Approximation Spaces of Deep Neural Networks

Julie Havbo Lund
Mads Bjerregaard Kjær

Master's thesis, Mathematics

Department of Mathematical Sciences
Skjernvej 4A
DK-9220 Aalborg Ø
http://math.aau.dk

AALBORG UNIVERSITY
STUDENT REPORT

Title
Approximation Spaces of
Deep Neural Networks

Theme
Analysis: Approximation Theory
& Machine Learning

Project Period
Fall Semester 2023 &
Spring Semester 2024

Participants
Julie Havbo Lund
Mads Bjerregaard Kjær

Supervisor
Morten Nielsen

Numbered Pages
91

Numbered Pages with Appendix
111

Date of Completion
May 30, 2024

Abstract

This master's thesis explores the relationship between the structure of deep neural networks and their expressivity, which is the ability to approximate functions from certain function classes. Approximation spaces are used as a theoretical framework for the expressivity and the complexity of the neural networks are measured by either the number of connections or the number of neurons. This thesis is divided into different segments; at first, neural networks and their activation functions are explored, and some elementary properties are considered. Then B-splines are introduced and connected to neural networks by proving that B-splines can be realized or approximated by neural networks if these networks are sufficiently complex. The results regarding B-splines are combined with B-spline approximation to construct a neural network with a structure equivalent to B-spline approximation. The performance of the equivalent neural network is explored in an experiment with simulated target functions. Lastly, approximation spaces related to the neural networks are established using general approximation theory. These approximation spaces for the ReLU activation function are discussed, and topics for further analysis are highlighted.

Contents

Preface

This master's thesis has been prepared by two students at the Department of Mathematical Sciences at Aalborg University. This thesis was prepared during the third and fourth semesters of the Mathematics Master's program in the period from September 1 2023, to June 3 2024. The project's theme is "Analysis" with a focus on "Approximation Spaces of Deep Neural Networks". The purpose of this thesis is, among other things, to be able to understand and reflect scientifically on the field of mathematical knowledge, as well as to build a project report according to the subject's norms. The academic purpose is to explore certain mathematical aspects of the structure of deep neural networks.

This thesis is numbered in sections, definitions, figures, etc. Furthermore, proofs are completed with ■ indicating that the proof is complete. The sources are cited using the IEEE method and can be found together at the back of this thesis. The notation environment is used for notation, which is always used in connection with a specific assumption, for example "*Let* Φ *be a neural network*", after which the assumption is not repeated in connection with the notation. These notations can also be found in the symbol overview. If more than one of the same symbol is needed within the same context, then a $\sim$ or an index is added. Combined with the notation environment, this means that if both Φ and $\tilde{\Phi}$ are used for neural network, the underlying sentence "*Let* Φ *and* $\tilde{\Phi}$ *be neural networks.*" is omitted. As a supplementary to some of the proofs illustrations are created using *Tikz* and the Python package `matplotlib`. These can differ from the notation in the proof but are chosen so they are simpler and easier to compare. The code constructed for the implementation is written in `Python 3.12.1`.

We want to thank Morten Nielsen, for being our supervisor throughout this master's thesis. We are also grateful for Morten being our supervisor or censor throughout every project since our bachelor's thesis. He is the best supervisor we could wish for. Additionally, we want to thank our co-students for creating an enjoyable studying environment and showing us that we have no reason to stress about deadlines.

Signatures

[illegible] [illegible]

Julie Havbo Lund Mads Bjerregaard Kjær

Introduction

Neural networks are a huge and developing area, impacting numerous fields of application, such as approximation, [12, p. 260]. Approximation using neural networks has demonstrated remarkable efficiency and is today a building block for many numerical algorithms in machine learning and artificial intelligence, [8, p. 328]. Going back, neural networks were developed in 1943 by Warren McCulloch and Walter Pitts, [12, p. 260]. Based on mathematics and algorithms, neural networks was used to introduce a theoretical framework for artificial intelligence. Today, the field has grown due to the significantly increased computer performance, and neural networks have become a major influence on a variety of machine learning problems, [6, p. 128]. Prominent examples are in computer vision, such as self-driving cars, in natural language processing, such as Google Translate, and in reinforcement learning, such as superhuman performance at Go, [8]. Moreover, fields of mathematics have been affected. Examples of these are inverse problems and numerical analysis of partial differential equations, [12, p. 260].

However, despite these empirical successes, the area is still under development, and neural networks are not yet well understood, [6], [8], [12]. Such as a profound and comprehensive mathematical foundation is still missing, and questions as to how and why they work could lead to practical improvements. Thus, there are still areas that are not properly covered. For a start, the understanding of the approximation properties of neural networks is important, since approximation is one of the main components of many algorithms design, [8]. Such an approach can lead to a better understanding of the expressivity of deep neural networks, which is the ability of deep neural networks to approximate functions from certain function classes. To be able to answer this question, an approach is to make a general theoretical framework, which is the focus of this report.

The main focus of this report is to explore certain mathematical aspects of the structure of deep neural networks. Here approximation spaces are used to provide a theoretical framework for studying the expressivity of deep neural networks. To associate an approximation space to a neural network, a measure of the neural network's complexity is required. This complexity can be given by the number of connections or by the number of neurons. Moreover, for ReLU-networks a relation to B-splines is established. This relation shows that B-splines can be well approximated by neural networks if these networks are sufficiently complex. The theory resulting from this approach will be attempted to be replicated in practice by measuring the precision of neural networks implemented with varying structures. This is included in a larger experiment that has the goal of connecting the purely theoretical results to application.

Despite the large and exciting area to work in, the report must be limited, but some fundamental work can be found in [12], [8], [10], [9], and [6]. Some of the simpler proofs, and proofs similar to those already stated are omitted from the report, but for the sake of the reader, they are included in Appendix B. Additionally, Appendix C will be used for definitions, theorems, etc. that are referred to during the report. The purpose of this is to let the reader view the referenced theory without having to track down the source. In addition, some errata to [12] are highlighted in Appendix D.

Chapter Overview

Below is a brief overview of the content of each chapter:

Chapter I - Intro to Approximation Theory:
This chapter introduces relevant terms within approximation theory, which are used throughout the report.

Chapter II - Neural Networks:
Neural networks and their realizations are introduced together with the notion of complexity in terms of the number of layers, connections, and neurons that neural networks consist of. Relations between these measures of complexity and the realizations of neural networks are explored. Additionally, the methods for combining and transforming neural networks are considered, along with the effect on a given neural network's complexity.

Chapter III - Activation Functions:
This chapter deals with activation functions and relevant properties. One such property is when an activation function can realize another activation function or the identity. What this means for the set of realizations is studied in both cases. Specifically, ReLU and its powers are highlighted.

Chapter IV - Realization of B-splines:
The chapter introduces B-splines, along with a few immediate properties, and an important decomposition. This decomposition is used to connect B-splines to the realization of neural networks.

Chapter V - Experiments:
Based on the result from the previous chapter two experiments are constructed, including an implementation of one of the main results in practice. The initial experiment attempts to construct neural networks, that can realize B-splines. The main experiment attempts to assess the performance of an alternative approximation approach using neural networks compared with the traditional B-spline approximation.

Chapter VI - Approximation Spaces:
The chapter provides an introduction to approximation theory and associates it with the knowledge from Chapters II and III. Approximation classes are constructed in a general setting and proven to be approximation spaces. Similar approximation classes are defined for functions associated with neural networks, and the general setup is used to prove that these are approximation spaces as well.

Chapter VII - Additional Discussion Points:
The chapter provides an overall introduction to further analysis in the focal area of the report. This includes discussing similar results and topics that are relevant to the findings established in this master's thesis.

Notation

Setup:

d	Input dimension.
k	Output dimension.
$\mathbb{N}$	Natural numbers.
$\mathbb{N}_0$	$\mathbb{N} \cup \{0\}$.
$\mathbb{N}_{\geq m}$	$\{n \in \mathbb{N} \mid n \geq m\}$.
$\mathcal{O}$	Big O-notation.
$\lfloor \cdot \rfloor$	Round down to the nearest integer.
$\lceil \cdot \rceil$	Round up to the nearest integer.
Ω	A subset of $\mathbb{R}^d$.
$\otimes$	Modified tensor product, see Definition II.1.
$\lesssim$	Less than or equivalent to.
$f(x) \xrightarrow[x \to 0]{} y$	$\lim_{x \to 0} f(x) = y$.
$f_m \xrightarrow[m \to c]{} f$	$\lim_{m \to c} f_m = f$, for $c \in \mathbb{R} \cup \{-\infty, \infty\}$.

Neural Network:

Φ	Neural network.
$d_{in}(\Phi)$	Input dimension of the neural network Φ.
$d_{out}(\Phi)$	Output dimension of the neural network Φ.
ϱ	Activation function.
$\varrho_j^{(\ell)}$	Activation function or identity function.
α_ℓ	Vector with entries $\varrho_j^{(\ell)}$.
ϱ_r	Rectified linear unit, ReLU and its power of exponent $r \in \mathbb{N}_0$.
T_ℓ	Affine-linear maps.
$A^{(\ell)}$	Weight matrices.
$b^{(\ell)}$	Bias vectors.
$\\|A\\|_{\ell^0}$	The number of non-zero entries of the matrix A.
$L(\Phi)$	The depth of the neural network Φ; The number of layers.
$L(\Phi) - 1$	The number of hidden layers in the neural network Φ.
$\mathtt{R}(\Phi)$	The realization of the neural network Φ.
$N(\Phi)$	The number of hidden neurons in the neural network Φ.
$W(\Phi)$	The number of connections or weights in the neural network Φ.
$\mathscr{L}$	Depth growth function.
$\mathscr{L}_1 \preceq \mathscr{L}_2$	$\mathscr{L}_2$ dominates $\mathscr{L}_1$, see Definition II.6.
$\mathscr{L}_1 \sim \mathscr{L}_2$	$\mathscr{L}_1$ is equivalent to $\mathscr{L}_2$, that is $\mathscr{L}_1 \preceq \mathscr{L}_2$ and $\mathscr{L}_1 \succeq \mathscr{L}_2$.
$\mathcal{L}$	Supremum of $\mathscr{L}(n)$ for $n \in \mathbb{N}$.

Relevant Sets of Neural Network and their Realization:

$\mathcal{NN}^{\varrho,d,k}_{W,L,N}$	The set of all generalized ϱ-networks Φ.
$\mathcal{SNN}^{\varrho,d,k}_{W,L,N}$	The subset of $\Phi \in \mathcal{NN}^{\varrho,d,k}_{W,L,N}$, which are strict.
$\mathtt{NN}^{\varrho,d,k}_{W,L,N}$	The set of f that can be realized by $\Phi \in \mathcal{NN}^{\varrho,d,k}_{W,L,N}$.
$\mathtt{SNN}^{\varrho,d,k}_{W,L,N}$	The set of f that can be realized by $\Phi \in \mathcal{SNN}^{\varrho,d,k}_{W,L,N}$.
$\mathtt{NN}^{\varrho,d,k}_{W,L,N}(\Omega)$	The set of $f\restriction_{\Omega}$, where $f \in \mathtt{NN}^{\varrho,d,k}_{W,L,N}$.
$\mathtt{SNN}^{\varrho,d,k}_{W,L,N}(\Omega)$	The set of $f\restriction_{\Omega}$, where $f \in \mathtt{SNN}^{\varrho,d,k}_{W,L,N}$.

Approximation Theory:

$\mathcal{X}$	Quasi-Banach space.
$\\|\cdot\\|_{\mathcal{X}}$	Quasi-norm associated with $\mathcal{X}$.
$\mathcal{X}_1 \hookrightarrow \mathcal{X}_2$	$\mathcal{X}_1$ is continuously embedded in $\mathcal{X}_2$, that is $\mathcal{X}_1 \subseteq \mathcal{X}_2$ and $\\|\cdot\\|_{\mathcal{X}_2} \leq C\\|\cdot\\|_{\mathcal{X}_1}$, see Definition VI.3.
$\Sigma := \{\Sigma_n\}_{n\in\mathbb{N}_0}$	Family of Σ_n, where Σ_n are subsets of a quasi-Banach space.
$\mathscr{E}(f,\Sigma_n)_{\mathcal{X}}$	Error of best approximation of $f \in \mathcal{X}$ from $\Sigma_n \subseteq \mathcal{X}$.
$A^{\alpha}_{q}(\mathcal{X},\Sigma)$	Approximation space, see Definition VI.2.

Approximation Theory Associated with Neural Networks:

$\mathtt{W}_n(\mathcal{X},\varrho,\mathscr{L})$	The set $\mathtt{NN}^{\varrho,d,k}_{n,L(n),\infty}(\Omega) \cap \mathcal{X}$.
$\mathtt{N}_n(\mathcal{X},\varrho,\mathscr{L})$	The set $\mathtt{NN}^{\varrho,d,k}_{\infty,L(n),n}(\Omega) \cap \mathcal{X}$.
$W^{\alpha}_{q}(\mathcal{X},\varrho,\mathscr{L})$	$A^{\alpha}_{q}(\mathcal{X},\Sigma)$ where Σ is the family of $\Sigma_n = \mathtt{W}_n(\mathcal{X},\varrho,\mathscr{L})$.
$N^{\alpha}_{q}(\mathcal{X},\varrho,\mathscr{L})$	$A^{\alpha}_{q}(\mathcal{X},\Sigma)$ where Σ is the family of $\Sigma_n = \mathtt{N}_n(\mathcal{X},\varrho,\mathscr{L})$.
$\mathtt{SW}_n(\mathcal{X},\varrho,\mathscr{L})$	The set $\mathtt{SNN}^{\varrho,d,k}_{n,L(n),\infty}(\Omega) \cap \mathcal{X}$.
$\mathtt{SN}_n(\mathcal{X},\varrho,\mathscr{L})$	The set $\mathtt{SNN}^{\varrho,d,k}_{\infty,L(n),n}(\Omega) \cap \mathcal{X}$.
$SW^{\alpha}_{q}(\mathcal{X},\varrho,\mathscr{L})$	$A^{\alpha}_{q}(\mathcal{X},\Sigma)$ where Σ is the family of $\Sigma_n = \mathtt{SW}_n(\mathcal{X},\varrho,\mathscr{L})$.
$SN^{\alpha}_{q}(\mathcal{X},\varrho,\mathscr{L})$	$A^{\alpha}_{q}(\mathcal{X},\Sigma)$ where Σ is the family of $\Sigma_n = \mathtt{SN}_n(\mathcal{X},\varrho,\mathscr{L})$.

Spaces:

X	General space.
$\\| \cdot \\|_X$	Norm associated with the space X.
$(X)^*$	The dual space of the space X.
$\ell^p(X)$	ℓ^p-space of sequences in X.
$L^p(\Omega,\mathbb{R}^n)$	L^p-space of function from $\Omega \subseteq \mathbb{R}^d$ to $\mathbb{R}^n$.
$L^p_{\mathrm{loc}}(\Omega,\mathbb{R}^n)$	The space of locally integrable function in $L^p(\Omega,\mathbb{R}^n)$.
$\mathcal{C}(\Omega,\mathbb{R}^n)$	The space of continuous function from $\Omega \subseteq \mathbb{R}^d$ to $\mathbb{R}^n$.
$\mathcal{C}^k(\Omega,\mathbb{R}^n)$	The space of k-times continuous differentiable function from $\Omega \subseteq \mathbb{R}^d$ to $\mathbb{R}^n$.
$\mathcal{C}^k_c(\Omega,\mathbb{R}^n)$	The space of compact supported functions in $\mathcal{C}^k(\Omega,\mathbb{R}^n)$.
$\mathcal{C}_0(\Omega,\mathbb{R}^n)$	The space of uniformly continuous function from Ω to $\mathbb{R}^n$ that vanish at infinity.
$X^k_p(\Omega)$	The space $L^p(\Omega,\mathbb{R}^k)$ for $0 < p < \infty$, and $\mathcal{C}_0(\Omega,\mathbb{R}^n)$ for $p = \infty$.

$\mathrm{GL}(\mathbb{R}^d)$	The generalized linear space, that consists of invertible real valued matrices.
$\mathbb{R}_{\deg \leq n}[x]$	The space of all polynomials of degree at most $n \in \mathbb{N}_0$.

B-spline:

$\beta_+^{(n)}$	Univariate B-spline of degree $n \in \mathbb{N}_0$.
$\beta_d^{(n)}$	Multivariate B-spline of degree $n \in \mathbb{N}_0$ in dimension $d \in \mathbb{N}$.

I | Intro to Approximation Theory

The study of approximation spaces of deep neural networks requires theories from both approximation theory and neural networks. Therefore, it makes sense to create a foundation from both worlds, establishing a solid base for understanding these spaces. Approximation theory is going to be a tool for understanding neural networks, so only the required tools are introduced. As neural networks are the primary focus, a broader foundation is needed, which is covered in the next chapter. This initial chapter serves as an introduction, guiding the reader through the overall concepts within approximation theory before diving into neural networks.

From the perspective of approximation, a central task is to approximate a function, often seen in applications where the aim is to approximate an unknown function f given a training data set $(x_i, f(x_i))_{i=1}^n$, [12]. This unknown function could for instance encode a classification problem, [3]. Exploring the possibility of discovering this approximating function introduces the realm of approximation theory, [15]. *Approximation theory* includes the studies of how possibly complicated *target functions* can best be approximated with simpler functions, [10]. These simpler functions are called *approximants* and can be created in different ways, corresponding to different *approximation methods*, [8]. Increasing the resolution of the target function can generally only be achieved by increasing the complexity of the approximants, which usually corresponds to the number of parameters. This trade-off between resolution and complexity is another important study within approximation theory. Comparison of approximation methods is typically done using the *rate of approximation*, which is the decrease in error versus the number of parameters in the approximant, [9]. From an applied point of view the computational time needed to construct the approximation is another major component for comparison, [9].

Overall approximation theory provides a framework for expressing the *approximation effiency*, which is how well these target functions can be approximated by the approximants of a certain type. Defining a set of target functions to be measured on the approximation efficiency is called a *model class*. By instead focusing on approximants, the *approximation properties* are in focus, which is how well the approximants can approximate the target function. The *expressivity* of the approximants refers to the ability of the approximants to approximate a target function from a certain class of functions. Compared to the approximation rate, an increase in the expressivity yields a better approximation rate. Unifying target functions based on their ability to be approximated by specific approximants yields an *approximation class*. This class is typically defined by target functions satisfying a specific approximation rate, and if the classes are well-defined spaces, they are called an *approximation space*.

In approximation theory, one usually assumes the target function to be fully accessible. This is in contrast to most real-world problems, where only a training data set is available. Thus using results from approximation theory in application, the performance would additionally depend on the amount and quality of the data. Typically the target function f is in a normed space $\mathcal{X}$ such as L^p-spaces and the approximants are in a small subset of $\mathcal{X}$, [10]. Classic examples of approximants are polynomials, splines, and wavelets, [8]. Another newer popular choice is neural networks. As the title of this master's thesis indicates, neural networks are the approximants of focus in this report. More specifically with emphasis on the expressivity of neural networks, which is explored through approximation spaces.

II | Neural Networks

To explore the expressivity of neural networks, the definition of the neural network itself is needed. This chapter presents this definition and the most basic terminology regarding neural networks. This includes the exploration of sets of neural networks and their properties. This foundation is also used to explore the properties of sets of neural networks. The chapter is based on [12, p. 259-276], and [6, p. 131], and is a foundation for the rest of the report.

As an initial step, the following notation is defined.

Definition II.1: Modified Tensor Product

Let $N \in \mathbb{N}$, and $\{f_n : X_n \to Y_n\}_{n=1}^N$. Then

$$\bigotimes_{n=1}^{N} f_n : X_1 \times \cdots \times X_N \to Y_1 \times \cdots \times Y_N,$$

is given by $[x_1, \ldots, x_N] \mapsto [f_1(x_1), \ldots, f_N(x_N)]$.

Note that compared to the classical use of tensor product, this stacks the outputs as a vector.

Definition II.2: Neural Networks

Let $\varrho : \mathbb{R} \to \mathbb{R}$, let $L \in \mathbb{N}$, and let $\{N_\ell\}_{\ell=0}^L \subseteq \mathbb{N}$. A **neural network with activation function** ϱ is a tuple

$$\Phi := \Big((T_1, \alpha_1), \ldots, (T_{L-1}, \alpha_{L-1}), (T_L, \mathrm{id}_{\mathbb{R}^{N_L}})\Big),$$

where for $1 \leq \ell \leq L$ the functions $T_\ell : \mathbb{R}^{N_{\ell-1}} \to \mathbb{R}^{N_\ell}$ are affine linear maps and for $1 \leq \ell < L$ the functions $\alpha_\ell : \mathbb{R}^{N_\ell} \to \mathbb{R}^{N_\ell}$ are given by

$$\alpha_\ell := \bigotimes_{n=1}^{N_\ell} \varrho_n^{(\ell)},$$

where $\varrho_n^{(\ell)} \in \{\varrho, \mathrm{id}_{\mathbb{R}}\}$.

Argumentation for the conventions in the definition can be found in Remark A.1. For convenience, a neural network with activation function ϱ will often be shorted to a *ϱ-network* or a *neural network* if the activation function holds no interest. In addition, this definition of the neural network is in this report also referred to as *generalized neural networks*. The affine linear maps T_ℓ in Definition II.2 is given by

$$T_\ell(x) := A^{(\ell)}x + b^{(\ell)},$$

where the matrices $A^{(\ell)} \in \mathbb{R}^{N_\ell \times N_{\ell-1}}$ are called *weight matrices* and the vectors $b^{(\ell)} \in \mathbb{R}^{N_\ell}$ are called *bias vectors*. These matrices and vectors are often used with a specific structure such as convolutional or Toeplitz. The *zero-norm of T_ℓ* is defined by

$$\|T_\ell\|_{\ell^0} := \|A^{(\ell)}\|_{\ell^0},$$

where $\|\cdot\|_{\ell^0}$ is the *zero-norm* which counts the number of non-zero entries in the matrix. Note this differs from the normal definition of the zero-norm for affine linear maps as it should count the number of non-zero entries in the bias vector as well. A general illustration of a neural network is highlighted in Figure II.1.

