Error bounds for approximations with deep ReLU neural networks in general norms

Author: Ingo Gühring
Student ID: 337031

Primary Reviewer: Prof. Dr. Gitta Kutyniok
Secondary Reviewer: Prof. Dr. Reinhold Schneider
Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und eigenhändig sowie ohne unerlaubte fremde Hilfe und ausschließlich unter Verwendung der aufgeführten Quellen und Hilfsmittel angefertigt habe.

Die selbstständige und eigenständige Anfertigung versichert an Eides statt:

Berlin, den 21.09.2018

..........................................................
Ingo Gühring
Contents

1 Introduction 1
   1.1 Neural networks 1
   1.2 Supervised learning with neural networks 2
   1.3 Motivation: Approximating functions and derivatives 3
   1.4 Our contribution 4
   1.5 Outline 4

2 Preliminaries 5

3 Neural networks 9
   3.1 Network concatenation 11
   3.2 Network parallelization 12
   3.3 Standard neural networks 13

4 Interpolation spaces 15

5 Sobolev spaces 17
   5.1 Composition estimate 20
   5.2 Product estimate 22
   5.3 Averaged Taylor polynomial 22
   5.4 Fractional Sobolev spaces 25

6 Upper bounds for approximations with deep ReLU neural networks in Sobolev type norms 29
   6.1 Proof of the main result 30

7 Discussion and Future Work 49
1 Introduction

Powered by today’s improved computational resources and the immense amount of accessible data, deep neural networks substantially outperform both traditional modelling approaches, e.g. based on differential equations, and classical machine learning methods in a wide range of applications. Prominent areas of application include image processing (see [25, 35]), speech recognition (see [27, 45]), and natural language processing (see [10, 11]).

There are still many open questions regarding the success of deep neural networks. In fact, traditional machine learning paradigms are challenged and have to be rethought (see [57]), as e.g. the number of trainable model parameters often exceeds the number of samples and the optimization problem is highly non-convex (see [15]). Empirical analysis shows that the network depth is a key factor in the rise of deep learning. It is common belief that deep architectures allow the re-use of features and promote a hierarchical data representation by building more abstract features with increasing depth (see [5]). However, the role of depth is mathematically still hardly understood and an active area of research (see [4, 16, 40, 41, 54]).

Viewing learning algorithms as function approximators, their ability to express a rich class of functions, called expressiveness, is a necessary condition for disentangling complicated feature dependencies (see [5]) and to perform well in challenging real-world applications, such as image processing.

In this thesis, we study the expressiveness of neural networks. We construct deep networks that approximate functions from Sobolev spaces and their (fractional) gradient and derive upper bounds for the complexity of the network.

A better theoretical understanding of deep learning can help craft more efficient architectures and allows the use of neural networks in safety-critical applications, such as autonomous driving.

In the remaining part of this introduction, we briefly describe neural network architectures relevant for this thesis. Then, we present the setting neural networks are typically used in. Afterwards, we give a practical and theoretical motivation for studying approximations of functions and their derivatives with neural networks. Finally, we state our main result.

1.1 Neural networks

There exists a wide variety of neural network architectures, each adapted to specific tasks. One of the most widely used architectures is a feedforward architecture also known as multi-layer perceptron which implements a function as a sequence of affine-linear transformations followed by a componentwise application of a non-linear function, called activation function. The length $L$ of the sequence is referred to as the number of layers of the network and the input of a layer consists of the output of the preceding layer. We denote by $N_0 \in \mathbb{N}$ the dimension of the input space and by $N_l \in \mathbb{N}$ the dimension of the output of the $l$-th layer for $l = 1, \ldots, L$. Thus, if $\varrho : \mathbb{R} \to \mathbb{R}$ is the activation function and

$$T_l : \mathbb{R}^{N_{l-1}} \to \mathbb{R}^{N_l}$$

\footnote{Some of the very successful neural networks used in image processing have hundreds of millions of weights (see e.g. [50]).}
the affine-linear transformation of the $l$-th layer, then the computation in that layer can be described as

$$f_l : \mathbb{R}^{N_{l-1}} \rightarrow \mathbb{R}^{N_l}, \quad x \mapsto \varrho(T_l(x))$$

for $l = 1, \ldots, L - 1$, and

$$f_L : \mathbb{R}^{N_{L-1}} \rightarrow \mathbb{R}^{N_L}, \quad x \mapsto T_L(x).$$

Note that in the last layer no activation is applied and $\varrho$ acts componentwise. The parameters defining the affine-linear transformations $T_l$ are referred to as weights and $\sum_{k=0}^{L} N_k$ is called the number of neurons of the network, since each output node of $f_1$ can be seen as a small computational unit, similar to neurons in the brain. If a network has 3 or more layers, then it is usually called deep and a 2-layer network is called shallow. The complexity of a neural network is typically measured in the number of layers, weights, and neurons (see [2]). The term deep learning refers to the subset of machine learning methods associated with deep neural networks.

Recently, more general feedforward architectures which allow connections between non-neighbouring layers, so-called skip connections, have been shown to yield state-of-the-art results in object recognition tasks (see [25, 31]). Here, the input of a layer consists of the output of all preceding layers. Note, that the case where no skip connections are allowed, is a special case of this more general architecture.

One of today’s most widely used activation functions is the Rectified Linear Unit (ReLU) (see [35]), defined as

$$\varrho : \mathbb{R} \rightarrow \mathbb{R}, \quad x \mapsto \max\{0, x\}.$$  

The popularity of the ReLU can be explained by a number of factors. It is cheap to compute, promotes sparsity in data representation (see [5]), alleviates the problem of vanishing-gradients (see [20]), and thus yields better optimization properties.

In this work, we will mostly focus on ReLU networks with a feedforward architecture allowing skip-connections.

### 1.2 Supervised learning with neural networks

Typically, neural networks are applied in supervised learning problems. The starting point is a dataset of input-output pairs $(x_i, f(x_i))_{i=1}^m$, called samples, where $f$ is in most cases an unknown function with values only given at sample points $x_i$. As an example, $x_i$ can be thought of as an image and $f(x_i)$ as a vector of scores, where each score is the probability of a certain category, e.g. "dog" or "cat", being associated with $x_i$. During training, one then seeks to learn the function $f$ by adapting the weights $w$ of $\mathcal{N}$ such that the empirical loss

$$\frac{1}{m} \sum_{i=1}^{m} l(\mathcal{N}(x_i|w), f(x_i))$$

is minimized for some loss function $l$. Ultimately, one is interested in how well the learning algorithm performs on formerly unseen data points $x$, which is called the generalization error.

---

1We will later see a case where $f$ is in fact known.
1.3 Motivation: Approximating functions and derivatives

Despite the current boost of attention, neural networks have been known and studied since the early 1940s (see [39]). For a compact overview about the history of neural networks we refer the interested reader to [21].

From a learning theory point of view, the task of estimating the generalization error decomposes into a statistical problem depending on the samples and an approximation theoretic problem, independent of the samples. A accessible introduction to learning theory from the perspective of approximation theory can be found in [12]. One of the most well-known approximation properties of neural networks shown by Cybenko in [13] and Hornik et al. in [30] is their universal approximation capacity. In detail, any continuous function can be approximated by a shallow neural network up to arbitrary precision on a compact subset in $L^\infty$-norm.

The aim of this work is to study the simultaneous approximation of a function and its derivative with a neural network. There are multiple scenarios in which this is possible and useful.

- In [14], the $j$-th order derivatives of $f$ are incorporated in the empirical loss function (1.1), resulting in

$$\frac{1}{m} \sum_{i=1}^{m} \left[ l(N(x_i|w), f(x_i)) + \sum_{j=1}^{k} l_j \left(D_j^x N(x_i|w), D_j^x f(x_i)\right) \right],$$

which encourages the network to also encode information about the derivatives in its weights. The authors of [14] call this method Sobolev training and reported improved generalization errors and better data-efficiency in a network compression task (see [28]) and in application to synthetic gradients (see [32]). In case of network compression, the approximated function $f$ is a possibly very large neural network $N_{\text{large}}$, that has been trained for some supervised learning task and is learnt by a smaller network $N_{\text{small}}$. In contrast to usual supervised learning settings, the approximated function $f = N_{\text{large}}$ is known and the derivatives can be computed.

- Motivated by the performance of deep learning-based solutions in classical machine learning tasks and, in particular, by their ability to overcome the curse of dimension, neural networks are now also applied for the approximative solution of partial differential equations (PDEs) (see [24, 34, 48, 53]).

In [48] the authors present their deep Galerkin method for approximating solutions of high-dimensional quasilinear parabolic PDEs. For this, a functional $J(f)$ encoding the differential operator, boundary conditions, and initial conditions is introduced. A neural network $N_{\text{PDE}}$ with weights $w$ is then trained to minimize the functional $J(N_{\text{PDE}}(w))$.

This is done by a discretization and randomly sampling spatial points.

The theoretical foundation for approximating a function and higher order derivatives with a neural network was already given in a less known version of the universal approximation theorem by Hornik in Theorem 3 in [29]. In particular, it was shown there that if the activation function $\varphi$ is $k$-times continuously differentiable, non-constant, and bounded, then any $k$-times continuously differentiable function $f$ and its derivatives up to order $k$ can be
uniformly approximated by a shallow neural network on compact sets. Note though that the conditions on the activation function are very restrictive and that, for example, the ReLU is not included in the above result. However, in [14], it was shown that the theorem also holds for shallow ReLU networks if \( k = 1 \). Theorem 3 in [29] was also used in [48] to show the existence of a shallow network approximating solutions of the PDEs considered in this paper.

An important aspect, that is untouched by the previous approximation results is how the complexity of a network and, in particular, its depth relates to its approximation properties.

1.4 Our contribution

In Theorem 1 in [54], Yarotsky showed upper complexity bounds for approximations in \( L^\infty \) norm of functions from the Sobolev space \( W^{n,\infty}((0,1)^d) \) with neural networks for a certain activation function. Precisely, it is shown there, that for any \( \varepsilon > 0 \) and any \( f \in W^{n,\infty}((0,1)^d) \) with \( \| f \|_{W^{n,\infty}((0,1)^d)} \leq 1 \) there exists a neural network \( N_{\varepsilon,f} \) with at most \( c \cdot \log_2(1/\varepsilon) \) layers and at most \( c \cdot \varepsilon^{-d/n} \log_2(1/\varepsilon) \) weights and neurons such that

\[
\| N_{\varepsilon,f} - f \|_{L^\infty((0,1)^d)} \leq \varepsilon,
\]

where \( c \) is a constant depending on \( d \) and \( n \).

In this thesis, we show that the approximation can also be done with respect to higher-order Sobolev norms and that there is a trade-off between the regularity used in the approximation norm and the regularity used in the upper complexity bounds. Specifically, we show that for any approximation accuracy \( \varepsilon > 0 \), regularity \( 0 \leq s \leq 1 \), and any \( f \in W^{n,\infty}((0,1)^d) \) with \( \| f \|_{W^{n,\infty}((0,1)^d)} \leq B \) there is a ReLU neural network \( N_{\varepsilon,f} \) with at most \( c \cdot \log_2(e^{-n/(n-s)}) \) layers and \( c \cdot \varepsilon^{-d/(n-s)} \cdot \log_2(e^{-n/(n-s)}) \) weights and neurons such that

\[
\| N_{\varepsilon,f} - f \|_{W^s,\infty((0,1)^d)} \leq \varepsilon,
\]

where \( c \) is a constant depending on \( d, n, s \), and \( B \). In the boundary case \( s = 0 \) our results corresponds to the theorem shown by Yarotsky and for \( s = 1 \) the network \( N_{\varepsilon,f} \) and its weak gradient uniformly approximate \( f \) and the weak gradient of \( f \), respectively. For \( 0 < s < 1 \) the space \( W^{s,\infty} \) is defined as the interpolation space \( (L^\infty, W^{1,\infty})_{s,\infty} \) which can, in this case, be identified with the Hölder space \( C^{0,s} \). Thus, for non-integer \( s \) the network \( N_{\varepsilon,f} \) uniformly approximates the function \( f \) and its fractional derivative of order \( s \).

1.5 Outline

As a preparation, we start by introducing notation and some definitions that will be used throughout the thesis in Section 2. In Section 3 we rigorously define a neural network in mathematical terms and develop a network calculus. In Section 4 we give a brief introduction to interpolation theory which is used in Section 5 where we define (fractional) Sobolev spaces. In Section 6 we present our main result (Theorem 6.1) which will be discussed in Section 7.
2 Preliminaries

In this subsection, we introduce the basic notation used throughout the thesis and recall some definitions. We set \( N := \{1, 2, \ldots \} \) and \( N_0 := N \cup \{0\} \). For \( k \in N_0 \) the set \( N_{\geq k} := \{k, k+1, \ldots \} \) consists of all natural numbers larger than or equal to \( k \). For a set \( A \) we denote its cardinality by \( |A| \in N \cup \{\infty\} \). If \( x \in \mathbb{R} \), then we write \( |x| := \min\{k \in \mathbb{Z} : k \geq x\} \) where \( \mathbb{Z} \) is the set of integers.

If \( d \in \mathbb{N} \) and \( \|\cdot\| \) is a norm on \( \mathbb{R}^d \), then we denote for \( x \in \mathbb{R}^d \) and \( r > 0 \) by
\[
B_{r,\|\cdot\|}(x) := \{y \in \mathbb{R}^d : \|x - y\| < r\}
\]
the open ball around \( x \) in \( \mathbb{R}^d \) with radius \( r \), where the distance is measured in \( \|\cdot\| \). By \( |x| \) we denote the euclidean norm of \( x \) and by \( \|x\|_{\ell^\infty} := \max_{i=1,\ldots,d} |x_i| \).

We endow \( \mathbb{R}^d \) with the standard topology and for \( A \subset \mathbb{R}^d \) we denote by \( \overline{A} \) the closure of \( A \) and by \( \partial A \) the boundary of \( A \). For the convex hull of \( A \), i.e. the intersection of all convex sets \( C \subset \mathbb{R}^d \) such that \( A \subset C \) we write \( \text{conv} A \). The diameter of a non-empty set \( A \subset \mathbb{R}^d \) is always taken with respect to the euclidean distance, i.e.
\[
\text{diam} A := \text{diam}_{|\cdot|} A := \sup_{x,y \in A} |x - y|.
\]

If \( A, B \subset \mathbb{R}^d \), then we write \( A \subset\subset B \) if \( A \) is compact in \( B \).

For \( d_1, d_2 \in \mathbb{N} \) and a matrix \( A \in \mathbb{R}^{d_1 \times d_2} \) the number of nonzero entries of \( A \) is counted by \( \|\cdot\|_{\ell^0} \), i.e.
\[
\|A\|_{\ell^0} := |\{(i,j) : A_{i,j} \neq 0\}|.
\]

If \( d_1, d_2, d_3 \in \mathbb{N} \) and \( A \in \mathbb{R}^{d_1 \times d_2} \), \( B \in \mathbb{R}^{d_1 \times d_3} \), then we use the common block matrix notation and write for the horizontal concatenation of \( A \) and \( B \)
\[
\begin{bmatrix}
A & B
\end{bmatrix} \in \mathbb{R}^{d_1 \times d_2 + d_3}
\text{ or } \begin{bmatrix}
A \\
B
\end{bmatrix} \in \mathbb{R}^{d_1, d_2 + d_3}.
\]

In the same way, we write for the vertical concatenation of \( A \in \mathbb{R}^{d_1, d_2} \) and \( B \in \mathbb{R}^{d_3, d_2} \)
\[
\begin{bmatrix}
A \\
B
\end{bmatrix} \in \mathbb{R}^{d_1 + d_3, d_2}.
\]

Of course, the same notation also applies to (block) vectors.

For a function \( f : X \to \mathbb{R} \) we denote by
\[
\text{supp} f := \{x \in X : f(x) \neq 0\}
\]
the support of \( f \). If \( f : X \to Y \) and \( g : Y \to Z \) are two functions, then we write for the composition
\[
g \circ f : X \to Z \quad \text{where} \quad (g \circ f)(x) := g(f(x)) \quad \text{for} \ x \in X.
\]
We use the usual multiindex notation, i.e. for $\alpha \in \mathbb{N}_0^d$ we write
\[ |\alpha| := \alpha_1 + \ldots + \alpha_d \quad \text{and} \quad \alpha! := \alpha_1! \cdot \ldots \cdot \alpha_d! \]
Moreover, if $x \in \mathbb{R}^d$, then we have
\[ x^\alpha := \prod_{i=1}^d x_i^{\alpha_i} \]
Let $\Omega \subset \mathbb{R}^d$ be open. We denote the space of bounded functions defined on $\Omega$ by
\[ B(\Omega) := \{ f : \Omega \to \mathbb{R} : f \text{ is bounded} \} \]
with the norm
\[ \|f\|_{\text{sup}} := \sup_{x \in \Omega} |f(x)|. \]
For $n \in \mathbb{N}_0$, we set
\[ C^n(\Omega) := \{ f : \Omega \to \mathbb{R} \mid f \text{ is } n \text{ times continuously differentiable} \} \]
The space of infinitely times continuously differentiable functions on $\Omega$ is denoted by
\[ C^\infty(\Omega) := \bigcap_{n \in \mathbb{N}} C^n(\Omega) \]
and the space of test functions by
\[ C^\infty_c(\Omega) := \{ f \in C^\infty(\Omega) \mid \text{supp } f \subset \subset \Omega \} \]
For $f \in C^n(\Omega)$ and $\alpha \in \mathbb{N}_0^d$ with $|\alpha| \leq n$ we write
\[ D^\alpha f := \frac{\partial^{|\alpha|} f}{\partial x_1^{\alpha_1} \partial x_2^{\alpha_2} \ldots \partial x_d^{\alpha_d}}. \]
We set
\[ C^{n,0}(\overline{\Omega}) := \left\{ f \in C^n(\Omega) : D^\alpha f \text{ is uniformly continuous and bounded } \forall \alpha \in \mathbb{N}_0^d \text{ with } |\alpha| \leq n \right\} \]
and endow it with the norm
\[ \|f\|_{C^{n,0}(\overline{\Omega})} := \max_{\alpha \in \mathbb{N}_0^d : |\alpha| \leq n} \|D^\alpha f\|_{\text{sup}}. \]
If $0 < s \leq 1$, then we say that a function $f : \Omega \to \mathbb{R}^m$ for $\Omega \subset \mathbb{R}^d$ is $s$-Hölder continuous if there is a constant $L > 0$ such that
\[ |f(x) - f(y)| \leq L|x - y|^s \]
for all $x, y \in \Omega$. If $f$ is 1-Hölder continuous, then we call $f$ Lipschitz continuous and if we want to emphasize the constant $L$, then we say that $f$ is $L$-Lipschitz. Note that the constant $L$ is taken with respect to the euclidean distance. Furthermore, we define Hölder spaces as
\[ C^{0,s}(\overline{\Omega}) := \{ f \in C^{0,0}(\overline{\Omega}) : f \text{ is } s\text{-Hölder continuous} \} \]
and
\[ C^{n,s}(\Omega) := \{ f \in C^{n,0}(\Omega) : D^{\alpha} f \in C^{0,s}(\Omega) \text{ for all } \alpha \in \mathbb{N}_0^d \text{ with } |\alpha| = n \} \]
with the norm
\[
\| f \|_{C^{n,s}(\Omega)} := \max \left\{ \| f \|_{C^{n,0}(\Omega)}, \max_{\alpha \in \mathbb{N}_0^d : |\alpha| = n} \sup_{x \neq y, x, y \in \Omega} \frac{|D^{\alpha} f(x) - D^{\alpha} f(y)|}{|x - y|^s} \right\},
\]
where \( n \in \mathbb{N} \) and \( 0 < s \leq 1 \). Note that \( C^{0,1}(\Omega) \) is the space of bounded and Lipschitz continuous functions.

