, Y. LeCun, et al., Scaling learning algorithms towards ai, Large-scale kernel machines 34 (5) (2007) 1–41.

[42] M. Bianchini, F. Scarselli, On the complexity of neural network classifiers: A comparison between shallow and deep architectures, IEEE transactions on neural networks and learning systems 25 (8) (2014) 1553–1565.

[43] L. Lu, Y. Su, G. E. Karniadakis, Collapse of deep and narrow neural nets, arXiv preprint arXiv:1808.04947 (2018).

[44] P. Grohs, F. Voigtlaender, Proof of the theory-to-practice gap in deep learning via sampling complexity bounds for neural network approximation spaces, arXiv preprint arXiv:2104.02746 (2021).

[45] A. R. Barron, Approximation and estimation bounds for artificial neural networks, Machine learning 14 (1) (1994) 115–133.

[46] V. Kurková, M. Sanguineti, Geometric upper bounds on rates of variable-basis approximation, IEEE Transactions on Information Theory 54 (12) (2008) 5681–5688.

[47] E. Lavretsky, On the geometric convergence of neural approximations, IEEE Transactions on Neural Networks 13 (2) (2002) 274–282.

[48] Y. Makovoz, Random approximants and neural networks, Journal of Approximation Theory 85 (1) (1996) 98–109.

[49] P. C. Kainen, V. Kurkova, M. Sanguineti, Dependence of computational models on input dimension: Tractability of approximation and optimization tasks, IEEE Transactions on Information Theory 58 (2) (2012) 1203–1214.

[50] N. J. Guliyev, V. E. Ismailov, On the approximation by single hidden layer feedforward neural networks with fixed weights, Neural Networks 98 (2018) 296–304.

[51] N. J. Guliyev, V. E. Ismailov, Approximation capability of two hidden layer feedforward neural networks with fixed weights, Neurocomputing 316 (2018) 262–269.

[52] S. Shalev-Shwartz, S. Ben-David, Understanding machine learning: From theory to algorithms, Cambridge university press, 2014.

[53] F. Cucker, S. Smale, On the mathematical foundations of learning, Bulletin of the American Mathematical Society 39 (1) (2002) 1–49.

[54] C. Beck, A. Jentzen, B. Kuckuck, Full error analysis for the training of deep neural networks, arXiv preprint arXiv:1910.00121 (2019).

[55] E. Weinan, Q. Wang, Exponential convergence of the deep neural network approximation for analytic functions, Science China Mathematics 61 (10) (2018) 1733–1740.

[56] H. N. Mhaskar, Neural networks for optimal approximation of smooth and analytic functions, Neural computation 8 (1) (1996) 164–177.

[57] D. S. Moak, Combinatorial multinomial matrices and multinomial Stirling numbers, Proceedings of the American Mathematical Society (1990) 1–8.

[58] H. Katsuura, Summations involving binomial coefficients, The College Mathematics Journal 40 (4) (2009) 275–278.

[59] K. Weierstrass, Über die analytische Darstellbarkeit sogenannter willkürlicher Functionen einer reellen veränderlichen, Sitzungsberichte der Königlich Preußischen Akademie der Wissenschaften zu Berlin 2 (1885) 633–639.

[60] K. R. Davidson, A. P. Donsig, Real analysis and applications: theory in practice, Springer Science & Business Media, 2009.

[61] M. Blanchard, M. Bennouna, The representation power of neural networks: Breaking the curse of dimensionality, arXiv preprint arXiv:2012.05451 (2020).

[62] L. Demanet, L. Ying, On Chebyshev interpolation of analytic functions, preprint (2010).

[63] E. Candes, L. Demanet, L. Ying, Fast computation of Fourier integral operators, SIAM Journal on Scientific Computing 29 (6) (2007) 2464–2493.

[64] E. Candes, L. Demanet, L. Ying, A fast butterfly algorithm for the computation of Fourier integral operators, Multiscale Modeling & Simulation 7 (4) (2009) 1727–1750.

[65] A. Jentzen, T. Welti, Overall error analysis for the training of deep neural networks via stochastic gradient descent with random initialisation, arXiv preprint arXiv:2003.01291 (2020).

[66] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, in: ICLR, 2015.

[67] B. Neyshabur, R. Tomioka, N. Srebro, In search of the real inductive bias: On the role of implicit regularization in deep learning., in: ICLR (Workshop), 2015.

[68] D. Costarelli, R. Spigler, Multivariate neural network operators with sigmoidal activation functions, Neural Networks 48 (2013) 72–77.

[69] L. Herrmann, C. Schwab, J. Zech, Deep neural network expression of posterior expectations in bayesian pde inversion, Inverse Problems 36 (12) (2020). `doi:https://doi.org/10.1088/1361-6420/abaf64`.

[70] S. Mishra, R. Molinaro, Physics-informed neural networks for simulating radiative transfer, arXiv preprint arXiv:2009.13291 (2020). `arXiv:2009.13291`.

[71] A. N. Kolmogorov, On the representation of continuous functions of many variables by superposition of continuous functions of one variable and addition, in: Doklady Akademii Nauk, Vol. 114, Russian Academy of Sciences, 1957, pp. 953–956.

[72] H. Montanelli, H. Yang, Error bounds for deep ReLU networks using the Kolmogorov-Arnold superposition theorem, Neural Networks 129 (2020) 1–6.

[73] P. C. Kainen, V. Kůrková, A. Vogt, Approximation by neural networks is not continuous, Neurocomputing 29 (1-3) (1999) 47–56.

[74] K. N. Boyadzhiev, Derivative polynomials for tanh, tan, sech and sec in explicit form, arXiv preprint arXiv:0903.0117 (2009).

[75] R. G. Durán, On polynomial approximation in Sobolev spaces, SIAM journal on numerical analysis 20 (5) (1983) 985–988.