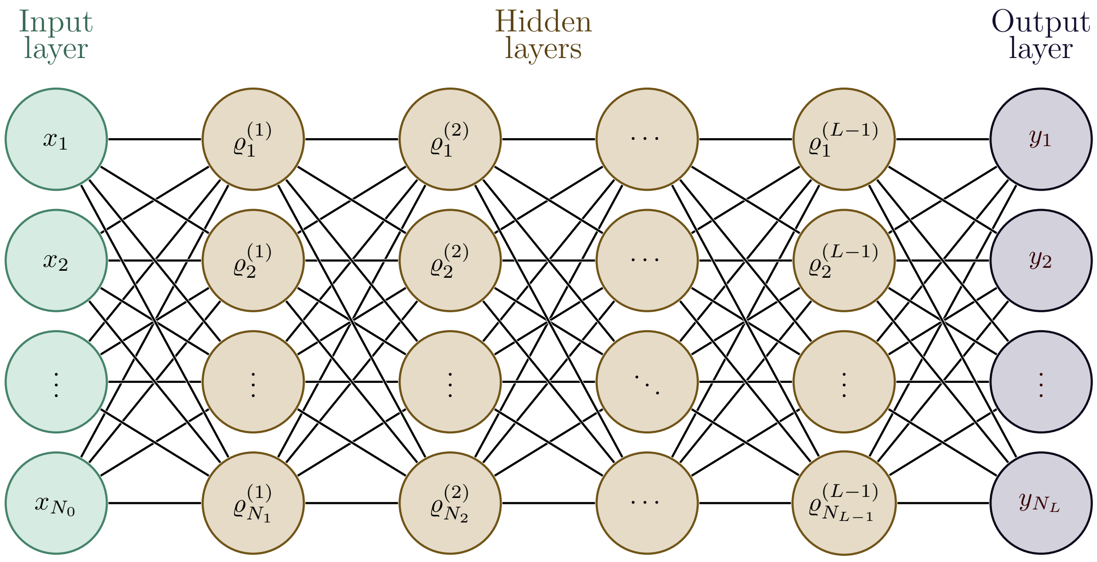

Figure II.1: *Illustration of a neural network with activation function ϱ.*

For future use, the neural network in Figure II.1 can also be illustrated in a more compact form, which is more in line with the tuple representation. This is illustrated in Figure II.2.

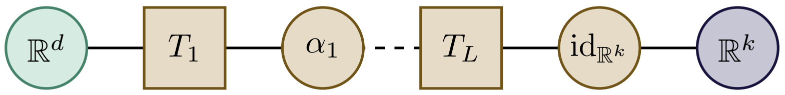

Figure II.2: *Illustration of a neural network by the tuple, where the dotted line illustrated more layers. Here $N_0 = d$ and $N_L = k$.*

This illustration is simpler and is useful in the context where many different neural networks are used at once. This is, for example, the case where illustrations are used to illustrate the constructive parts of a proof.

To associate an approximation space to a neural network, a measure of the neural network's *complexity* is required. To understand complexity, the following definition includes several characterizing features.

Definition II.3:

Let Φ be a neural network. Then the following features are defined:

(i) **Input dimension**: $d_{in}(\Phi) := N_0 \in \mathbb{N}$.

(ii) **Output dimension**: $d_{out}(\Phi) := N_L \in \mathbb{N}$.

(iii) **Number of layers**: $L(\Phi) := L \in \mathbb{N}$.

(iv) **Number of hidden neurons**: $N(\Phi) := \sum_{\ell=1}^{L-1} N_\ell \in \mathbb{N}_0$.

(v) **Number of connections**: $W(\Phi) := \sum_{\ell=1}^{L} \|T_\ell\|_{\ell^0} \in \mathbb{N}_0$.

The number of connections can also be called *the number of weights*. Note that by definition connections of value 0 are not considered as connections. Furthermore, the number of layers is called the *depth*, and by definition, the *number of hidden layers* is given by $L-1$. If $L \geq 2$ a neural network is called *deep*. A special class of neural networks with less flexibility is the *strict* neural networks.

Definition II.4: Strict Neural Network

A ϱ-network is called **strict**, if and only if $\varrho_n^{(\ell)} = \varrho$ for all $1 \leq \ell < L$ and $1 \leq n \leq N_\ell$.

As for the generalized neural network, a comment for the strict neural networks can be found in Remark A.1 as well.

The quantities given in Definition II.3 all describe the complexity of the neural network, and are all used as an expression for it. However, to be able to work with it later on, something more concrete is required. This can be done by restricting the complexity with an upper bound. More concretely this can be done by bounding the number of neurons $N(\Phi) \leq n$ or the number of weights $W(\Phi) \leq n$, where $n \in \mathbb{N}_0$. If needed the number of layers can be restricted depending on n. This is typically done with a *depth growth function*, which controls how the depth $L(\Phi)$ grows with n.

Definition II.5: Depth Growth Function

A **depth grown function** is a non-decreasing function $\mathscr{L} : \mathbb{N}_0 \to \mathbb{N} \cup \{\infty\}$ given by $n \mapsto \mathscr{L}(n)$.

This allows the number of layers to depend on the parameter, that restricts the number of connections or neurons. Note this definition allows the number of layers to be ∞. Later on, the relation between approximation classes associated with different depth growth functions is explored. For this, a comparison rule between depth growth functions is needed.

Definition II.6:

Let $\mathscr{L}_1$ and $\mathscr{L}_2$ be depth growth functions. Then $\mathscr{L}_1$ is said to be **dominated** by $\mathscr{L}_2$, denoted by $\mathscr{L}_1 \preceq \mathscr{L}_2$ or $\mathscr{L}_2 \succeq \mathscr{L}_1$, if

$$\exists\, c, n_0 \in \mathbb{N}_0 : \forall n \geq n_0 : \; \mathscr{L}_1(n) \leq \mathscr{L}_2(cn).$$

Moreover, $\mathscr{L}_1$ and $\mathscr{L}_2$ are **equivalent**, denoted by $\mathscr{L}_1 \sim \mathscr{L}_2$, if $\mathscr{L}_1 \preceq \mathscr{L}_2$ and $\mathscr{L}_1 \succeq \mathscr{L}_2$, which is equivalent with

$$\exists\, c, n_0 \in \mathbb{N}_0 : \forall n \geq n_0 : \; \mathscr{L}_1(n) \leq \mathscr{L}_2(cn) \,\wedge\, \mathscr{L}_1(cn) \geq \mathscr{L}_2(n).$$

Note that $\mathscr{L}_1(n) \leq \mathscr{L}_2(n)$ for all $n \in \mathbb{N}_0$, implies that $\mathscr{L}_1 \preceq \mathscr{L}_2$.

With the complexity and the ways to characterize a neural network in place, functions implemented by neural networks are considered. These are defined by applying the different maps in the tuple in an iterative order. Formally this is given as follows:

Definition II.7: Realization of a Neural Networks

Let Φ be a neural network. Then $\mathtt{R}(\Phi) : \mathbb{R}^{N_0} \to \mathbb{R}^{N_\ell}$ is called the **realization of** Φ and is given by

$$\mathtt{R}(\Phi) := T_L \circ \alpha_{L-1} \circ T_{L-1} \circ \cdots \circ \alpha_1 \circ T_1.$$

Since $\mathrm{id}_{\mathbb{R}^{N_L}} \circ T_L = T_L$, the identity is omitted from the definition of the realization but can be reinstated if needed. The argumentation for the convention in the definition of neural networks also relies on the definition of the realization, which is commented in Remark A.1 as well. Moreover, the remark highlights the reasoning for distinguishing between a neural network and its realization. The realization is going to play an important role and allow a connection between the neural networks and approximation theory.

An example of a simple neural network is when $N(\Phi) = 0$. In this case $\mathtt{R}(\Phi)$ is an affine linear map. Given different constraints on the features for complexity different families occur. Some of the relevant families are stated in the following definition.

Definition II.8:

Let $d, k \in \mathbb{N}$, let $L \in \mathbb{N} \cup \{\infty\}$, and let $W, N \in \mathbb{N}_0 \cup \{\infty\}$. Then

$$\begin{aligned}
\mathcal{NN}^{\varrho,d,k}_{W,L,N} &:= \left\{ \Phi \text{ is a } \varrho\text{-network} \;\middle|\; \begin{array}{c} d_{in}(\Phi) = d,\, d_{out}(\Phi) = k, \\ W(\Phi) \leq W,\, L(\Phi) \leq L,\, N(\Phi) \leq N \end{array} \right\}, \\
\mathcal{SNN}^{\varrho,d,k}_{W,L,N} &:= \left\{ \Phi \in \mathcal{NN}^{\varrho,d,k}_{W,L,N} \;\middle|\; \Phi \text{ is strict} \right\}, \\
\mathtt{NN}^{\varrho,d,k}_{W,L,N} &:= \left\{ \mathtt{R}(\Phi) \;\middle|\; \Phi \in \mathcal{NN}^{\varrho,d,k}_{W,L,N} \right\}, \\
\mathtt{SNN}^{\varrho,d,k}_{W,L,N} &:= \left\{ \mathtt{R}(\Phi) \;\middle|\; \Phi \in \mathcal{SNN}^{\varrho,d,k}_{W,L,N} \right\}.
\end{aligned}$$

By definition $\mathtt{NN}^{\varrho,d,k}_{W,L,N}$ is the set of all functions $f : \mathbb{R}^d \to \mathbb{R}^k$ that can be represented by a realization of a ϱ-network with at most W weights, L layers and N neurons, and similarly for $\mathtt{SNN}^{\varrho,d,k}_{W,L,N}$. An example of such a family is $\mathtt{NN}^{\varrho,d,k}_{\infty,L,0}$, which is the set of affine linear functions.

Moreover, for a non-empty set $\Omega \subseteq \mathbb{R}^d$ the restriction of the introduced set to Ω is denoted by $\mathtt{NN}^{\varrho,d,k}_{W,L,N}(\Omega)$, and $\mathtt{SNN}^{\varrho,d,k}_{W,L,N}(\Omega)$ respectively. Often Ω is chosen to be a simple bounded compact subset of $\mathbb{R}^d$, for instance, $\Omega = [0,1]^d$.

These classes are later used with an $n \in \mathbb{N}_0$ and a depth growth function $\mathscr{L}$, to make an upper bound of the complexity. For instance the class

$$\mathtt{NN}^{\varrho,d,k}_{n,\mathscr{L}(n),\infty} := \left\{ \mathtt{R}(\Phi) \;\middle|\; \Phi \in \mathcal{NN}^{\varrho,d,k}_{n,\mathscr{L}(n),\infty} \right\}.$$

Notation:

For the rest of the report, let $\mathcal{NN}^{\varrho,\,d,\,k}_{W,\,L,\,N}$, $\mathcal{SNN}^{\varrho,\,d,\,k}_{W,\,L,\,N}$, $\mathtt{NN}^{\varrho,\,d,\,k}_{W,\,L,\,N}$, and $\mathtt{SNN}^{\varrho,\,d,\,k}_{W,\,L,\,N}$ be defined as in Definition II.8.

II.1 Relations Depending on the Complexity

This section is devoted to exploring the relationships between the quantities describing the complexity, as these are tightly connected. First of all the number of connections and layers can be bounded by the number of neurons in a neural network.

Proposition II.9:

Let Φ be a neural network. Then

$$L(\Phi) \leq 1 + N(\Phi), \tag{II.1}$$
$$W(\Phi) \leq \Big(N(\Phi) + d_{in}(\Phi)\Big)\Big(N(\Phi) + d_{out}(\Phi)\Big). \tag{II.2}$$

If $L(\Phi) = 2$, then $W(\Phi) \leq (d_{in}(\Phi) + d_{out}(\Phi))N(\Phi)$.

Proof, see Appendix B.

Proposition II.9 gives a boundary of $L(\Phi)$ and $W(\Phi)$ in terms of $N(\Phi)$. Therefore one could try to consider a boundary of $N(\Phi)$ and $L(\Phi)$ in terms of $W(\Phi)$. However, such a bound does not exist as it is always possible to bloat a neural network with 'dead neurons'. By considering the class of realizations one could make such a bound.

The next result shows that when the number of non-zero weights is less than the number of layers the realization of the neural network is guaranteed to be constant.

Proposition II.10:

Let Φ be a neural network with $d_{out}(\Phi) = k$. If $W(\Phi) < L(\Phi)$, then $\mathtt{R}(\Phi) = c \in \mathbb{R}^k$.

Proof, see Appendix B.

Note Proposition II.10 also guarantees $\mathtt{R}(\Phi)$ is constant if $W(\Phi) = 0$, since $L(\Phi) \geq 1$.

Now it is considered how the set of constant functions relates to sets of realizations of neural networks.

Proposition II.11:

Let $\varrho : \mathbb{R} \to \mathbb{R}$ be an activation function, let $d, k \in \mathbb{N}$, $W, N \in \mathbb{N}_0 \cup \{\infty\}$, and let $L \in \mathbb{N} \cup \{\infty\}$. Then

$$\Big\{ f : \mathbb{R}^d \to \mathbb{R}^k \;\Big|\; \exists c \in \mathbb{R}^k : f = c \Big\} \subseteq \mathtt{SNN}^{\varrho,\,d,\,k}_{W,\,L,\,N} \subseteq \mathtt{NN}^{\varrho,\,d,\,k}_{W,\,L,\,N}.$$

Proof, see Appendix B.

In the case where the number of connections is zero, there is equality, so the classes of realizations of neural networks with no connections are the constant map.

Lemma II.12:

Let $\varrho : \mathbb{R} \to \mathbb{R}$ be an activation function, let $d, k \in \mathbb{N}$, $N \in \mathbb{N}_0 \cup \{\infty\}$, and let $L \in \mathbb{N} \cup \{\infty\}$. Then

$$\mathtt{NN}^{\varrho,\, d,\, k}_{0,\, L,\, N} = \mathtt{SNN}^{\varrho,\, d,\, k}_{0,\, L,\, N} = \left\{ f : \mathbb{R}^d \to \mathbb{R}^k \;\middle|\; \exists\, c \in \mathbb{R}^k : f = c \right\}.$$

Proof, see Appendix B.

Given $W \in \mathbb{N}$, the next result shows that any function that is a realization of a neural network with at most $W \geq 1$ connections can also be realized by a neural network with W connections and at most W layers and hidden neurons. Some notations for matrices are used to prove the result, these can be found in Definition A.4.

Lemma II.13:

Let $\varrho : \mathbb{R} \to \mathbb{R}$ be an activation function, let $d, k \in \mathbb{N}$, $L \in \mathbb{N} \cup \{\infty\}$, and let $W \in \mathbb{N}$. Then

$$\mathtt{NN}^{\varrho,\, d,\, k}_{W,\, L,\, \infty} = \mathtt{NN}^{\varrho,\, d,\, k}_{W,\, L,\, W} \subseteq \mathtt{NN}^{\varrho,\, d,\, k}_{W,\, W,\, W}.$$

For $L \geq W$ the inclusion becomes an equality. An analogous statement holds for strict neural networks.

Proof
First, it is desired to show that

$$\mathtt{NN}^{\varrho,\, d,\, k}_{W,\, L,\, W} \subseteq \mathtt{NN}^{\varrho,\, d,\, k}_{W,\, W,\, W}.$$

Therefore assume $f \in \mathtt{NN}^{\varrho,\, d,\, k}_{W,\, L,\, W}$. Then there exists a $\Phi \in \mathcal{NN}^{\varrho,\, d,\, k}_{W,\, L,\, W}$ such that $\mathtt{R}(\Phi) = f$. The statement is now divided into two cases:

$$L(\Phi) \leq W(\Phi), \quad L(\Phi) > W(\Phi).$$

First assume $L(\Phi) \leq W(\Phi)$. Then it holds that $\Phi \in \mathcal{NN}^{\varrho,\, d,\, k}_{W,\, W,\, W}$, so $f \in \mathtt{NN}^{\varrho,\, d,\, k}_{W,\, W,\, W}$. Now assume $W(\Phi) < L(\Phi)$. Then f is constant according to Proposition II.10, so Proposition II.11 gives that $f \in \mathtt{NN}^{\varrho,\, d,\, k}_{W,\, W,\, W}$. This shows that $\mathtt{NN}^{\varrho,\, d,\, k}_{W,\, L,\, W} \subseteq \mathtt{NN}^{\varrho,\, d,\, k}_{W,\, W,\, W}$.

To prove the last statement in the lemma, assume $L \geq W$. Then it follows directly from the definition of the sets that $\mathtt{NN}^{\varrho,\, d,\, k}_{W,\, L,\, W} \supseteq \mathtt{NN}^{\varrho,\, d,\, k}_{W,\, W,\, W}$, which gives equality.

Next, it is desired to prove that $\mathtt{NN}^{\varrho,\, d,\, k}_{W,\, L,\, \infty} = \mathtt{NN}^{\varrho,\, d,\, k}_{W,\, L,\, W}$, by showing that they are subspaces of each other. By definition of the classes $\mathtt{NN}^{\varrho,\, d,\, k}_{W,\, L,\, \infty} \supseteq \mathtt{NN}^{\varrho,\, d,\, k}_{W,\, L,\, W}$. Therefore it remains to show that $\mathtt{NN}^{\varrho,\, d,\, k}_{W,\, L,\, \infty} \subseteq \mathtt{NN}^{\varrho,\, d,\, k}_{W,\, L,\, W}$. For realizations in $\mathtt{NN}^{\varrho,\, d,\, k}_{W,\, L,\, \infty}$, where the neural network satisfies

$N(\Phi) \leq W$, the inclusion holds by definition, so assume that $N(\Phi) > W$. To establish the inclusion it is desired to prove that that for a neural network $\Phi \in \mathcal{NN}^{\varrho,d,k}_{W,L,\infty}$ with $N(\Phi) > W$, there exists a neural network $\tilde{\Phi} \in \mathcal{NN}^{\varrho,d,k}_{W,L,\infty}$ with $\mathtt{R}(\Phi) = \mathtt{R}(\tilde{\Phi})$ such that $N(\tilde{\Phi}) \leq W$. This is done by an "compression" argument, which consists of showing that there exists a $\tilde{\Phi} \in \mathcal{NN}^{\varrho,d,k}_{W,L,\infty}$ with $\mathtt{R}(\Phi) = \mathtt{R}(\tilde{\Phi})$ such that $N(\tilde{\Phi}) < N(\Phi)$, and then repeating the argument until the case where $N(\tilde{\Phi}) \leq W$.

The "compression" argument is given as follows: Consider $\Phi \in \mathcal{NN}^{\varrho,d,k}_{W,L,\infty}$. Since $W(\Phi) \leq W$ and $N(\Phi) > W$ it follows that

$$\begin{aligned}\sum_{\ell=1}^{L(\Phi)-1} N_\ell = N(\Phi)&\\ &> W\\ &\geq W(\Phi)\\ &= \sum_{\ell=1}^{L(\Phi)} \left\| A^{(\ell)} \right\|_{\ell^0}\\ &= \sum_{\ell=1}^{L(\Phi)} \sum_{i=1}^{N_\ell} \left\| A^{(\ell)}_{i,-} \right\|_{\ell^0}\\ &\geq \sum_{\ell=1}^{L(\Phi)-1} \sum_{i=1}^{N_\ell} \left\| A^{(\ell)}_{i,-} \right\|_{\ell^0}.\end{aligned}$$

This shows that $L(\Phi) > 1$ and the existence of some $\ell_0 \in \{1, \ldots, L(\Phi) - 1\}$ and $i \in \{1, \ldots, N_{\ell_0}\}$ such that $A^{(\ell_0)}_{i,-} = 0$. By reindexing assume without loss of generality that $i = N_{\ell_0}$. The rest of the proof will be split into two cases:

$$N_{\ell_0} = 1, \quad N_{\ell_0} > 1.$$

First, consider the case where $N_{\ell_0} = 1$. Hence $A^{(\ell_0)} = 0$, and therefore $T_{\ell_0} = b^{(\ell_0)}$. Thus the realization is constant, so there exists a $c \in \mathbb{R}^k$ such that $\mathtt{R}(\Phi) = c$. Then the strict neural network

$$\tilde{\Phi} := \left(\left(\tilde{T}_1, \mathrm{id}_{\mathbb{R}^k} \right) \right),$$

where $\tilde{T}_1 := c$ satisfy $\mathtt{R}(\tilde{\Phi}) = \mathtt{R}(\Phi)$ and $\tilde{\Phi} \in \mathcal{NN}^{\varrho,d,k}_{0,1,0} \subseteq \mathcal{NN}^{\varrho,d,k}_{W,L,W}$.