If \( X \) is a linear space and \( \| \cdot \|_1 \) and \( \| \cdot \|_2 \) are two norms on \( X \), then we say that \( \| \cdot \|_1 \) and \( \| \cdot \|_2 \) are \textit{equivalent} if there exist constants \( C_1, C_2 > 0 \) such that
\[
C_1 \| x \|_1 \leq \| x \|_2 \leq C_2 \| x \|_1 \quad \text{for all } x \in X.
\]

For two normed linear spaces \( X, Y \) we denote by \( \mathcal{L}(X,Y) \) the set of bounded linear operators mapping \( X \) to \( Y \) and for \( T \in \mathcal{L}(X,Y) \) the induced operator norm of \( T \) is denoted by
\[
\| T \|_{\mathcal{L}(X,Y)} := \sup \{ \| Tx \|_Y : x \in X, \| x \|_X \leq 1 \}.
\]

Let \( a > 0 \), then we say for two functions \( f : (0,a) \to [0,\infty) \) and \( g : (0,a) \to [0,\infty) \) that \( f = O(g) \) or that \( f \) is in \( O(g) \) if there exists \( 0 < \delta < a \) and \( C > 0 \) such that \( f(\varepsilon) \leq C g(\varepsilon) \) for all \( \varepsilon \in (0,\delta) \).
3 Neural networks

In this section we introduce the notion of neural networks used in this thesis. As in [54],
we will consider a general type of feedforward architecture that also allows for connections
of neurons in non-neighbouring layers. It can be seen, though, that any function realized by such
a network can also be realized by a network with a more restrictive feedforward architecture
where only neurons from neighbouring layers can be connected (Lemma 3.12). As in [41],
we draw a distinction between the neural network (architecture) and the function that
the network realizes. This gives us the possibility to develop a network calculus in the spirit
of [11 Chapter 2]. As in that paper we will introduce the notion of network concatenation
and parallelization.

The following definition is similar to [41, Definition 2.1], where the difference is that we
also allow connections between non-neighbouring layers.

Definition 3.1. Let \( d, L \in \mathbb{N} \). A neural network \( \Phi \) with input dimension \( d \) and \( L \) layers
is a sequence of matrix-vector tuples

\[
\Phi = ((A_1, b_1), (A_2, b_2), \ldots, (A_L, b_L)),
\]

where \( N_0 = d \) and \( N_1, \ldots, N_L \in \mathbb{N} \), and where each \( A_l \) is an \( N_l \times \sum_{k=0}^{l-1} N_k \) matrix, and
\( b_l \in \mathbb{R}^{N_l} \).

If \( \Phi \) is a neural network as above, and if \( \varrho : \mathbb{R} \to \mathbb{R} \) is arbitrary, then we define the
associated realization of \( \Phi \) with activation function \( \varrho \) as the map \( R_{\varrho}(\Phi) : \mathbb{R}^d \to \mathbb{R}^{N_L} \) such
that

\[
R_{\varrho}(\Phi)(x) = x_L,
\]

where \( x_L \) results from the following scheme:

\[
\begin{align*}
x_0 &:= x, \\
x_l &:= \varrho \left( A_l \left[ x_T^0, \ldots, x_T^{l-1} \right]^T + b_l \right), \quad \text{for } l = 1, \ldots, L-1, \\
x_L &:= A_L \left[ x_T^0, \ldots, x_T^{L-1} \right]^T + b_L,
\end{align*}
\]

where \( \varrho \) acts componentwise, i.e., \( \varrho(y) = [\varrho(y^1), \ldots, \varrho(y^m)] \) for \( y = [y^1, \ldots, y^m] \in \mathbb{R}^m \). We
sometimes write \( A_l \) in block-matrix form as

\[
A_l = \left[ \begin{array}{c|c}
A_l,x_0 & \cdots & A_l,x_{l-1} \\
\end{array} \right],
\]

where \( A_{l,k} \) is an \( N_l \times N_k \) matrix for \( k = 0, \ldots, l-1 \) and \( l = 1, \ldots, L \). Then

\[
\begin{align*}
x_l &:= \varrho \left( A_l,x_0 x_0 \ldots + A_l,x_{l-1} x_{l-1} + b_l \right), \quad \text{for } l = 1, \ldots, L-1, \\
x_L &:= A_L,x_0 x_0 \ldots + A_L,x_{L-1} x_{L-1} + b_L.
\end{align*}
\]

We call \( N(\Phi) := d + \sum_{j=1}^{L} N_j \) the number of neurons of the network \( \Phi \), \( L = L(\Phi) \) the
number of layers, and finally \( M(\Phi) := \sum_{j=1}^{L} (\|A_j\|_\infty + \|b_j\|_\infty) \) denotes the number of nonzero
entries of all \( A_l, b_l \) which we call the number of weights of \( \Phi \). Moreover, we refer to \( N_L \) as the
dimension of the output layer of \( \Phi \).
We will now define the class of neural networks where only connections between neighbouring layers are allowed. The following definition is essentially the same as [11, Definition 2.1].

**Definition 3.2.** If \( \Phi = (((A_1, b_1), (A_2, b_2), \ldots, (A_L, b_L)) \) is a neural network as above and we have for
\[ A_l = \begin{bmatrix} A_{l,x_0} & \cdots & A_{l,x_{l-1}} \end{bmatrix}, \]
that \( A_{l,x_i} = 0 \) for \( l = 1, \ldots, L \) and \( i = 0, \ldots, l-2 \), then we call \( \Phi \) a standard neural network. The computation scheme then reduces to the following:

\[
\begin{align*}
x_0 & := x, \\
x_l & := \varphi \left( A_{l,x_{l-1},x_l} + b_l \right), \quad \text{for } l = 1, \ldots, L-1, \\
x_L & := A_{L,x_{L-1},x_L} + b_L.
\end{align*}
\]

In practice, before training a neural network, i.e. adjusting the weights of the network, one has to decide which network architecture to use. The following definition will clarify the notion of a network architecture.

**Definition 3.3.** Let \( L \in \mathbb{N} \) and let \( N_1, \ldots, N_L \in \mathbb{N} \), then we call
\[ \mathcal{A} = (L, (N_1, \ldots, N_L)) \]
a network architecture. If \( \Phi \) is a (standard) neural network such that \( L(\Phi) = L \) and \( N_j(\Phi) = N_j \) for \( j = 1, \ldots, L \), then we say that \( \mathcal{A} \) is the (standard) architecture of the network \( \Phi \).

Moreover, we define one of the in practice commonly used (see [35]) activation functions.

**Definition 3.4.** The function
\[ \varphi : \mathbb{R} \to \mathbb{R}, \quad x \mapsto \max(0, x), \]
is called ReLU (Rectified Linear Unit) activation function.

In [34], Proposition 1] the following lemma is proven which shows that in terms of approximation and upper complexity bounds for networks with continuous and piecewise linear activation functions it suffices to focus on the ReLU.

**Lemma 3.5.** There exists an absolute constant \( c > 0 \) with the following property:

For any continuous piecewise linear activation function \( \hat{\varphi} \) with \( M \) breakpoints, where \( 1 \leq M < \infty \), any neural network \( \Phi \) with \( d \)-dimensional input, with \( L = L(\Phi) \) layers, \( N = N(\Phi) \) neurons and \( M = M(\Phi) \) nonzero weights, and any bounded subset \( \Omega \subset \mathbb{R}^d \), there is a neural network \( \Phi' \) with \( d \)-dimensional input, with \( L \) layers, at most \( cN \) neurons and at most \( cM \) nonzero weights such that
\[ R_{\hat{\varphi}(\hat{\Phi}')}(x) = R_{\hat{\varphi}(\hat{\Phi})}(x) \quad \text{for all } x \in \Omega, \]
where \( \varphi \) is the ReLU activation function.
3.1 Network concatenation

In this subsection we introduce the notion of concatenation of two networks which allows us to realize compositions of network realizations. In detail, if \( \Phi^1 \) and \( \Phi^2 \) are two networks, then we are interested in a network \( \Phi^1 \circ \Phi^2 \) such that \( R_{\rho}(\Phi^1 \circ \Phi^2)(x) = R_{\rho}(\Phi^1) \circ R_{\rho}(\Phi^2)(x) \) for all \( x \in \mathbb{R}^d \). We will start by constructing a network \( \Phi^1 \circ \Phi^2 \) for general activation functions \( \rho \). In the case where the activation function is the ReLU we use the fact, that it is possible to find a network which realizes the identity function and construct a so called sparse concatenation which gives us control over the complexity of the resulting network.

**Definition/Lemma 3.6.** Let \( L_1, L_2 \in \mathbb{N} \) and let

\[
\Phi^1 = ((A^1_1, b^1_1), \ldots, (A^1_{L_1}, b^1_{L_1})), \quad \Phi^2 = ((A^2_1, b^2_1), \ldots, (A^2_{L_2}, b^2_{L_2}))
\]

be two neural networks such that the input layer of \( \Phi^1 \) has the same dimension as the output layer of \( \Phi^2 \). Then, \( \Phi^1 \circ \Phi^2 \) denotes the following \( L_1 + L_2 - 1 \) layer network:

\[
\Phi^1 \circ \Phi^2 := \left( (A^2_1, b^2_1), \ldots, (A^2_{L_2-1}, b^2_{L_2-1}), (\tilde{A}^1_1, \tilde{b}^1_1), \ldots, (\tilde{A}^1_{L_1}, \tilde{b}^1_{L_1}) \right),
\]

where

\[
\tilde{A}^1_l := \left[ A^1_{l,x_0} A^2_{L_2} \middle| A^1_{l,x_1} \middle| \cdots \middle| A^1_{l,x_{L_2-1}} \right] \quad \text{and} \quad \tilde{b}^1_l := A^1_{l,x_0} b^2_{L_2} + b^1_l,
\]

for \( l = 1, \ldots, L_1 \). We call \( \Phi^1 \circ \Phi^2 \) the concatenation of \( \Phi^1 \) and \( \Phi^2 \).

Then, \( R_{\rho}(\Phi^1 \circ \Phi^2) = R_{\rho}(\Phi^1) \circ R_{\rho}(\Phi^2) \).

**Proof.** This can be verified with a direct computation. \( \square \)

Next, we adapt [11] Lemma 2.3] to the network architecture used in this thesis and define a network which realizes an identity function if the activation function is the ReLU.

**Lemma 3.7.** Let \( \rho : \mathbb{R} \to \mathbb{R} \) be the ReLU and \( d \in \mathbb{N} \). We define

\[
\Phi^{Id} := ((A_1, b_1), (A_2, b_2))
\]

with

\[
A_1 := \begin{bmatrix} \text{Id}_{d \times d} \\ -\text{Id}_{d \times d} \end{bmatrix}, \quad b_1 := 0, \quad A_2 := \begin{bmatrix} 0_{d \times d} & \text{Id}_{d \times d} & -\text{Id}_{d \times d} \end{bmatrix}, \quad b_2 := 0.
\]

Then, \( R_{\rho}(\Phi^{Id}) = \text{Id}_{d \times d} \).

Using the identity network \( \Phi^{Id} \) from the previous lemma we can now (as in [11] Definition 2.5)) define an alternative network concatenation which allows us to control the complexity of the resulting network for the case of the ReLU activation function.

**Definition 3.8.** Let \( \rho : \mathbb{R} \to \mathbb{R} \) be the ReLU, let \( L_1, L_2 \in \mathbb{N} \) and let

\[
\Phi^1 = ((A^1_1, b^1_1), \ldots, (A^1_{L_1}, b^1_{L_1})), \quad \Phi^2 = ((A^2_1, b^2_1), \ldots, (A^2_{L_2}, b^2_{L_2}))
\]

be two neural networks such that the input layer of \( \Phi^1 \) has the same dimension as the output layer of \( \Phi^2 \). Let \( \Phi^{Id} \) be as in Lemma 3.7.

Then the *sparse concatenation* of \( \Phi^1 \) and \( \Phi^2 \) is defined as

\[
\Phi^1 \circ \Phi^2 := \Phi^1 \circ \Phi^{Id} \circ \Phi^2.
\]
Remark 3.9. We have
\[
\Phi^1 \circ \Phi^2 = \left((A_1^1, b_1^1), \ldots, (A_{L_1}^1, b_{L_1}^1), \left(\begin{bmatrix} A_{L_2}^1, b_{L_2}^1 \\ -A_{L_2}^2 \end{bmatrix}, \begin{bmatrix} b_{L_2}^2 \\ -b_{L_2}^2 \end{bmatrix}\right), (\tilde{A}_1^1, b_1^1), \ldots, (\tilde{A}_{L_1}^1, b_{L_1}^1)\right),
\]
where
\[
\tilde{A}_l^i := \begin{bmatrix} 0_{N_l^i, \sum_{k=0}^{l-2} N_k^i} & A_{l,x_0}^i & -A_{l,x_0}^i & A_{l,x_1}^i & \cdots & A_{l,x_{l-1}}^i \end{bmatrix},
\]
for \( l = 1, \ldots, L_1 \). Furthermore, it holds that \( L(\Phi^1 \circ \Phi^2) = L_1 + L_2 \), and \( M(\Phi^1 \circ \Phi^2) \leq 2M(\Phi^1) + 2M(\Phi^2) \) and \( N(\Phi^1 \circ \Phi^2) \leq 2N(\Phi^1) + 2N(\Phi^2) \).

3.2 Network parallelization

In some cases it is useful to combine several networks into one larger network. In particular, if \( \Phi^1 \) and \( \Phi^2 \) are two networks, then we are looking for a network \( P(\Phi^1, \Phi^2) \) such that \( R\Phi(P(\Phi^1, \Phi^2))(x) \) for all \( x \in \mathbb{R}^d \).

Definition/Lemma 3.10. Let \( L_1, L_2 \in \mathbb{N} \) and let
\[
\Phi^1 = ((A_1^1, b_1^1), \ldots, (A_{L_1}^1, b_{L_1}^1), \quad \Phi^2 = ((A_1^2, b_1^2), \ldots, (A_{L_2}^2, b_{L_2}^2))
\]
be two neural networks with \( d \)-dimensional input. If \( L_1 \leq L_2 \), then we define
\[
P(\Phi^1, \Phi^2) := \left(\begin{bmatrix} \tilde{A}_1^1, b_1^1 \\ \vdots \\ \tilde{A}_{L_1}^1, b_{L_1}^1\end{bmatrix}\right),
\]
where
\[
\tilde{A}_l := \begin{bmatrix} A_l^0 & A_l^{i,x_0} & 0 & \cdots & A_l^{i,x_{l-1}} & 0 \\ A_l^{i,x_0} & 0 & A_l^{i,x_1} & \cdots & 0 & A_l^{i,x_{l-1}} \end{bmatrix}, \quad \tilde{b}_l := \begin{bmatrix} b_l^1 \\ b_l^2 \end{bmatrix}
\]
for \( 1 \leq l < L_1 \), and
\[
\tilde{A}_l := \begin{bmatrix} A_l^0 & 0 & \cdots & A_l^{i,x_{L_1-1}} & A_l^{i,x_{L_1}} & \cdots & A_l^{i,x_{L_2-1}} \end{bmatrix}, \quad \tilde{b}_l := b_l^2
\]
for \( L_1 \leq l < L_2 \) and
\[
\tilde{A}_{L_2} := \begin{bmatrix} A_{L_2}^0 & A_{L_2}^{i,x_0} & \cdots & A_{L_2}^{i,x_{L_1-1}} & 0 & 0 & \cdots & 0 \\ A_{L_2}^{i,x_0} & 0 & A_{L_2}^{i,x_1} & \cdots & 0 & A_{L_2}^{i,x_{L_1-1}} & \cdots & A_{L_2}^{i,x_{L_2-1}} \end{bmatrix},
\]
\[
\tilde{b}_{L_2} := \begin{bmatrix} b_{L_2}^1 \\ b_{L_2}^2 \end{bmatrix}.
\]
A similar construction is used for the case \( L_1 > L_2 \). Then \( P(\Phi^1, \Phi^2) \) is a neural network with \( d \)-dimensional input and \( \max\{L_1, L_2\} \) layers, which we call the parallelization of \( \Phi^1 \) and \( \Phi^2 \). We have \( M(P(\Phi^1, \Phi^2)) = M(\Phi^1) + M(\Phi^2) \), \( N(P(\Phi^1, \Phi^2)) = N(\Phi^1) + N(\Phi^2) - d \) and
\[
R\Phi(P(\Phi^1, \Phi^2))(x) = (R\Phi(\Phi^1))(x), R\Phi(\Phi^2))(x).
\]
The previous definition can easily be generalized to a parallelization of an arbitrary number of neural networks.

**Remark 3.11.** Let \( n \in \mathbb{N} \) and let \( \Phi^i \) be a neural network for \( i = 1, \ldots, n \) with \( d \)-dimensional input and \( L_i \) layers. The network

\[
P(\Phi^i : i = 1, \ldots, n) := P(\Phi^1, \ldots, \Phi^n) := P(\Phi^1, P(\Phi^2, P(\ldots, P(\Phi^{n-1}, \Phi^n) \ldots)))
\]

with \( d \)-dimensional input and \( L = \max\{L_1, \ldots, L_n\} \) layers is called the parallelization of \( \Phi^1, \Phi^2, \ldots, \Phi^n \). We have

\[
M(P(\Phi^1, \ldots, \Phi^n)) = \sum_{i=1}^{n} M(\Phi^i) \quad \text{and} \quad N(P(\Phi^1, \ldots, \Phi^n)) = \sum_{i=1}^{n} N(\Phi^i) - (n-1)d,
\]

and \( R_\varphi(P(\Phi^1, \ldots, \Phi^n))(x) = (R_\varphi(\Phi^1)(x), \ldots, R_\varphi(\Phi^n)(x)) \).

### 3.3 Standard neural networks

In this subsection, we will show that if a function is a realization of a neural network, then it can also be realized by a standard neural network with similar complexity. The proof of this result is heavily based on the idea of the construction of an identity function with ReLU networks (Lemma 3.7).

**Lemma 3.12.** Let \( \varphi \) be the ReLU. Then there exist absolute constants \( C_1, C_2 > 0 \) such that for any neural network \( \Phi \) with \( d \)-dimensional input and \( L = L(\Phi) \) layers, \( N = N(\Phi) \) neurons and \( M = M(\Phi) \) nonzero weights, there is a standard neural network \( \Phi_{st} \) with \( d \)-dimensional input, and \( L \) layers, at most \( C_1LN \) neurons and \( C_2 \cdot (LN + M) \) nonzero weights such that

\[
R_\varphi(\Phi)(x) = R_\varphi(\Phi_{st})(x)
\]

for all \( x \in \mathbb{R}^d \).

**Proof.** If \( L = 1 \), then there is nothing to show. For \( L > 1 \) we start by defining the matrices

\[
A_{1,x_0}^{st} := \begin{bmatrix}
\text{Id}_{\mathbb{R}^{N_0}} \\
-\text{Id}_{\mathbb{R}^{N_0}} \\
A_{1,x_0}
\end{bmatrix}, \quad b_0^{st} := \begin{bmatrix} 0 \\
0 \\
b_0
\end{bmatrix}
\]

and

\[
A_{l,x_l-1}^{st} := \begin{bmatrix}
\text{Id}_{\mathbb{R}^{2N_0}} & 0 \\
0 & \text{Id}_{\mathbb{R}^{(L-1)N_k}} \\
A_{l,x_0} & -A_{l,x_0} & A_{l,x_1} & \ldots & A_{l,x_{l-1}}
\end{bmatrix}, \quad b_l^{st} := \begin{bmatrix} 0 \\
0 \\
b_l
\end{bmatrix}
\]

for \( l = 1, \ldots, L - 1 \) and

\[
A_{L,x_{L-1}}^{st} := \begin{bmatrix}
A_{L,x_0} & -A_{L,x_0} & A_{L,x_1} & \ldots & A_{L,x_{L-1}}
\end{bmatrix}, \quad b_L^{st} := b_L.
\]
Now we set
\[ A_{st}^l := \begin{bmatrix} 0 & \cdots & 0 & A_{l,x_l-1}^st \end{bmatrix}, \]
and
\[ \Phi_{st} := ((A_{1st},b_{1st}),\ldots,(A_{Lst},b_{Lst})). \]
Then \( \Phi_{st} \) is a standard neural network with \( d \)-dimensional input and \( L \) layers and one can verify that \( R_\varrho(\Phi)(x) = R_\varrho(\Phi_{st})(x) \) for all \( x \in \mathbb{R}^d \). For the number of neurons in the \( l \)-th layer \( N_{st}^l \) we have \( N_{st}^l = 2N_0 + \sum_{k=1}^{l} N_l \leq 2N \). Hence, \( N(\Phi_{st}) \leq 2LN \). To estimate the number of weights, note that \( \|A_{st}^l\|_\varrho \leq 2N + 2\|A_l\|_\varrho \) and \( \|b_{st}^l\|_\varrho = \|b_l\|_\varrho \), from which we conclude that \( N(\Phi_{st}) \leq 2(LN + M) \).

In the next remark we collect some properties of the class of functions that can be realized by a neural network.

**Remark 3.13.** If \( \Phi \) is a neural network with \( d \)-dimensional input and \( m \)-dimensional output and \( \varrho \) is the ReLU, then \( R_\varrho(\Phi) \) is a Lipschitz-continuous, piecewise affine-linear function. This follows easily from Definition 3.2 and Lemma 3.12 which show that \( R_\varrho(\Phi) \) can be expressed as the composition of Lipschitz-continuous, piecewise affine functions.

In [3, Theorem 2.1] also the converse was shown, i.e. every piecewise affine-linear function can be realized by a neural network with the ReLU as activation function.
4 Interpolation spaces

In this section, we will give a short introduction to interpolation theory in Banach spaces. We will only cover the necessary definitions and pick from the wide theory the results that are used in this thesis. In particular, we present the K-method of real interpolation.

For two Banach spaces $B_0$ and $B_1$ interpolation techniques allow the definition and study a family of spaces that in some sense bridge the gap between $B_0$ and $B_1$. This rather abstract concept will be applied to Sobolev spaces in Section 5.4 to derive spaces of fractional regularity. Interpolation theory is a valuable tool in harmonic analysis and the study of partial differential equations. A small summary about the history of the development of the theory of interpolation spaces can be found in [51, Lecture 21]. Interested readers are recommended to read the monographs [6, 36, 51] and [52] for a detailed treatment in the context of partial differential equations.

The basic notions of functional analysis are assumed to be known and we refer to [55] for an introduction.

If $B_0, B_1$ are two Banach spaces such that $B_1$ is continuously embedded in $B_0$, then we write $B_1 \subset B_0$ and call $(B_0, B_1)$ an interpolation couple. One can also consider more general pairs of Banach spaces but this setting is enough for our purposes.

**Definition 4.1.** Let $(B_0, B_1)$ be an interpolation couple. For every $u \in B_1$ and $t > 0$ we define

$$K(t, u, B_0, B_1) := \inf_{v \in B_1} \left( \|u - v\|_{B_0} + t\|v\|_{B_1} \right).$$

The first term measures how well $u$ can be approximated by elements from the smaller space $B_1$ in $\|\cdot\|_{B_0}$ and the second term is a penalty term weighted by $t$.

Now we can define interpolation spaces.

**Definition 4.2.** Let $(B_0, B_1)$ be an interpolation couple. Let $0 < \theta < 1$ and $1 \leq p \leq \infty$. We set

$$\|u\|_{(B_0, B_1)_{\theta,p}} := \begin{cases} \left( \int_0^\infty t^{-\theta p} K(t, u, B_0, B_1)^p \frac{dt}{t} \right)^{1/p}, & \text{for } 1 \leq p < \infty \\ \sup_{0 < t < \infty} t^{-\theta} K(t, u, B_0, B_1), & \text{for } p = \infty. \end{cases}$$

Moreover, we define the set

$$(B_0, B_1)_{\theta,p} := B_{\theta,p} := \left\{ u \in B_0 : \|u\|_{(B_0, B_1)_{\theta,p}} < \infty \right\}.$$

The mapping $u \mapsto \|u\|_{(B_0, B_1)_{\theta,p}}$ turns $(B_0, B_1)_{\theta,p}$ into a Banach space (cf. [36, Proposition 1.5]) which is called a real interpolation space.

**Remark 4.3.** Note that if $\|\cdot\|_{B_i}^{(1)}$ and $\|\cdot\|_{B_i}^{(2)}$ are two equivalent norms on $B_i$ for $i = 0, 1$, then

$$(B_0, \|\cdot\|_{B_0}^{(1)}), (B_1, \|\cdot\|_{B_1}^{(1)})_{\theta,p} = (B_0, \|\cdot\|_{B_0}^{(2)}), (B_1, \|\cdot\|_{B_1}^{(2)})_{\theta,p}$$

with equivalence of the respective norms.
There exist many results relating interpolation spaces to each other from which we only mention a few. The next lemma shows that in the sense of a continuous embedding the spaces $B_{\theta,p}$ form a family of nested Banach spaces.

**Lemma 4.4.** Let $1 \leq p \leq \infty$, then the following holds:

(i) If $(B_0, B_1)$ is an interpolation couple and $0 < \theta_1 < \theta_2 < 1$, then

$$B_1 \subset B_{\theta_2,p} \subset B_{\theta_1,p} \subset B_0;$$

(ii) if $B$ is a Banach space and $0 < \theta < 1$, then $(B,B)_{\theta,p} = B$.

**Proof.** For (i) see [36, p. 2] and for (ii) [36, Proposition 1.4].

One of the most important theorems in interpolation theory shows how the norm of an operator defined on interpolation couples relates to the operator norm with respect to the corresponding interpolation spaces. The theorem can be found in [36, Theorem 1.6].

**Theorem 4.5.** Let $(A_0, A_1)$ and $(B_0, B_1)$ be two interpolation couples. If $T \in \mathcal{L}(A_0, B_0) \cap \mathcal{L}(A_1, B_1)$, then $T \in \mathcal{L}(A_{\theta,p}, B_{\theta,p})$ for every $0 < \theta < 1$ and $1 \leq p \leq \infty$. Furthermore,

$$\|T\|_{\mathcal{L}(A_{\theta,p}, B_{\theta,p})} \leq \|T\|^{1-\theta}_{\mathcal{L}(A_0, B_0)} \cdot \|T\|^\theta_{\mathcal{L}(A_1, B_1)} \quad (4.1)$$

As a corollary from the previous theorem the following useful corollary can be obtained (see [36, Corollary 1.7]).

**Corollary 4.6.** Let $(B_0, B_1)$ be an interpolation couple. Moreover, let $0 < \theta < 1$ and $1 \leq p \leq \infty$, then there is a constant $c = c(\theta,p) > 0$ such that for all $u \in B_1$ we have

$$\|u\|_{B_{\theta,p}} \leq c \|u\|^{1-\theta}_{B_0} \cdot \|u\|^\theta_{B_1}. $$

For the case $p = \infty$ we have $c(\theta, \infty) = 1$. 

16
5 Sobolev spaces

In this section, we give a brief introduction to Sobolev spaces and present related concepts which are important for the proof of our main result in Section 6. In detail, we show estimates for compositions and products of functions from certain Sobolev spaces in Subsection 5.1 and 5.2 respectively. In Subsection 5.3 we introduce a generalization of Taylor polynomials which is one of the key tools in the proof of our main result. In the last subsection, we apply Banach space interpolation theory to (integer) Sobolev spaces in order to define fractional Sobolev spaces.

Spaces of functions that admit generalized derivatives fulfilling suitable integrability properties are a crucial concept in modern theory of partial differential equations (cf. e.g. [1, 19, 42]). In order to study properties of PDEs using functional analytic tools, a differential equation is reformulated via a differential operator mapping one function space to another. For a wide range of differential equations the appropriate spaces in this formulation are Sobolev spaces. A historical background of the development of Sobolev spaces in the context of PDEs can be found in [51, Lecture 1]. For a detailed treatment of the broad theory of Sobolev spaces we refer the reader to [1, 8, 19].

For this entire section, let \( d \in \mathbb{N} \) and, if not stated otherwise, \( \Omega \subset \mathbb{R}^d \) denote an open subset of \( \mathbb{R}^d \). We start by introducing the notion of \( L^p \) spaces which form the basis for defining Sobolev spaces. The reader is assumed to be familiar with the basic concepts of measure and integration theory and we refer to [43] for an in-depth treatment. The following definition can, for example, be found in [8, Chapter 4].

**Definition 5.1 (Lebesgue space).** We set

\[
L^p(\Omega) := \left\{ f : \Omega \to \mathbb{R} \mid f \text{ is measurable and } \int_\Omega |f|^p \, dx < \infty \right\},
\]

with

\[
\|f\|_{L^p(\Omega)} := \left( \int_\Omega |f|^p \, dx \right)^{1/p}
\]

for \( 1 \leq p < \infty \), and

\[
L^\infty(\Omega) := \left\{ f : \Omega \to \mathbb{R} \mid f \text{ is measurable and exists a constant } C \text{ such that } |f(x)| \leq C \text{ a.e. on } \Omega \right\}
\]

with

\[
\|f\|_{L^\infty(\Omega)} := \text{ess sup}_{x \in \Omega} |f(x)| := \inf\{C \geq 0 : |f(x)| \leq C \text{ a.e. on } \Omega\}.
\]

As is common, we identify functions that are equal a.e. on \( \Omega \), so that \( \|\cdot\|_{L^p(\Omega)} \) is a norm on \( L^p(\Omega) \) and turns it into a Banach space which is called a *Lebesgue space*. In a strict sense, \( L^p(\Omega) \) comprises equivalence classes of functions. However, most of the time it is convenient to ignore this technicality. We frequently say that \( f \) has property \( A \), if there is a representative of \( f \) that has property \( A \). For example, we say that \( f \) is continuous, if there is a continuous representative of \( f \). Now, we define Sobolev spaces and weak derivatives up to order \( n \), which generalize the concept of classical differentiability to the setting of Lebesgue spaces and \( n \) can be seen as a means to measure the regularity of a function. As a reference see [8, Chapter 9].
**Definition 5.2** (Sobolev space). Let \( n \in \mathbb{N}_0 \) and \( 1 \leq p \leq \infty \). Then we define

\[
W^{n,p}(\Omega) := \left\{ f \in L^p(\Omega) \mid \forall \alpha \in \mathbb{N}_d^0 \text{ with } |\alpha| \leq n, \exists g_\alpha \in L^p(\Omega) \text{ such that } \int_\Omega f D^\alpha \phi dx = (-1)^{|\alpha|} \int_\Omega g_\alpha \phi dx \ \forall \phi \in C_c^\infty(\Omega) \right\}.
\]

For \( f \in W^{n,p}(\Omega) \) and \( \alpha \in \mathbb{N}_d^0 \) with \(|\alpha| \leq n\) we set

\[
D^\alpha f := g_\alpha
\]

and call it the weak derivative of \( f \) of order \( \alpha \).

Furthermore, for \( f \in W^{n,p}(\Omega) \) and \( 1 \leq p < \infty \) we define the norm

\[
\|f\|_{W^{n,p}(\Omega)} := \left( \sum_{0 \leq |\alpha| \leq n} \|D^\alpha f\|_{L^p(\Omega)}^p \right)^{1/p}
\]

and

\[
\|f\|_{W^{n,\infty}(\Omega)} := \max_{0 \leq |\alpha| \leq n} \|D^\alpha f\|_{L^\infty(\Omega)}.
\]

It can be shown that the weak derivative \( D^\alpha f \) is unique (see [8, Corollary 4.24]) and it is easy to see that \( D^\alpha : W^{n,p}(\Omega) \to L^p(\Omega) \) is linear, i.e. if \( f, g \in W^{n,p}(\Omega) \), and \( \lambda \in \mathbb{R} \), and \( \alpha \in \mathbb{N}_d^0 \) such that \(|\alpha| \leq n\) then

\[
D^\alpha (f + \lambda g) = D^\alpha f + \lambda D^\alpha g.
\]

We write \( \nabla f := (D^1 f, \ldots, D^d f) \). Moreover, the space \( W^{n,p}(\Omega) \) equipped with the norm \( \|\cdot\|_{W^{n,p}(\Omega)} \) is a Banach space. For \( f \in W^{n,p}(\Omega) \) we shall use the notation

\[
\|f\|_{W^{n,p}(\Omega)} \text{ if } \|f\|_{W^{n,p}(\Omega)} = \|f\|_{W^{n,p}(\Omega)} = \|f(x)\|_{W^{n,p}(\Omega; dx)}
\]

if the domain is clear from the context, or if we want to emphasize the variable \( x \) that the function \( f \) depends on, respectively.

As a generalization of the previous definition, we can now also introduce Sobolev spaces of vector-valued functions (see [42, Chapter 1.2.3]).

**Definition 5.3.** We set for \( n \in \mathbb{N}_0, m \in \mathbb{N} \) and \( 1 \leq p \leq \infty \)

\[
W^{n,p}(\Omega; \mathbb{R}^m) := \{(f_1, \ldots, f_m) : f_i \in W^{n,p}(\Omega)\}
\]

with

\[
\|f\|_{W^{n,p}(\Omega; \mathbb{R}^m)} := \left( \sum_{i=1}^m \|f_i\|_{W^{n,p}(\Omega)}^p \right)^{1/p} \text{ for } 1 \leq p < \infty
\]

and

\[
\|f\|_{W^{n,\infty}(\Omega; \mathbb{R}^m)} := \max_{i=1,\ldots,m} \|f_i\|_{W^{n,\infty}(\Omega)}.
\]
These spaces are again Banach spaces. To simplify the exposition, we introduce notation for a family of semi-norms.

**Definition 5.4** (Sobolev semi-norm). For \( n, k \in \mathbb{N}_0 \) with \( k \leq n \), \( m \in \mathbb{N} \) and \( 1 \leq p \leq \infty \) we define for \( f \in W^{n,p}(\Omega; \mathbb{R}^m) \) the Sobolev semi-norm

\[
|f|_{W^{k,p}(\Omega; \mathbb{R}^m)} := \left( \sum_{i=1}^{m} \|D^{\alpha} f_i\|_{L^p(\Omega)}^p \right)^{1/p}
\]

for \( 1 \leq p < \infty \) and

\[
|f|_{W^{k,\infty}(\Omega; \mathbb{R}^m)} := \max_{i=1}^{m} \|D^{\alpha} f_i\|_{L^\infty(\Omega)}
\]

For \( m = 1 \) we only write \(|·|_{W^{k,p}(\Omega)}\). Note that \(|·|_{W^{0,p}(\Omega)}\) corresponds to \(|·|_{L^p(\Omega)}\).

Many results about function spaces defined on a domain \( \Omega \) require \( \Omega \) to fulfill certain regularity conditions. Different geometrical conditions and the resulting properties have been intensively studied and can for example be found in [1]. For our purposes it is enough to focus on the condition introduced in the next definition which can be found in [19, Appendix C.1].

**Definition 5.5** (Lipschitz-domain). We say that a bounded and open set \( \Omega \subset \mathbb{R}^d \) is a Lipschitz-domain if for each \( x_0 \in \partial \Omega \) there exists \( r > 0 \) and a Lipschitz continuous function \( g : \mathbb{R}^d - 1 \rightarrow \mathbb{R} \) such that

\[
\Omega \cap B_{r,\{1\}}(x_0) = \{ x \in B_{r,\{1\}}(x_0) : x_d > g(x_1, \ldots, x_{d-1}) \}
\]

after possibly relabeling and reorienting the coordinate axes.

In this thesis, we will only encounter convex domains \( \Omega \), which can be shown to have some regularity (see [23, Corollary 1.2.2.3]).

**Lemma 5.6.** Let \( \Omega \subset \mathbb{R}^d \) be open, bounded, and convex, then \( \Omega \) is a Lipschitz-domain.

A fundamental class of results in the theory of function spaces are extension theorems. Properties of function spaces defined on the whole \( \mathbb{R}^d \) can be transferred to spaces defined on a domain \( \Omega \subset \mathbb{R}^d \), if \( \Omega \) admits an extension.

**Theorem 5.7** (extension operator). Let \( \Omega \subset \mathbb{R}^d \) be a Lipschitz-domain and \( 1 \leq p \leq \infty \). Then there exists a bounded linear operator \( E \) such that for any \( n \in \mathbb{N}_0 \)

\[
E : W^{n,p}(\Omega) \rightarrow W^{n,p}(\mathbb{R}^d)
\]

with \( Ef = f \) a.e. on \( \Omega \) and \( \|Ef\|_{W^{n,p}(\mathbb{R}^d)} \leq C_E \|f\|_{W^{n,p}(\Omega)} \) for all \( f \in W^{n,p}(\Omega) \) where \( C_E = C_E(d, n, p, \Omega) \).

A proof can be found in [19, Theorem VI.3.1.5].

Rademacher’s Theorem states that a Lipschitz continuous function \( f : \Omega \rightarrow \mathbb{R} \) is differentiable (in the classical sense) almost everywhere in \( \Omega \) (see e.g. [19, Theorem 5.8.6]). The following theorem yields an alternative characterization for the space \( W^{1,\infty} \) using Lipschitz continuous functions and connects the notion of the classical and the weak derivative.
Theorem 5.8. Let $\Omega \subset \mathbb{R}^d$ be open and convex. If $f \in W^{1,\infty}(\Omega)$, then $f$ is Lipschitz. Conversely, if $f$ is bounded and $L$-Lipschitz on $\Omega$, then $f \in W^{1,\infty}(\Omega)$, the weak gradient of $f$ agrees almost everywhere with the classical gradient, and

$$\|\nabla f(x)\|_{L^\infty(\Omega;dx)} \leq L.$$ (5.2)

A proof can be found in [20, Theorem 4.1] together with [20, Remark 4.2].

Remark 5.9. Note that the previous theorem gives us a convenient way to compute the weak derivative of a function $f \in W^{1,\infty}$ using the almost everywhere existing classical derivative of the Lipschitz continuous representative of $f$.

Corollary 5.10. If $\Omega \subset \mathbb{R}^d$ is open and convex, then

$$W^{1,\infty}(\Omega) = C^{0,1}(\Omega)$$

with equivalent norms.\(^1\)

The next remark generalizes the previous corollary and yields an equivalent characterization of $W^{n,\infty}$ for convex domains (see [1] Chapter 1.3).

Remark 5.11. If $\Omega \subset \mathbb{R}^d$ is open and convex, and $n \in \mathbb{N}$, then we have

$$W^{n,\infty}(\Omega) = C^{n-1,1}(\Omega),$$

with equivalence of the respective norms.

5.1 Composition estimate

In this subsection, we derive an estimate for the semi-norm $| \cdot |_{W^{1,\infty}}$ of the composition $g \circ f$ of two functions $f \in W^{1,\infty}(\Omega; \mathbb{R}^m)$ and $g \in W^{1,\infty}(\mathbb{R}^m)$. Since in general this is not a well-defined notion we need to clarify what $g \circ f$ denotes in this case. To demonstrate the problem let $f$ and $g$ be as above and recall that $f$ and $g$ are equivalence classes of functions. Let now $\hat{f}$ be a representative for the equivalence class $f$ and $\hat{g}_1, \hat{g}_2$ be two representatives for $g$. Then $\hat{g}_1 = \hat{g}_2$ almost everywhere but in general we do not have $\hat{g}_1 \circ \hat{f} = \hat{g}_2 \circ \hat{f}$ almost everywhere. To circumvent this problem we denote by $g \circ f$ the equivalence class of $\hat{g} \circ \hat{f}$ where $\hat{g}$ and $\hat{f}$ are Lipschitz continuous representatives of $g$ and $f$, respectively.

The next lemma recalls a well known fact about the composition of Lipschitz functions.

Lemma 5.12. Let $d, m \in \mathbb{N}$ and $\Omega_1 \subset \mathbb{R}^d$, $\Omega_2 \subset \mathbb{R}^m$. Let $f : \Omega_1 \to \Omega_2$ be a $L_1$-Lipschitz function and $g : \Omega_2 \to \mathbb{R}$ be a $L_2$-Lipschitz function for some $L_1, L_2 > 0$. Then the composition $g \circ f : \Omega_1 \to \mathbb{R}$ is a $(L_1 \cdot L_2)$-Lipschitz function.

\(^1\)The above equality holds after identifying each equivalence class with its Lipschitz continuous representative.
Proof. Let \( x, y \in \Omega_1 \) then

\[
|g(f(x)) - g(f(y))| \leq L_2|f(x) - f(y)| \leq L_2L_1|x - y|,
\]

which proves the claim.

The following corollary introduces a chain rule estimate for \( W^{1,\infty} \).