For the case where $N_{\ell_0} > 1$ it is desired to construct a ϱ-network based on the original with i slight change for the index ℓ_0. Start by defining

$$\begin{aligned}\tilde{\alpha}_\ell &:= \alpha_\ell, \quad \ell \in \{1, \ldots, L(\Phi) - 1\} \setminus \{\ell_0\},\\ \tilde{T}_\ell &:= T_\ell, \quad \ell \in \{1, \ldots, L(\Phi)\} \setminus \{\ell_0, \ell_0 + 1\}.\end{aligned}$$

For ℓ_0 define $\tilde{\alpha}_{\ell_0} : \mathbb{R}^{N_{\ell_0}-1} \to \mathbb{R}^{N_{\ell_0}-1}$ given by

$$\tilde{\alpha}_{\ell_0}(x) := \left(\bigotimes_{\ell=1}^{N_{\ell_0}-1} \varrho^{\ell_0}_\ell \right)(x),$$

and $\tilde{T}_{\ell_0} : \mathbb{R}^{N_{\ell_{0-1}}} \to \mathbb{R}^{N_{\ell_0}-1}$ given by

$$\tilde{T}_{\ell_0} x := (T_{\ell_0} x)_{(N_{\ell_0})}.$$

Moreover, define $\tilde{T}_{\ell_0+1} : \mathbb{R}^{N_{\ell_0}-1} \to \mathbb{R}^{N_{\ell_0+1}}$ given by

$$\begin{aligned}\tilde{T}_{\ell_0+1}x &:= T_{\ell_0+1}\left(x_1, \dots, x_{N_{\ell_0}-1}, \varrho^{(\ell_0)}_{N_{\ell_0}}\left(b^{(\ell_0)}_{N_{\ell_0}}\right)\right) \\ &= A^{(\ell_0+1)}_{[N_{\ell_0}]}x + \tilde{b}^{(\ell_0+1)},\end{aligned}$$

where

$$\tilde{b}^{(\ell_0+1)} := b^{(\ell_0+1)} + \varrho^{(\ell_0)}_{N_{\ell_0}}\left(b^{(\ell_0)}_{N_{\ell_0}}\right)\left(A^{(\ell_0+1)}_{-,N_{\ell_0}}\right)^T.$$

Thus the constructed neural network is given by

$$\tilde{\Phi} := \left(\left(\tilde{T}_1, \tilde{\alpha}_1\right), \dots, \left(\tilde{T}_{L(\Phi)-1}, \tilde{\alpha}_{L(\Phi)-1}\right), \left(\tilde{T}_{L(\Phi)}, \mathrm{id}_{\mathbb{R}^{N_\ell}}\right)\right).$$

Now consider some $x \in \mathbb{R}^{N_{\ell_0}-1}$ and define

$$y := T_{\ell_0}x \in \mathbb{R}^{N_{\ell_0}}, \quad \tilde{y} := \tilde{T}_{\ell_0}x \in \mathbb{R}^{N_{\ell_0}-1}.$$

Notice that $\tilde{y}_j = y_j$ for $1 \le j < N_{\ell_0}$ and $y_{N_{\ell_0}} = b^{(\ell_0)}_{N_{\ell_0}}$ since $A^{(\ell_0)}_{N_{\ell_0},-} = 0$. From these observations, it follows that

$$\begin{aligned}\left(\tilde{T}_{\ell_0+1} \circ \tilde{\alpha}_{\ell_0} \circ \tilde{T}_{\ell_0}\right)(x) &= \left(\tilde{T}_{\ell_0+1} \circ \tilde{\alpha}_{\ell_0}\right)(\tilde{y}) \\ &= T_{\ell_0+1}\left(\varrho^{(\ell_0)}_1\left(\tilde{y}_1\right), \dots, \varrho^{(\ell_0)}_{N_{\ell_0}-1}\left(\tilde{y}_{N_{\ell_0}-1}\right), \varrho^{(\ell_0)}_{N_{\ell_0}}\left(b^{(\ell_0)}_{N_{\ell_0}}\right)\right) \\ &= T_{\ell_0+1}\left(\varrho^{(\ell_0)}_1(y_1), \dots, \varrho^{(\ell_0)}_{N_{\ell_0}}\left(y_{N_{\ell_0}}\right)\right) \\ &= \left(T_{\ell_0+1} \circ \alpha_{\ell_0}\right)(y) \\ &= \left(T_{\ell_0+1} \circ \alpha_{\ell_0} \circ T_{\ell_0}\right)(x).\end{aligned}$$

This implies that $\tilde{\Phi}$ satisfies $\mathtt{R}(\tilde{\Phi}) = \mathtt{R}(\Phi)$, $W(\tilde{\Phi}) \le W(\Phi)$ and $N(\tilde{\Phi}) = N(\Phi) - 1$ by construction. It is therefore possible to "compress" until the desired inclusion is achieved. Additionally $\tilde{\Phi}$ inherits strictness from Φ, so the proof for strict neural networks follows analogously. ■

Consider a neural network $\Phi \in \mathcal{NN}^{\varrho,d,k}_{\infty,\infty,\infty}$. Using Lemma II.13 with $L = L(\Phi)$ and $W = W(\Phi)$ there exists another neural network $\tilde{\Phi} \in \mathcal{NN}^{\varrho,d,k}_{W(\Phi),L(\Phi),W(\Phi)}$ with $\mathtt{R}(\tilde{\Phi}) = \mathtt{R}(\Phi)$, and

$$N(\tilde{\Phi}) \le W(\tilde{\Phi}) \le W(\Phi) \le N^2(\Phi) + (d+k)N(\Phi) + dk, \tag{II.3}$$

where the first inequality holds by the compression argument in the proof of Lemma II.13, and the last inequality holds according to (II.2). In the case where $L = 2$, it holds that

$$N(\tilde{\Phi}) \le W(\tilde{\Phi}) \le W(\Phi) \le (d+k)N(\Phi), \tag{II.4}$$

according to Proposition II.9.

With the focus on the depth growth function, Lemma II.13 gives that

$$\mathtt{NN}^{\varrho,d,k}_{n,\mathscr{L}(n),\infty} = \mathtt{NN}^{\varrho,d,k}_{n,\tilde{\mathscr{L}}(n),\infty}$$

for $\tilde{\mathscr{L}}(n) := \min\{n, \mathscr{L}(n)\}$. Similarly Proposition II.9, and (II.1) gives that

$$\mathtt{NN}^{\varrho,d,k}_{\infty,\mathscr{L}(n),n} = \mathtt{NN}^{\varrho,d,k}_{\infty,\tilde{\mathscr{L}}(n),n}$$

for $\tilde{\mathscr{L}}(n) := \min\{n+1, \mathscr{L}(n)\}$. These conclusions will be used when associating neural networks with approximation spaces.

II.2 Operations on Generalized Neural Networks

This section explores different operations on generalized neural networks, such as addition and composition. For this, an extended version of a neural network is needed. This is constructed in the next result, which shows that it is possible to increase the depth of generalized neural networks while controlling the increase in complexity.

Lemma II.14: Extended Neural Networks

Let $\varrho : \mathbb{R} \to \mathbb{R}$ be an activation function, let $d, k \in \mathbb{N}$ and let $L_0 \in \mathbb{N}_0$. Consider a neural network $\Phi \in \mathcal{NN}^{\varrho, d, k}_{\infty, \infty, \infty}$ with layer sizes $\{N_\ell\}_{\ell=0}^{L(\Phi)}$, and define $c := \min\{N_\ell\}_{\ell=0}^{L(\Phi)}$. Then there exists a $\Psi \in \mathcal{NN}^{\varrho, d, k}_{\infty, \infty, \infty}$ such that

$$\begin{aligned} \mathtt{R}(\Psi) &= \mathtt{R}(\Phi), & W(\Psi) &= W(\Phi) + cL_0, \\ L(\Psi) &= L(\Phi) + L_0, & N(\Psi) &= N(\Phi) + cL_0. \end{aligned}$$

Proof

Assume $\Phi \in \mathcal{NN}^{\varrho, d, k}_{\infty, \infty, \infty}$. Then it holds that

$$\Phi := \Big((T_1, \alpha_1), \dots, (T_{L(\Phi)-1}, \alpha_{L(\Phi)-1}), \big(T_{L(\Phi)}, \mathrm{id}_{\mathbb{R}^k}\big) \Big).$$

If $L_0 = 0$, then choose $\Psi := \Phi$. If $L_0 > 0$ then choose $\ell_0 := \arg\min\{N_\ell\}_{\ell=0}^{L(\Phi)}$ and define

$$\begin{aligned} \Psi := \Big((T_1, \alpha_1), \dots, (T_{\ell_0}, \alpha_{\ell_0}), \underbrace{\big(\mathrm{id}_{\mathbb{R}^{N_{\ell_0}}}, \mathrm{id}_{\mathbb{R}^{N_{\ell_0}}}\big), \dots, \big(\mathrm{id}_{\mathbb{R}^{N_{\ell_0}}}, \mathrm{id}_{\mathbb{R}^{N_{\ell_0}}}\big)}_{L_0 \text{ terms}}, \\ (T_{\ell_0+1}, \alpha_{\ell_0+1}), \dots, (T_{L(\Phi)-1}, \alpha_{L(\Phi)-1}), \big(T_{L(\Phi)}, \mathrm{id}_{\mathbb{R}^k}\big) \Big). \end{aligned}$$

Note Ψ is just Φ, with additional L_0 terms of $\big(\mathrm{id}_{\mathbb{R}^{N_{\ell_0}}}, \mathrm{id}_{\mathbb{R}^{N_{\ell_0}}}\big)$, where each terms add one layer, N_{ℓ_0} neurons and N_{ℓ_0} weights. Thus

$$\begin{aligned} \mathtt{R}(\Psi) &= \mathtt{R}(\Phi), & W(\Psi) &= W(\Phi) + N_{\ell_0} L_0, \\ L(\Psi) &= L(\Phi) + L_0, & N(\Psi) &= N(\Phi) + N_{\ell_0} L_0, \end{aligned}$$

which was the desired. The principle of the extended neural network Ψ in the case where $\ell_0 = L(\Phi)$ is illustrated in Figure II.3. ■

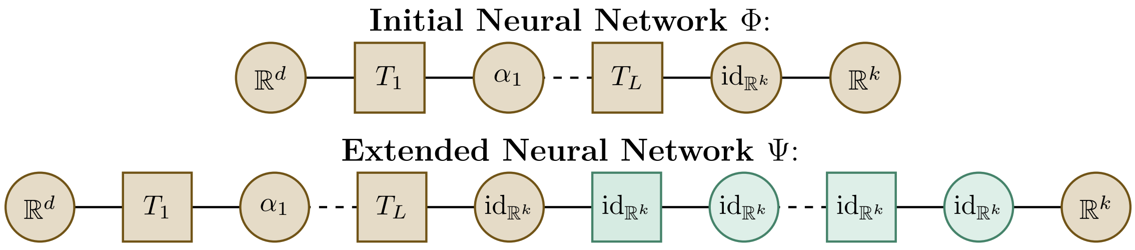
Initial Neural Network Φ:
$\mathbb{R}^d$
T_1
α_1
T_L
$\mathrm{id}_{\mathbb{R}^k}$
$\mathbb{R}^k$
Extended Neural Network Ψ:
$\mathbb{R}^d$
T_1
α_1
T_L
$\mathrm{id}_{\mathbb{R}^k}$
$\mathrm{id}_{\mathbb{R}^k}$
$\mathrm{id}_{\mathbb{R}^k}$
$\mathrm{id}_{\mathbb{R}^k}$
$\mathrm{id}_{\mathbb{R}^k}$
$\mathbb{R}^k$

Figure II.3: *Illustration of the principle of the proof for Lemma II.14. The first neural network is the initial neural network, and the last is an example of the extended neural network, where $\ell_0 = L(\Phi)$. Here the green boxes consist L_0 additional layers, $L_0 k$ connections and $L_0 k$ neurons compared to the initial neural network.*

This result has the consequence that the class of generalized neural networks is closed under linear combinations and Cartesian products. However, this does not hold for the class of strict neural networks. Already in proof of Lemma II.14 it is used that the identity can be chosen as an activation function. First, the statement on linear combinations is formalized in the following lemma.

Lemma II.15: Linear Combination

Let $\varrho : \mathbb{R} \to \mathbb{R}$ be an activation function, let $d, k, n \in \mathbb{N}$, $W, N \in \mathbb{N}_0 \cup \{\infty\}$, $L \in \mathbb{N} \cup \{\infty\}$, and $c \in \mathbb{R}$.

(i) **Scalar Multiplication**:
If $\Phi \in \mathcal{NN}^{\varrho,d,k}_{W,L,N}$ then there exists a $\Psi \in \mathcal{NN}^{\varrho,d,k}_{W,L,N}$ such that

$$\begin{aligned} \mathtt{R}(\Psi) &= c\mathtt{R}(\Phi), & W(\Psi) &\leq W(\Phi), \\ L(\Psi) &= L(\Phi), & N(\Psi) &= N(\Phi). \end{aligned}$$

(ii) **Addition**:
If $\{\Phi_m\}_{m=1}^n \subseteq \mathcal{NN}^{\varrho,d,k}_{W,L,N}$, then there exists a $\Psi \in \mathcal{NN}^{\varrho,d,k}_{C(L-1)+nW,L,(C+n)N}$ such that

$$\begin{aligned} \mathtt{R}(\Psi) &= \sum_{m=1}^{n} \mathtt{R}(\Phi_m), & W(\Psi) &\leq \delta + \sum_{m=1}^{n} W(\Phi_m) \\ L(\Psi) &= \max\{L(\Phi_m)\}_{m=1}^n, & N(\Psi) &\leq \delta + \sum_{m=1}^{n} N(\Phi_m), \end{aligned}$$

where $\delta \coloneqq C(L(\Psi) - \min\{L(\Phi_m)\}_{m=1}^n)$ and $C \leq \min\{d, k\}$.

Proof

(i) Let

$$\Phi \coloneqq \Big((T_1, \alpha_1), \dots, (T_{L(\Phi)-1}, \alpha_{L(\Phi)-1}), (T_{L(\Phi)}, \mathrm{id}_{\mathbb{R}^k})\Big) \in \mathcal{NN}^{\varrho,d,k}_{W,L,N}.$$

Then the neural network

$$\Psi \coloneqq \Big((T_1, \alpha_1), \dots, (T_{L(\Phi)-1}, \alpha_{L(\Phi)-1}), (cT_{L(\Phi)}, \mathrm{id}_{\mathbb{R}^k})\Big) \in \mathcal{NN}^{\varrho,d,k}_{W,L,N}$$

satisfy $\mathtt{R}(\Psi) = c\mathtt{R}(\Phi)$, $L(\Psi) = L(\Phi)$, and $N(\Psi) = N(\Phi)$. If $c \neq 0$ then it also holds that $W(\Psi) = W(\Phi)$, but if $c = 0$, then

$$W(\Psi) = W(\Phi) - \left\|T_{L(\Phi)}\right\|_{\ell^0} \leq W(\Phi),$$

which completes this part of the proof.

(ii) To prove this part induction is used on $i \in \{1, \dots, n\}$. Before beginning with this, some assumptions are made. Assume without loss of generality that $L(\Phi_1) \leq \cdots \leq L(\Phi_n)$. This can be done by permutating the index, which does not affect the desired realization, since addition is commutative. Next, it is desired to show that it can be assumed that there only is one neural network with one layer, so $L(\Phi_2) \geq 2$, compared to the previous assumption. To see this assume there there exists a $j \in \{2,\dots,n\}$ such that $L(\Phi_j) = 1$. Then it holds that $\Phi_m = \big((T_{m,1}, \mathrm{id}_{\mathbb{R}^k})\big)$, where $T_{m,1} \coloneqq A^{(m,1)} \bullet + b^{(m,1)}$ for $1 \le m \le j$. Moreover, $N(\Phi_m) = 0$ and $\mathtt{R}(\Phi_m)$ is an affine linear map. By defining

$$\tilde{\Phi}_i \coloneqq \begin{cases} \left(\left(\sum_{m=1}^{j} T_{m,1}, \mathrm{id}_{\mathbb{R}^k}\right)\right), & i = 1, \\ \Phi_{i+j-1}, & 2 \le i \le n-j+1, \end{cases}$$

it holds that

$$L(\tilde{\Phi}_1) = 1, \qquad \mathtt{R}(\tilde{\Phi}_1) = \sum_{m=1}^{j} \mathtt{R}(\Phi_m), \qquad N(\tilde{\Phi}_1) = 0,$$

$$W(\tilde{\Phi}_1) = \left\| \sum_{m=1}^{j} T_{m,1} \right\|_{\ell^0} \le \min\left\{dk, \sum_{m=1}^{j} \|T_{m,1}\|_{\ell^0}\right\} = \min\left\{dk, \sum_{m=1}^{j} W(\Phi_m)\right\}.$$

Then for $\{\tilde{\Phi}_i\}_{i=1}^{n-j+1}$ there exists no such j, so $L(\tilde{\Phi}_2) \ge 2$. It is therefore possible to assume without loss of generality that $L(\Phi_2) \ge 2$.

Now the induction part is considered for $i \in \{1,\dots,n\}$. The result holds for $i = 1$ by choosing $\Psi \coloneqq \Phi_1$. For the induction step, a constructive argument is needed. Assume that Ψ_i satisfies the result for some $i \in \{1,\dots,n-1\}$. That is

$$\mathtt{R}(\Psi_i) = \sum_{m=1}^{i} \mathtt{R}(\Phi_m), \qquad W(\Psi_i) \le \delta_i + \sum_{m=1}^{i} W(\Phi_m),$$
$$L(\Psi_i) = \max\{L(\Phi_m)\}_{m=1}^{i} = L(\Phi_i), \qquad N(\Psi_i) \le \delta_i + \sum_{m=1}^{i} N(\Phi_m),$$

where $\delta_i \coloneqq C_i(L(\Psi_i) - \min\{L(\Phi_m)\}_{m=1}^{i})$ and $C_i \le \min\{d,k\}$. Then it holds that $L(\Psi_i) = L(\Phi_i) \le L(\Phi_{i+1})$, where

$$\Phi_{i+1} \coloneqq \big((T_1,\alpha_1),\dots,(T_{L(\Phi_{i+1})-1},\alpha_{L(\Phi_{i+1})-1}),(T_{L(\Phi_{i+1})},\mathrm{id}_{\mathbb{R}^k})\big).$$

Let

$$\tilde{\Psi}_i \coloneqq \left(\left(\tilde{S}_1,\tilde{\beta}_1\right),\dots,\left(\tilde{S}_{L(\Phi_{i+1})-1},\tilde{\beta}_{L(\Phi_{i+1})-1}\right),\left(\tilde{S}_{L(\Phi_{i+1})},\mathrm{id}_{\mathbb{R}^k}\right)\right)$$

be the result of extending Ψ_i using Lemma II.14 with

$$L_0 = L(\Phi_{i+1}) - L(\Psi_i), \quad \mathtt{R}(\tilde{\Psi}_i) = \mathtt{R}(\Psi_i),$$

and let c_i be the minimal layer size of Ψ_i. Define

$$S_1 : \mathbb{R}^d \to \mathbb{R}^{N_1+\tilde{N}_1}, \qquad x \mapsto \left(\tilde{S}_1 x, T_1 x\right),$$
$$S_m : \mathbb{R}^{N_{m-1}+\tilde{N}_{m-1}} \to \mathbb{R}^{N_m+\tilde{N}_m}, \qquad (x,y) \mapsto \left(\tilde{S}_m x, T_m y\right),$$
$$S_{L(\Phi_{i+1})} : \mathbb{R}^{N_{L(\Phi_{i+1})-1}+\tilde{N}_{L(\Phi_{i+1})-1}} \to \mathbb{R}^k, \qquad (x,y) \mapsto \tilde{S}_{L(\Phi_{i+1})} x + T_{L(\Phi_{i+1})} y,$$

for $1 < m < L(\Phi_{i+1})$ and define

$$\beta_m : \mathbb{R}^{N_m+\tilde{N}_m} \to \mathbb{R}^{N_m+\tilde{N}_m}, \qquad (x,y) \mapsto \left(\tilde{\beta}_m x, \alpha_m y\right),$$

for $1 \leq m < L(\Phi_{i+1})$. Now define

$$\Psi_{i+1} \coloneqq \Big((S_1, \beta_1), \dots, (S_{L(\Phi_{i+1})-1}, \beta_{L(\Phi_{i+1})-1}), (S_{L(\Phi_{i+1})}, \mathrm{id}_{\mathbb{R}^k})\Big).$$

This neural network plays the role of the sum of the original neural networks Φ_i. The principle of the construction is illustrated in the case of two neural networks in Figure II.4. Now it holds that

$$\begin{aligned} L(\Psi_{i+1}) &= L(\Phi_{i+1}) = \max\{L(\Phi_m)\}_{m=1}^{i+1}, \\ \mathtt{R}(\Psi_{i+1}) &= \mathtt{R}(\Psi_i) + \mathtt{R}(\Phi_{i+1}) = \sum_{m=1}^{i+1} \mathtt{R}(\Phi_m). \end{aligned}$$

Using the properties of the extended neural network and the induction assumption it holds that

$$\begin{aligned} N(\Psi_{i+1}) &= N(\tilde{\Psi}_i) + N(\Phi_{i+1}) \\ &\leq N(\Psi_i) + c_i(L(\Phi_{i+1}) - L(\Psi_i)) + N(\Phi_{i+1}) \\ &\leq \left(\delta_i + \sum_{m=1}^{i} N(\Phi_m)\right) + c_i(L(\Phi_{i+1}) - L(\Psi_i)) + N(\Phi_{i+1}) \\ &= C_i(L(\Psi_i) - \min\{L(\Phi_m)\}_{m=1}^{i}) + c_i(L(\Phi_{i+1}) - L(\Psi_i)) + \sum_{m=1}^{i+1} N(\Phi_m) \\ &\leq C_i(L(\Phi_{i+1}) - \min\{L(\Phi_m)\}_{m=1}^{i}) + c_i(L(\Phi_{i+1}) - L(\Psi_i)) + \sum_{m=1}^{i+1} N(\Phi_m) \\ &\leq \max\{C_i, c_i\}(L(\Phi_{i+1}) - \min\{L(\Phi_m)\}_{m=1}^{i}) + \sum_{m=1}^{i+1} N(\Phi_m) \\ &\leq \delta_{i+1} + \sum_{m=1}^{i+1} N(\Phi_m), \end{aligned}$$

where

$$\begin{aligned} \delta_{i+1} &\coloneqq C\left(L(\Phi_{i+1}) - \min\{L(\Phi_m)\}_{m=1}^{i+1}\right), \\ C &\coloneqq \max\{C_i, c_i\} \leq \min\{d, k\}. \end{aligned}$$

By similar arguments, it holds that

$$\begin{aligned} W(\Psi_{i+1}) &= W(\tilde{\Psi}_i) + W(\Phi_{i+1}) \\ &\leq \delta_i + c_i(L(\Phi_{i+1}) - L(\Psi_i)) + \sum_{m=1}^{i+1} W(\Phi_m) \\ &\leq \delta_{i+1} + \sum_{m=1}^{i+1} W(\Phi_m). \end{aligned}$$

This concludes the induction. It remains to show that the constructed Ψ is in the right family. Since $\delta_n \leq C(L-1) \leq CN$, it holds that $\Psi \in \mathcal{NN}^{\varrho, d, k}_{C(L-1)+nW, L, (C+n)N}$, which concludes the proof. ■

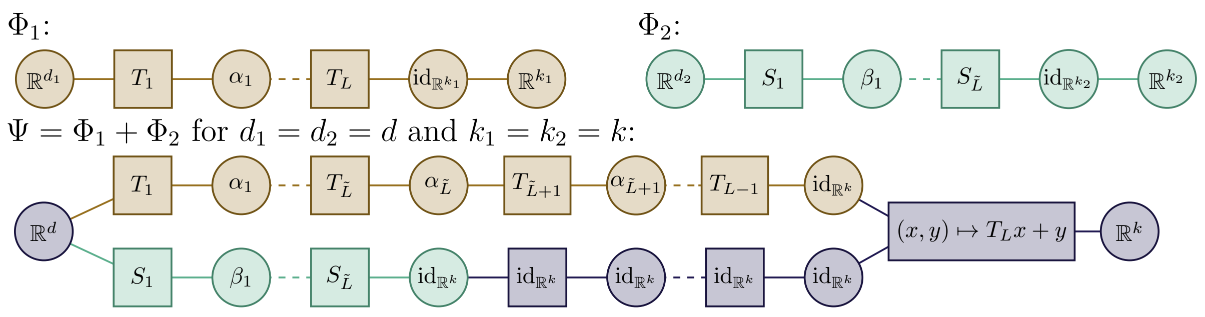
Φ_1:
$\mathbb{R}^{d_1}$
T_1
α_1
T_L
$\mathrm{id}_{\mathbb{R}^{k_1}}$
$\mathbb{R}^{k_1}$
Φ_2:
$\mathbb{R}^{d_2}$
S_1
β_1
$S_{\tilde{L}}$
$\mathrm{id}_{\mathbb{R}^{k_2}}$
$\mathbb{R}^{k_2}$
$\Psi = \Phi_1 + \Phi_2$ for $d_1 = d_2 = d$ and $k_1 = k_2 = k$:
$\mathbb{R}^d$
T_1
α_1
$T_{\tilde{L}}$
$\alpha_{\tilde{L}}$
$T_{\tilde{L}+1}$
$\alpha_{\tilde{L}+1}$
T_{L-1}
$\mathrm{id}_{\mathbb{R}^k}$
S_1
β_1
$S_{\tilde{L}}$
$\mathrm{id}_{\mathbb{R}^k}$
$\mathrm{id}_{\mathbb{R}^k}$
$\mathrm{id}_{\mathbb{R}^k}$
$\mathrm{id}_{\mathbb{R}^k}$
$\mathrm{id}_{\mathbb{R}^k}$
$(x,y) \mapsto T_L x + y$
$\mathbb{R}^k$

Figure II.4: *Illustration of the principle of adding two neural networks, as constructed in the proof for Lemma II.15(ii).*

Note that Lemma II.15 assumes that the input and output dimensions of the neural networks are equal. However, for the result stating that the class of generalized neural networks is closed under Cartesian products, it is only assumed that the input dimensions are equal. The proof of this result follows the same ideas as the previous and will be dealt with in less detail.