Corollary 5.13. Let \( d, m \in \mathbb{N} \) and \( \Omega_1 \subset \mathbb{R}^d, \Omega_2 \subset \mathbb{R}^m \) both be open, bounded, and convex. Then, there is a constant \( C = C(d, m) > 0 \) with the following property:

If \( f \in W^{1,\infty}(\Omega_1; \mathbb{R}^m) \) and \( g \in W^{1,\infty}(\Omega_2) \) are Lipschitz continuous representatives such that \( \text{ran } f \subset \Omega_2 \), then for the composition \( g \circ f \) it holds that \( g \circ f \in W^{1,\infty}(\Omega_1) \) and we have

\[
|g \circ f|_{W^{1,\infty}(\Omega_1)} \leq C|g|_{W^{1,\infty}(\Omega_2)}|f|_{W^{1,\infty}(\Omega_1; \mathbb{R}^m)}
\]

and

\[
\|g \circ f\|_{W^{1,\infty}(\Omega_1)} \leq C \max \left\{ \|g\|_{L^\infty(\Omega_2)}, |g|_{W^{1,\infty}(\Omega_2)}|f|_{W^{1,\infty}(\Omega_1; \mathbb{R}^m)} \right\}.
\]

Proof. We set for \( j = 1, \ldots, m \)

\[
L_j := \| \nabla f_j \|_{L^\infty(\Omega_1)} \quad \text{and} \quad L_f := (L_1, \ldots, L_m).
\]

It then follows from Theorem 5.8 that \( f_j \) is \( L_j \)-Lipschitz and hence, \( f \) is \( |L_f| \)-Lipschitz. In the same manner we define

\[
L_g := \| \nabla g \|_{L^\infty(\Omega_2)}
\]

and have that \( g \) is \( L_g \)-Lipschitz. With Lemma 5.12 we conclude that also the composition \( g \circ f \) is \( (|L_f| \cdot L_g) \)-Lipschitz. Since \( g \) is bounded and hence also \( g \circ f \) applying Theorem 5.8 again yields that \( g \circ f \in W^{1,\infty}(\Omega_1) \) and furthermore

\[
\| \nabla (g \circ f) \|_{L^\infty(\Omega_1)} \leq |L_f| \cdot L_g.
\]

We get

\[
|g \circ f|_{W^{1,\infty}(\Omega_1)} \leq \| \nabla (g \circ f) \|_{L^\infty(\Omega_1)} \\
\leq |L_f| \cdot L_g \\
\leq \sqrt{m}|L_f|_{L^\infty} \cdot \sqrt{m}|g|_{W^{1,\infty}(\Omega_2)} \\
\leq \sqrt{dm}|f|_{W^{1,\infty}(\Omega_1; \mathbb{R}^m)} \cdot |g|_{W^{1,\infty}(\Omega_2)},
\]

where we used Equation (5.2) in the first step and the previous estimate (5.3) in the second. The third step is due to estimating the \( \ell^2 \) norm with the \( \ell^\infty \) norm on \( \mathbb{R}^m \), and applying (5.2) to the definition of \( L_g \). The last step uses Equation (5.2) again. This shows the first estimate. The second estimate is now an easy consequence of the first.
5.2 Product estimate

The following lemma shows that under some conditions the product rule of differentiation also holds for weak derivatives.

**Lemma 5.14** (differentiation of a product). Let \( 1 \leq p \leq \infty \) and \( f, g \in W^{1,p}(\Omega) \cap L^\infty(\Omega) \). Then \( fg \in W^{1,p}(\Omega) \cap L^\infty(\Omega) \) and
\[
D_i(fg) = (D_i f)g + f(D_i g)
\]
for \( i = 1, 2, \ldots, d \).

We refer to [8, Proposition 9.4] for a proof.

As a direct consequence of the previous Lemma, we obtain an estimate for the semi-norm of a product of functions in \( W^{1,\infty}(\Omega) \).

**Corollary 5.15.** Let \( f, g \in W^{1,\infty}(\Omega) \). Then \( fg \in W^{1,\infty}(\Omega) \) and
\[
|fg|_{W^{1,\infty}(\Omega)} \leq |f|_{W^{1,\infty}(\Omega)} \|g\|_{L^\infty(\Omega)} + \|f\|_{L^\infty(\Omega)} |g|_{W^{1,\infty}(\Omega)}
\]

**Proof.** From Lemma 5.14 it follows that \( fg \in W^{1,\infty}(\Omega) \) and for the semi-norm we have
\[
|fg|_{W^{1,\infty}(\Omega)} = \max_{i=1,\ldots,d} \|(D_i^f)g + f(D_i^g)\|_{L^\infty(\Omega)}
\leq \max_{i=1,\ldots,d} \left( \|D_i^f\|_{L^\infty(\Omega)} \|g\|_{L^\infty(\Omega)} + \|f\|_{L^\infty(\Omega)} \|D_i^g\|_{L^\infty(\Omega)} \right)
\leq |f|_{W^{1,\infty}(\Omega)} \|g\|_{L^\infty(\Omega)} + \|f\|_{L^\infty(\Omega)} |g|_{W^{1,\infty}(\Omega)},
\]
where we used the product rule from Lemma 5.14 for the first step.

5.3 Averaged Taylor polynomial

In this subsection we develop a polynomial approximation in the spirit of Taylor polynomials but appropriate for Sobolev spaces. A reference for this entire subsection is [7, Chapter 4.1].

**Definition 5.16** (averaged Taylor polynomial). Let \( n \in \mathbb{N} \), \( 1 \leq p \leq \infty \) and \( f \in W^{n-1,p}(\Omega) \), and let \( x_0 \in \Omega \), \( r > 0 \) such that for the ball \( B := B_{r,\|x_0\|} \) it holds that \( B \subset \subset \Omega \). The corresponding **Taylor polynomial of order** \( n \) **of** \( f \) **averaged over** \( B \) **is defined for** \( x \in \Omega \) **as**
\[
Q^nf(x) := \int_B T^n_y f(x) \phi(y) dy,
\]
where
\[
T^n_y f(x) := \sum_{|\alpha| \leq n-1} \frac{1}{\alpha!} D^\alpha f(y)(x-y)^\alpha
\]
and \( \phi \) is an arbitrary cut-off function supported in \( \overline{B} \), i.e.
\[
\phi \in C_c^\infty(\mathbb{R}^d) \quad \text{with} \quad \phi(x) \geq 0 \text{ for all } x \in \mathbb{R}^d, \quad \supp \phi = \overline{B} \quad \text{and} \quad \int_{\mathbb{R}^d} \phi(x) dx = 1.
\]
A cut-off function as used in the previous definition always exists. A possible choice is
\[
\psi(x) = \begin{cases}
  e^{-(1-||x-x_0||/r)^2}, & \text{if } ||x-x_0|| < r \\
  0, & \text{else}
\end{cases}
\]
normalized by \(\int_{\mathbb{R}^d} \psi(x)dx\). Next, we derive some properties of the averaged Taylor polynomial and start with a remark.

**Remark 5.17.** From the linearity of the weak derivative (cf. Equation (5.1)) we can easily conclude that the averaged Taylor polynomial is linear in \(f\).

Recall that so far the averaged Taylor polynomial is defined via an integral and some cut-off function (cf. (5.4)) that perform an averaging of a polynomial expression (cf. (5.5)). The following lemma shows that the averaged Taylor polynomial of order \(n\) is actually a polynomial of degree less than \(n\) in \(x\).

**Lemma 5.18.** Let \(n \in \mathbb{N}, 1 \leq p \leq \infty\) and \(f \in W^{n-1,p}(\Omega)\), and let \(x_0 \in \Omega, r > 0, R \geq 1\) such that for the ball \(B := B_r, ||x_0||(x_0)\) it holds that \(B \subset \subset \Omega\) and \(B \subset B_{R||\cdot||_\infty}(0)\). Then the Taylor polynomial of order \(n\) of \(f\) averaged over \(B\) can be written as
\[
Q^nf(x) = \sum_{|\alpha| \leq n-1} c_\alpha x^\alpha
\]
for \(x \in \Omega\).

Moreover, if \(f \in W^{n-1,\infty}(\Omega)\), then there exists a constant \(c = c(n, R) > 0\) such that the coefficients \(c_\alpha\) are bounded with \(|c_\alpha| \leq c\|f\|_{W^{n-1,\infty}(\Omega)}\) for all \(\alpha\) with \(|\alpha| \leq n - 1\).

**Proof.** The first part of the proof of this lemma follows closely the chain of arguments in [7 4.1.5 - 4.1.9]. We write for \(f \in \mathbb{N}^0_0\)
\[
(x - y)^\alpha = \prod_{i=1}^d (x_i - y_i)^{\alpha_i} = \sum_{\gamma, \beta \in \mathbb{N}^d_0, \gamma + \beta = \alpha} a_{(\gamma, \beta)} x^\gamma y^\beta,
\]
where \(a_{(\gamma, \beta)} \in \mathbb{R}\) are suitable constants with
\[
|a_{(\gamma, \beta)}| \leq \binom{\gamma + \beta}{\gamma} = \frac{(\gamma + \beta)!}{\gamma! \beta!} (5.6)
\]
in multi-index notation. Then, combining Equation (5.4) and (5.5) yields
\[
Q^n f(x) = \sum_{|\alpha| \leq n-1} \sum_{\gamma + \beta = \alpha} \frac{1}{\alpha!} a_{(\gamma, \beta)} x^\gamma \int_B D^\alpha f(y) y^\beta \phi(y)dy
\]
\[
= \sum_{|\gamma| \leq n-1} x^\gamma \sum_{|\gamma + \beta| \leq n-1} \frac{1}{(\gamma + \beta)!} a_{(\gamma, \beta)} \int_B D^{\gamma + \beta} f(y) y^\beta \phi(y)dy.
\]

For the second part, note that
\[
\left| \int_B D^{\gamma + \beta} f(y) y^\beta \phi(y) dy \right| \leq \int_B \left| D^{\gamma + \beta} f(y) \right| |y^\beta| \phi(y) dy \\
\leq \|f\|_{W^{n-1,\infty}(\Omega)} R^{\beta} \int_B \phi(y) dy \\
\leq R^{n-1} \|f\|_{W^{n-1,\infty}(\Omega)}, \tag{5.7}
\]
where we used the non-negativity of \(\phi\) in the first step, \(B \subset \mathbb{B}_R\|\cdot\|_{\infty}(0)\) in the second step and \(\int_B \phi(y) dy = 1\) in the last step. To estimate the absolute value of the coefficients \(c_\gamma\), we have
\[
|c_\gamma| \leq \sum_{|\gamma + \beta| \leq n-1} \frac{1}{(\gamma + \beta)_+ !} \left| \int_B D^{\gamma + \beta} f(y) y^\beta \phi(y) dy \right| \\
\leq R^{n-1} \|f\|_{W^{n-1,\infty}(\Omega)} \leq c \|f\|_{W^{n-1,\infty}(\Omega)}.
\]
Here, the second step used Equation (5.7) together with Equation (5.6), and \(c = c(n,R) > 0\) is a constant.

The next step is to derive approximation properties of the averaged Taylor polynomial. To this end, recall that for the (standard) Taylor expansion of some function \(f\) defined on a domain \(\Omega\) in \(x_0\) to yield an approximation at some point \(x_0 + th\) for \(0 \leq t \leq 1\) has to be contained in \(\Omega\) (see [38, Theorem 6.8.10]). In case of the averaged Taylor polynomial the expansion point \(x_0\) is replaced by a ball \(B\) and we require that the path between each \(x_0 \in B\) and each \(x \in \Omega\) is contained in \(\Omega\). This geometrical condition is made precise in the following definition.

**Definition 5.19.** Let \(\Omega, B \subset \mathbb{R}^d\). Then \(\Omega\) is called star-shaped with respect to \(B\) if,
\[
\overline{\text{conv} (\{x\} \cup B)} \subset \Omega \quad \text{for all } x \in \Omega.
\]

The next definition introduces a geometric notion which becomes important when given a family of subdivisions \(T^h\), \(0 < h \leq 1\) of a domain \(\Omega\) which becomes finer for smaller \(h\). One typically needs to control the nondegeneracy of \((T^h)_h\) which can be done e.g. with a uniformly bounded chunkiness parameter.

**Definition 5.20.** Let \(\Omega \subset \mathbb{R}^d\) be bounded. We define the set
\[
\mathcal{R} := \left\{ r > 0 : \text{ there exists } x_0 \in \Omega \text{ such that } \Omega \text{ is star-shaped with respect to } B_{r,\|\cdot\|}(x_0) \right\}.
\]
If \(\mathcal{R} \neq \emptyset\), then we define
\[
r_{\max}^* := \sup \mathcal{R} \quad \text{and call } \gamma := \frac{\text{diam}(\Omega)}{r_{\max}^*}
\]
the chunkiness parameter of \(\Omega\).
To emphasize the dependence on the set \( \Omega \), we will occasionally write \( r^*_{\text{max}}(\Omega) \) and \( \gamma(\Omega) \).

Finally, the next lemma shows approximation properties of the averaged Taylor polynomial.

**Lemma 5.21 (Bramble-Hilbert).** Let \( \Omega \subset \mathbb{R}^d \) be open and bounded, \( x_0 \in \Omega \) and \( r > 0 \) such that \( \Omega \) is star-shaped with respect to \( B := B_r(x_0) \), and \( r > (1/2)r^*_{\text{max}} \). Moreover, let \( n \in \mathbb{N} \), \( 1 \leq p \leq \infty \) and denote by \( \gamma \) the chunkiness parameter of \( \Omega \). Then there exists a constant \( C = C(n, d, \gamma) > 0 \) such that for all \( f \in W^{n,p}(\Omega) \)

\[
|f - Q^nf|_{W^{n,p}(\Omega)} \leq Ch^n|f|_{W^{n,p}(\Omega)} \quad \text{for } k = 0, 1, \ldots, n,
\]

where \( Q^nf \) denotes the Taylor polynomial of order \( n \) of \( f \) averaged over \( B \) and \( h = \text{diam}(\Omega) \).

A proof can be found in [7, Lemma 4.3.8].

### 5.4 Fractional Sobolev spaces

In this subsection, we derive a generalization of Sobolev spaces characterized by fractional-order derivatives using two different approaches. First, we interpolate integer-valued Sobolev spaces, and secondly, we define an intrinsic norm. Both approaches are then shown to be equivalent under some regularity condition on the domain \( \Omega \).

Fractional Sobolev spaces play an important role in the analysis of partial differential equations. In particular, they characterize the regularity of functions from Sobolev spaces defined on a domain \( \Omega \) restricted to the boundary \( \partial \Omega \) of the domain where the restriction is realized by a so-called trace operator (see e.g. [8]). Moreover, a detailed description of various areas of further application is listed in [9]. For a more in-depth analysis of these spaces the interested reader is referred to [36] and [51].

In view of the main result in Section 6, we are mostly interested in the case \( p = \infty \). We will see that in this particular case the resulting spaces coincide with Hölder spaces which can thus be seen as a special case of the broader theory of fractional Sobolev spaces.

We start by defining fractional-order Sobolev spaces \( W^{s,p} \) for \( 0 < s < 1 \) via Banach space interpolation (see Section 4).

**Definition 5.22.** Let \( 0 < s < 1 \) and \( 1 \leq p \leq \infty \), then we set

\[
W^{s,p}(\Omega) := \left( L^p(\Omega), W^{1,p}(\Omega) \right)_{s,p}.
\]

Next, we define fractional-order spaces in terms of an intrinsic norm.

**Definition 5.23 (Sobolev-Slobodeckij spaces).** We set for \( 0 < s < 1 \) and \( 1 \leq p \leq \infty \)

\[
\widetilde{W}^{s,p}(\Omega) := \left\{ f \in L^p(\Omega) : \|f\|_{\widetilde{W}^{s,p}(\Omega)} < \infty \right\}
\]

with

\[
\|f\|_{\widetilde{W}^{s,p}(\Omega)} := \left( \|f\|_{L^p(\Omega)}^p + \int_\Omega \int_\Omega \left( \frac{|f(x) - f(y)|}{|x - y|^{d+sp/p}} \right)^p dxdy \right)^{1/p}
\]

for \( 1 \leq p < \infty \)

and

\[
\|f\|_{\widetilde{W}^{s,\infty}(\Omega)} := \max \left\{ \|f\|_{L^\infty(\Omega)}, \sup_{x,y \in \Omega} \frac{|f(x) - f(y)|}{|x - y|^{s}} \right\}.
\]
The space $\tilde{W}^{s,p}(\Omega)$ endowed with the norm $\| \cdot \|_{\tilde{W}^{s,p}(\Omega)}$ is a Banach space called Sobolev-Slobodeckij space. Note that for $p = \infty$ the space coincides with the space of bounded $s$-Hölder continuous functions, denoted by $C^{0,s}(\Omega)$.

We now show that given suitable regularity conditions of the domain $\Omega$, Definitions 5.22 and 5.23 yield the same spaces with equivalent norms. We believe that the result is well-known but could not find a reference for the case $p = \infty$ so that we will give a proof here.

**Theorem 5.24.** Let $\Omega \subset \mathbb{R}^d$ be a Lipschitz-domain. Then we have for $0 < s < 1$ and $1 \leq p \leq \infty$ that

$$W^{s,p}(\Omega) = \tilde{W}^{s,p}(\Omega)$$

with equivalence of the respective norms.

**Proof.** For $1 \leq p < \infty$ the statement coincides with [7, Theorem 14.2.3]. So we only need to prove the case $p = \infty$. To this end, we use the same strategy as was used for $1 \leq p < \infty$ in [7], i.e. we show the statement for $\Omega = \mathbb{R}^d$ and then use an extension operator to deduce the statement for a Lipschitz-domain $\Omega$.

In [36, Remark after Example 1.8] it is shown that

$$\left( L^\infty(\mathbb{R}^d), C^{0,1}(\mathbb{R}^d) \right)_{s,\infty} = \tilde{W}^{s,\infty}(\mathbb{R}^d) \tag{5.8}$$

with equivalence of the respective norms. From Corollary 5.10 we have that

$$C^{0,1}(\mathbb{R}^d) = W^{1,\infty}(\mathbb{R}^d)$$

with norm equivalence. It now follows that we can use Remark 4.3 to exchange the space Lip for $W^{1,\infty}$ in Equation (5.8) and finally get

$$\left( L^\infty(\mathbb{R}^d), W^{1,\infty}(\mathbb{R}^d) \right)_{s,\infty} = \tilde{W}^{s,\infty}(\mathbb{R}^d) \tag{5.9}$$

with equivalence of the norms.

Let now $\Omega \subset \mathbb{R}^d$ be a Lipschitz-domain. Then, Theorem 5.7 yields that there exists an extension operator $E$ such that

$$E \in L(L^\infty(\Omega), L^\infty(\mathbb{R}^d)) \cap L(W^{1,\infty}(\Omega), W^{1,\infty}(\mathbb{R}^d))$$

and $Ef = f$ a.e. on $\Omega$ for all $f \in L^\infty(\Omega)$. Using Theorem 4.5 for $A_0 = L^\infty(\Omega)$, $A_1 = W^{1,\infty}(\Omega)$ and $B_0 = L^\infty(\mathbb{R}^d)$, $B_1 = W^{1,\infty}(\mathbb{R}^d)$ we get that

$$E \in L(W^{s,\infty}(\Omega), W^{s,\infty}(\mathbb{R}^d)) \tag{5.10}$$

is an extension operator for $W^{s,\infty}(\Omega)$. We now show that $W^{s,\infty}(\Omega) \subset \tilde{W}^{s,\infty}(\Omega)$ and have

$$\| u \|_{\tilde{W}^{s,\infty}(\Omega)} = \| Eu \|_{\tilde{W}^{s,\infty}(\Omega)}$$

$$\leq \| Eu \|_{\tilde{W}^{s,\infty}(\mathbb{R}^d)}$$

$$\leq C_1 \| Eu \|_{W^{s,\infty}(\mathbb{R}^d)}$$

$$\leq C'_1 \| u \|_{W^{s,\infty}(\Omega)}.$$
where the extension property of $E$ is used in the first step and the second step follows easily from the definition of the norm in Definition 5.23. The third step is a consequence of the norm equivalence if $\Omega = \mathbb{R}^d$ (see (5.9)) and the last step uses the boundedness of $E$ (see (5.10)). Here $C' = C'(s, d, \Omega) > 0$ is a constant.