Lemma II.16: Cartesian Products

Let $\varrho : \mathbb{R} \to \mathbb{R}$ be an activation function, let $d, n \in \mathbb{N}$, $W, N \in \mathbb{N}_0 \cup \{\infty\}$, $L \in \mathbb{N} \cup \{\infty\}$, and let $\{k_m\}_{m=1}^n \subset \mathbb{N}$. If $\Phi_m \in \mathcal{NN}^{\varrho, d, k_m}_{W, L, N}$ for $m \in \{1, \dots, n\}$, then there exists a $\Psi \in \mathcal{NN}^{\varrho, d, k}_{CL+nW, L, (C+n)N}$, where $k := \sum_{m=1}^n k_m$, such that

$$\mathtt{R}(\Psi) = \Big(\mathtt{R}(\Phi_1), \dots, \mathtt{R}(\Phi_n)\Big), \qquad W(\Psi) \le \delta + \sum_{m=1}^n W(\Phi_m)$$

$$L(\Psi) = \max\{L(\Phi_m)\}_{m=1}^n, \qquad N(\Psi) \le \delta + \sum_{m=1}^n N(\Phi_m),$$

where $\delta := C(L(\Psi) - \min\{L(\Phi_m)\}_{m=1}^n)$ and $C \le \min\{d, k-1\}$.

Proof

First, consider a permutation $\sigma \in S_n$ such that $L(\Phi_{\sigma(1)}) \le \cdots \le L(\Phi_{\sigma(n)})$ and a permutation matrix $P \in \mathrm{GL}(\mathbb{R}^k)$ such that

$$P \circ \Big(\mathtt{R}(\Phi_{\sigma(1)}), \dots, \mathtt{R}(\Phi_{\sigma(n)})\Big) = \Big(\mathtt{R}(\Phi_1), \dots, \mathtt{R}(\Phi_n)\Big).$$

By these considerations, assume without loss of generality, that $L(\Phi_1) \le \cdots \le L(\Phi_n)$. In contrast to the proof of Lemma II.15(ii) this permutation matrix is crucial since the realization depends on the order of the neural networks.

The result follows from an induction argument similar to the proof of Lemma II.15(ii). The only differences lie in the affine-linear map

$$S_m : \mathbb{R}^{N_{m-1} \times \tilde{N}_{m-1}} \to \mathbb{R}^{N_m \times \tilde{N}_m}, \quad (x, y) \mapsto \Big(\tilde{S}_m x, T_m y\Big),$$

for $1 < m \le L(\Phi_{i+1})$, the constant $C := \max\{C_i, c_i\} \le \min\{d, k-1\}$, and the realization $\mathtt{R}(\Psi_{i+1}) = \Big(\mathtt{R}(\Psi_1), \dots, \mathtt{R}(\Psi_{i+1})\Big)$. Similarly as in the proof of Lemma II.15(ii), the neural networks constructed from these S_m plays the role of the Cartesian product, and the principle of the construction is illustrated in the case of two neural networks in Figure II.5. ■

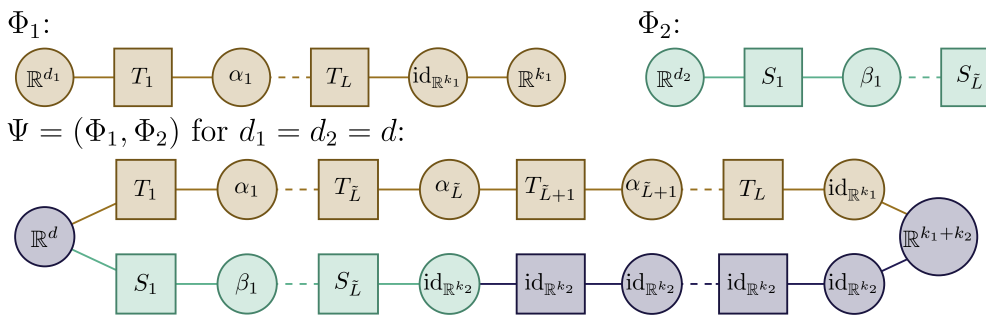
Φ_1:
$\mathbb{R}^{d_1}$
T_1
α_1
T_L
$\mathrm{id}_{\mathbb{R}^{k_1}}$
$\mathbb{R}^{k_1}$
Φ_2:
$\mathbb{R}^{d_2}$
S_1
β_1
$S_{\tilde{L}}$
$\mathrm{id}_{\mathbb{R}^{k_2}}$
$\mathbb{R}^{k_2}$
$\Psi = (\Phi_1, \Phi_2)$ for $d_1 = d_2 = d$:
$\mathbb{R}^d$
T_1
α_1
$T_{\tilde{L}}$
$\alpha_{\tilde{L}}$
$T_{\tilde{L}+1}$
$\alpha_{\tilde{L}+1}$
T_L
$\mathrm{id}_{\mathbb{R}^{k_1}}$
$\mathbb{R}^{k_1+k_2}$
S_1
β_1
$S_{\tilde{L}}$
$\mathrm{id}_{\mathbb{R}^{k_2}}$
$\mathrm{id}_{\mathbb{R}^{k_2}}$
$\mathrm{id}_{\mathbb{R}^{k_2}}$
$\mathrm{id}_{\mathbb{R}^{k_2}}$
$\mathrm{id}_{\mathbb{R}^{k_2}}$

Figure II.5: *Illustration of the principle of Cartesian products for two neural networks, as constructed in the proof for Lemma II.16.*

The next result deals with neural networks and composition. New notation is needed before stating the result. For an affine linear map $T : \mathbb{R}^d \to \mathbb{R}^k$, given by $T(x) := Ax + b$ define

$$\|A\|_{\ell^{0,\infty}} := \max\{\|A_{-,m}\|_{\ell^0}\}_{m=1}^{d}, \qquad \|T\|_{\ell^{0,\infty}} := \|A\|_{\ell^{0,\infty}},$$
$$\|A\|_{\ell^{0,\infty}_*} := \max\left\{\|A_{m,-}\|_{\ell^0}\right\}_{m=1}^{k}, \qquad \|T\|_{\ell^{0,\infty}_*} := \|A\|_{\ell^{0,\infty}_*}.$$

Lemma II.17: Composition

Let $\varrho : \mathbb{R} \to \mathbb{R}$ be an activation function, let $d_1, d_2, k_1, k_2 \in \mathbb{N}$, $W, N \in \mathbb{N}_0 \cup \{\infty\}$, and let $L \in \mathbb{N} \cup \{\infty\}$,

(i) If $\Phi \in \mathcal{NN}^{\varrho, d_1, k_1}_{W, L, N}$ and $P : \mathbb{R}^{d_2} \to \mathbb{R}^{d_1}$, $Q : \mathbb{R}^{k_1} \to \mathbb{R}^{k_2}$ are affine linear maps, then there exists a $\Psi \in \mathcal{NN}^{\varrho, d_2, k_2}_{W_1, L, N}$, where $W_1 := \|Q\|_{\ell^{0,\infty}} \|P\|_{\ell^{0,\infty}_*} W$ such that

$$\mathtt{R}(\Psi) = Q \circ \mathtt{R}(\Phi) \circ P, \qquad L(\Psi) = L(\Phi),$$
$$W(\Psi) \leq \|Q\|_{\ell^{0,\infty}} \|P\|_{\ell^{0,\infty}_*} W(\Phi), \qquad N(\Psi) = N(\Phi).$$

(ii) If $\Phi_1 \in \mathcal{NN}^{\varrho, d_1, k_1}_{W, L, N}$ and $\Phi_2 \in \mathcal{NN}^{\varrho, k_1, k_2}_{W, L, N}$, then there exists a $\Psi_1 \in \mathcal{NN}^{\varrho, d_1, k_2}_{2W, 2L, 2N+k_1}$ such that

$$\mathtt{R}(\Psi_1) = \mathtt{R}(\Phi_2) \circ \mathtt{R}(\Phi_1), \qquad L(\Psi_1) = L(\Phi_1) + L(\Phi_2),$$
$$N(\Psi_1) = N(\Phi_1) + N(\Phi_2) + k_1, \qquad W(\Psi_1) = W(\Phi_1) + W(\Phi_2),$$

and a $\Psi_2 \in \mathcal{NN}^{\varrho, d_1, k_2}_{W_2, 2L, 2N}$, where $W_2 := W + \max\{N(\Phi_1), d_1\} W$ such that

$$\mathtt{R}(\Psi_2) = \mathtt{R}(\Phi_2) \circ \mathtt{R}(\Phi_1), \qquad L(\Psi_2) = L(\Phi_1) + L(\Phi_2) - 1,$$
$$N(\Psi_2) = N(\Phi_1) + N(\Phi_2), \qquad W(\Psi_2) \leq W(\Phi_1) + \max\{N(\Phi_1), d_1\} W(\Phi_2).$$

If Φ_1 and Φ_2 are strict neural networks, then Ψ_2 is strict as well.

Proof

(i) Let

$$\Phi := \Big((T_1, \alpha_1), \ldots, (T_{L(\Phi)-1}, \alpha_{L(\Phi)-1}), (T_{L(\Phi)}, \mathrm{id}_{\mathbb{R}^k})\Big).$$

In the case where $\|P\|_{\ell^0} = 0$ or $\|Q\|_{\ell^0} = 0$ and the case where $\|P\|_{\ell^0} \geq 1$ and $\|Q\|_{\ell^0} \geq 1$ er dealt with separately.

If $\|P\|_{\ell^0} = 0$ or $\|Q\|_{\ell^0} = 0$ then $Q \circ \mathtt{R}(\Phi) \circ P = c \in \mathbb{R}^{k_2}$. Then by defining

$$\Psi := \Big((0T_1, \alpha_1), \dots, (0T_{L(\Phi)-1}, \alpha_{L(\Phi)-1}), (c, \mathrm{id}_{\mathbb{R}^k})\Big)$$

it holds that

$$\begin{aligned} \mathtt{R}(\Psi) &= c, & L(\Psi) &= L(\Phi), \\ W(\Psi) &= 0, & N(\Psi) &= N(\Phi). \end{aligned}$$

Now assume $\|P\|_{\ell^0} \geq 1$ and $\|Q\|_{\ell^0} \geq 1$, from which it follows that $\|P\|_{\ell^{0,\infty}} \geq 1$ and $\|Q\|_{\ell^{0,\infty}_*} \geq 1$. Thus for any general affine linear map T it holds that

$$\begin{aligned} \|T\|_{\ell^0} &\leq \|P\|_{\ell^{0,\infty}_*} \|T\|_{\ell^0}, \\ \|T\|_{\ell^0} &\leq \|Q\|_{\ell^{0,\infty}} \|T\|_{\ell^0}. \end{aligned}$$

According to Lemma C.2 with the weight matrices $T = T_i$ for $i \in \{1, \dots, L(\Phi)\}$ and $S = P$ or $S = Q$ it holds that

$$\begin{aligned} \|T_1 P\|_{\ell^0} &\leq \|T_1\|_{\ell^0} \|P\|_{\ell^{0,\infty}_*}, \\ \|Q T_{L(\Phi)}\|_{\ell^0} &\leq \|Q\|_{\ell^{0,\infty}} \|T_{L(\Phi)}\|_{\ell^0}. \end{aligned}$$

By defining

$$\Psi := \Big((T_1 \circ P, \alpha_1), (T_2, \alpha_2) \dots, (T_{L(\Phi)-1}, \alpha_{L(\Phi)-1}), (Q \circ T_{L(\Phi)}, \mathrm{id}_{\mathbb{R}^k})\Big)$$

it holds that

$$\mathtt{R}(\Psi) = Q \circ \mathtt{R}(\Phi) \circ P, \qquad L(\Psi) = L(\Phi), \qquad N(\Psi) = N(\Phi),$$

and

$$\begin{aligned} W(\Psi) &= \|T_1 \circ P\|_{\ell^0} + \|Q \circ T_{L(\Phi)}\|_{\ell^0} + \sum_{n=2}^{L(\Phi)-1} \|T_n\|_{\ell^0} \\ &\leq \|Q\|_{\ell^{0,\infty}} \|P\|_{\ell^{0,\infty}_*} \sum_{n=1}^{L(\Phi)} \|T_n\|_{\ell^0} \\ &\leq \|Q\|_{\ell^{0,\infty}} \|P\|_{\ell^{0,\infty}_*} W(\Phi). \end{aligned}$$

By defining $W_1 := \|Q\|_{\ell^{0,\infty}} \|P\|_{\ell^{0,\infty}_*} W$, part (i) is proven.

(ii) Define $L_1 := L(\Phi_1)$ and $L_2 := L(\Phi_2)$ and let

$$\begin{aligned} \Phi_1 &:= \Big((T_1, \alpha_1), \dots, (T_{L_1-1}, \alpha_{L_1-1}), (T_{L_1}, \mathrm{id}_{\mathbb{R}^{k_1}})\Big), \\ \Phi_2 &:= \Big((S_1, \beta_1), \dots, (S_{L_2-1}, \beta_{L_2-1}), (S_{L_2}, \mathrm{id}_{\mathbb{R}^{k_2}})\Big). \end{aligned}$$

Moreover, define

$$\begin{aligned} \Psi_1 := \Big(&(T_1, \alpha_1), \dots, (T_{L_1-1}, \alpha_{L_1-1}), (T_{L_1}, \mathrm{id}_{\mathbb{R}^{k_1}}), \\ &(S_1, \beta_1), \dots, (S_{L_2-1}, \beta_{L_2-1}), (S_{L_2}, \mathrm{id}_{\mathbb{R}^{k_2}})\Big), \\ \Psi_2 := \Big(&(T_1, \alpha_1), \dots, (T_{L_1-1}, \alpha_{L_1-1}), (S_1 \circ T_{L_1}, \beta_1) \\ &(S_2, \beta_2), \dots, (S_{L_2-1}, \beta_{L_2-1}), (S_{L_2}, \mathrm{id}_{\mathbb{R}^{k_2}})\Big). \end{aligned}$$

These neural networks play the role of the two different compositions, and the principle of the construction is illustrated in Figure II.6. Note that since T_ℓ, where $\ell \in \{1, \dots, L_1\}$, and S_ℓ, where $\ell \in \{1, \dots, L_2\}$, are affine-linear maps with the right dimensions, and since α_ℓ and β_ℓ still satisfies the condition in the definition, both Ψ_1 and Ψ_2 are generalized ϱ-networks. Moreover, it holds that

$$\begin{aligned}
\mathtt{R}(\Psi_1) &= \mathtt{R}(\Psi_2) = \mathtt{R}(\Phi_2) \circ \mathtt{R}(\Phi_1),\\
L(\Psi_1) &= L_1 + L_2,\\
L(\Psi_2) &= L_1 + L_2 - 1,\\
N(\Psi_1) &= N(\Phi_1) + N(\Phi_2) + k_1,\\
N(\Psi_2) &= N(\Phi_1) + N(\Phi_2),\\
W(\Psi_1) &= W(\Phi_1) + W(\Phi_2).
\end{aligned}$$

Since $\|T_{L_1}\|_{\ell_*^{0,\infty}} \le \max\{d_1, N(\Phi_1)\}$, it holds that

$$\|S_1 \circ T_{L_1}\|_{\ell^0} \le \|S_1\|_{\ell^0} \|T_{L_1}\|_{\ell_*^{0,\infty}} \le \|S_1\|_{\ell^0} \max\{d, N(\Phi_1)\},$$

according to Lemma C.2. As $1 \le \max\{d, N(\Phi_1)\}$ it follows that

$$\begin{aligned}
W(\Psi_2) &= \sum_{\ell=1}^{L_1-1} \|T_\ell\|_{\ell^0} + \|S_1 \circ T_{L_1}\|_{\ell^0} + \sum_{\ell=2}^{L_2} \|S_\ell\|_{\ell^0}\\
&\le \sum_{\ell=1}^{L_1-1} \|T_\ell\|_{\ell^0} + \max\{d, N(\Phi_1)\} \sum_{\ell=1}^{L_2} \|S_\ell\|_{\ell^0}\\
&\le W(\Phi_1) + \max\{d, N(\Phi_1)\} W(\Phi_2).
\end{aligned}$$

Note that if Φ_1 and Φ_2 are strict neural networks, then Ψ_2 is a strict neural network. It does not hold for Ψ_1 because the last term in Φ_1 contains an identity map, which breaks the strictness. ■

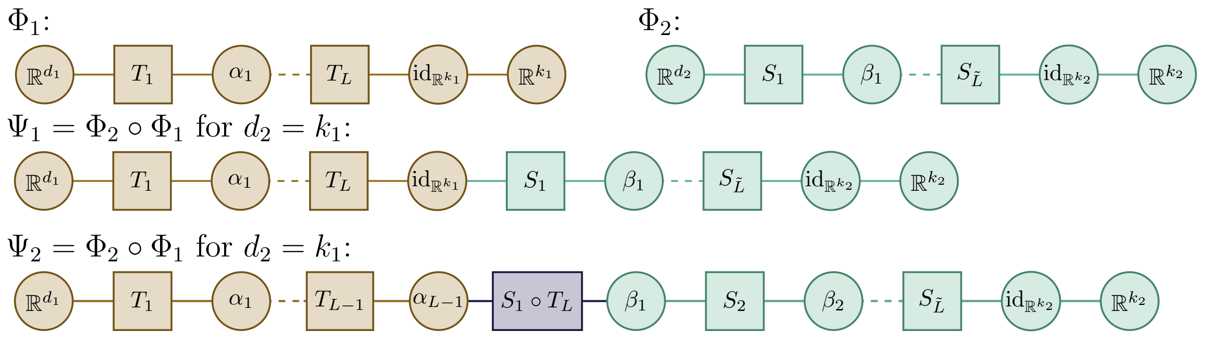

Figure II.6: *Illustration of the principle of the two possible implementations of composition, as constructed in the proof of Lemma II.17.*

Note that the previous results all consider an upper bound, but this is not necessarily an optimal upper bound. These results will mostly be used for realization, where there exists an underlying neural network, which is not explicitly mentioned at the time.

The previous results imply that the class of realizations of generalized neural networks admits good closure properties under linear combinations and compositions of functions, which do not hold in general for realizations of strict neural networks.

III | Activation Functions

The choice of activation function influences the approximation properties of a neural network. Therefore the choice of activation function is important, as well as the properties they have. The chapter is based on [12, p. 259-276].

A specific activation function that is widely used is the *rectified linear unit*, abbreviated *ReLU*, and it is defined in the following:

Definition III.1: ReLU & its Powers

Let $r \in \mathbb{N}$. Then $\varrho_r : \mathbb{R} \to \mathbb{R}$ given by

$$\varrho_r(x) := x_+^r := (\max\{0, x\})^r$$

is called **ReLU** for $r = 1$, its powers are called **ReLU-r** for $r > 1$. Additionally, **ReLU-**0 is defined by the **Heaviside functions** given by $\varrho_0 := \chi_{[0,\infty)}$.

Neural networks with ReLU activation function or its powers are also called *ReLU-networks*, or *ReLU-r-networks*. Notice per definition that $\varrho_r \in \mathcal{C}^{r-1}(\mathbb{R})$, for $r \in \mathbb{N}$. The motivation for using ReLU as an activation function is based on its wide use in other practical applications such as areas within machine learning. Other examples of activation functions as well as ReLU and one of its powers are illustrated in Figure III.1.

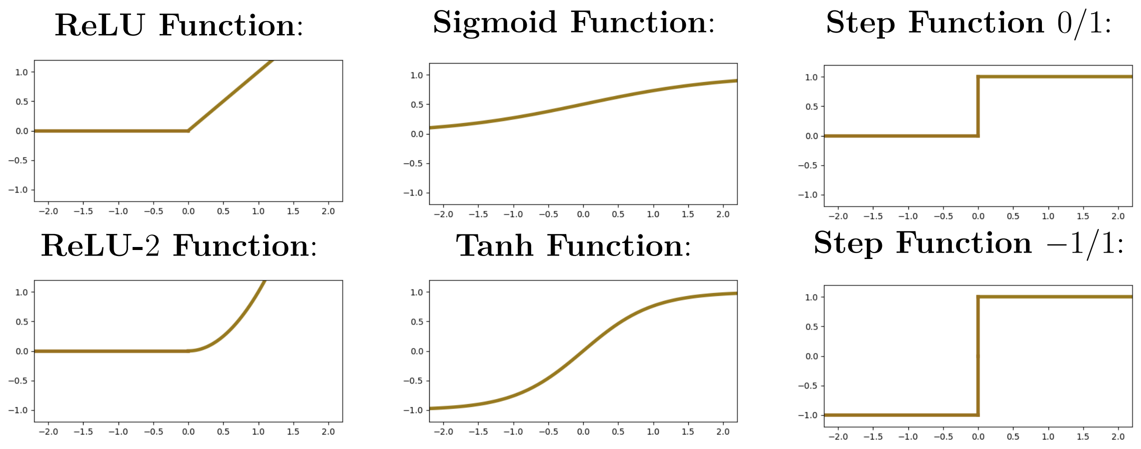

Figure III.1: *Illustration of different activation functions.*

Notation:

For the rest of the report let ϱ_r be ReLU-r for $r \in \mathbb{N}_0$.

III.1 Connections Between Activation Functions

This section explores general connections between activation functions such as when one is a realization of a neural network, and when one is a composition of another activation function.

The first result shows that if an activation function σ is a realization of a ϱ-network, then σ-networks can be converted to a ϱ-networks with controlled complexity.

Lemma III.2:

Let $\varrho, \sigma : \mathbb{R} \to \mathbb{R}$ be activation functions where σ is non-constant. Assume that there exists a $\Phi_\sigma \in \mathcal{NN}^{\varrho, 1, 1}_{w, \ell, n}$, with $L(\Phi_\sigma) = \ell, w, n \in \mathbb{N}$ such that $\mathtt{R}(\Phi_\sigma) = \sigma$. Then for all $W, N \in \mathbb{N}_0 \cup \{\infty\}$, $L \in \mathbb{N} \cup \{\infty\}$, and $d, k \in \mathbb{N}$, it holds that:

(i) If $\ell = 2$, then $\mathtt{NN}^{\sigma, d, k}_{W, L, N} \subseteq \mathtt{NN}^{\varrho, d, k}_{n^2 W, L, nN}$.

(ii) If $\ell > 2$, then $\mathtt{NN}^{\sigma, d, k}_{W, L, N} \subseteq \mathtt{NN}^{\varrho, d, k}_{nW + wN, (L-1)\ell + 1, (n+1)N}$.

Proof

(i) Let $\Psi \in \mathcal{NN}^{\sigma, d, k}_{W, L, N}$ be given by

$$\Psi := \Big((T_1, \alpha_1), \ldots, (T_{L(\Psi)-1}, \alpha_{L(\Psi)-1}), (T_{L(\Psi)}, \mathrm{id}_{\mathbb{R}^k})\Big).$$

It is desired to prove that there exists some

$$\Phi \in \mathcal{NN}^{\varrho, d, k}_{n^2 W, L, nN},$$

such that $\mathtt{R}(\Psi) = \mathtt{R}(\Phi)$. If $L(\Phi) = 1$, then $N(\Phi) = 0$, so $\Psi \in \mathcal{NN}^{\varrho, d, k}_{n^2 W, L, nN}$. Therefore assume that $L(\Psi) > 1$.

Let N_m be the number of neurons in the m'th layer of Ψ. By Lemma C.3, with $d = N_m$ there exists a

$$\Phi_m := \Big((T_{m,1}, \beta_m), (T_{m,2}, \mathrm{id}_{\mathbb{R}^{N_m}})\Big) \in \mathcal{NN}^{\varrho, N_m, N_m}_{wN_m, 2, nN_m}$$

for each $m \in \{1, \ldots, L(\Psi) - 1\}$ such that $\mathtt{R}(\Phi_m) = \alpha_m$, where

$$\beta_m : \mathbb{R}^{N(\Phi_m)} \to \mathbb{R}^{N(\Phi_m)},$$

with $N(\Phi_m) \leq nN_m$, $\|T_{m,1}\|_{\ell^{0,\infty}} \leq n$, and $\|T_{m,2}\|_{\ell^{0,\infty}_*} \leq n$. Now define

$$\begin{aligned} S_1 &:= T_{1,1} \circ T_1, \\ S_m &:= T_{m,1} \circ T_m \circ T_{m-1,2} \\ S_{L(\Psi)} &:= T_{L(\Psi)} \circ T_{L(\Psi)-1,2}, \end{aligned}$$

for $1 < m < L(\Psi)$ and

$$\Phi := \Big((S_1, \beta_1), \ldots, (S_{L(\Psi)-1}, \beta_{L(\Psi)-1}), (S_{L(\Psi)}, \mathrm{id}_{\mathbb{R}^k})\Big).$$

Then per construction, it holds that $L(\Phi) = L(\Psi)$. By considering the first five terms in the tuple one can see that

$$\begin{aligned} S_3 \circ \beta_2 \circ S_2 \circ \beta_1 \circ S_1 &= (T_{3,1} \circ T_3 \circ T_{2,2}) \circ \beta_2 \circ (T_{2,1} \circ T_2 \circ T_{1,2}) \circ \beta_1 \circ (T_{1,1} \circ T_1) \\ &= T_{3,1} \circ T_3 \circ (T_{2,2} \circ \beta_2 \circ T_{2,1}) \circ T_2 \circ (T_{1,2} \circ \beta_1 \circ T_{1,1}) \circ T_1 \\ &= T_{3,1} \circ T_3 \circ \alpha_2 \circ T_2 \circ \alpha_1 \circ T_1, \end{aligned}$$

since $\mathtt{R}(\Phi_m) = \alpha_m$ per construction. Thus by following the same steps for all terms in the tuple, it holds that $\mathtt{R}(\Phi) = \mathtt{R}(\Psi)$. Per assumption $n \geq 1$, so by Lemma C.2 it holds that

$$\begin{aligned} \|S_1\|_{\ell^0} &\leq \|T_{1,1}\|_{\ell^{0,\infty}} \|T_1\|_{\ell^0} \leq n^2 \|T_1\|_{\ell^0}, \\ \|S_m\|_{\ell^0} &\leq \|T_{m,1}\|_{\ell^{0,\infty}} \|T_m\|_{\ell^0} \|T_{m-1,2}\|_{\ell^{0,\infty}_*} \leq n^2 \|T_m\|_{\ell^0}, \\ \|S_{L(\Phi)}\|_{\ell^0} &\leq \|T_{L(\Phi)}\|_{\ell^0} \|T_{L(\Phi)-1,2}\|_{\ell^{0,\infty}_*} \leq n^2 \|T_{L(\Phi)}\|_{\ell^0}, \end{aligned}$$

for $1 < m < L(\Phi)$. From this and the construction of Φ it follows that

$$\begin{aligned} W(\Phi) &= \sum_{m=1}^{L(\Phi)} \|S_m\|_{\ell^0} \leq n^2 \sum_{m=1}^{L(\Phi)} \|T_m\|_{\ell^0} = n^2 W(\Psi), \\ N(\Phi) &= \sum_{m=1}^{L(\Phi)-1} N(\Phi_m) \leq n \sum_{m=1}^{L(\Phi)-1} N_m = nN(\Psi), \end{aligned}$$

which proves (i).

(ii) The proof of

$$\mathtt{NN}^{\sigma,\, d,\, k}_{W,\, L,\, N} \subseteq \mathtt{NN}^{\varrho,\, d,\, k}_{nW+wN,\, (L-1)\ell+1,\, (n+1)N}.$$

is done by induction on $L \in \mathbb{N}$. For $L = 1$ it is clear that

$$\mathtt{NN}^{\sigma,\, d,\, k}_{W,\, 1,\, N} \subseteq \mathtt{NN}^{\varrho,\, d,\, k}_{nW+wN,\, 1,\, (n+1)N}$$

since $w \geq 1$, $n \geq 1$ and the activation function plays no role for neural networks with no hidden layers and neurons.

Now for the induction step assume that the statement holds for L and consider $f \in \mathtt{NN}^{\sigma,\, d,\, k}_{W,\, L+1,\, N}$. If $f \in \mathtt{NN}^{\sigma,\, d,\, k}_{W,\, L,\, N}$ then by the induction assumption

$$\begin{aligned} \mathtt{NN}^{\sigma,\, d,\, k}_{W,\, L,\, N} &\subseteq \mathtt{NN}^{\varrho,\, d,\, k}_{nW+wN,\, (L-1)\ell+1,\, (n+1)N} \\ &\subseteq \mathtt{NN}^{\varrho,\, d,\, k}_{nW+wN,\, L\ell+1,\, (n+1)N}, \end{aligned}$$

so assume that $f \notin \mathtt{NN}^{\sigma,\, d,\, k}_{W,\, L,\, N}$. Then there exists a neural network

$$\Psi := \Big((T_1, \alpha_1), \ldots, (T_L, \alpha_L), (T_{L+1}, \mathrm{id}_{\mathbb{R}^k}) \Big) \in \mathcal{NN}^{\sigma,\, d,\, k}_{W,\, L+1,\, N},$$

with $\mathtt{R}(\Psi) = f$, and $T_{L+1} : \mathbb{R}^{k_L} \to \mathbb{R}^k$, where k_L is the number of neurons in layer L. Define

$$\Psi_L := \Big((T_1, \alpha_1), \ldots, (T_{L-1}, \alpha_{L-1}), (T_L, \mathrm{id}_{\mathbb{R}^{k_L}}) \Big) \in \mathcal{NN}^{\sigma,\, d,\, k_L}_{W_L,\, L,\, N_L},$$

where $W_L := W(\Psi_L)$ and $N_L := N(\Psi_L)$. Thus it follows that

$$\begin{aligned} W_L + \|T_{L+1}\|_{\ell^0} &= W(\Psi) \leq W, \\ N_L + k_L &= N(\Psi) \leq N. \end{aligned}$$

By the induction hypothesis, it holds that

$$\mathtt{R}(\Psi_L) \in \mathtt{NN}^{\sigma,\, d,\, k_L}_{W_L,\, L,\, N_L} \subseteq \mathtt{NN}^{\varrho,\, d,\, k}_{nW_L+wN_L,\, (L-1)\ell+1,\, (n+1)N_L}.$$

Consider

$$\Phi_L := \Big((S_1,\beta_1),\dots,(S_{\tilde{L}-1},\beta_{\tilde{L}-1}),(S_{\tilde{L}},\mathrm{id}_{\mathbb{R}^{k_L}})\Big) \in \mathcal{NN}^{\varrho,\,d,\,k_L}_{nW_L+wN_L,\,(L-1)\ell+1,\,(n+1)N_L}$$

with $\mathtt{R}(\Phi_L)=\mathtt{R}(\Psi_L)$. Thus $f = T_{L+1}\circ\alpha_L\circ\mathtt{R}(\Phi_L)$. Furthermore by Lemma C.3 it follows that there exists a

$$\Phi_\alpha := \Big((U_1,\gamma_1),\dots,(U_{\ell-1},\gamma_{\ell-1}),(U_\ell,\mathrm{id}_{\mathbb{R}^{k_L}})\Big) \in \mathcal{NN}^{\varrho,\,k_L,\,k_L}_{wk_L,\,\ell,\,nk_L}$$

with $\mathtt{R}(\Phi_\alpha)=\alpha_L$ and $\|U_\ell\|_{\ell^{0,\infty}_*}\le n$. Thus for the neural network

$$\begin{aligned}\Phi := \Big(&(S_1,\beta_1),\dots,(S_{\tilde{L}-1},\beta_{\tilde{L}-1}),(S_{\tilde{L}},\mathrm{id}_{\mathbb{R}^{k_L}}),\\ &(U_1,\gamma_1),\dots,(U_{\ell-1},\gamma_{\ell-1}),(T_{L+1}\circ U_\ell,\mathrm{id}_{\mathbb{R}^{k}})\Big)\end{aligned}$$

it holds that $\mathtt{R}(\Phi)=f$. Moreover,

$$\begin{aligned}L(\Phi) &= \tilde{L}+\ell\\ &\le (L-1)\ell+1+\ell\\ &= L\ell+1,\end{aligned}$$

$$\begin{aligned}N(\Phi) &= N(\Phi_\alpha)+N(\Phi_L)+k_L\\ &\le nk_L+(n+1)N_L+k_L\\ &= (n+1)(N_L+k_L)\\ &\le (n+1)N,\end{aligned}$$

and by Lemma C.2

$$\begin{aligned}W(\Phi) &= W(\Phi_\alpha)+(W(\Phi_L)-\|U_\ell\|_{\ell^0})+\|T_{L+1}\circ U_\ell\|_{\ell^0}\\ &\le wk_L+(nW_L+wN_L)+\|T_{L+1}\|_{\ell^0}\|U_\ell\|_{\ell^{0,\infty}_*}\\ &\le w(k_L+N_L)+(W_L+\|T_{L+1}\|_{\ell^0})n\\ &\le wN+nW.\end{aligned}$$

Thus $f\in\mathtt{NN}^{\varrho,\,d,\,k}_{nW+wN,\,L\ell+1,\,(n+1)N}$, which was desired. ■

In the case where σ is an s-fold composition of ϱ, an improvement of Lemma III.2 can be improved to the following:

Lemma III.3:

Let $\varrho,\sigma:\mathbb{R}\to\mathbb{R}$ be an activation function, let $s\in\mathbb{N}$, and let

$$\sigma := \underbrace{\varrho\circ\cdots\circ\varrho}_{s\text{ terms}}.$$

Then for all $W,N\in\mathbb{N}_0\cup\{\infty\}$ and $L\in\mathbb{N}\cup\{\infty\}$ it holds that

$$\mathtt{NN}^{\sigma,\,d,\,k}_{W,\,L,\,N}\subseteq\mathtt{NN}^{\varrho,\,d,\,k}_{W+(s-1)N,\,(L-1)s+1,\,sN}.$$

An analogous statement holds for strict neural networks, where `NN` is replaced with `SNN`.

Proof
Let

$$\Phi \coloneqq \Big((T_1,\alpha_1),\dots,(T_{L(\Phi)-1},\alpha_{L(\Phi)-1}),(T_{L(\Phi)},\mathrm{id}_{\mathbb{R}^k})\Big) \in \mathcal{NN}^{\sigma,\,d,\,k}_{W,\,L,\,N}.$$

It is desired to prove that there exists some

$$\Psi \in \mathcal{NN}^{\varrho,\,d,\,k}_{W+(s-1)N,\,(L-1)s+1,\,sN}$$

such that $\mathtt{R}(\Psi) = \mathtt{R}(\Phi)$. If $s = 1$ or $L(\Phi) = 1$, then the result follows trivially, so assume that $s \geq 2$ and $L(\Phi) \geq 2$.

For $\ell \in \{1,\dots,L(\Phi)-1\}$ let N_ℓ denote the number of neurons in the ℓ'th layer of Φ, and define $\alpha_\ell \coloneqq \alpha_\ell^{(1)} \otimes \cdots \otimes \alpha_\ell^{(N_\ell)}$, where $\alpha_\ell^{(i)} \in \{\sigma, \mathrm{id}_{\mathbb{R}}\}$. For $\ell \in \{1,\dots,L(\Phi)-1\}$, $j \in \{1,\dots,N_\ell\}$, and $i \in \{1,\dots,s\}$ define

$$\beta^{(j)}_{(\ell-1)s+i} \coloneqq \begin{cases} \varrho, & \text{if } \alpha_\ell^{(j)} = \sigma, \\ \mathrm{id}_{\mathbb{R}}, & \text{if } \alpha_\ell^{(j)} \neq \sigma, \end{cases}$$
$$\beta_{(\ell-1)s+i} \coloneqq \bigotimes_{j=1}^{N_\ell} \beta^{(j)}_{(\ell-1)s+i}.$$

Moreover, for $\ell \in \{1,\dots,L(\Phi)-1\}$, and $i \in \{2,\dots,s\}$, define

$$S_{(\ell-1)s+1} \coloneqq T_\ell : \mathbb{R}^{N_{\ell-1}} \to \mathbb{R}^{N_\ell},$$
$$S_{(\ell-1)s+i} \coloneqq \mathrm{id}_{\mathbb{R}^{N_\ell}}.$$

By construction, it holds that

$$\begin{aligned}\alpha_\ell \circ T_\ell &= \beta_{(\ell-1)s+s} \circ \cdots \circ \beta_{(\ell-1)s+1} \circ S_{(\ell-1)s+1} \\ &= \beta_{(\ell-1)s+s} \circ S_{(\ell-1)s+s} \circ \cdots \circ \beta_{(\ell-1)s+1} \circ S_{(\ell-1)s+1} \\ &= \beta_{\ell s} \circ S_{\ell s} \circ \cdots \circ \beta_{(\ell-1)s+1} \circ S_{(\ell-1)s+1}\end{aligned}$$

and thus

$$\begin{aligned}\mathtt{R}(\Phi) &= T_{L(\Phi)} \circ \alpha_{L(\Phi)-1} \circ T_{L(\Phi)-1} \circ \cdots \circ T_1 \circ \alpha_1 \\ &= T_{L(\Phi)} \circ \beta_{s(L(\Phi)-1)} \circ S_{s(L(\Phi)-1)} \circ \cdots \circ S_1 \circ \beta_1.\end{aligned}$$

Then the ϱ-network

$$\begin{aligned}\Psi \coloneqq \Big((S_1,\beta_1),\dots,(S_{(L(\Phi)-1)s},\beta_{(L(\Phi)-1)s}),(T_{L(\Phi)},\mathrm{id}_{\mathbb{R}^k})\Big) &\in \mathcal{NN}^{\varrho,\,d,\,k}_{W_\Psi,\,(L(\Phi)-1)s+1,\,sN} \\ &\subseteq \mathcal{NN}^{\varrho,\,d,\,k}_{W_\Psi,\,(L-1)s+1,\,sN},\end{aligned}$$

where

$$\begin{aligned}W_\Psi &\coloneqq W(\Psi) \\ &= \|T_{L(\Phi)}\|_{\ell^0} + \sum_{j=1}^{(L(\Phi)-1)s} \|S_j\|_{\ell^0} \\ &= \|T_{L(\Phi)}\|_{\ell^0} + \sum_{\ell=1}^{L(\Phi)-1} \sum_{i=1}^{s} \|S_{(\ell-1)s+i}\|_{\ell^0}\end{aligned}$$

$$\begin{aligned}
&= \|T_{L(\Phi)}\|_{\ell^0} + \sum_{\ell=1}^{L(\Phi)-1} \left(\|S_{(\ell-1)s+1}\|_{\ell^0} + \sum_{i=2}^{s} \|S_{(\ell-1)s+i}\|_{\ell^0} \right) \\
&= \|T_{L(\Phi)}\|_{\ell^0} + \sum_{\ell=1}^{L(\Phi)-1} \left(\|T_\ell\|_{\ell^0} + (s-1)N_\ell \right) \\
&= \sum_{\ell=1}^{L(\Phi)} \|T_\ell\|_{\ell^0} + (s-1) \sum_{\ell=1}^{L(\Phi)-1} N_\ell \\
&= W(\Phi) + (s-1)N(\Phi) \\
&\le W + (s-1)N.
\end{aligned}$$

Therefore $\Psi \in \mathcal{NN}^{\varrho,\,d,\,k}_{W+(s-1)N,\,(L-1)s+1,\,sN}$. Additionally, if Φ is a strict neural network, then Ψ is strict as well. ■

The previous results consider the case where σ can be implemented by a ϱ-network. The next result considers the case where σ is the limit of a sequence of ϱ-networks.

Lemma III.4:

Let $\varrho, \sigma : \mathbb{R} \to \mathbb{R}$ be activation functions, and let σ be continuous. Assume that there exists a family $\Phi_m \in \mathcal{NN}^{\varrho,\,1,\,1}_{w,\,\ell,\,n}$ parameterized by $m \in \mathbb{R}$, with $L(\Phi_m) = \ell \in \mathbb{N}$, and $w, n \in \mathbb{N}_0$ such that

$$\sigma_m := \mathtt{R}(\Phi_m) \xrightarrow[m\to 0]{} \sigma,$$

with locally uniformly on $\mathbb{R}$. Then, for any $d, k \in \mathbb{N}$, $W, N \in \mathbb{N}_0$, and $L \in \mathbb{N}$, it holds that

(i) If $\ell = 2$, then $\mathtt{NN}^{\sigma,\,d,\,k}_{W,\,L,\,N} \subseteq \overline{\mathtt{NN}^{\varrho,\,d,\,k}_{n^2W,\,L,\,nN}}$.