To finish the proof it remains to show that $\tilde{W}^{s, \infty}(\Omega) \subset W^{s, \infty}(\Omega)$. In [23, 6.2.4] the existence of an extension operator $\tilde{E}$ is shown with $\tilde{E} \in \mathcal{L}(\tilde{W}^{s, \infty}(\Omega), \tilde{W}^{s, \infty}(\mathbb{R}^d))$. We get

$$
\|u\|_{W^{s, \infty}(\Omega)} = \|\tilde{E}u\|_{W^{s, \infty}(\Omega)} \\
\leq C_2 \|\tilde{E}u\|_{W^{s, \infty}(\mathbb{R}^d)} \\
\leq C'_2 \|\tilde{E}u\|_{\tilde{W}^{s, \infty}(\mathbb{R}^d)} \\
\leq C''_2 \|u\|_{\tilde{W}^{s, \infty}(\Omega)}.
$$

The first step is again due to the extension property and the second step follows from

$$
W^{s, \infty}(\Omega') \subset W^{s, \infty}(\Omega)
$$

for $\Omega \subset \Omega'$ (see [7, 14.1.2]). The third step is a consequence of (5.9) and the last step follows from the boundedness of $\tilde{E}$. As above $C'_2 = C'_2(s, d, \Omega) > 0$ is a constant.
6 Upper bounds for approximations with deep ReLU neural networks in Sobolev type norms

In this section, we derive upper complexity bounds for approximations of functions from certain Sobolev spaces with deep ReLU networks. Our result is a generalization of [54, Theorem 1] to the case where the approximation error is measured in a Sobolev-type norm.

From now on we assume that \( \varrho : \mathbb{R} \to \mathbb{R} \) is the ReLU activation function (cf. Definition 3.4). As in [54] we are interested in approximating functions in subsets of the Sobolev space \( W^{n,\infty}((0,1)^d) \) with realizations of neural networks. For this we define the set:

\[
F_{n,d,B} := \left\{ f \in W^{n,\infty}((0,1)^d) : \| f \|_{W^{n,\infty}((0,1)^d)} \leq B \right\}.
\]

Recall from Remark 5.11 that \( W^{n,\infty}((0,1)^d) \) can be identified with \( C^{n-1,1}([0,1]^d) \). In [54, Theorem 1] it is shown that for \( B = 1 \) and an arbitrary function \( f \in F_{n,d,B} \) a ReLU network can be constructed that realizes an approximation with \( L^\infty \)-error at most \( \varepsilon \) using \( O(\log_2(1/\varepsilon)) \) layers and \( O(\varepsilon^{-d/n} \log_2(1/\varepsilon)) \) nonzero weights and neurons.

We will show that the approximation can also be performed with respect to a continuous scale of higher order Sobolev-type norms and additionally, that there is a trade-off between the regularity used in the norm in which the approximation error is measured and the regularity used in the bounds. In particular, we will show the following theorem:

**Theorem 6.1.** Let \( d \in \mathbb{N}, n \in \mathbb{N} \geq 2, B > 0, \) and \( 0 \leq s \leq 1 \). Then, there exists a constant \( c = c(d,n,B,s) > 0 \) with the following properties:

For any \( f \in F_{n,d,B} \) and any \( \varepsilon \in (0,1/2) \), there is a neural network \( \Phi^f_\varepsilon \) with \( d \)-dimensional input and one-dimensional output, with at most \( c \cdot \log_2(\varepsilon^{-n/(n-s)}) \) layers and \( c \cdot \varepsilon^{-d/(n-s)} \cdot \log_2(\varepsilon^{-n/(n-s)}) \) weights and neurons, such that

\[
\| R_\varepsilon(\Phi^f_\varepsilon^f) - f \|_{W^{s,\infty}((0,1)^d)} \leq \varepsilon.
\]

Furthermore, for the architecture \( A \) of \( \Phi^f_\varepsilon \) it holds that \( A = A(d,n,B,s) \), i.e. the architecture is independent of \( f \).

Clearly, if \( s = 0 \), then Theorem 6.1 corresponds to the theorem shown by Yarotsky in [54].

Note that if \( \Phi \) is a neural network and \( \varphi \) is the ReLU, then the restriction of \( R_\varepsilon(\Phi) \) to \((0,1)^d\) is bounded and Lipschitz continuous (cf. Remark 3.13) and hence in view of Corollary 5.10 an element of the Sobolev space \( W^{1,\infty}((0,1)^d) \). Therefore, the expressions in the previous theorem are well-defined.

**Remark.** In view of Theorem 5.24, the same asymptotic bounds can be obtained for approximations with \( C^{0,s} \)-error at most \( \varepsilon \), i.e. by replacing the Sobolev norm with the corresponding Hölder norm.

Furthermore, Theorem 6.1 holds for all activation functions that are in some sense similar to the ReLU. In particular, it follows directly from Lemma 3.5 that the statement holds for any continuous piecewise linear activation function \( \hat{\varphi} \) with \( M \) breakpoints, where \( 1 \leq M < \infty \).

As a consequence of Theorem 6.1 and Lemma 3.12, we can easily derive a similar statement for standard neural networks.
Corollary 6.2. Let $d \in \mathbb{N}$, $n \in \mathbb{N}_{\geq 2}$, $B > 0$, and $0 \leq s \leq 1$. Then, there exists a constant $c = c(d,n,B,s) > 0$ with the following properties:

For any $f \in \mathcal{F}_{n,d,B}$ and any $\varepsilon \in (0,1/2)$, there is a standard neural network $\Phi_{st,\varepsilon}$ with $d$-dimensional input and one-dimensional output, with at most $c \cdot \log_2 \left( \varepsilon^{-n/(n-s)} \right)$ layers and $c \cdot \varepsilon^{-d/(n-s)} \cdot \log^2_2 \left( \varepsilon^{-n/(n-s)} \right)$ weights and neurons, such that

$$\| R_\varepsilon(\Phi_{st,\varepsilon}^f) - f \|_{W^{s,\infty}((0,1)^d)} \leq \varepsilon.$$ 

Furthermore, for the standard architecture $\mathcal{A}$ of $\Phi_{st,\varepsilon}^f$, it holds that $\mathcal{A} = \mathcal{A}(d,n,B,s)$, i.e. $\mathcal{A}$ is independent of $f$.

6.1 Proof of the main result

The strategy of the proof of our main result follows the idea of the proof of Theorem 1 in [54]. As in this paper, we start by constructing a neural network that approximates the square function on the interval $(0,1)$ up to an approximation error at most $\varepsilon$. In our result the error is measured in the $W^{1,\infty}$ norm (Proposition 6.3). This result is then used to obtain an approximation of a multiplication operator (Proposition 6.4). Next, we derive a partition of unity (Lemma 6.5) as a product of univariate functions that can be realized by neural networks. Using this construction we then build a localized Taylor approximation for a function $f \in \mathcal{F}$ (Proposition 6.3).

There exist constants $c_1, c_2, c_3, c_4 > 0$, such that for all $\varepsilon \in (0,1/2)$ there is a neural network $\Phi_{\varepsilon}^{sq}$ with at most $c_1 \cdot \log_2 \left( 1/\varepsilon \right)$ nonzero weights, at most $c_2 \cdot \log_2 \left( 1/\varepsilon \right)$ layers, at most $c_3 \cdot \log_2 \left( 1/\varepsilon \right)$ neurons, and with one-dimensional input and output such that

$$\| R_\varepsilon(\Phi_{\varepsilon}^{sq}) - x^2 \|_{W^{1,\infty}((0,1);dx)} \leq \varepsilon$$

(6.1)

and $R_\varepsilon(\Phi_{\varepsilon}^{sq})(0) = 0$. Furthermore, it holds that

$$| R_\varepsilon(\Phi_{\varepsilon}^{sq}) |_{W^{1,\infty}((0,1))} \leq c_4.$$ 

(6.2)

Proof. In the proof of [54] Proposition 2 it is shown that there exist constants $c_1, c_2, c_3 > 0$, such that for each $m \in \mathbb{N}$ there is a neural network $\Phi_m$ with at most $c_1 \cdot m$ nonzero weights, at most $c_2 \cdot m$ layers, at most $c_3 \cdot m$ neurons the realization of which is a piecewise linear
interpolation of $x \mapsto x^2$ on $(0, 1)$. In detail, it is shown there, that the network $\Phi_m$ satisfies for $k \in \{0, \ldots, 2^m - 1\}$ and $x \in \left[ \frac{k}{2^m}, \frac{k+1}{2^m} \right]$

$$R_\varphi(\Phi_m)(x) = \left( \frac{(k+1)^2}{2^m} - \frac{k^2}{2^m} \right) \left( x - \frac{k}{2^m} \right) + \left( \frac{k}{2^m} \right)^2.$$  \hfill (6.3)

Thus, $R_\varphi(\Phi_m)$ interpolates $f$ piecewise linear with $2^m + 1$ uniformly distributed breakpoints $\frac{k}{2^m}, k = 0, \ldots, 2^m$. In particular, $R_\varphi(\Phi_m)(0) = 0$. Furthermore, it is shown in the proof of [54, Proposition 2] that

$$\|R_\varphi(\Phi_m)(x) - x^2\|_{L^\infty((0,1);dx)} \leq 2^{-2-2^m}.$$  \hfill (6.4)

We will now show that the approximation error of the derivative can be bounded in a similar way. In particular, we show the estimate

$$|R_\varphi(\Phi_m) - x^2|_{W^{1,\infty}((0,1);dx)} \leq 2^{-m}.$$  \hfill (6.5)

Note that $R_\varphi(\Phi_m)$ is by Equation (6.3) piecewise affine-linear. Thus, by application of Remark 5.9, we get for $k = 0, \ldots, 2^m$

$$|R_\varphi(\Phi_m)(x) - x^2|_{W^{1,\infty}((0,1);dx)} = \left\| \frac{(k+1)^2}{2^m} - \frac{k^2}{2^m} - 2x \right\|_{L^\infty((k/2^m,(k+1)/2^m);dx)} = \max\left\{ \left\| \frac{2k+1}{2^m} - 2\frac{k}{2^m} \right\|_{L^\infty((k/2^m,(k+1)/2^m);dx)}, \left\| \frac{2k+1}{2^m} - \frac{k+1}{2^m} \right\|_{L^\infty((k/2^m,(k+1)/2^m);dx)} \right\} = 2^{-m}.$$  \hfill (6.6)

Combining Equation (6.4) and (6.5) yields

$$\|R_\varphi(\Phi_m)(x) - x^2\|_{W^{1,\infty}((0,1);dx)} \leq \max\{2^{-2m-2}, 2^{-m}\} = 2^{-m}.$$  \hfill (6.7)

Clearly, the weak derivative of $\Phi_m$ is a piecewise constant function, which reaches its maximum on the last piece. Hence,

$$|R_\varphi(\Phi_m)|_{W^{1,\infty}((0,1))} \leq \frac{(2^m)^2 - (2^m - 1)^2}{2^m} = 2 - \frac{1}{2^m} \leq 2.$$  \hfill (6.8)

Let now $\varepsilon \in (0, 1/2)$ and choose $m = \left\lceil \log_2(1/\varepsilon) \right\rceil$. Setting $\Phi^\varepsilon_m := \Phi_m$ fulfills the approximation bound in Equation (6.1) and $R_\varphi(\Phi^\varepsilon_m)(0) = 0$. The estimate (6.2) is fulfilled because of Equation (6.6). The number of weights can be bounded by

$$M(\Phi^\varepsilon_m) \leq c_1 \cdot m \leq c_1 \cdot (\log_2(1/\varepsilon) + 1) \leq 2 \cdot c_1 \cdot \log_2(1/\varepsilon).$$

In the same way, the number of neurons and layers can be bounded, which concludes the proof.
As in [54], we will now use Proposition 6.3 and the polarization identity
\[ xy = \frac{1}{2}(x + y)^2 - x^2 - y^2 \quad \text{for } x, y \in \mathbb{R}, \] (6.7)
to define an approximate multiplication where the approximation error is again (and in contrast to [54]) measured in the \( W^{1,\infty} \) norm.

**Proposition 6.4.** For any \( M \geq 1 \), there exist constants \( c, c_1 > 0 \) and \( c_2 = c_2(M) > 0 \) such that for any \( \varepsilon \in (0, 1/2) \), there is a neural network \( \tilde{x} \) with two-dimensional input and one-dimensional output that satisfies the following properties:

1. \( \| R_\varepsilon(\tilde{x})(x, y) - xy \|_{W^{1,\infty}((-M, M)^2, dx dy)} \leq \varepsilon; \)
2. if \( x = 0 \) or \( y = 0 \), then \( R_\varepsilon(\tilde{x})(x, y) = 0; \)
3. \( \| R_\varepsilon(\tilde{x} \varepsilon)|_{W^{1,\infty}((-M, M)^2)} \leq cM; \)
4. the depth and the number of weights and neurons in \( \tilde{x} \) is at most \( c_1 \log_2(1/\varepsilon) + c_2. \)

**Proof.** Let \( \delta := \varepsilon/(6M^2C) \), where \( C \) is the constant from Corollary 5.13 for \( n = 2 \) and \( m = 1 \), and let \( \Phi^\text{sq}_\delta \) be the approximate squaring network from Proposition 6.3 such that
\[ \| R_\varepsilon(\Phi^\text{sq}_\delta) - x^2 \|_{W^{1,\infty}((0, 1), dx)} < \delta. \] (6.8)
As in the proof of [54] Proposition 3], we use the fact that \( |x| = g(x) + g(-x) \) to see that a network \( \tilde{x} \) can be constructed with two-dimensional input and one-dimensional output that satisfies
\[ R_\varepsilon(\tilde{x})(x, y) = 2M^2 \left( R_\varepsilon(\Phi^\text{sq}_\delta) \left( \frac{|x + y|}{2M} \right) - R_\varepsilon(\Phi^\text{sq}_\delta) \left( \frac{|x|}{2M} \right) - R_\varepsilon(\Phi^\text{sq}_\delta) \left( \frac{|y|}{2M} \right) \right). \]

As a consequence of Proposition 6.3 there exists a constant \( c_0 > 0 \) such that \( \tilde{x} \) has at most \( c_0 \log_2(1/\varepsilon) + c_0 \log_2(6M^2) + 3 \leq c_0 \log_2(1/\varepsilon) + c_1 \) layers, \( 3c_0 \log_2(1/\varepsilon) + 3c_0 \log_2(6M^2) + 9 \leq c' \log_2(1/\varepsilon) + c_2 \) neurons and \( 3c_0 \log_2(1/\varepsilon) + 3c_0 \log_2(6M^2) + 17 \leq c'' \log_2(1/\varepsilon) + c_3 \) nonzero weights. Here \( c', c'' > 0 \) and \( c_1 = c_1(M), c_2 = c_2(M), c_3 = c_3(M) > 0 \) are suitable constants and \( [54] \) is hence satisfied.
Since \( R_\varepsilon(\Phi^\text{sq}_\delta)(0) = 0, [54] \) easily follows.

Using the polarization identity (6.7), we can write
\[ xy = 4M^2 \frac{x}{2M} \frac{y}{2M} \]
\begin{align*}
&= 4M^2 \frac{1}{2} \left( \frac{x}{2M} + \frac{y}{2M} \right)^2 - \left( \frac{x}{2M} \right)^2 - \left( \frac{y}{2M} \right)^2 \\
&= 2M^2 \left( \frac{|x + y|}{2M} \right)^2 - \left( \frac{|x|}{2M} \right)^2 - \left( \frac{|y|}{2M} \right)^2. \quad (6.9)
\end{align*}

To keep the following calculations simple, we introduce some notation. In particular, we define
\[ u_{xy}: (-M, M)^2 \to (0, 1), \quad (x, y) \mapsto \frac{|x + y|}{2M}. \]
\[ u_x : (-M, M)^2 \rightarrow (0, 1), \quad (x, y) \mapsto \frac{|x|}{2M}, \]

and

\[ u_y : (-M, M)^2 \rightarrow (0, 1), \quad (x, y) \mapsto \frac{|y|}{2M}. \]

Setting \( f : (0, 1) \rightarrow \mathbb{R}, x \mapsto x^2 \) and using (6.9) we get

\[ xy = 2M^2 \left( f \circ u_{xy}(x, y) - f \circ u_x(x, y) - f \circ u_y(x, y) \right). \tag{6.10} \]

Now, we can estimate

\[
\|R_\epsilon(\widetilde{x})(x, y) - xy\|_{W^{1, \infty}((-M, M)^2; dx dy)}
\]

\[
= \left\| 2M^2 \left( R_\epsilon(\Phi^\delta) \circ u_{xy} - R_\epsilon(\Phi^\delta) \circ u_x - R_\epsilon(\Phi^\delta) \circ u_y \right) \right\|_{W^{1, \infty}((-M, M)^2)}
\]

\[
- \left\| 2M^2 \left( f \circ u_{xy} - f \circ u_x - f \circ u_y \right) \right\|_{W^{1, \infty}((-M, M)^2)}
\]

\[
\leq 2M^2 \left( \| (R_\epsilon(\Phi^\delta) - f) \circ u_{xy} \|_{W^{1, \infty}((-M, M)^2)} + \| (R_\epsilon(\Phi^\delta) - f) \circ u_x \|_{W^{1, \infty}((-M, M)^2)} \right)
\]

\[
+ \| (R_\epsilon(\Phi^\delta) - f) \circ u_y \|_{W^{1, \infty}((-M, M)^2)} \right) \tag{6.11}
\]

Note that for the inner functions in the compositions in Equation (6.11), it holds that

\[
|u_{xy}|_{W^{1, \infty}((-M, M)^2)} = |u_x|_{W^{1, \infty}((-M, M)^2)} = |u_y|_{W^{1, \infty}((-M, M)^2)} = 1/(2M). \tag{6.12}
\]

Hence, to finally prove (iii), we apply Corollary 5.13 to (6.11) and get

\[
\|R_\epsilon(\widetilde{x})(x, y) - xy\|_{W^{1, \infty}((-M, M)^2; dx dy)}
\]

\[
\leq 6M^2 C \delta = \varepsilon,
\]

where we used (6.8) in the second step.

To show (iii) we use the chain rule estimate from Corollary 5.13 and get

\[
|R_\epsilon(\widetilde{x})|_{W^{1, \infty}((-M, M)^2)}
\]

\[
\leq 2M^2 C \sum_{x \in \{x, y, xy\}} |R_\epsilon(\Phi^\delta)|_{W^{1, \infty}((0,1))} |u_x|_{W^{1, \infty}((-M, M)^2)}
\]

\[
\leq 2M^2 C 3C' \frac{1}{2M} = 3CC',
\]

where we used Equation (6.12) in the third step and \( C' \) is the absolute constant from Equation (6.2) in Proposition 6.3. \(\square\)
Next, we construct a partition of unity (in the same way as in [54, Theorem 1]) that can be defined as a product of piecewise linear functions, such that each factor of the product can be realized by a neural network. We will use this product structure in Lemma 6.9 together with a generalized version of the approximate multiplication from Proposition 6.4 to approximate localized polynomials with ReLU networks.

**Lemma 6.5.** For any \( d, N \in \mathbb{N} \) there exists a collection of functions
\[
\Psi = \{ \phi_m : m \in \{0, \ldots, N\}^d \}
\]
with \( \phi_m : \mathbb{R}^d \to \mathbb{R} \) for all \( m \in \{0, \ldots, N\}^d \) with the following properties:

(i) \( 0 \leq \phi_m(x) \leq 1 \) for every \( \phi_m \in \Psi \) and every \( x \in \mathbb{R}^d \);

(ii) \( \sum_{\phi_m \in \Psi} \phi_m(x) = 1 \) for every \( x \in [0, 1]^d \);

(iii) \( \text{supp } \phi_m \subset B_{1/N}^{\mathbb{R}^d} \) for every \( \phi_m \in \Psi \).

(iv) There exists a constant \( c \geq 1 \) such that \( \|\phi_m\|_{W^{k, \infty}(\mathbb{R}^d)} \leq (c \cdot N)^k \) for \( k \in \{0, 1\} \).

(v) There exist absolute constants \( C, c \geq 1 \) such that for each \( \phi_m \in \Psi \) there is a neural network \( \Phi_m \) with \( d \)-dimensional input and \( d \)-dimensional output, with at most three layers, \( C_d \) nonzero weights and neurons, that satisfies
\[
\prod_{l=1}^d [R_{\psi}(\Phi_m)]_l \phi_m
\]
and \( \|[R_{\psi}(\Phi_m)]_l W^{k, \infty}((0,1)^d) \leq (cN)^k \) for all \( l = 1, \ldots, d \) and \( k \in \{0, 1\} \).