(ii) If $\ell > 2$, then $\mathtt{NN}^{\sigma,\,d,\,k}_{W,\,L,\,N} \subseteq \overline{\mathtt{NN}^{\varrho,\,d,\,k}_{nW+wN,\,(L-1)\ell+1,\,(n+1)N}}$.

The closure is with respect to locally uniform convergence.

Proof
Let $f \in \mathtt{NN}^{\sigma,\,d,\,k}_{W,\,L,\,N}$. Then there exists a

$$\Psi := \Big((T_1, \alpha_1), \dots, (T_{\tilde{L}-1}, \alpha_{\tilde{L}-1}), (T_{\tilde{L}}, \mathrm{id}_{\mathbb{R}^k}) \Big) \in \mathcal{NN}^{\sigma,\,d,\,k}_{W,\,\tilde{L},\,N},$$

such that $\mathtt{R}(\Psi) = f$, with $L(\Psi) = \tilde{L} \le L$, and

$$\alpha_\ell := \bigotimes_{n=1}^{N_\ell} \sigma_n^{(\ell)},$$

where $\sigma_n^{(\ell)} \in \{\sigma, \mathrm{id}_\mathbb{R}\}$. It is desired to construct a sequence f_m that converges to f and use Lemma III.2 on this sequence.

For $1 \le \ell \le \tilde{L}$ construct

$$\alpha_{m,\ell}^{(n)} := \begin{cases} \sigma_m, & \text{if } \sigma_n^{(\ell)} = \sigma, \\ \mathrm{id}_\mathbb{R}, & \text{if } \sigma_n^{(\ell)} = \mathrm{id}_\mathbb{R}, \end{cases}$$

$$\alpha_{m,\ell} := \bigotimes_{n=1}^{N_\ell} \alpha_{m,\ell}^{(n)}$$

and

$$\Psi_m := \Big((T_1, \alpha_{m,1}), \dots, (T_{\tilde{L}-1}, \alpha_{m,\tilde{L}-1}), (T_{\tilde{L}}, \mathrm{id}_{\mathbb{R}^k})\Big) \in \mathcal{NN}^{\sigma_m, d, k}_{W, \tilde{L}, N}.$$

Note that $\alpha_{m,\ell}$ is just α_ℓ, where σ is replaces by σ_m. Define $f_m := \mathtt{R}(\Psi_m) \in \mathtt{NN}^{\sigma_m, d, k}_{W, \tilde{L}, N}$. By Lemma C.4 $f_m \to f$ locally uniformly since σ is continuous and $\sigma_m \to \sigma$ locally uniformly on $\mathbb{R}$ as $m \to 0$.

Per assumption $\sigma_m \in \mathtt{NN}^{\varrho, 1, 1}_{w, \ell, n}$ so Lemma III.2 gives that

$$f_m \in \mathtt{NN}^{\sigma_m, d, k}_{W, \tilde{L}, N} \subseteq \begin{cases} \mathtt{NN}^{\varrho, d, k}_{n^2 W, \tilde{L}, nN}, & \text{if } \ell = 2, \\ \mathtt{NN}^{\varrho, d, k}_{nW + wN, (\tilde{L}-1)\ell + 1, (n+1)N}, & \text{if } \ell > 2. \end{cases}$$

This shows the desired result, since $\tilde{L} \leq L$ and $f_m \to f$ locally uniformly. ■

Next, a relation between the strict and generalized neural networks is highlighted under specific assumptions on the activation function. To prove this, the *network compatible topology family* is required, see Definition C.5.

Lemma III.5:

Let $d, k \in \mathbb{N}$, $L \in \mathbb{N} \cup \{\infty\}$, $N \in \mathbb{N}_0 \cup \{\infty\}$ and $W \in \mathbb{N}_0$. Moreover, let $\varrho : \mathbb{R} \to \mathbb{R}$ be a continuous activation function and assume that ϱ is differentiable at some $x_0 \in \mathbb{R}$ with $\varrho'(x_0) \neq 0$. Then it holds that

$$\mathtt{NN}^{\varrho, d, k}_{W, L, W} \subseteq \overline{\mathtt{SNN}^{\varrho, d, k}_{4W, L, 2N}},$$

where the closure is with respect to locally uniform convergence.

Proof

At first, it is desired to construct a network compatible topology family. For $d, k \in \mathbb{N}$, define

$$\mathcal{G}_{d,k} := \{f : \mathbb{R}^d \to \mathbb{R}^k \mid f \text{ continuous}\},$$

and let $\mathcal{T}_{d,k}$ denote the topology of locally uniform convergence on $\{f : \mathbb{R}^d \to \mathbb{R}^k\}$. For $\mathcal{T}_{d,k}$ to be a network compatible topology family, the conditions (i)-(iii) in Definition C.5 have to hold. Since affine-linear functions are continuous, (i) holds, and (iii) holds by Lemma C.4. To check if (ii) holds, consider a $f_i^{(n)} : \mathbb{R} \to \mathbb{R}$ satisfying

$$f_i^{(n)} \xrightarrow[n \to \infty]{} f_i^{(0)}$$

locally uniformly for all $i \in \{1, \dots, p\}$. Then it holds that

$$f_1^{(n)} \otimes \cdots \otimes f_p^{(n)} \xrightarrow[n \to \infty]{} f_1^{(0)} \otimes \cdots \otimes f_p^{(0)}$$

locally uniformly. Thus by defining $\mathcal{G} := \{\mathcal{G}_{d,k}\}_{d,k \in \mathbb{N}}$ and $\mathcal{T} := \{\mathcal{T}_{d,k}\}_{d,k \in \mathbb{N}}$, it holds that $(\mathcal{G}, \mathcal{T})$ is network compatible topology family.

It is now desired to use Proposition C.6. Therefore it is required to check that $\varrho \in \mathcal{G}_{1,1}$, and that there exists a $n \in \mathbb{N}$ such that for all $m \in \mathbb{N}$ there exist affine linear maps $F_m : \mathbb{R} \to \mathbb{R}$, $D_m : \mathbb{R}^n \to \mathbb{R}$, and $E_m : \mathbb{R} \to \mathbb{R}^n$ such that

$$F_m := D_m \circ (\varrho \otimes \cdots \otimes \varrho) \circ E_m$$

satisfying $F_m \to \mathrm{id}_{\mathbb{R}}$ as $m \to \infty$ with locally uniform convergence. The first condition holds since ϱ is continuous.

To check the last condition let $c := \varrho'(x_0) \neq 0$, and $\varepsilon_m := |c|/m$ for all $m \in \mathbb{N}$. By definition of the derivative, there exists a $\delta_m > 0$ such that

$$\left| \frac{\varrho(x_0 + h) - \varrho(x_0)}{h} - c \right| \leq \varepsilon_m = \frac{|c|}{m}, \tag{III.1}$$

for $0 < |h| \leq \delta_m$. Now, define the affine-linear maps

$$\begin{aligned}
E_m &: \mathbb{R} \to \mathbb{R}^2, & x &\mapsto \left(x_0 + \frac{\delta_m x}{\sqrt{m}}, x_0 \right)^T, \\
D_m &: \mathbb{R}^2 \to \mathbb{R}, & (x_1, x_2) &\mapsto \frac{\sqrt{m}(x_1 - x_2)}{\delta_m c}, \\
F_m &: \mathbb{R} \to \mathbb{R}, & x &\mapsto \left(D_m \circ (\varrho \otimes \varrho) \circ E_m \right)(x).
\end{aligned}$$

Then let $x \in \mathbb{R}$ satisfy $0 < |x| \leq \sqrt{m}$ and define $h := \delta_m x/\sqrt{m}$ for which it holds that $0 < |h| \leq \delta_m$. By multiplying the left term in (III.1) with $|h|/|c|$ it holds that

$$\begin{aligned}
\left| \frac{\varrho(x_0 + h) - \varrho(x_0)}{h} - c \right| \frac{|h|}{|c|} &= \left| \frac{\varrho(x_0 + h) - \varrho(x_0)}{h} \frac{h}{c} - c\frac{h}{c} \right| \\
&= \left| \frac{\varrho(x_0 + h) - \varrho(x_0)}{c} - h \right|.
\end{aligned}$$

Therefore (III.1) implies that

$$\left| \frac{\varrho(x_0 + h) - \varrho(x_0)}{c} - h \right| \leq \frac{|c|}{m} \frac{|h|}{|c|} = \frac{|h|}{m}.$$

Thus for $x \in \mathbb{R}$ satisfying $0 < |x| \leq \sqrt{m}$ it holds that

$$\begin{aligned}
|F_m(x) - x| &= \left| \left(\varrho \left(x_0 + \frac{\delta_m x}{\sqrt{m}} \right) - \varrho(x_0) \right) \frac{\sqrt{m}}{\delta_m c} - x \right| \\
&= \left| \left(\varrho\left(x_0 + h\right) - \varrho(x_0) \right) \frac{\sqrt{m}}{\delta_m c} - h \frac{\sqrt{m}}{\delta_m} \right| \\
&= \left| \frac{\varrho(x_0 + h) - \varrho(x_0)}{c} - h \right| \frac{\sqrt{m}}{\delta_m} \\
&\leq \frac{|h|}{m} \frac{\sqrt{m}}{\delta_m} \\
&= \frac{|h|}{\sqrt{m}\delta_m} \\
&= \frac{|x|}{m} \\
&\leq \frac{1}{\sqrt{m}}.
\end{aligned}$$

Additionally the inequality $|F_m(x) - x| \leq 1/\sqrt{m}$ holds for $x = 0$. Therefore it holds that $|F_m(x) - x| \leq 1/\sqrt{m}$ for all $x \in \mathbb{R}$, where $|x| \leq \sqrt{m}$, which is $F_m \to \mathrm{id}_{\mathbb{R}}$ as $m \to \infty$ with locally uniform convergence. Now Proposition C.6 gives the desired result. ∎

Note that the convergence in the lemma is only locally uniformly. Later on, it is proven that this is not strong enough to ensure equality of the associated approximation spaces on unbounded domains. Therefore a specific condition on the activation function is needed, to ensure that strict and generalized neural networks yield the same approximation spaces on unbounded domains. This is the focus of the next section.

III.2 Functions that can Represent the Identity

This section is devoted to a condition on the activation function that is needed to get the desired result in the unbounded domain. This condition is that functions can *represent a function with n term.* This is given in the following definition:

Definition III.6: Representation of a Function with n Terms

Let $\varrho : \mathbb{R} \to \mathbb{R}$, and let $n \in \mathbb{N}$. Then ϱ **can represent** $f : \mathbb{R} \to \mathbb{R}$ **with** n **terms**, if $f \in \mathtt{SNN}^{\varrho,1,1}_{\infty,2,n}$, that is if

$$\exists \left\{a^{(1)}_m\right\}_{m=1}^n, \left\{a^{(2)}_m\right\}_{m=1}^n, \left\{b^{(1)}_m\right\}_{m=1}^n \subset \mathbb{R},\, b^{(2)} \in \mathbb{R} :$$
$$f(x) = b^{(2)} + \sum_{m=1}^n a^{(2)}_m \varrho\left(a^{(1)}_m x + b^{(1)}_m\right) \quad \forall x \in \mathbb{R}.$$

A special case of interest is when ϱ *can represent the identity* $\mathrm{id}_\mathbb{R} : \mathbb{R} \to \mathbb{R}$ *with* n *terms.* The primary example of an activation function with these properties is ReLU and its powers. This is stated in the following lemma since $\mathrm{id}_\mathbb{R}$ is a monomial of degree 1. As a preliminary denote the space of all polynomials of degree at most n, by $\mathbb{R}_{\deg\leq n}[x]$.

Lemma III.7:

Let $r \in \mathbb{N}$. Then ϱ_r can represent any polynomial of degree less or equal to r with $2r+2$ terms.

Proof
As a preliminary, note that for $x \geq 0$

$$\varrho_r(x) + (-1)^r \varrho_r(-x) = \varrho_r(x) = x_+^r = x^r,$$

and for $x < 0$

$$\varrho_r(x) + (-1)^r \varrho_r(-x) = (-1)^r \varrho_r(-x) = (-1)^r (-x)_+^r = (-1)^r(-x)^r = x^r.$$

Let T_y be the translation operator given by $T_y f(\cdot) := f(\cdot - y)$, for functions $f : \mathbb{R} \to \mathbb{R}$ and $y \in \mathbb{R}$, and let $g_n : \mathbb{R} \to \mathbb{R}$ be given by $x \mapsto x^n$ for $n \in \mathbb{N}_0$, with convention $g_0 = 1$. Then by defining

$$\mathcal{V}_n := \operatorname{span}\{T_y g_n \mid y \in \mathbb{R}\}$$

Lemma C.7 gives that $\mathcal{V}_r = \mathbb{R}_{\deg \leq r}[x]$ with dimension $r+1$. Now assume $f \in \mathbb{R}_{\deg \leq r}[x]$. Then there exist $c_i, y_i \in \mathbb{R}$ for $\{1, \dots, r+1\}$ such that for all $x \in \mathbb{R}$

$$\begin{aligned} f(x) &= \sum_{m=1}^{r+1} c_m (T_{y_m} g_r)(x) \\ &= \sum_{m=1}^{r+1} c_m (\varrho_r(x - y_m) + (-1)^r \varrho_r(-(x - y_m))), \end{aligned}$$

where the last equality holds from the preliminary step at the start of the proof. Thus every polynomial of degree at most r can be represented by ϱ_r with $2(r+1) = 2r+2$ terms, which was desired. ■

The next lemma considers the effect of an activation function being able to represent the identity.

Lemma III.8:

Let $\varrho : \mathbb{R} \to \mathbb{R}$ be an activation function, let $d, k \in \mathbb{N}$, $W, N \in \mathbb{N}_0$ and let $L \in \mathbb{N} \cup \{\infty\}$. Assume that ϱ can represent the identity with n terms. Then

$$\mathtt{NN}^{\varrho,\, d,\, k}_{W,\, L,\, N} \subseteq \mathtt{SNN}^{\varrho,\, d,\, k}_{n^2 W,\, L,\, nN}.$$

Proof, see Appendix B.

These representations of polynomials now allow for the representation of multiplication maps, which is stated in the following lemma.

Lemma III.9:

Let $d \in \mathbb{N}_{\geq 2}$ and $k \in \mathbb{N}$. Assume that $\varrho : \mathbb{R} \to \mathbb{R}$ can represent all polynomials of degree 2 with n terms. Then the following holds:

(i) The multiplication function $M_d : \mathbb{R}^d \to \mathbb{R}$, given by $x \mapsto \prod_{j=1}^{d} x_j$ satisfies

$$M_d \in \mathtt{NN}^{\varrho,\, d,\, 1}_{6n(2^m - 1),\, 2m,\, (2n+1)(2^m - 1) - 1}, \quad m := \lceil \log_2(d) \rceil.$$

In the case where $d = 2$, it holds that $M_2 \in \mathtt{NN}^{\varrho,\, 2,\, 1}_{6n,\, 2,\, 2n}$.

(ii) The multiplication map $M : \mathbb{R} \times \mathbb{R}^k \to \mathbb{R}^k$, given by $(c, x) \mapsto cx$ satisfies

$$M \in \mathtt{NN}^{\varrho,\, 1+k,\, k}_{6kn,\, 2,\, 2kn}.$$

Proof

(i) For simplicity define

$$w_m := 6n(2^m - 1), \quad v_m := (2n+1)(2^m - 1) - 1,$$

for $m \in \mathbb{N}$. At first, the goal is to prove that

$$M_{2^m} \in \mathtt{NN}^{\varrho,\, d,\, 1}_{w_m,\, 2m,\, v_m}.$$

This part is done by induction on $m \in \mathbb{N}$. At first, consider the setup for $m = 1$. Per assumption, ϱ can represent all polynomials of degree 2 with n terms. Therefore there exist $c, \alpha_i, \beta_i, \in \mathbb{R}$ for $i \in \{1, \dots, n\}$, such that for all $x \in \mathbb{R}$ it holds that

$$x^2 = c + \sum_{j=1}^{n} \beta_j \varrho(x - \alpha_j).$$

By defining the affine-linear maps

$$\begin{aligned} T_1 &: \mathbb{R} \to \mathbb{R}^n, \quad x \mapsto (x - \alpha_j)_{j=1}^n, \\ T_2 &: \mathbb{R}^n \to \mathbb{R}, \quad y \mapsto c + \sum_{j}^{n} \beta_j y_j, \end{aligned}$$

x^2 can be expressed as

$$x^2 = T_2 \circ \underbrace{(\varrho \otimes \cdots \otimes \varrho)}_{n \text{ factors}} \circ T_1(x) \quad \forall x \in \mathbb{R}.$$

Furthermore, define

$$\begin{aligned} T_0 &: \mathbb{R}^2 \to \mathbb{R}^2, \quad (x, y) \mapsto (x + y, x - y), \\ T_3 &: \mathbb{R}^2 \to \mathbb{R}, \quad (\tilde{x}, \tilde{y}) \mapsto \frac{1}{4}(\tilde{x} - \tilde{y}). \end{aligned}$$

In general, it holds that

$$xy = \frac{1}{4}\left((x+y)^2 - (x-y)^2\right) \quad \forall x, y \in \mathbb{R},$$

so by defining $S_1 : \mathbb{R}^2 \to \mathbb{R}^{2n}$ and $S_2 : \mathbb{R}^{2n} \to \mathbb{R}$ by

$$\begin{aligned} S_1 &:= (T_1 \otimes T_1) \circ T_0, \\ S_2 &:= T_3 \circ (T_2 \otimes T_2) \end{aligned}$$

respectively, it holds that for all $x, y \in \mathbb{R}$

$$xy = (S_2 \circ \underbrace{(\varrho \otimes \cdots \otimes \varrho)}_{2n \text{ factors}} \circ S_1)(x, y).$$

Note that the maps S_1 and S_2 satisfies that

$$\begin{aligned} \|S_1\|_{\ell^0} &\le 4n \le 6n, \\ \|S_2\|_{\ell^0} &\le 2n. \end{aligned}$$

Thus by defining

$$\Phi_1 := ((S_1, \varrho \otimes \cdots \otimes \varrho), (S_2, \mathrm{id}_{\mathbb{R}}))$$

it holds that $\Phi_1 \in \mathcal{NN}^{\varrho,2,1}_{6n,2,2n}$, and $\mathtt{R}(\Phi_1) = M_2$. Hence

$$M_2 \in \mathtt{SNN}^{\varrho,2,1}_{6n,2,2n} \subseteq \mathtt{NN}^{\varrho,2,1}_{6n,2,2n},$$

as desired for $m = 1$. Note this completes the case where $d = 2$ in (i).

For the induction step assume that the statement holds for m and define the affine linear maps $U_1, U_2 : \mathbb{R}^{2^{m+1}} \to \mathbb{R}^{2^m}$ by

$$\begin{aligned} U_1(x) &:= (x_1, \dots, x_{2^m}) =: \hat{x}, \\ U_2(x) &:= (x_{2^m+1}, \dots, x_{2^{m+1}}) =: \tilde{x} \end{aligned}$$

for $x \in \mathbb{R}^{2^{m+1}}$. Then

$$\begin{aligned} M_{2^{m+1}}(x) &= M_{2^m}(\hat{x})M_{2^m}(\tilde{x}) \\ &= M_2(M_{2^m}(U_1(x)),\, M_{2^m}(U_2(x))). \end{aligned}$$

By the induction hypothesis, there exists a neural network

$$\Phi_m := ((V_1, \alpha_1), \dots, (V_{L-1}, \alpha_{L-1}), (V_L, \mathrm{id}_{\mathbb{R}})) \in \mathcal{NN}^{\varrho,2^m,1}_{w_m,2m,v_m},$$

with $L := L(\Phi_m) \leq 2m$ such that $M_{2^m} = \mathtt{R}(\Phi_m)$. Since $\|U_j\|_{\ell^{0,\infty}_*} = 1$ for $j \in \{1,2\}$ Lemma C.2 gives that

$$\|V_1 \circ U_j\|_{\ell^0} \leq \|V_1\|_{\ell^0}, \quad j \in \{1,2\}.$$

For $j \in \{1,2\}$ define

$$\Psi_j := ((V_1 \circ U_j, \alpha_1), (V_2, \alpha_2), \dots, (V_{L-1}, \alpha_{L-1}), (V_L, \mathrm{id}_{\mathbb{R}})),$$

for which it holds that $W(\Psi_j) \leq W(\Phi_j)$, $N(\Psi_j) \leq N(\Phi_j)$, and $L(\Psi_j) = L$, that is

$$\Psi_j \in \mathcal{NN}^{\varrho,2^m,1}_{w_m,2m,v_m},$$

and $\mathtt{R}(\Psi_j) = M_{2^m} \circ U_j$. Hence

$$(M_{2^m} \circ U_1, M_{2^m} \circ U_2) \in \mathtt{NN}^{\varrho,2^{m+1},2}_{2w_m,2m,2v_m},$$

per Lemma II.16. Now since $M_2 \in \mathtt{NN}^{\varrho,2,1}_{6n,2,2n}$, Lemma II.17(i) gives that

$$M_{2^{m+1}} = M_2 \circ (M_{2^m} \circ U_1, M_{2^m} \circ U_2) \in \mathtt{NN}^{\varrho,2^{m+1},1}_{2w_m+6n,2m+2,2v_m+2n+2}.$$

By rewriting the upper bound for neurons and connections, it follows that

$$2w_m + 6n = 12n(2^m - 1) + 6n = 6n(2^{m+1} - 1) = w_{m+1},$$

and

$$\begin{aligned} 2v_m + 2n + 2 &= 2(2n+1)(2^m - 1) + 2n \\ &= (2n+1)(2^{m+1} - 2) + 2n + 1 - 1 \\ &= (2n+1)(2^{m+1} - 1) - 1 \\ &= v_{m+1}, \end{aligned}$$

which shows that

$$M_{2^{m+1}} \in \mathtt{NN}^{\varrho,\,2^{m+1},\,1}_{w_{m+1},\,2(m+1),\,v_{m+1}},$$

which completes the induction part.

Now let $d \in \mathbb{N}_{\geq 2}$ and define $P : \mathbb{R}^d \to \mathbb{R}^{2^m}$ given by

$$x \mapsto (x, 1_{2^m-d}) = (x, 0_{2^m-d}) + (0_d, 1_{2^m-d}),$$

where $m := \lceil \log_2(d) \rceil$. Per construction it is an affine linear map, and $\|P\|_{\ell^{0,\infty}_*} = 1$. Moreover, it holds that $M_d = M_{2^m} \circ P$, and per Lemma II.17(i) it holds that

$$M_d \in \mathtt{NN}^{\varrho,\,d,\,1}_{w_m,\,2m,\,v_m},$$

which was desired.

(ii) Consider $\Phi \in \mathcal{SNN}^{\varrho,\,2,\,1}_{6n,\,2,\,2n}$, with $L(\Phi) = 2$. Per (i) it holds that $M_2 : \mathbb{R}^2 \to \mathbb{R}$ given by $(x_1, x_2) \mapsto x_1 x_2$, satisfies that $M_2 = \mathtt{R}(\Phi)$. For $m \in \{1, \dots, k\}$ define $P_m : \mathbb{R} \times \mathbb{R}^k \to \mathbb{R} \times \mathbb{R}$, given by $(c, x) \mapsto (c, x_m)$. Note P_m is linear with

$$\|P_m\|_{\ell^{0,\infty}} = 1 = \|P_m\|_{\ell^{0,\infty}_*}.$$

Therefore Lemma II.17(i) gives that $M_2 \circ P_m = \mathtt{R}(\Phi_m)$ where $\Phi_m \in \mathcal{SNN}^{\varrho,\,1+k,\,1}_{6n,\,2,\,2n}$ and $L(\Phi_m) = L(\Phi) = 2$. By defining $M : \mathbb{R} \times \mathbb{R}^k \to \mathbb{R}^k$ given by $(c, x) \mapsto cx$ it holds that

$$(M_2 \circ P_m)(c, x) = cx_m = [M(c, x)]_m.$$

Hence

$$M = (M_2 \circ P_1, \dots, M_2 \circ P_k)$$

and Lemma II.16 implies that $M \in \mathtt{NN}^{\varrho,\,1+k,\,k}_{6kn,\,2,\,2kn}$, as desired. ∎

Remark III.10:

To use Lemma III.9 the activation function must be able to represent all polynomials of degree 2. By Lemma III.7 this holds for ϱ_r with $r \geq 2$. However, per Lemma III.7 this does not hold for ϱ_1, since it is only guaranteed that ϱ_1 can represent polynomials of degree less or equal to 1.

IV | Realization of B-splines

The so-called B-splines are a well-studied area of approximation theory and they are widely used in applications. Based on the previous chapters, this chapter aims to establish a theory that allows neural networks to realize B-splines. At first, B-splines are introduced. Next, realizations of B-splines are highlighted in the main result. The chapter is based on [12, p. 290-295, 303-304, 338-340] and [5, p. 633-639].

Definition IV.1: B-splines

Let $n \in \mathbb{N}_0$. The **B-spline of degree** 0 is given by $\beta_+^{(0)} := \chi_{[0,1)}$. The **B-spline of degree** n is given by the convolution of $\beta_+^{(0)}$ with itself $n+1$ times, that is

$$\beta_+^{(n)} := \underbrace{\beta_+^{(0)} * \cdots * \beta_+^{(0)}}_{n+1 \text{ terms}}.$$

The initial four B-splines are illustrated in Figure IV.1, visualizing that the B-splines become smoother, as the degree increases.

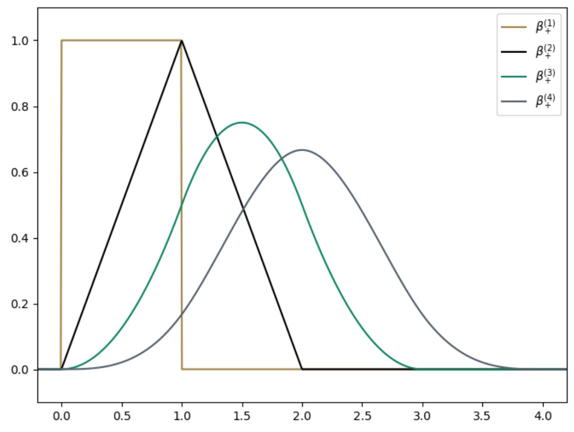

Figure IV.1: *Plot of B-splines* $\beta_+^{(n)}$ *with* $n \in \{0, 1, 2, 3\}$.

The next result shows that B-splines are non-negative and the support is restricted based on the degree.

Proposition IV.2:

Let $n \in \mathbb{N}_0$, and let $\beta_+^{(n)}$ be a B-spline of degree n. Then $\beta_+^{(n)}$ is non-negative and $\operatorname{supp}\left(\beta_+^{(n)}\right) \subseteq [0, n+1]$.

Proof, see Appendix B.

The first result, with a focus on approximating B-splines, is the following, which gives a decomposition of B-splines using ReLU activation functions. This decomposition of the B-splines is also called the *Irwin–Hall distribution.*

Theorem IV.3: Irwin-Hall Distribution

Let $n \in \mathbb{N}_0$, and let $\beta_+^{(n)}$ be a B-spline of degree n. Then $\beta_+^{(n)} \in \mathtt{SNN}^{\varrho_n,1,1}_{2(n+2),2,n+2}$ and it can be decomposed as

$$\beta_+^{(n)} = \frac{1}{n!}\sum_{k=0}^{n+1}\binom{n+1}{k}(-1)^k\varrho_n(x-k).$$

Proof

The claim is proven through an induction argument for $n \in \mathbb{N}_0$. For $n = 0$ consider the claimed decomposition:

$$\begin{aligned}\frac{1}{0!}\sum_{k=0}^{1}\binom{1}{k}(-1)^k\varrho_0(x-k) &= \varrho_0(x) - \varrho_0(x-1)\\ &= \chi_{[0,\infty)} - \chi_{[1,\infty)}\\ &= \chi_{[0,1)}\\ &= \beta_+^{(0)},\end{aligned}$$

which proves the initial step.

Now for the induction step assume that the statement holds for $n-1$. Consider $\beta_+^{(n)}$, where the three cases $x < 0$, $x \in [0, n+1]$, and $x > n+1$ are considered.

First, consider $x < 0$. Then both $\beta_+^{(n)}(x) = 0$ and $\varrho_n(x-k) = 0$ for $k = 0, \dots, n+1$. Thus the decomposition holds for $x < 0$.

Now consider $x \in [0, n+1]$. By the induction hypothesis, it holds that

$$\begin{aligned}\beta_+^{(n)}(x) &= \beta_+^{(n-1)} * \beta_+^{(0)}(x)\\ &= \int_{-\infty}^{\infty}\beta_+^{(n-1)}(x-t)\beta_+^{(0)}(t)\,dt\\ &= \int_0^1 \beta_+^{(n-1)}(x-t)\,dt\\ &= \frac{1}{(n-1)!}\sum_{k=0}^{n}\binom{n}{k}(-1)^k\int_0^1\varrho_{n-1}(x-k-t)\,dt.\end{aligned}$$

Now split the interval $[0, n+1]$ into sub-intervals and consider $x \in [K, K+1]$ for some arbitrary fixed $K \in \{0, 1, \dots, n\}$. The index k is considered in different cases: $k = K$, $k < K$, and $k > K$. With a focus on the integral, it holds that

$$\begin{aligned}\int_0^1\varrho_{n-1}(x-K-t)\,dt &= \int_0^{x-K}(x-K-t)^{n-1}\,dt\\ &= \frac{1}{n}(x-K)^n,\end{aligned}$$

for $k = K$, and

$$\begin{aligned}\int_0^1\varrho_{n-1}(x-k-t)\,dt &= \int_0^1(x-k-t)^{n-1}\,dt\\ &= \frac{1}{n}\left((x-k)^n - (x-1-k)^n\right),\end{aligned}$$

for $k = 0, 1, \dots, K-1$. Consider the partial sum for $k = 0, 1, \dots, K-1$: Splitting the sum and reordering the terms implies that

$$\begin{aligned}
&\sum_{k=0}^{K-1} \binom{n}{k} (-1)^k \int_0^1 \varrho_{n-1}(x-k-t)\,dt \\
&\quad = \frac{1}{n} \sum_{k=0}^{K-1} \binom{n}{k} (-1)^k \left((x-k)^n - (x-1-k)^n\right) \\
&\quad = \frac{1}{n} \left(\sum_{k=0}^{K-1} \binom{n}{k} (-1)^k (x-k)^n - \sum_{k=0}^{K-1} \binom{n}{k} (-1)^k (x-1-k)^n \right) \\
&\quad = \frac{1}{n} \left(\sum_{k=0}^{K-1} \binom{n}{k} (-1)^k (x-k)^n - \sum_{k=1}^{K} \binom{n}{k-1} (-1)^{k-1} (x-k)^n \right) \\
&\quad = \frac{1}{n} \left(x^n + \sum_{k=1}^{K-1} \left(\binom{n}{k} + \binom{n}{k-1} \right) (-1)^k (x-k)^n + \binom{n}{K-1} (-1)^K (x-K)^n \right) \\
&\quad = \frac{1}{n} \left(x^n + \sum_{k=1}^{K-1} \binom{n+1}{k} (-1)^k \varrho_n(x-k) + \binom{n}{K-1} (-1)^K \varrho_n(x-K) \right).
\end{aligned}$$

For $k = K+1, \dots, n$

$$\int_0^1 \varrho_{n-1}(x-k-t)\,dt = 0,$$

since the input in ReLU is non-positive. Thus, the partial sum for $k = K+1, \dots, n$ is equal to zero. Overall for $x \in [K, K+1]$ it holds that

$$\begin{aligned}
\beta_+^{(n)}(x) =& \frac{1}{(n-1)!} \frac{1}{n} \left(x^n + \sum_{k=1}^{K-1} \binom{n+1}{k} (-1)^k \varrho_n(x-k) \right. \\
&\left. + \binom{n}{K-1} (-1)^K \varrho_n(x-K) + \binom{n}{K} (-1)^K \varrho_n(x-K) \right) \\
=& \frac{1}{n!} \left(x^n + \sum_{k=1}^{K-1} \binom{n+1}{k} (-1)^k \varrho_n(x-k) + \left(\binom{n}{K-1} + \binom{n}{K} \right) (-1)^K \varrho_n(x-K) \right) \\
=& \frac{1}{n!} \sum_{k=0}^{K} \binom{n+1}{k} (-1)^k \varrho_n(x-k),
\end{aligned}$$

where the binomial coefficients are summed using Pascal's rule. Notice

$$0 = \sum_{k=K+1}^{n+1} \binom{n+1}{k} (-1)^k \varrho_n(x-k),$$

since $x \in [K, K+1]$, and the input in ReLU is non-negative. Using this it holds that

$$\beta_+^{(n)}(x) = \frac{1}{n!} \sum_{k=0}^{n+1} \binom{n+1}{k} (-1)^k \varrho_n(x-k),$$

for $x \in [K, K+1]$. Since K was arbitrary this gives the decomposition for $x \in [0, n+1]$.

For $x > n+1$ the proof follows by similar arguments as for $x \in [0, n+1]$: By setting $K = n+1$ and repeating the same steps for $k = K$ and $k < K$, it follows that the claim holds.

With the decomposition in place, it remains to show that $\beta_+^{(n)} \in \mathtt{SNN}^{\varrho_n,1,1}_{2(n+2),2,n+2}$. By defining $\alpha_1 : \mathbb{R}^{n+2} \to \mathbb{R}^{n+2}$ given by $\alpha_1 \coloneqq \varrho_n \otimes \cdots \otimes \varrho_n$, and

$$T_1 : \mathbb{R} \to \mathbb{R}^{n+2}, \qquad T_1(x) \coloneqq (x-k)_{k=0}^{n+1}$$
$$T_2 : \mathbb{R}^{n+2} \to \mathbb{R}, \qquad T_2(y) \coloneqq \frac{1}{n!}\sum_{k=0}^{n+1}\binom{n+1}{k}(-1)^k y_k,$$

it holds that $\beta_+^{(n)} = T_2 \circ \alpha_1 \circ T_1$. Note that $\|T_i\|_{\ell^0} = n+2$ for $i \in \{1,2\}$. Thus by defining $\Phi \coloneqq ((T_1,\alpha_1),(T_2,\mathrm{id}_\mathbb{R}))$ it holds that

$$L(\Phi) = 2, \quad N(\Phi) = n+2, \quad W(\Phi) = 2(n+2).$$

Overall $\Phi \in \mathcal{SNN}^{\varrho_n,1,1}_{2(n+2),2,n+2}$. ■

Remark IV.4:

Note that Theorem IV.3 deduce that B-splines can be decomposed by ReLU activation functions, which is the first motivation for approximating B-splines by ReLU-networks. This decomposition is an important property, that is going to play a key role in connecting B-splines and ReLU-networks.

Since $\varrho_n \in \mathcal{C}^{n-1}$, the decomposition in Theorem IV.3 implies the following:

Corollary IV.5:

Let $n \in \mathbb{N}$, and let $\beta_+^{(n)}$ be a B-spline of degree n. Then $\beta_+^{(n)} \in \mathcal{C}_c^{n-1}(\mathbb{R})$.

B-splines are originally univariate functions, however, through the application of the tensor product, a multivariate version can be generated.

Definition IV.6: Multivariate B-spline

Let $n \in \mathbb{N}_0$, $d \in \mathbb{N}$ and let $\beta_+^{(n)}$ be a B-spline of degree n. The **multivariate B-spline of degree** n denoted by $\beta_d^{(n)} : \mathbb{R}^d \to \mathbb{R}$ is given by the tensor product

$$\beta_d^{(n)}(x_1,\ldots,x_d) \coloneqq \beta_+^{(n)}(x_1)\cdots\beta_+^{(n)}(x_d)$$

Notation:

For the rest of the report let $\beta_+^{(n)}$ be a B-spline of degree n, and $\beta_d^{(n)}$ be a multivariate B-spline of degree n, where $n \in \mathbb{N}_0$ and $d \in \mathbb{N}$.

Note that $\beta_d^{(0)} \coloneqq \chi_{[0,1)^d}$. It is desired to generalize the decomposition in Theorem IV.3 to multivariate B-splines. First, a realization of multivariate B-splines is considered for degree $n \in \mathbb{N}$.

Theorem IV.7: Realization of Multivariate B-Splines

Let $d, n \in \mathbb{N}$ and assume $n \geq \min\{d, 2\}$. Then $\beta_d^{(n)} \in \mathtt{NN}_{W,L,N}^{\varrho_n,d,1}$ with $L := 2 + 2\lceil \log_2(d) \rceil$, and

$$W := \begin{cases} 28d(n+1), & \text{if } d > 1, \\ 2(n+2), & \text{if } d = 1, \end{cases}$$

$$N := \begin{cases} 13d(n+1), & \text{if } d > 1, \\ n+2, & \text{if } d = 1. \end{cases}$$

Proof

Per Theorem IV.3, it holds that

$$\beta_+^{(n)} \in \mathtt{NN}_{2(n+2),\,2,\,n+2}^{\varrho_n,1,1},$$

which gives the claim for $d = 1$. Therefore assume $d > 1$. Thus $n \geq \min\{d, 2\} = 2$, per assumption. It is intended to construct the multivariate B-splines by using the established operations on generalized neural networks and $\beta_+^{(n)}$. Define $f_j : \mathbb{R}^d \to \mathbb{R}$ given by

$$f_j := \beta_+^{(n)} \circ \pi_j$$

with $\pi_j : \mathbb{R}^d \to \mathbb{R}$ given by $\pi_j(x) = x_j$, for $j \in \{1, \dots, d\}$. Since $\|\pi_j\|_{\ell_*^{0,\infty}} = 1$, Lemma II.17(i) with $P = \pi_j$ and $Q = I$, implies that

$$f_j \in \mathtt{NN}_{2(n+2),\,2,\,n+2}^{\varrho_n,1,1}.$$

By forming the vector function $f := (f_1, \dots, f_d)$, Lemma II.16 shows that

$$f \in \mathtt{NN}_{2d(n+2),\,2,\,d(n+2)}^{\varrho_n,d,d}.$$

Since $n \geq 2$, Lemma III.7 gives that ϱ_n can represent any polynomial of degree two with $m := 2(n+1)$ terms. Thus the assumptions in Lemma III.9 are satisfied. Therefore there exists a multiplication function

$$M_d \in \mathtt{NN}_{6m(2^j-1),\,2j,\,(2m+1)(2^j-1)-1}^{\varrho_n,d,1},$$

where $j := \lceil \log_2(d) \rceil$, such that $\beta_d^{(n)} = M_d \circ f$. Next, the aim is to express the upper bounds in terms of d. Using $2^{j-1} < d \leq 2^j$, the following three inequalities holds:

$$2^j - 1 \leq 2(d-1),$$

$$\begin{aligned} 6m(2^j - 1) &\leq 12m(d-1) \\ &= 24(n+1)(d-1), \end{aligned}$$

$$\begin{aligned} (2m+1)(2^j - 1) - 1 &\leq (4m+2)(d-1) - 1 \\ &= (8n+10)(d-1) - 1. \end{aligned}$$

Thus

$$M_d \in \mathtt{NN}^{\varrho_n,\,d,\,1}_{24(n+1)(d-1),\,2j,\,(8n+10)(d-1)-1}.$$

Using Lemma II.17(ii) on the underlying neural networks for M_d and f, it holds that

$$\beta_d^{(n)} \in \mathtt{NN}^{\varrho_n,\,d,\,1}_{2d(n+2)+24(n+1)(d-1),\,2j+2,\,d(n+2)+(8n+10)(d-1)-1+d}.$$

Now, the goal is to simplify the upper bounds, to get the claim. Using basic calculus it holds that

$$\begin{aligned} 2d(n+2)+24(n+1)(d-1) &\le d(2n+4+24n+24) \\ &= d(26n+28) \\ &\le 28d(n+1), \end{aligned}$$

and

$$\begin{aligned} d(n+2)+(8n+10)(d-1)-1+d &\le d(n+2+8n+10+1) \\ &= d(9n+13) \\ &\le 13d(n+1). \end{aligned}$$

All in all, this proves the claim. ■

Some preliminary lemmas are needed to construct a similar result for B-splines of degree zero. For this, a function with specific properties is required. These are given by:

(Pσ) $\sigma : \mathbb{R} \to \mathbb{R}$ such that for all $x \in \mathbb{R}$ it holds that $0 \le \sigma(x) \le 1$ and

$$\sigma(x) = \begin{cases} 0, & \text{if } x \le 0 \\ 1, & \text{if } x \ge 1. \end{cases}$$

Lemma IV.8:

Let $r \in \mathbb{N}$. Then there exists a $\sigma_r \in \mathtt{SNN}^{\varrho_r,\,1,\,1}_{2(r+1),\,2,\,r+1}$ which satisfies (Pσ).

Proof

The goal is to construct a function that satisfies the claim. First, define

$$g_n(x) := \int_0^x \beta_+^{(n)}(t)\,dt.$$

Per Proposition IV.2 $\beta_+^{(n)}$ is non-negative and zero except for $x \in [0, n+1]$, where $n \in \mathbb{N}_0$. Thus g_n is non-decreasing. Since the integrants are non-negative, Tonelli's theorem implies that $\|\beta_+^{(n)}\|_{L^1(\mathbb{R},\mathbb{R})} = 1$ for all $n \in \mathbb{N}_0$, since $\|\beta_+^{(0)}\|_{L^1(\mathbb{R},\mathbb{R})} = 1$. Choosing $x \ge n+1$, the integral $g_n(x)$ include the support of $\beta_+^{(n)}$. Thus $g_n(x) = \|\beta_+^{(n)}\|_{L^1(\mathbb{R},\mathbb{R})} = 1$ for $x \ge n+1$, culminating in

$$g_n(x) = \begin{cases} 0, & \text{if } x \le 0, \\ 1, & \text{if } x \ge n+1. \end{cases}$$

From Corollary IV.5 $\beta_+^{(n)} \in \mathcal{C}_c^{n-1}(\mathbb{R},\mathbb{R})$ for $n \geq 1$. Hence $g_n \in \mathcal{C}^n(\mathbb{R},\mathbb{R})$ for $n \neq 1$. Moreover, $g_0 \in \mathcal{C}^0(\mathbb{R},\mathbb{R})$, since $\beta_+^{(n)}$ is bounded.