**Proof.** As in [54], we define the functions
\[
\psi : \mathbb{R} \to \mathbb{R}, \quad \psi(x) := \begin{cases} 1, & |x| < 1, \\ 0, & 2 < |x|, \\ 2 - |x|, & 1 \leq |x| \leq 2, \end{cases}
\]
and \( \phi_m : \mathbb{R}^d \to \mathbb{R} \) as a product of scaled and shifted versions of \( \psi \). In detail, we set
\[
\phi_m(x) := \prod_{i=1}^d \psi \left( 3N \left( x_i - \frac{m_i}{N} \right) \right), \quad (6.13)
\]
for \( m = (m_1, \ldots, m_d) \in \{0, \ldots, N\}^d \). Then, (i), (ii), (iii) follow easily from the definition.

To show (iv), note that \( \|\phi_m\|_{L^\infty} \leq 1 \) follows already from (ii). It now suffices to show the claim for the \( W^{1,\infty} \) semi-norm. For this, let \( l \in \{1, \ldots, d\} \) and \( x \in \mathbb{R}^d \), then
\[
\left| \frac{\partial}{\partial x_l} \phi_m(x) \right| = \prod_{i=1, i \neq l}^d \psi \left( 3N \left( x_i - \frac{m_i}{N} \right) \right) \left| \psi' \left( 3N \left( x_l - \frac{m_l}{N} \right) \right) \right| \leq 3N.
\]
It follows that $|\phi_m|_{W^{1,\infty}} \leq 3N$.

To show (v), we start by constructing a network $\Phi_{\psi}$ that realizes the function $\psi$. For this we set

$$A_1 := \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \quad b_1 := \begin{bmatrix} 2 \\ 1 \\ -1 \\ 1 \end{bmatrix} \quad \text{and} \quad A_2 := \begin{bmatrix} 0 & 1 & -1 & -1 & 1 \end{bmatrix}, \quad b_2 := 0,$$

and $\Phi_{\psi} := ((A_1, b_1), (A_2, b_2))$. Then $\Phi_{\psi}$ is a two-layer network with one-dimensional input and one-dimensional output, with 12 nonzero weights and 6 neurons such that

$$R_{\psi}(\Phi_{\psi})(x) = \psi(x) \quad \text{for all} \quad x \in \mathbb{R}.$$

We denote by $\Phi_{m,l}$ the one-layer network with $d$-dimensional input and one-dimensional output, with 2 nonzero weights and $d+1$ neurons such that $R_{\psi}(\Phi_{m,l}) = 3N(x_l - m_l/N)$ for all $x \in \mathbb{R}^d$ and $m \in \{0, \ldots, N\}^d$, $l = 1, \ldots, d$. Finally, we define

$$\Phi_m := P(\Phi_{\psi} \circ \Phi_{m,l} : l = 1, \ldots, d),$$

which is a three-layer network with $d$-dimensional input and $d$-dimensional output, with at most $d \cdot 2 \cdot (12 + 2) = d \cdot 28$ nonzero weights and at most $9d$ neurons, and

$$|R_{\psi}(\Phi_m)|_l(x) = R_{\psi}(\Phi_{\psi} \circ \Phi_{m,l})(x) = \psi \left( 3N \left( x_l - \frac{m_l}{N} \right) \right)$$

for $l = 1, \ldots, d$ and $x \in \mathbb{R}^d$. Clearly, it follows that $\prod_{l=1}^d |R_{\psi}(\Phi_m)|_l(x) = \phi_m(x)$ for all $x \in \mathbb{R}^d$. The last part of (v) can be shown similarly as (iv).

The following Lemma uses the partition of unity from Lemma 6.5 and the Bramble-Hilbert Lemma in a classical way to derive an approximation with localized polynomials in the $L^\infty$ norm and the $W^{1,\infty}$ norm.

**Lemma 6.6.** Let $d, n, N \in \mathbb{N}$, $n \geq 2$ and $\Psi = \Psi(d, N) = \{\phi_m : m \in \{0, \ldots, N\}^d\}$ be the partition of unity from Lemma 6.5. Then there is a constant $C = C(d, n) > 0$ such that for any $f \in W^{n,\infty}((0,1)^d)$, there exist polynomials $p_{f,m}(x) = \sum_{|\alpha| \leq n-1} c_{f,m,\alpha} x^\alpha$ for $m \in \{0, \ldots, N\}^d$ with the following properties: Set $f_N := \sum_{m \in \{0, \ldots, N\}^d} \phi_m p_{f,m}$, then

(i) the operator $T_0 : W^{n,\infty}((0,1)^d) \to L^\infty((0,1)^d)$ with $T_0 f = f - f_N$ is linear and bounded with

$$\|T_0 f\|_{L^\infty((0,1)^d)} \leq C \left( \frac{1}{N} \right)^n \|f\|_{W^{n,\infty}((0,1)^d)}.$$

(ii) The operator $T_1 : W^{n,\infty}((0,1)^d) \to W^{1,\infty}((0,1)^d)$ with $T_1 f = f - f_N$ is linear and bounded with

$$\|T_1 f\|_{W^{1,\infty}((0,1)^d)} \leq C \left( \frac{1}{N} \right)^{n-1} \|f\|_{W^{n,\infty}((0,1)^d)}.$$
Furthermore, there is a constant $c = c(d, n) > 0$ such that for any $f \in W^{n, \infty}((0, 1)^d)$ the coefficients of the polynomials $p_{f,m}$ satisfy $|c_{f,m,\alpha}| \leq c\|f\|_{W^{n,\infty}((0,1)^d)}$ for all $\alpha \in \mathbb{N}_d^d$ with $|\alpha| \leq n - 1$ and $m \in \{0, \ldots, N\}^d$.

**Proof.** The idea of the proof is similar to the first part of the proof of [54, Theorem 1]. We use approximation properties of averaged Taylor polynomials (see Bramble-Hilbert Lemma 5.21) to derive local estimates and then combine them using a partition of unity to obtain a global estimate. In order to use this strategy also near the boundary, we make use of an extension operator.

For this, let $E : W^{n, \infty}((0, 1)^d) \to W^{n, \infty}(\mathbb{R}^d)$ be the extension operator from Theorem 5.7 and set $f := Ef$. Note that for arbitrary $\Omega \subset \mathbb{R}^d$ and $1 \leq k \leq n$ it holds
\[
|\tilde{f}|_{W^{k, \infty}(\Omega)} \leq \|\tilde{f}\|_{W^{n, \infty}(\mathbb{R}^d)} \leq C_E\|f\|_{W^{n, \infty}((0,1)^d)},
\] (6.14)
where $C_E = C_E(n, d)$ is the norm of the extension operator.

For each $m \in \{0, \ldots, N\}^d$ we set
\[
\Omega_{m,N} := B_{\frac{1}{N}}|\|_{L^\infty}(\frac{m}{N}) \quad \text{and} \quad B_{m,N} := B_{\frac{1}{N}}|\|_{L^1}(\frac{m}{N}),
\]
denote by $p_m = p_{f,m}$ the Taylor polynomial of order $n$ of $\tilde{f}$ averaged over $B_{m,N}$ (cf. Definition 5.16). It follows from Proposition 5.18 (for $\Omega = \Omega_{m,N}$, $B = B_{m,N}$ and $R = 2$) that we can write $p_m = \sum_{|\alpha| \leq n-1} c_{m,\alpha} x^\alpha$ and that there is a constant $c' = c'(n) > 0$ such that
\[
|c_{m,\alpha}| \leq c'_n \|f\|_{W^{n, \infty}(\Omega_{m,N})} \leq c' \cdot C_E\|f\|_{W^{n, \infty}((0,1)^d)},
\]
for $m \in \{0, \ldots, N\}^d$, where (6.14) was used for the second step and $c'' = c''(n, d) > 0$ is a suitable constant. It now suffices to show (i) and (ii).

To check that the conditions of the Bramble-Hilbert Lemma 5.21 are fulfilled, note that $B_{m,N} \subset \subset \Omega_{m,N}$. Furthermore, $B_{m,N}$ is a ball in $\Omega_{m,N}$ such that $\Omega_{m,N}$ is star-shaped with respect to $B_{m,N}$. We have $\text{diam}_{|\|}(\Omega_{m,N}) = (2\sqrt{d})/N$ and $r_{\text{max}}(\Omega_{m,N}) = 1/N$ and, thus,
\[
r_{|\|}(B_{m,N}) = \frac{3}{4N} > \frac{1}{2} \cdot \frac{1}{N} = \frac{1}{2} \cdot r_{\text{max}}(\Omega_{m,N}).
\]
Finally, we have for the chunkiness parameter of $\Omega_{m,N}$
\[
\gamma(\Omega_{m,N}) = \text{diam}(\Omega_{m,N}) \cdot \frac{1}{r_{\text{max}}(\Omega_{m,N})} = \frac{2\sqrt{d}}{N} \cdot N = 2\sqrt{d},
\]
(6.15)
Applying the Bramble-Hilbert Lemma 5.21 yields for each $m \in \{0, \ldots, N\}^d$ the local estimate
\[
\|\tilde{f} - p_m\|_{L^\infty(\Omega_{m,N})} \leq C_1 \left(\frac{2\sqrt{d}}{N}\right)^n |\tilde{f}|_{W^{n, \infty}(\Omega_{m,N})} \leq C_1(2\sqrt{d})^n \left(\frac{1}{N}\right)^n C_E\|f\|_{W^{n, \infty}((0,1)^d)} = C_2 \left(\frac{1}{N}\right)^n \|f\|_{W^{n, \infty}((0,1)^d)},
\]
(6.16)
Here, $C_1 = C_1(n, d) > 0$ is the constant from Lemma 5.21 which only depends on $n$ and $d$, since the chunkiness parameter of $\Omega_{m,N}$ is a constant depending only on $d$ (see (6.15)) and $C_2 = C_2(n, d) > 0$. In the same way, we get

$$\left| \tilde{f} - p_m \right|_{W^{1,\infty}((0,1)^d)} \leq C_3 \left( \frac{1}{N} \right)^{n-1} \| f \|_{W^{n,\infty}((0,1)^d)},$$

(6.17)

where $C_3 = C_3(n, d) > 0$ is a suitable constant. The first step towards a global estimate is now to combine Equation (6.16) and (6.17) with the cut-off functions from the partition of unity. We have

$$\| \phi_m (\tilde{f} - p_m) \|_{L^\infty(\Omega_{m,N})} \leq \| \phi_m \|_{L^\infty(\Omega_{m,N})} \cdot \| \tilde{f} - p_m \|_{L^\infty(\Omega_{m,N})}$$

(Lemma 5.5, Equation (6.16)) \leq C_2 \left( \frac{1}{N} \right)^n \| f \|_{W^{n,\infty}((0,1)^d)}.$$

(6.18)

Furthermore, using the product inequality for weak derivatives from Corollary 5.15 we get

$$\| \phi_m (\tilde{f} - p_m) \|_{W^{1,\infty}((0,1)^d)} \leq \| \phi_m \|_{W^{1,\infty}((0,1)^d)} \cdot \| \tilde{f} - p_m \|_{L^\infty(\Omega_{m,N})}$$

$$+ \| \phi_m \|_{L^\infty(\Omega_{m,N})} \cdot \| \tilde{f} - p_m \|_{W^{1,\infty}((0,1)^d)}$$

$$\leq \tilde{c}N \cdot C_2 \left( \frac{1}{N} \right)^n \| f \|_{W^{n,\infty}((0,1)^d)} + C_3 \left( \frac{1}{N} \right)^{n-1} \| f \|_{W^{n,\infty}((0,1)^d)}$$

$$= C_4 \left( \frac{1}{N} \right)^{n-1} \| f \|_{W^{n,\infty}((0,1)^d)}.$$

where the first part of the second inequality follows from Lemma 6.15 (iv) with $k = 1$ and an absolute constant $\tilde{c} \geq 1$, together with Equation (6.16) and the second part from Lemma 6.15 (iv) with $k = 0$, together with Equation (6.17). Here, $C_4 = C_4(n, d) > 0$. Now it easily follows that

$$\| \phi_m (\tilde{f} - p_m) \|_{W^{1,\infty}((0,1)^d)} \leq C_5 \left( \frac{1}{N} \right)^{n-1} \| f \|_{W^{n,\infty}((0,1)^d)},$$

(6.19)

for some constant $C_5 = C_5(n, d) > 0$.

To derive the global estimate, we start by noting that with property (iii) from Lemma 6.5 we have

$$\tilde{f}(x) = \sum_{m \in \{0, \ldots, N\}^d} \phi_m(x) \tilde{f}(x), \quad \text{for a.e. } x \in (0,1)^d.$$

(6.20)
Using that \( \tilde{f} \) is an extension of \( f \) on \((0,1)^d\) we can write for \( k \in \{0,1\} \)

\[
\bigg\| f - \sum_{m \in \{0, \ldots, N\}^d} \phi_m p_m \bigg\|_{W^{k,\infty}((0,1)^d)} = \bigg\| \tilde{f} - \sum_{m \in \{0, \ldots, N\}^d} \phi_m p_m \bigg\|_{W^{k,\infty}((0,1)^d)}
\]

(Equation (6.20)) = \[
\sum_{m \in \{0, \ldots, N\}^d} \phi_m (\tilde{f} - p_m) \bigg\|_{W^{k,\infty}((0,1)^d)}
\]

\[
\leq \max_{\tilde{m} \in \{0, \ldots, N\}^d} \bigg\| \sum_{m \in \{0, \ldots, N\}^d} \phi_m (\tilde{f} - p_m) \bigg\|_{W^{k,\infty}(\Omega_{\tilde{m}}, N)}
\]

(6.21)

where the last step follows from \((0,1)^d \subset \bigcup_{\tilde{m} \in \{0, \ldots, N\}^d} \Omega_{\tilde{m}, N}\). Now we obtain for each \( \tilde{m} \in \{0, \ldots, N\}^d \) and \( k \in \{0,1\} \)

\[
\bigg\| \sum_{m \in \{0, \ldots, N\}^d} \phi_m (\tilde{f} - p_m) \bigg\|_{W^{k,\infty}(\Omega_{\tilde{m}}, N)} \leq \sum_{m \in \{0, \ldots, N\}^d} \|\phi_m (\tilde{f} - p_m)\|_{W^{k,\infty}(\Omega_{\tilde{m}}, N)}
\]

\[
\leq 3^d \max_{m \in \{0, \ldots, N\}^d} \|\phi_m (\tilde{f} - p_m)\|_{W^{k,\infty}(\Omega_{\tilde{m}}, N)}
\]

\[
\leq C_6 \left( \frac{1}{N} \right)^{n-k} \|f\|_{W^{n,\infty}((0,1)^d)},
\]

(6.22)

where the triangle inequality together with the support property (iii) from Lemma 6.5 is used in the first step and the third step follows from (6.18) for \( k = 0 \) and from (6.19) for \( k = 1 \). Here \( C_6 = C_6(n, d) > 0 \) can be chosen independent of \( k \) (e.g. \( C_6 := 3^d \max\{C_2, C_5\} \)).

Finally, the boundedness claim in (i) and (ii) follows from using the definition of \( f_N \) and combining Equation (6.21) with Equation (6.22):

\[
\| f - f_N \|_{W^{k,\infty}((0,1)^d)} \leq C_T \left( \frac{1}{N} \right)^{n-k} \|f\|_{W^{n,\infty}((0,1)^d)},
\]

(6.23)

for \( k \in \{0,1\} \), where \( C_T = C_T(n, d) > 0 \). The linearity of \( T_k, k \in \{0,1\} \) is a consequence of the linearity of the averaged Taylor polynomial (cf. Remark 5.17).

Using an interpolation argument, we can generalize the above result to the case where the approximation is performed with respect to the \( W^{s,\infty} \) norm, where \( 0 \leq s \leq 1 \).

**Corollary 6.7.** Let \( d, N \in \mathbb{N} \), \( n \in \mathbb{N}_{\geq 2} \) and \( \Psi = \Psi(d, N) = \{ \phi_m : m \in \{0, \ldots, N\}^d \} \) be the partition of unity from Lemma 6.5. Then there is a constant \( C = C(d, n) > 0 \) such that for any \( f \in W^{n,\infty}((0,1)^d) \), there exist polynomials \( p_{f,m}(x) = \sum_{|\alpha| \leq n-1} c_{f,m,\alpha} x^\alpha \) for \( m \in \{0, \ldots, N\}^d \) with the following properties:
Let $0 \leq s \leq 1$ and set $f_N := \sum_{m \in \{0, \ldots, N\}} \phi_m p_{f,m}$, then the operator $T_s : W^{n,\infty}((0,1)^d) \to W^{s,\infty}((0,1)^d)$ with $T_s f = f - f_N$ is linear and bounded with

$$\|T_s f\|_{W^{s,\infty}((0,1)^d)} \leq C \left( \frac{1}{N} \right)^{n-s} \|f\|_{W^{n,\infty}((0,1)^d)}.$$  

Furthermore, there is a constant $c = c(d,n) > 0$ such that for any $f \in W^{n,\infty}((0,1)^d)$ the coefficients of the polynomials $p_{f,m}$ satisfy $|c_{f,m,\alpha}| \leq c \|f\|_{W^{n,\infty}((0,1)^d)}$ for all $\alpha \in \mathbb{N}_0^d$ with $|\alpha| \leq n-1$ and $m \in \{0, \ldots, N\}^d$.

**Proof.** Using the polynomials from Lemma 6.6 the boundedness of the coefficients and the case $s \in \{0,1\}$ immediately follows.

For $0 < s < 1$ we use a Banach space interpolation argument. Set $A_0 = A_1 = W^{n,\infty}((0,1)^d)$ together with $B_0 = L^{\infty}((0,1)^d)$ and $B_2 = W^{1,\infty}((0,1)^d)$.

Then, Lemma 6.6 implies that we can apply Theorem 4.5 and get

$$\|\Phi\|_{L^1(W^{n,\infty}(0,1)^d, W^{s,\infty}((0,1)^d))} \leq \|\Phi\|_{L^1(W^{n,\infty}(0,1)^d, L^{\infty}((0,1)^d))} \leq C \left( \frac{1}{N} \right)^{n-s} \|f\|_{W^{n,\infty}((0,1)^d)},$$

where we used Lemma 4.4 (ii) to see that $(A_0, A_1)_{s,\infty} = W^{n,\infty}((0,1)^d)$. Here $C = C(n,d) > 0$ is a suitable constant.

The following technical lemma lays the foundation for approximating localized monomials with neural networks. Using the notation and statement (v) from Lemma 6.5, a localized monomial $\phi_m x^\alpha$ can be expressed as the product of the output components of a network $\Phi_{(m,\alpha)}$ with a suitable output dimension $n$, i.e.

$$\phi_m(x) x^\alpha = \prod_{l=1}^n [R_l(\Phi_{(m,\alpha)})]_l(x).$$

Given a network $\Phi$ with $n$-dimensional output, we construct a network $\Psi \Phi$ that approximates the product of the output components of $\Phi$.

**Lemma 6.8.** Let $d, m, K \in \mathbb{N}$ and $N \geq 1$ be arbitrary. Then there is a constant $C = C(m) > 0$ such that the following holds:

For any $\varepsilon \in (0, 1/2)$, and any neural network $\Phi$ with $d$-dimensional input and $n$-dimensional output where $n \leq m$, and with number of layers, neurons and weights all bounded by $K$, such that

$$\|\Phi\|_{L^1(W^{s,\infty}((0,1)^d))} \leq N^k \quad \text{for} \quad k \in \{0,1\} \quad \text{and} \quad l = 1, \ldots, n$$

39
there exists a neural network $\Psi_{\epsilon,\Phi}$ with $d$-dimensional input and one-dimensional output, and with number of layers, neurons and weights all bounded by $KC\log(1/\epsilon)$, such that

$$
\left\| R_{\phi}(\Psi_{\epsilon,\Phi}) - \prod_{l=1}^{n}[R_{\phi}(\Phi)]_{l} \right\|_{W^{k,\infty}((0,1)^d)} \leq cN^k\epsilon
$$

(6.24)

for $k \in \{0,1\}$ and some constant $c = c(d, m, k)$. Moreover,

$$
R_{\phi}(\Psi_{\epsilon,\Phi})(x) = 0 \quad \text{if} \quad \prod_{l=1}^{n}[R_{\phi}(\Phi)]_{l}(x) = 0
$$

(6.25)

for $x \in (0,1)^d$.

Proof. Let $d, K \in \mathbb{N}$ and $N \geq 1$ be fixed. We show by induction over $m \in \mathbb{N}$ that the statement holds. To make the induction argument easier we will additionally show that $c = c(d, m, k) = m^{1-k}c_1^k$, where $c_1 = c_1(d, m) > 0$, and that the network $\Psi_{\epsilon,\Phi}$ can be chosen such that the first $L(\Phi) - 1$ layers of $\Psi_{\epsilon,\Phi}$ and $\Phi$ coincide and $|R_{\phi}(\Psi_{\epsilon,\Phi})|_{W^{1,\infty}((0,1)^d)} \leq C_1N$ for a constant $C_1 = C_1(d, m) > 0$.

If $m = 1$, then we can choose $\Psi_{\epsilon,\Phi} = \Phi$ for any $\epsilon \in (0,1/2)$ and the claim holds.

Now, assume that the claim holds for some $m \in \mathbb{N}$. We show that it also holds for $m + 1$. For this, let $\epsilon \in (0,1/2)$ and let $\Phi = ((A_1, b_1), (A_2, b_2), \ldots, (A_L, b_L))$ be a neural network with $d$-dimensional input and $n$-dimensional output, where $n \leq m + 1$, and with number of layers, neurons and weights all bounded by $K$, where each $A_l$ is an $N_l \times \sum_{k=0}^{l-1} N_k$ matrix, and $b_l \in \mathbb{R}^{N_l}$ for $l = 1, \ldots, L$.

Case 1: If $n \leq m$, then we use the induction hypothesis and get that there is a constant $C_0 = C_0(m) > 0$ and a neural network $\Psi_{\epsilon,\Phi}$ with $d$-dimensional input and one-dimensional output, and at most $KC_0\log(1/\epsilon)$ layers, neurons and weights such that

$$
\left\| R_{\phi}(\Psi_{\epsilon,\Phi}) - \prod_{l=1}^{n}[R_{\phi}(\Phi)]_{l} \right\|_{W^{k,\infty}((0,1)^d)} \leq m^{1-k}c_1^kN^k\epsilon \leq (m + 1)^{1-k}c_1^kN^k\epsilon
$$

for $k \in \{0,1\}$ and $c_1 = c_1(d, m)$. Moreover,

$$
R_{\phi}(\Psi_{\epsilon,\Phi})(x) = 0 \quad \text{if} \quad \prod_{l=1}^{n}[R_{\phi}(\Phi)]_{l}(x) = 0,
$$

for any $x \in (0,1)^d$. Furthermore, we have $|R_{\phi}(\Psi_{\epsilon,\Phi})|_{W^{1,\infty}((0,1)^d)} \leq C_1N$, for $C_1 = C_1(d, m)$.

Case 2: Now, we assume that $n = m + 1$ and show the claim for constants $\tilde{C}_0$, $\tilde{c}_1$ and $\tilde{C}_1$ depending on $m + 1$, possibly different from the constants $C_0$, $c_1$ and $C_1$ from Case 1, respectively. The maximum of each pair of constants fulfills then the claim for networks with output dimension $n \leq m + 1$.

We denote by $\Phi_m$ the neural network with $d$-dimensional input and $m$-dimensional output which results from $\Phi$ by removing the last output neuron and corresponding weights. In detail, we write

$$
A_L = \begin{bmatrix} A_L^{(1,m)} \\ a_L^{(m+1)} \end{bmatrix} \quad \text{and} \quad b_L = \begin{bmatrix} b_L^{(1,m)} \\ b_L^{(m+1)} \end{bmatrix},
$$

40
where \( A_L^{(1,m)} \) is a \( m \times \sum_{k=0}^{L-1} N_k \) matrix and \( a_L^{(m+1)} \) is a \( 1 \times \sum_{k=0}^{L-1} N_k \) vector, and \( b_L^{(1,m)} \in \mathbb{R}^m \) and \( b_L^{(m+1)} \in \mathbb{R}^1 \). Now we set
\[
\Phi_m := \left( (A_1, b_1), (A_2, b_2), \ldots, (A_{L-1}, b_{L-1}), (A_L^{(1,m)}, b_L^{(1,m)}) \right).
\]
Using the induction hypothesis and the constants \( C_0, c_1 \) and \( C_1 \) from Case 1, we get that there is a neural network \( \Psi_{\varepsilon, \Phi} = ((A_1', b_1'), (A_2', b_2'), \ldots, (A_L', b_L')) \) with \( d \)-dimensional input and one-dimensional output, and at most \( KC_0 \log_2(1/\varepsilon) \) layers, neurons and weights such that
\[
\left\| R_\varepsilon(\Psi_{\varepsilon, \Phi}) - \prod_{l=1}^m [R_\varepsilon(\Phi_m)]_l \right\|_{W^{k,\infty}((0,1)^d)} \leq m^{1-k} c_1^k N^k \varepsilon \tag{6.26}
\]
for \( k \in \{0, 1\} \). Moreover,
\[
R_\varepsilon(\Psi_{\varepsilon, \Phi})(x) = 0 \quad \text{if} \quad \prod_{l=1}^m [R_\varepsilon(\Phi_m)]_l(x) = 0, \tag{6.27}
\]
for any \( x \in (0,1)^d \). Furthermore, we can assume that \( |R_\varepsilon(\Psi_{\varepsilon, \Phi})|_{W^{1,\infty}((0,1)^d)} \leq C_1 N \), and that the first \( L(\Phi) - 1 \) layers of \( \Psi_{\varepsilon, \Phi} \) and \( \Phi_m \) coincide and, thus, also the first \( L(\Phi) - 1 \) layers of \( \Psi_{\varepsilon, \Phi} \) and \( \Phi \), i.e. \( A_l = A_l' \) for \( l = 1, \ldots, L(\Phi) - 1 \).

Now, we add the formerly removed neuron with corresponding weights back to the last layer of \( \Psi_{\varepsilon, \Phi} \). For the resulting network
\[
\tilde{\Psi}_{\varepsilon, \Phi} := \left( (A_1', b_1'), (A_2', b_2'), \ldots, (A_L', b_L'), \left( \begin{array}{cc}
A_L' \\
n_{L+1} \oplus \sum_{k=L}^{L-1} n_k
\end{array} \right), \left( \begin{array}{c}
b_L' \\
0 \in \mathbb{R}^{(m+1)}
\end{array} \right) \right)
\]
it holds that the first \( L - 1 \) layers of \( \tilde{\Psi}_{\varepsilon, \Phi} \) and \( \Phi \) coincide, and \( \tilde{\Psi}_{\varepsilon, \Phi} \) is a neural network with two-dimensional output. Note that
\[
\left\| [R_\varepsilon(\tilde{\Psi}_{\varepsilon, \Phi})]_1 \right\|_{L^{\infty}((0,1)^d)} = \left\| R_\varepsilon(\Psi_{\varepsilon, \Phi}) \right\|_{L^{\infty}((0,1)^d)}
\leq \left\| R_\varepsilon(\Psi_{\varepsilon, \Phi}) - \prod_{l=1}^m [R_\varepsilon(\Phi_m)]_l \right\|_{L^{\infty}((0,1)^d)} + \left\| \prod_{l=1}^m [R_\varepsilon(\Phi_m)]_l \right\|_{L^{\infty}((0,1)^d)}
\leq m \varepsilon + 1 < m + 1,
\]
where we used Equation \((6.20)\) for \( k = 0 \). Additionally we have
\[
\left\| [R_\varepsilon(\tilde{\Psi}_{\varepsilon, \Phi})]_2 \right\|_{L^{\infty}((0,1)^d)} = \left\| [R_\varepsilon(\Phi)]_{m+1} \right\|_{L^{\infty}((0,1)^d)} \leq 1.
\]
Now, we denote by \( \tilde{x} \) the network from Proposition \([6, 4] \) with \( M = m + 1 \) and accuracy \( \varepsilon \) and define
\[
\Psi_{\varepsilon, \Phi} := \tilde{x} \circ \tilde{\Psi}_{\varepsilon, \Phi}.
\]
Consequently, $\Psi_{\varepsilon, \Phi}$ has $d$-dimensional input, one-dimensional output and, combining the induction hypothesis with statement (iv) of Proposition 6.4 and Remark 3.9, at most
\[ 2KC_0 \log_2(1/\varepsilon) + 2(c'/\log_2(1/\varepsilon) + c'') \leq KC \log_2(1/\varepsilon) \]
layers, number of neurons and weights. Here $c'$ and $c'' = c''(m + 1)$ are the constants from Proposition 6.4 (iv) for the choice $M = m + 1$ and $C = C(m + 1) > 0$ is a suitable constant.

Clearly, the first $L - 1$ layers of $\Psi_{\varepsilon, \Phi}$ and $\Phi$ coincide and for the approximation properties it holds that
\[ R_{\varepsilon}(\Psi_{\varepsilon, \Phi}) = \prod_{l=1}^{m+1} [R_{\varepsilon}(\Phi)]_{\ell} \]
\[ = R_{\varepsilon}(\tilde{x}) \circ R_{\varepsilon}(\tilde{\Psi}_{\varepsilon, \Phi}) - [R_{\varepsilon}(\Phi)]_{m+1} \cdot \prod_{l=1}^{m} [R_{\varepsilon}(\Phi)]_{\ell} \]
\[ \leq R_{\varepsilon}(\tilde{x}) \circ (R_{\varepsilon}(\tilde{\Psi}_{\varepsilon, \Phi}), [R_{\varepsilon}(\Phi)]_{m+1}) - R_{\varepsilon}(\tilde{\psi}_{\varepsilon, \Phi}) \cdot [R_{\varepsilon}(\Phi)]_{m+1} \]
\[ + \left[ [R_{\varepsilon}(\Phi)]_{m+1} \cdot (R_{\varepsilon}(\tilde{\psi}_{\varepsilon, \Phi}) - \prod_{l=1}^{m} [R_{\varepsilon}(\Phi)]_{\ell}) \right]_{W^{k, \infty}(0, 1)^d}, \quad (6.28) \]
for $k \in \{0, 1\}$. We continue by considering the first term of the Inequality (6.28) for $k = 0$ and obtain
\[ \left\| R_{\varepsilon}(\tilde{x}) \circ (R_{\varepsilon}(\tilde{\psi}_{\varepsilon, \Phi}), [R_{\varepsilon}(\Phi)]_{m+1}) - R_{\varepsilon}(\tilde{\psi}_{\varepsilon, \Phi}) \cdot [R_{\varepsilon}(\Phi)]_{m+1} \right\|_{L^\infty((0, 1)^d)} \]
\[ \leq \| R_{\varepsilon}(\tilde{x})(x, y) - x \cdot y \|_{L^\infty((-m+1,m+1)^2; dxdy)} \]
\[ \leq \varepsilon. \quad (6.29) \]

Next, we consider the same term for $k = 1$ and apply the chain rule from Corollary 5.13. For this, let $\tilde{C} = \tilde{C}(d)$ be the constant from Corollary 5.13 (for $n = d$ and $m = 2$). We get
\[ \left[ R_{\varepsilon}(\tilde{x}) \circ (R_{\varepsilon}(\tilde{\psi}_{\varepsilon, \Phi}), [R_{\varepsilon}(\Phi)]_{m+1}) - R_{\varepsilon}(\tilde{\psi}_{\varepsilon, \Phi}) \cdot [R_{\varepsilon}(\Phi)]_{m+1} \right]_{W^{1, \infty}(0, 1)^d} \]
\[ \leq \tilde{C} \cdot \| R_{\varepsilon}(\tilde{x})(x, y) - x \cdot y \|_{W^{1, \infty}((-m+1,m+1)^2; dxdy)} \]
\[ \leq \tilde{C} \cdot \varepsilon \max\{C_1 N, N\} \]
\[ = C'_1 \varepsilon N, \quad (6.30) \]
where we used the induction hypothesis together with $\| [R_{\varepsilon}(\Phi)]_{m+1} \|_{W^{1, \infty}(0, 1)^d} \leq N$ in the third step, and $C'_1 = C'_1(d, m + 1) > 0$ is a suitable constant.
To estimate the second term of (6.28) for \( k = 0 \) we use the induction hypothesis (for \( k = 0 \)) and get
\[
\left\| R_\varrho(\Phi)_{m+1} \cdot \left( R_\varrho(\Psi_\varepsilon,\Phi_m) - \prod_{l=1}^{m} [R_\varrho(\Phi)]_l \right) \right\|_{L^\infty((0,1)^d)} \\
\leq \left\| [R_\varrho(\Phi)]_{m+1} \right\|_{L^\infty((0,1)^d)} \cdot \left\| R_\varrho(\Psi_\varepsilon,\Phi_m) - \prod_{l=1}^{m} [R_\varrho(\Phi)]_l \right\|_{L^\infty((0,1)^d)} \\
\leq 1 \cdot m \cdot \varepsilon. \tag{6.31}
\]

For \( k = 1 \) we apply the product rule from Corollary 5.15 together with \( \| [R_\varrho(\Phi)]_{m+1} \|_{L^\infty} \leq 1 \) and get
\[
\left\| R_\varrho(\Phi)_{m+1} \cdot \left( R_\varrho(\Psi_\varepsilon,\Phi_m) - \prod_{l=1}^{m} [R_\varrho(\Phi)]_l \right) \right\|_{W^{1,\infty}((0,1)^d)} \\
\leq \| [R_\varrho(\Phi)]_{m+1} \|_{W^{1,\infty}((0,1)^d)} \cdot \left\| R_\varrho(\Psi_\varepsilon,\Phi_m) - \prod_{l=1}^{m} [R_\varrho(\Phi)]_l \right\|_{W^{1,\infty}((0,1)^d)} \\
+ \| [R_\varrho(\Phi)]_{m+1} \|_{L^\infty((0,1)^d)} \cdot \left\| R_\varrho(\Psi_\varepsilon,\Phi_m) - \prod_{l=1}^{m} [R_\varrho(\Phi)]_l \right\|_{W^{1,\infty}((0,1)^d)} \\
\leq N \cdot m \varepsilon + 1 \cdot c_1 N \varepsilon = c_1' N \varepsilon, \tag{6.32}
\]
where we used the induction hypothesis for \( k = 1 \), and \( c_1' = c_1'(d,m+1) > 0 \).

Combining (6.28) with (6.29) and (6.31) we have
\[
\left\| R_\varrho(\Psi_\varepsilon,\Phi) - \prod_{l=1}^{m+1} [R_\varrho(\Phi)]_l \right\|_{L^\infty((0,1)^d)} \leq \varepsilon + m \cdot \varepsilon = (m+1) \cdot \varepsilon, \tag{6.33}
\]
and in the same way a combination of (6.28) with (6.30) and (6.32) yields
\[
\left\| R_\varrho(\Psi_\varepsilon,\Phi) - \prod_{l=1}^{m+1} [R_\varrho(\Phi)]_l \right\|_{W^{1,\infty}((0,1)^d)} \leq (C_1' + c_1') \cdot N \cdot \varepsilon = c_1'' N \varepsilon,
\]
where \( c_1'' = c_1''(d,m+1) > 0 \). Putting together the two previous estimates yields
\[
\left\| R_\varrho(\Psi_\varepsilon,\Phi) - \prod_{l=1}^{m+1} [R_\varrho(\Phi)]_l \right\|_{W^{1,\infty}((0,1)^d)} \leq c_1''' N \varepsilon, \tag{6.34}
\]
for a suitable constant \( c_1''' = c_1'''(d,m+1) > 0 \).

We now show Equation (6.25) for \( m+1 \). To this end, assume that \( [R_\varrho(\Phi)]_l(x) = 0 \) for some \( l \in \{1, \ldots, m+1\} \) and \( x \in (0,1)^d \). If \( l \leq m \), then Equation (6.27) implies that
\[
[R_\varrho(\tilde{\Psi}_\varepsilon,\Phi)]_l(x) = R_\varrho(\Psi_\varepsilon,\Phi_m)(x) = 0.
\]
If \( l = m + 1 \), then we have
\[
[R_\varepsilon(\tilde{\Psi},\delta)]_2(x) = [R_\varepsilon(\Phi)]_{m+1}(x) = 0.
\]
Hence, by application of Proposition 6.3, we have
\[
R_\varepsilon(\Psi_\varepsilon,\Phi)(x) = R_\varepsilon(\tilde{\Psi})([R_\varepsilon(\tilde{\Psi},\delta)]_1(x), [R_\varepsilon(\tilde{\Psi},\delta)]_2(x)) = 0.
\]
Finally, we need to show that there is a constant \( C''_4 = C'_4(d, m + 1) > 0 \) such that
\[
|R_\varepsilon(\Psi_\varepsilon,\Phi)|_{W^{1,\infty}((0,1)^d)} \leq C''_4 N.
\]
Similarly as in (6.30) we have
\[
|R_\varepsilon(\Psi_\varepsilon,\Phi)|_{W^{1,\infty}((0,1)^d)} = \left| R_\varepsilon(\tilde{\Psi}) \circ R_\varepsilon(\tilde{\Psi},\delta) \right|_{W^{1,\infty}((0,1)^d)} \leq \hat{C} \cdot |R_\varepsilon(\tilde{\Psi})|_{W^{1,\infty}((0,1)^d; \mathbb{R}^2)} \leq \hat{C} \cdot \hat{c} \cdot (m + 1) \cdot \max \{ C_1 N, N \} = C''_4 N,
\]
where Corollary 5.13 was used for the second step and \( \hat{c} \) is the constant from Proposition 6.4 which together with an argument as in (6.30) implies the third step. Here, \( C''_4 = C_4'(d, m + 1) \) is a suitable constant.

Taking the maximum of the each pair of constants derived in Case 1 and Case 2 concludes the proof. \( \square \)

In the next lemma, we approximate a sum of localized polynomials with a neural network. One of the difficulties is to control the derivative of the localizing functions from the partition of unity.

Lemma 6.9. Let \( d, n, N \in \mathbb{N} \), let \( B > 0 \) and \( 0 \leq s \leq 1 \), and let \( \Psi = \Psi(d, N) = \{ \phi_m : m \in \{0, \ldots, N\}^d \} \) be the partition of unity from Lemma 5.4. Then, there are constants \( C_1 = C_1(n, d, B, s) > 0 \) and \( C_2 = C_2(n, d) \), \( C_3 = C_3(n, d) > 0 \) such that for any sequence of coefficients \( \{ c_{m,\alpha} : m \in \{0, \ldots, N\}^d, \alpha \in \mathbb{N}_0^d, |\alpha| \leq n - 1 \} \subset [-B, B] \) and any \( \varepsilon \in (0, 1/2) \) there is a neural network \( \Phi_{P, \varepsilon} \) (independent of \( s \)) with \( d \)-dimensional input and one-dimensional output, with at most \( C_2 \cdot \log_2(1/\varepsilon) \) layers and \( C_3(N + 1)^d \log_2(1/\varepsilon) \) weights and neurons, such that
\[
\left\| \sum_{m \in \{0, \ldots, N\}^d} \phi_m p_m - R_\varepsilon(\Phi_{P, \varepsilon}) \right\|_{W^{s,\infty}((0,1)^d)} \leq C_1 N^s \varepsilon,
\]
where \( p_m(x) = \sum_{|\alpha| \leq n-1} c_{m,\alpha} x^\alpha \).

Furthermore, for the architecture \( A \) of \( \Phi_{P, \varepsilon} \) it holds that \( A = A(d, n, N) \), i.e. the architecture of \( \Phi_{P, \varepsilon} \) is independent of the elements \( c_{m,\alpha} \) of the sequence of coefficients.

Proof. It is easy to see that there is a neural network \( \Phi_{\alpha} \) with \( d \)-dimensional input and \(|\alpha|\)-dimensional output, with one layer and at most \( n - 1 \) weights and \( n - 1 + d \) neurons such that
\[
x^\alpha = \prod_{i=1}^{n} [R_\varepsilon(\Phi_{\alpha})]_i(x) \quad \text{for all } x \in (0,1)^d
\]
and 
\[ \| [R_\varepsilon(\Phi_\alpha)]_l \|_{W^{k,\infty}((0,1)^d)} \leq 1 \quad \text{for all } l = 1, \ldots, |\alpha| \text{ and } k \in \{0,1\}. \] (6.35)

Let now \( \Phi_{\varepsilon m} \) be the neural network and \( C, c \geq 1 \) the constants from Lemma 6.5 (v) and define the network
\[ \Phi_{m,\alpha} := \Phi_{\varepsilon m} \left( \Phi_{m,\alpha} \right). \]

Then \( \Phi_{m,\alpha} \) has at most \( 3 \leq K_0 \) layers, \( Cd + n - 1 \leq K_0 \) nonzero weights, and \( Cd + n - 1 + d \leq K_0 \) neurons for a suitable constant \( K_0 = K_0(n,d) \in \mathbb{N} \), and
\[ \prod_{l=1}^{|\alpha|+d} \| [R_\varepsilon(\Phi_{m,\alpha})]_l(x) \|_{L^2} = m,\alpha \text{ for all } x \in (0,1)^d. \] Moreover, as a consequence of Lemma 6.5 (v) together with Equation (6.35) we have
\[ \| [R_\varepsilon(\Phi_{m,\alpha})]_l \|_{W^{k,\infty}((0,1)^d)} \leq (cN)^k \quad \text{for all } l = 1, \ldots, |\alpha| + d \text{ and for } k \in \{0,1\}. \]

To construct an approximation of the localized monomials \( \phi_{m,\alpha} \), let \( \Psi_{\varepsilon,m,\alpha} \) be the neural network provided by Lemma 6.8 (with \( \Phi_{m,\alpha} \) instead of \( \Phi \), \( m = n - 1 + d \in \mathbb{N} \), \( K = K_0 \in \mathbb{N} \) and \( cN \) instead of \( N \)) for \( m \in \{0, \ldots, N\}^d \) and \( \alpha \in \mathbb{N}_0^d, |\alpha| \leq n - 1 \). There exists a constant \( C_0 = C_0(n,d) > 0 \) such that \( \Psi_{\varepsilon,m,\alpha} \) has at most \( K_0C_0 \log_2(1/\varepsilon) \leq C_1 \log_2(1/\varepsilon) \) layers, number of neurons and weights. Here \( C_1 = C_1(n,d) \geq 1 \) is a suitable constant. Moreover,
\[ \| \phi_{m}(x)x^\alpha - R_\varepsilon(\Psi_{\varepsilon,m,\alpha})(x) \|_{W^{k,\infty}((0,1)^d, dx)} \leq c'N^k\varepsilon \] (6.36)
for a constant \( c' = c'(n,d,k) > 0 \) and \( k \in \{0,1\} \), and
\[ R_\varepsilon(\Psi_{\varepsilon,m,\alpha})(x) = 0 \quad \text{if} \quad \phi_{m}(x)x^\alpha = 0 \quad \text{for all} \quad x \in (0,1)^d. \] (6.37)

We set
\[ M := |\{(m, \alpha) : m \in \{0, \ldots, N\}^d, \alpha \in \mathbb{N}_0^d, |\alpha| \leq n - 1\}| \]
and define the matrix \( A_{\text{sum}} \in \mathbb{R}^{1 \times M} \) with \( A_{\text{sum}} = [c_{m,\alpha} : m \in \{0, \ldots, N\}^d, \alpha \in \mathbb{N}_0^d, |\alpha| \leq n - 1] \) and the neural network \( \Phi_{\text{sum}} := ((A_{\text{sum}}, 0)) \). Finally, we set
\[ \Phi_{\varepsilon,m} := \Phi_{\text{sum}} \circ P(\Psi_{\varepsilon,m,\alpha} : m \in \{0, \ldots, N\}^d, \alpha \in \mathbb{N}_0^d, |\alpha| \leq n - 1). \]

Then, \( \Phi_{\varepsilon,m} \) is a neural network with \( d \)-dimensional input and one-dimensional output, with at most \( 1 + C_1 \log_2(1/\varepsilon) \leq C_2 \log_2(1/\varepsilon) \) layers,
\[ 2(M + MC_1 \log_2(1/\varepsilon)) \leq 4MC_1 \log_2(1/\varepsilon) \leq C_3(N + 1)^d \log_2(1/\varepsilon) \]
nonzero weights and neurons for constants \( C_2 = C_2(n,d), C_3 = C_3(n,d) > 0 \), and
\[ R_\varepsilon(\Phi_{\varepsilon,m}) = \sum_{m \in \{0, \ldots, N\}^d} \sum_{|\alpha| \leq n - 1} c_{m,\alpha} \Psi_{\varepsilon,m,\alpha}. \]

Clearly, the architecture of \( \Phi_{\varepsilon,m} \) is independent of the elements \( c_{m,\alpha} \) of the sequence of coefficients.

For each \( m \in \{0, \ldots, N\}^d \) we set
\[ \Omega_{m,N} := B_\frac{1}{\pi} \| \frac{m}{N} \| \left( \frac{m}{N} \right) \]

45
and get for $k \in \{0, 1\}$

$$
\left\| \sum_{m \in \{0, \ldots, N\}^d} \phi_m(x) p_m(x) - R_\epsilon(\Phi_{P, \epsilon})(x) \right\|_{W^{k, \infty}((0,1)^d; dx)}
$$

$$
= \left\| \sum_{m \in \{0, \ldots, N\}^d} \sum_{|\alpha| \leq n-1} c_{m, \alpha} \left( \phi_m(x)x^\alpha - R_\epsilon(\Psi_{\epsilon, (m, \alpha)})(x) \right) \right\|_{W^{k, \infty}((0,1)^d; dx)}
$$

$$
= \max_{\tilde{m} \in \{0, \ldots, N\}^d} \left\| \sum_{m \in \{0, \ldots, N\}^d} \sum_{|\alpha| \leq n-1} c_{m, \alpha} \left( \phi_m(x)x^\alpha - R_\epsilon(\Psi_{\epsilon, (m, \alpha)})(x) \right) \right\|_{W^{k, \infty}(\tilde{\Omega}_{\tilde{m}} \cap (0,1)^d; dx)},
$$

where the last step is a consequence of $(0,1)^d \subset \bigcup_{\tilde{m} \in \{0, \ldots, N\}^d} \Omega_{\tilde{m}}$. For $\tilde{m} \in \{0, \ldots, N\}^d$ we have

$$
\left\| \sum_{m \in \{0, \ldots, N\}^d} \sum_{|\alpha| \leq n-1} c_{m, \alpha} \left( \phi_m(x)x^\alpha - R_\epsilon(\Psi_{\epsilon, (m, \alpha)})(x) \right) \right\|_{W^{k, \infty}(\tilde{\Omega}_{\tilde{m}} \cap (0,1)^d; dx)}
$$

$$
\leq B \sum_{m \in \{0, \ldots, N\}^d} \sum_{|\alpha| \leq n-1} \left\| \phi_m(x)x^\alpha - R_\epsilon(\Psi_{\epsilon, (m, \alpha)})(x) \right\|_{W^{k, \infty}(\tilde{\Omega}_{\tilde{m}} \cap (0,1)^d; dx)}
$$

$$
\leq B 3^d \max_{\|\tilde{m} - m\|_\infty \leq 1} \sum_{|\alpha| \leq n-1} \left\| \phi_m(x)x^\alpha - R_\epsilon(\Psi_{\epsilon, (m, \alpha)})(x) \right\|_{W^{k, \infty}((0,1)^d; dx)},
$$

where we used Lemma 6.5 and 6.6 together with Equation 6.37 for the last step. A combination of the last two estimates together with Equation 6.36 yields

$$
\left\| \sum_{m \in \{0, \ldots, N\}^d} \phi_m(x) p_m(x) - R_\epsilon(\Phi_{P, \epsilon})(x) \right\|_{W^{k, \infty}((0,1)^d; dx)} \leq B 3^d n^d c' N^k \epsilon
$$

$$
= c'' N^k \epsilon,
$$

for $c'' = c''(n,d,B,k)$. Hence the case $s = 0$ and $s = 1$ is proven. To show the general statement for $0 \leq s \leq 1$ we use the interpolation inequality from Corollary 4.6 and directly get

$$
\left\| \sum_{m \in \{0, \ldots, N\}^d} \phi_m(x) p_m(x) - R_\epsilon(\Phi_{P, \epsilon})(x) \right\|_{W^{s, \infty}((0,1)^d; dx)} \leq c''' N^s \epsilon
$$

for a constant $c''' = c'''(n,d,B,s) > 0$. This concludes the proof of the Lemma.

Finally, we are ready to proof the main result.

**Proof of Theorem 6.1.** The proof can be divided into two steps: First, we approximate the function $f$ by a sum of localized polynomials and then approximate this sum by a network.
For the first step, we set
\[ N := \left\lceil \left( \frac{\varepsilon}{2CB} \right)^{-1/(n-s)} \right\rceil, \tag{6.38} \]
where \( C = C(n, d) > 0 \) is the constant from Corollary 6.7. Without loss of generality we may assume that \( CB \geq 1 \). The same corollary yields that if \( \Psi = \Psi(d, N) = \{ \phi_m : m \in \{0, \ldots, N\}^d \} \) is the partition of unity from Lemma 6.5, then there exist polynomials \( p_m(x) = \sum_{|\alpha| \leq n-1} c_{m, \alpha} x^\alpha \) for \( m \in \{0, \ldots, N\}^d \) such that
\[
\left\| f - \sum_{m \in \{0, \ldots, N\}^d} \phi_m p_m \right\|_{W^{s,\infty}((0,1)^d)} \leq CB \left( \frac{1}{N} \right)^{n-s} \leq CB \frac{\varepsilon}{2CB} = \frac{\varepsilon}{2}, \tag{6.39} \]
Moreover, we have \( \{c_{m, \alpha} : m \in \{0, \ldots, N\}^d, \alpha \in \mathbb{N}_0^d, |\alpha| \leq n-1\} \subset [-Bc_0, Bc_0] \) for some constant \( c_0 = c_0(n, d) > 0 \).

For the second step, let \( C_1 = C_1(n, d, B, s) > 0 \), \( C_2 = C_2(n, d) > 0 \) and \( C_3 = C_3(n, d) > 0 \) be the constants from Lemma 6.9 and \( \Phi_{P, \varepsilon} \) be the neural network provided by Lemma 6.9 with \( \varepsilon^{n/(n-s)}/(8CBC_1) \) instead of \( \varepsilon \) and \( Bc_0 \) instead of \( B \). Then \( \Phi_{P, \varepsilon} \) has at most
\[
C_2 \log_2 \left( 8CBC_1 \varepsilon^{-n/(n-s)} \right) \leq C_2 \left( \log_2(8CBC_1) + \log_2 \left( \varepsilon^{-n/(n-s)} \right) \right) \leq C' \log_2 \left( \varepsilon^{-n/(n-s)} \right) \]
layers for a constant \( C' = C'(n, d, B, s) > 0 \) and at most
\[
C_3 \left( \left( \frac{\varepsilon}{2CB} \right)^{-1/(n-s)} + 2 \right)^d \log_2 \left( 8CBC_1 \varepsilon^{-n/(n-s)} \right) \leq C3 \left( \frac{\varepsilon}{2CB} \right)^{-d/(n-s)} \log_2 \left( 8CBC_1 \varepsilon^{-n/(n-s)} \right) \leq C'' \varepsilon^{-d/(n-s)} \log_2 \left( \varepsilon^{-n/(n-s)} \right) \]
nonzero weights and neurons. Here \( C'' = C''(n, d, B, s) \) is a suitable constant and we used \((2CB)/\varepsilon \geq 1\) in the first step. It holds that the architecture of \( \Phi_{P, \varepsilon} \) is independent of the function \( f \). Furthermore, we have
\[
\left\| \sum_{m \in \{0, \ldots, N\}^d} \phi_m p_m - R_\mathcal{E}(\Phi_{P, \varepsilon}) \right\|_{W^{s,\infty}((0,1)^d)} \leq C_1 N^s \frac{\varepsilon^{n/(n-s)}}{8CBC_1},
\]
\[
\leq \left( \left( \frac{\varepsilon}{2CB} \right)^{-1/(n-s)} + 1 \right) \varepsilon^{n/(n-s)} \frac{8CB}{8CBC_1},
\]
\[
\leq \left( \left( \frac{\varepsilon}{2CB} \right)^{-s/(n-s)} + 1 \right) \varepsilon^{n/(n-s)} \frac{8CB}{8CBC_1},
\]
where we used the inequality \((x + y)^p \leq x^p + y^p\) for \(x, y \geq 0\) and \(0 \leq p \leq 1\) in the last step. We now continue the above computation using that \(s/(n-s) \leq 1\) and \(2CB \geq 1\) and therefore \((2CB)^{s/(n-s)} \leq 2CB\) and get

\[
\left(\left(\frac{\varepsilon}{2CB}\right)^{-s/(n-s)} + 1\right) \frac{\varepsilon n/(n-s)}{8} \leq \frac{1}{8CB} \left(2CBe + \varepsilon n/(n-s)\right)
\leq \frac{1}{8CB} (2CB\varepsilon + 2B\varepsilon)
= \frac{\varepsilon}{2}.
\]

Combining the previous computations we get

\[
\left\| \sum_{m \in \{0, \ldots, N\}^d} \phi_m p_m - R_{Q}(\Phi, \varepsilon) \right\|_{W^{s,\infty}((0,1)^d)} \leq \frac{\varepsilon}{2},
\tag{6.40}
\]

Using the triangle inequality and Equations (6.39) and (6.40) we finally obtain

\[
\|f - R_{Q}(\Phi, \varepsilon)\|_{W^{s,\infty}((0,1)^d)}
\leq \|f - \sum_{m \in \{0, \ldots, N\}^d} \phi_m p_m\|_{W^{s,\infty}((0,1)^d)} + \|\sum_{m \in \{0, \ldots, N\}^d} \phi_m p_m - R_{Q}(\Phi, \varepsilon)\|_{W^{s,\infty}((0,1)^d)}
\leq \frac{\varepsilon}{2} + \frac{\varepsilon}{2} = \varepsilon,
\]

which concludes the proof. \(\square\)
7 Discussion and Future Work

In the end, the motivation behind deriving theoretical results for deep neural networks is to improve our understanding of their empirical success in a broad area of applications. In this section, we briefly describe the practical relevance of our result, but mostly focus on aspects where our theoretical framework differs from empirical observations or fails to incorporate computational limitations met in real implementations. Furthermore, we will discuss possible extensions of our results.

- **Practical relevance.** Our result gives upper bounds for the complexity of networks resulting from Sobolev training (see Section 1.3 and [14]). Furthermore, we think that it can be used to integrate complexity bounds in the theoretical foundation of using deep learning-based approaches for the approximative solution of PDEs. We leave an in-depth analysis of this topic for future research.

- **Curse of dimension.** One of the main reasons for the current interest in deep neural networks is their outstanding performance in high-dimensional problems. As an example, consider the popular ILSVRC challenge (see [44]), an image recognition task on the ImageNet database (see [18]) containing variable-resolution images which are typically downsampled before training to yield a roughly $(240 \times 240 \times 3)$-dimensional input space (cf. e.g. [33, 47, 56]).

  Our asymptotic upper complexity bounds for the number of weights and neurons are in $O(\varepsilon^{-d/(n-s)})$, and thus depend strongly on the dimension $d$ of the input space. Moreover, the constants in the upper bounds for the number of layers, weights, and neurons also increase exponentially with increasing $d$.

  Several ways have been proposed so far to tackle this common problem. One idea is to think of the data as residing on or near a manifold $\mathcal{M}$ embedded in $\mathbb{R}^d$ with dimension $d_{\mathcal{M}} \ll d$ (see [5]). Thinking, for example, of image classification problems this idea seems to be rather intuitive, since most of the elements in $\mathbb{R}^{240 \times 240 \times 3}$ are not perceived as images. A similar approach is to narrow down the approximated function space by incorporating invariances (see [37]). If, for example, the approximated function $f$ maps images to labels, then it makes sense to assume that $f$ is translation and rotation invariant. The additional structure of the function space can then be exploited in the approximation (see [41, Section 5]). In particular, the last approach could also be suitable to tackle the curse of dimension in our result.

- **The power of depth.** For $s = 0$, the approximations obtained in Theorem 6.1 and [54, Theorem 1] coincide. In this case, Yarotsky showed in [54] that the constructed unbounded depth approximations for functions in $W^{n,\infty}((0,1)^d)$ (with $n > 2$) are asymptotically more efficient in the number of weights and neurons than approximations with fixed length $L$ if

  \[
  \frac{d}{n} \leq \frac{1}{2(L - 1)}.
  \]

  As a consequence, to be more efficient than a shallow network, i.e. a network with depth $L = 2$, one needs $n > 2d$ regularity. Even if this result does not completely

---

1Typically abbreviated with ILSRC which stands for ImageNet Large Scale Visual Recognition Challenge.
explain the success of deep networks over shallow ones, since $d$ is typically very large, it would be interesting to obtain similar results for higher-order Sobolev norms.

- **Lower complexity bounds.** By establishing lower complexity bounds in [54, Theorem 3], Yarotsky showed that under the assumption of a fixed network architecture and continuous weight selection the bounds derived in Theorem 6.1 for $s = 0$, which correspond to [54, Theorem 1], are optimal up to a factor $O(\log_2(1/\varepsilon))$. It is subject to further research if the same techniques as in that paper can be used to derive lower bounds for approximations in higher-order Sobolev type norms.

- **Unbounded complexity of weights.** When training a neural network on a computer, the weights have to be stored in memory. In practice, storing weights up to an arbitrary precision or even unbounded weights (in absolute value) is infeasible. In the construction of the neural network $R_{\varepsilon}(\Phi_f)$ that approximates a function $f$ up to an $W^{s,\infty}$-error $\varepsilon$ and realizes the upper complexity bound from Theorem 6.1 we used weights $W_\varepsilon$ with $|W_\varepsilon| \to \infty$ as $\varepsilon \downarrow 0$ (see construction of $\Phi_{m,l}$ in the proof of Lemma 6.5 (v)). In [41], neural networks are restricted to possess quantised weights (see [41, Definition 2.9]) which controls the complexity of the weights depending on the approximation error $\varepsilon$. It would be interesting to see if the upper bounds derived in Theorem 6.1 can also be achieved with quantised weights.

- **General $W^{s,p}$.** Results for approximations in $L^2$ norm in a setting similar as in this thesis were, for example, obtained in [41]. In particular, from the perspective of possibly applying our result in the context of partial differential equations, complexity bounds for approximations in general $W^{s,p}$ with $1 \leq p < \infty$ are desirable.

To conclude, we briefly summarize our result. We presented a network calculus for a general type of feedforward ReLU neural networks and derived upper complexity bounds for approximations of functions from the Sobolev space $W^{n,\infty}((0,1)^d)$ with deep ReLU networks in $W^{s,\infty}$ norm, where $0 \leq s \leq 1$. 

50
References


Deutsche Zusammenfassung

Auf Grund der immensen Menge an verfügbaren Daten und ausreichender Rechenkapazität, übertreffen künstliche neuronale Netze heutzutage sowohl klassische Modellierungsverfahren, basierend beispielsweise auf partiellen Differentialgleichungen, als auch konventionelle Algorithmen aus dem Bereich des Maschinelles Lernens, in einem breiten Anwendungsspektrum. Zahlreiche Beispiele finden sich in dem Bereich der Bildverarbeitung oder der Computerlinguistik.


Ein aktuelles Forschungsgebiet ist dabei die Expressivität neuronaler Netze. Dabei wird untersucht, welche Funktionen durch neuronale Netze approximiert werden können und wie die Komplexität des approximierenden Netzwerkes mit der Approximationsgüte in Zusammenhang steht.

In dieser Arbeit verallgemeinern wir ein in [54] bewiesenes Approximationsresultat für neuronale Netze. Wir werden zeigen, dass die in [54] gezeigten oberen Komplexitätsschranken für Approximationen von Funktionen aus dem Sobolevräum $W^{n,\infty}$ mit neuronalen Netzen in der $L^\infty$-Norm für $n \geq 2$ in ähnlicher Form auch für Approximationen in der Sobolevnorm $W^{s,\infty}$ für $0 \leq s \leq 1$ gelten. Dabei muss die Regularität $s$ der Norm in die Schranke miteinbezogen werden, wodurch sich ein Trade-off aus der in der Approximationsnorm verwendeten Regularität und der in der Schranke verwendeten Regularität ergibt.