Now define

$$\sigma_r(x) \coloneqq g_{r-1}(rx).$$

Using the properties for g_{r-1} it holds that $\sigma_r \in \mathcal{C}^{r-1}(\mathbb{R},\mathbb{R})$. To verify that σ_r satisfies (Pσ) consider the cases: $x \leq 0$ and $x \geq 1$. For $x \leq 0$, it holds that $\sigma_r = 0$. For $x \geq 1$, it holds that $rx \geq r = (r-1)+1$. Therefore

$$\sigma_r(x) = g_{r-1}((r-1)+1) = 1.$$

Next, it is desired to show that $\sigma_r \in \mathtt{SNN}^{\varrho_r,1,1}_{2(r+1),2,r+1}$. To do so the goal is to construct a Φ where $\sigma_r = \mathtt{R}(\Phi)$. Therefore let $0 \leq k \leq n+1$. Then it holds that

$$\int_0^x \varrho_n(t-k)\,dt = 0$$

for $x \leq k$, and

$$\begin{aligned}\int_0^x \varrho_n(t-k)\,dt &= \int_0^x (t-k)^n\,dt \\ &= \int_0^{x-k} t^n\,dt \\ &= \frac{(x-k)^{n+1}}{n+1}\end{aligned}$$

for $x > k$. Overall

$$\int_0^x \varrho_n(t-k)\,dt = \frac{\varrho_{n+1}(x-k)^{n+1}}{n+1}.$$

Using the decomposition of $\beta_+^{(n)}$ in Theorem IV.3, g_n can be rewritten as

$$g_n(x) = \frac{1}{(n+1)!}\sum_{k=0}^{n+1}\binom{n+1}{k}(-1)^k\varrho_{n+1}(x-k).$$

Thus σ_r can be rewritten as

$$\sigma_r = \frac{1}{r!}\sum_{k=0}^{r}\binom{r}{k}(-1)^k\varrho_r(rx-k). \tag{IV.1}$$

By defining $\alpha_1 : \mathbb{R}^{r+1} \to \mathbb{R}^{r+1}$ given by $\alpha_1 \coloneqq \varrho_r \otimes \cdots \otimes \varrho_r$, and

$$\begin{aligned} &T_1 : \mathbb{R} \to \mathbb{R}^{r+1}, && T_1(x) \coloneqq (rx-k)_{k=0}^r \\ &T_2 : \mathbb{R}^{r+1} \to \mathbb{R}, && T_2(y) \coloneqq \frac{1}{r!}\sum_{k=0}^{r}\binom{r}{k}\end{aligned}$$

it holds that $\sigma_r = T_2 \circ \alpha_1 \circ T_1$. Note that $\|T_i\|_{\ell^0} = r+1$ for $i \in \{1,2\}$. Thus by defining the neural network $\Phi \coloneqq ((T_1,\alpha_1),(T_2,\mathrm{id}_{\mathbb{R}}))$ it holds that

$$L(\Phi) = 2, \quad N(\Phi) = r+1, \quad W(\Phi) = 2(r+1).$$

Overall $\Phi \in \mathcal{SNN}^{\varrho_r,1,1}_{2(r+1),2,r+1}$, which completes the proof. ■

Lemma IV.8 constructs a function that satisfies (Pσ). The next lemma approximates indicator functions using functions that satisfy (Pσ).

Lemma IV.9:

Let $W, N, L \in \mathbb{N}$, and assume $\sigma \in \mathtt{NN}^{\varrho,1,1}_{W,L,N}(\Omega)$ satisfies (Pσ). Then the following holds for $d \in \mathbb{N}$:

(i) For $0 \in (0, 1/2)$, there exists a function $h : \mathbb{R}^d \to \mathbb{R}$ such that

$$h \in \mathtt{NN}^{\varrho,d,1}_{2dW(N+1),\,2L-1,\,(2d+1)N}$$

where $0 \leq h(x) \leq 1$ for $x \in \mathbb{R}^d$, $\operatorname{supp}(h) \subseteq [0, 1]^d$, and

$$\left|h(x) - \chi_{[0,1]^d}(x)\right| \leq \chi_{[0,1]^d/[\varepsilon,1-\varepsilon]^d}(x) \quad \forall x \in \mathbb{R}^d. \tag{IV.2}$$

(ii) There exists a $\tilde{L} \leq 2L - 1$ such that for all hyper-rectangle

$$[a, b] := \prod_{n=1}^{d} [a_n, b_n], \quad d \in \mathbb{N},\, a_n, b_n \in \mathbb{R},$$

each $p \in (0, \infty)$, and each $\varepsilon \in (0, 1/2)$, there exists a compactly supported, non-negative function $g : \mathbb{R}^d \to \mathbb{R}$ such that $0 \leq g(x) \leq 1$ for $x \in \mathbb{R}^d$, $\operatorname{supp}(g) \subseteq [a, b]$,

$$\left\|g - \chi_{[a,b]}\right\|_{L^p(\mathbb{R}^d,\mathbb{R})} < \varepsilon, \tag{IV.3}$$

and where $g = \mathtt{R}(\Phi)$ for some $\Phi \in \mathcal{NN}^{\varrho,d,1}_{2dW(N+1),\,\tilde{L},\,(2d+1)N}$, where $L(\Phi) = \tilde{L}$.

If $d = 1$ (i) holds for $h \in \mathtt{NN}^{\varrho,1,1}_{2W,L,2N}$, and (ii) holds for $\Phi \in \mathcal{NN}^{\varrho,1,1}_{2W,\tilde{L},2N}$ with $L(\Phi) = \tilde{L}$, where $\tilde{L} \leq L$.

Proof

(i) The goal is to find a function h satisfying the claim in (i). At first, (IV.2) is proven for $d = 1$. Define $h_1 : \mathbb{R} \to \mathbb{R}$ given by

$$h_1(x) := \sigma\left(\frac{x}{\varepsilon}\right) - \sigma\left(1 + \frac{x-1}{\varepsilon}\right), \tag{IV.4}$$

where $\varepsilon \in (0, 1/2)$. For $x \leq 0$, both terms in h_1 are zero. For $x \geq 1$, both terms are one, so h_1 is zero. For $x \in [0, \varepsilon]$, the first term is between zero and one, and the last term is zero. For $x \in [1-\varepsilon, 1]$ the first term is one and the last is between zero and one, so subtracted from each other gives a term between zero and one. Lastly, for $x \in [\varepsilon, 1-\varepsilon]$, the first term is one and the last is zero. These considerations imply that

$$h_1(x) = \begin{cases} 0, & \text{if } x \in \mathbb{R} \setminus [0, 1] \\ 1 & \text{if } x \in [\varepsilon, 1-\varepsilon] \end{cases} \quad \wedge \quad 0 \leq h_1(x) \leq 1 \;\forall x \in \mathbb{R}.$$

All in all, $\operatorname{supp}(h_1) \subseteq [0, 1]$ and

$$\left|h_1(x) - \chi_{[0,1]}(x)\right| \leq \chi_{[0,1]/[\varepsilon,1-\varepsilon]}(x) \quad \forall x \in \mathbb{R}.$$

By defining $h := h_1$, this proves (IV.2) for $d = 1$.

Next, the goal is to prove (IV.2) for $d \in \mathbb{N}$. Define $h_2 : \mathbb{R}^d \to \mathbb{R}$, given by

$$h_2(x) := \sigma\left(1 - d + \sum_{i=1}^{d} h_1(x_i)\right), \tag{IV.5}$$

for $x := (x_1, \dots, x_d)$. From the properties of σ it hold that $0 \leq h_2(x) \leq 1$, for $x \in \mathbb{R}^d$. For $x \in [\varepsilon, 1-\varepsilon]^d$, it holds that $h_1(x_i) = 1$ for all $i \in \{1, \dots d\}$. For $x \notin [0,1]^d$, there at least exists one $i \in \{1, \dots, d\}$ such that $h_1(x_i) = 0$. Consequently

$$\sum_{i=1}^{d} h_1(x_i) \leq d - 1,$$

since $0 \leq h_1(x_i) \leq 1$ for all $i \in \{1, \dots, d\}$. Therefore $h_2(x) = 0$ for $x \notin [0,1]^d$, which proves $\operatorname{supp}(h_2) \subseteq [0,1]^d$. Moreover

$$\left| h_2(x) - \chi_{[0,1]^d}(x) \right| \leq \chi_{[0,1]^d / [\varepsilon, 1-\varepsilon]^d}(x) \quad \forall x \in \mathbb{R}^d.$$

Thus by defining, $h := h_2$ (IV.2) is satisfied.

It remains to show that h is in the right function space. To verify this (ii) is proven in the special case where $[a,b] = [0,1]^d$. Let λ^d denote the d-dimensional Lebesgue measure. Using h as constructed above,

$$\left\| h - \chi_{[0,1]^d} \right\|^p_{L^p(\mathbb{R}^d, \mathbb{R})} \leq \lambda^d\left([0,1]^d \setminus [\varepsilon, 1-\varepsilon]^d\right) = 1 - (1 - 2\varepsilon)^d. \tag{IV.6}$$

The right-hand side tends to zero, as $\varepsilon \to 0$, which proves (IV.3) for $[a,b] = [0,1]^d$. Per assumption on σ, there exists an $L_\sigma \leq L$ such that $\sigma = \mathtt{R}(\Phi_\sigma)$ for some $\Phi_\sigma \in \mathcal{NN}^{\varrho, 1, 1}_{W, L_\sigma, N}$, with $L(\Phi_\sigma) = L_\sigma$. For $i \in \{1, \dots, d\}$, define $f_{i,1}, f_{i,2} : \mathbb{R}^d \to \mathbb{R}$ given by

$$f_{i,1}(x) := \sigma\left(\frac{x_i}{\varepsilon}\right), \qquad f_{i,2}(x) := -\sigma\left(1 + \frac{x_i - 1}{\varepsilon}\right).$$

Using Lemma II.17(i), with

$$\begin{aligned} &\Phi = \Phi_\sigma, \quad P : x \mapsto \frac{x}{\varepsilon}, && Q = I, \\ &\Phi = \Phi_\sigma, \quad P : x \mapsto 1 + \frac{x-1}{\varepsilon}, && Q = -I, \end{aligned}$$

for $f_{i,1}$ and $f_{i,2}$ respectively, there exist $\Psi_{i,1}, \Psi_{i,2} \in \mathcal{NN}^{\varrho, d, 1}_{W, L_\sigma, N}$ with

$$L(\Psi_{i,1}) = L(\Psi_{i,2}) = L_\sigma$$

for any $i \in \{1, \dots, d\}$ such that $f_{i,1} = \mathtt{R}(\Psi_{i,1})$, and $f_{i,2} = \mathtt{R}(\Psi_{i,2})$. Now define the function $F : \mathbb{R}^d \to \mathbb{R}$ given by

$$F(x) := \sum_{i=1}^{d} h_1(x_i) = \sum_{i=1}^{d} f_{i,1}(x) + \sum_{i=1}^{d} f_{i,2}(x).$$

Then Lemma II.15(ii) with $\{\Phi_m\}_{m=1}^{2d} = \{\Psi_{m,1}, \Psi_{m,2}\}_{m=1}^{d}$, implies that there exists a $\Phi_F \in \mathcal{NN}^{\varrho, d, 1}_{2dW, L_\sigma, 2dN}$ with $L(\Phi_F) = L_\sigma$ such that $F = \mathtt{R}(\Phi_F)$. Note that compared to h for $d = 1$ it holds that $h = h_1 = F$ and $L(\Phi_F) = L_\sigma \le L$. Thus there exists a $h = \mathtt{R}(\Phi_F)$ where $\Phi_F \in \mathcal{NN}^{\varrho,1,1}_{2W,L,2N}$. This proves (i) for $d = 1$.

For $d \in \mathbb{N}$, Lemma II.17(i) with $\Phi = \Phi_F$, $P = I$ and $Q = I + 1 - d$, gives that there exists a $\Phi_G \in \mathcal{NN}^{\varrho,d,1}_{2dW,L_\sigma,2dN}$ with $L(\Phi_F) = L_\sigma$ such that $\mathtt{R}(\Phi_G) = G$, where $G : \mathbb{R}^d \to \mathbb{R}$ is given by

$$G(x) \coloneqq 1 - d + \sum_{i=1}^{d} h_1(x_i).$$

Note that per construction of h for general d, it holds that $h = h_2 = \sigma \circ G$. Using Ψ_2 in Lemma II.17(ii), with $\Phi_1 = \Phi_\sigma$ and $\Phi_2 = \Phi_G$ it holds that there exists a

$$\Phi_h \in \mathcal{NN}^{\varrho,d,1}_{W_2,\,2L_\sigma-1,\,(2d+1)N}$$

where $L_\sigma \le L$ and

$$W_2 \coloneqq 2dW + \max\{2dN, d\}W \le 2dW(N+1).$$

This concludes (i) for general d. By defining $\tilde{L} \coloneqq 2L_\sigma - 1$ for general $d \in \mathbb{N}$ and $\tilde{L} \coloneqq L_\sigma$ for $d = 1$ the special case for (ii) is proven.

(ii) By using the special case of (ii), proven in part (i), it remains to expand the result from $[0,1]^d$ to general domains $[a,b]$, where $a, b \in \mathbb{R}^d$. To do so, define the affine-linear map $T : \mathbb{R}^d \to \mathbb{R}^d$, given by

$$T(x) \coloneqq \left(\frac{x_i - a_i}{b_i - a_i}\right)_{i=1.}^{d}$$

Note, that T is invertible since it is bijective, and the inverse T^{-1} is given by

$$T^{-1}(y) = ((b_i - a_i)y_i + a_i)_{i=1.}^{d}$$

Additionally

$$\chi_{[0,1]^d} \circ T = \chi_{T^{-1}([0,1]^d)} = \chi_{[a,b]},$$

and compared to the general definition of an affine-linear map, the weight matrix A is the diagonal matrix with entries $(b_i - a_i)^{-1}$, ensuring that $\|T\|_{\ell^{0,\infty}_*} = 1$. It is desired to use h constructed in part (i). Note that h is defined by h_2 in (IV.5) for general $d \in \mathbb{N}$, and h_1 in (IV.4) for the special case $d = 1$. For general $d \in \mathbb{N}$ Lemma II.17(i) with $P = T$ and $Q = I$ implies that there exists a

$$\Phi \in \mathcal{NN}^{\varrho,d,1}_{2dW(N+1),\,\tilde{L},\,(2d+1)N},$$

such that $h \circ T = \mathtt{R}(\Phi)$ and $\tilde{L} = L(\Phi) = 2L_\sigma - 1$. If $d = 1$, then Lemma II.17(i) gives that there exists a $\Phi \in \mathcal{NN}^{\varrho,1,1}_{2W,\tilde{L},2N}$, such that $h \circ T = \mathtt{R}(\Phi)$ and $\tilde{L} = L(\Phi) = 2L_\sigma - 1$. By defining $g \coloneqq h \circ T$, (ii) is proven except for (IV.3). Using the change of variables formula, it holds that

$$\begin{aligned}
\left\|g - \chi_{[a,b]}\right\|_{L^p(\mathbb{R}^d,\mathbb{R})} &= \left\|h \circ T - \chi_{[0,1]^d} \circ T\right\|_{L^p(\mathbb{R}^d,\mathbb{R})} \\
&= \left|\det \operatorname{diag}\left((b_i - a_i)^{-1}\right)_{i\in\{1,\dots,d\}}\right|^{-1/p} \left\|g - \chi_{[0,1]^d}\right\|_{L^p(\mathbb{R}^d,\mathbb{R})} \\
&= \left\|g - \chi_{[0,1]^d}\right\|_{L^p(\mathbb{R}^d,\mathbb{R})} \prod_{i=1}^{d} (b_i - a_i)^{1/p}.
\end{aligned}$$

Per (IV.6), the first factor can be made arbitrarily small by choosing $\varepsilon \in (0, 1/2)$ suitably. As the second factor is constant, this proves (IV.3) for $[a,b]$. ∎

With these lemmas in place, multivariate B-splines of degree 0 are approximation by ReLU-networks:

Theorem IV.10: Approximation of B-Splines of degree 0

Let $d, r \in \mathbb{N}$. Then $\beta_d^{(0)} \in \overline{\mathtt{NN}_{W,\,L,\,N}^{\varrho_r,\,d,\,1}}$ where the closure is taken with respect to $\|\cdot\|_{L^p(\mathbb{R}^d,\mathbb{R})}$ and where

$$L := \begin{cases} 3, & \text{if } d > 1, \\ 2, & \text{if } d = 1, \end{cases}$$

$$W := \begin{cases} 4d(r+1)(r+2), & \text{if } d > 1, \\ 4(r+1), & \text{if } d = 1, \end{cases}$$

$$N := \begin{cases} (2d+1)(r+1), & \text{if } d > 1, \\ 2(r+1), & \text{if } d = 1. \end{cases}$$

Proof

Using Lemma IV.8 for each ϱ_r there exists a $\sigma_r \in \mathtt{SNN}_{2(r+1),\,2,\,r+1}^{\varrho_r,\,1,\,1}$ for $r \in \mathbb{N}$, which satisfies (Pσ). Per Lemma IV.9(ii) with $\sigma = \sigma_r$, there exists a

$$L = \begin{cases} 3, & \text{if } d > 1, \\ 2, & \text{if } d = 1. \end{cases},$$

such that for $\varepsilon > 0$ there exists a function $g_\varepsilon = \mathtt{R}(\Phi_\varepsilon)$ with

$$\left\| g_\varepsilon - \chi_{[0,\,1]^d} \right\|_{L^p(\mathbb{R}^d,\mathbb{R})} < \varepsilon,$$

where $L(\Phi_\varepsilon) \le L$ and $\Phi_\varepsilon \in \mathtt{NN}_{W,\,L,\,N}^{\varrho_r,\,d,\,1}$ with

$$W = \begin{cases} 4d(r+1)(r+2), & \text{if } d > 1, \\ 4(r+1), & \text{if } d = 1, \end{cases}$$

$$N = \begin{cases} (2d+1)(r+1), & \text{if } d > 1, \\ 2(r+1), & \text{if } d = 1. \end{cases}$$

Since $\beta_d^{(0)} = \chi_{[0,1]^d}$, it holds that g_ε can approximate $\beta_d^{(0)}$. ■

Comparing $n \in \mathbb{N}$ and $n = 0$ it should be noted that Theorem IV.7, provides a better estimate than Theorem IV.10. In fact, Theorem IV.7 gives a realization, where Theorem IV.10 only is an approximation.

V | Experiments

In application, algorithms based on B-splines are commonly used to approximate functions. According to Theorem IV.7 and Theorem IV.10 neural networks can approximate multivariate B-splines arbitrarily well under specific assumptions. Thus intuitively there should exist equivalent algorithms based on neural networks to approximate functions. However, this assumption is purely based on a theoretical point of view, and the connection between theory and application does not always go hand in hand. Thus the goal of this section is to design an experiment that demonstrates the results in practice. As an initial experiment, it is also desired to demonstrate that the neural network, can realize univariate B-splines, as given in Theorem IV.3. The code, data, and plots constructed for these experiments can be found at *https://github.com/Frizlerrr/SpecialeKode*. Additionally, an overview of Python files used for each step can be found in Appendix E.

V.1 Initial Experiment: Realization of B-Splines

The goal of the initial experiment is to implement the decomposition from Theorem IV.3 in practice. This experiment aims to evaluate the realization of univariate B-splines using neural networks. The initial experiment is structured as follows:

Initial Experiment:

(i) Generating B-splines:
- Generating univariate B-splines of different degrees.

(ii) Construction of Neural Networks:
- Constructing the neural network that can realize B-splines.

(iii) Comparison:
- Comparing B-splines to neural networks before and after training.

This section explains the setup for the initial experiment and presents the findings.

(i) Generating B-splines:

The attempted methods for generating B-splines are the following, mentioned in the order in which they were attempted:

- ***Irwin-Hall distribution***: The decomposition in Theorem IV.3.
- ***Recursive***: The naive implementation of the recursive formula for B-splines from [7] given by $\beta_+^{(0)} = \chi_{[0,1)}$ and

$$\beta_+^{(n)}(x) = \frac{x}{n}\beta_+^{(n-1)}(x) + \frac{(n+1)-x}{n}\beta_+^{(n-1)}(x-1), \quad n \in \mathbb{N}. \tag{V.1}$$

- ***De Boor's algorithm***: Algorithm equivalent with ***Recursive***, [7], using the implementation `scipy.interpolate.BSpline.basis_element` from the Python package `SciPy`.

As the decomposition in Theorem IV.3 allows neural networks to realize univariate B-splines, generating the B-splines using ***Irwin-Hall distribution*** is a natural starting point. Plotting these shows that for degrees higher than 25 the generated B-splines oscillate near the support edges. These problems are illustrated in Figure V.1, where the $\beta_+^{(27)}$ and $\beta_+^{(30)}$ are generated using ***Irwin-Hall distribution*** and plotted in grey. A reason for the oscillations could be due to numerous errors such as rounding errors.

Considering the generation of B-splines using ***Recursive***, Figure V.1 illustrates that the generated B-splines using ***Recursive*** do not suffer from oscillation. However, the computational time for $\beta_+^{(30)}$ is 14 hours when using ***Recursive***.

The problems with oscillation and computation time can be circumvented by using ***de Boor's algorithm***. This algorithm is equivalent to ***Recursive*** but with significantly better computation time. By plotting the B-splines using ***de Boor's algorithm*** for degrees up to 1000 there is no visual indication that any oscillation occurs. This is illustrated for $\beta_+^{(27)}$ and $\beta_+^{(30)}$ in Figure V.1 in doted black, where it is clear that ***de Boor's algorithm*** is as exact as ***Recursive***. The B-splines used in the initial experiment are therefore generated using ***de Boor's algorithm***.

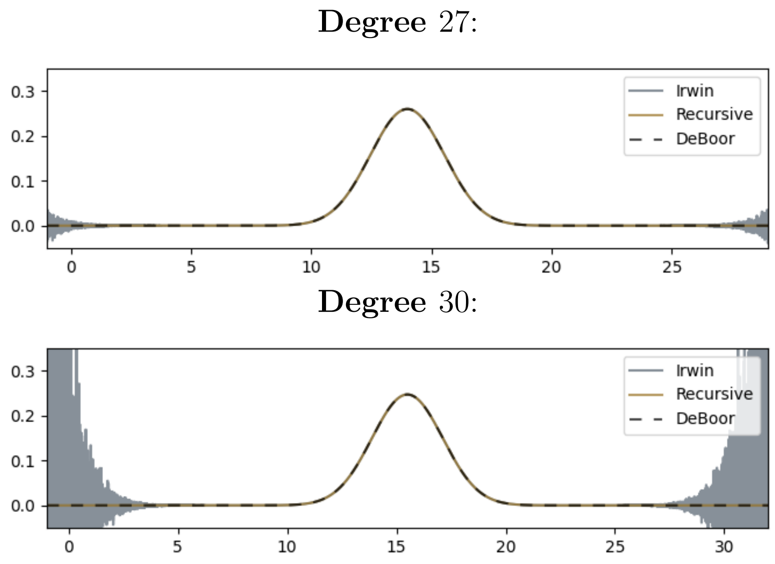

Figure V.1: *The B-splines $\beta_+^{(27)}$ and $\beta_+^{(30)}$ generated using **Irwin–Hall distribution**, **Recursive**, and **de Boor's algorithm** denoted by Irwin, Recursive, and DeBoor, respectively.*

(ii) Construction of Neural Networks:

The neural networks constructed to realize the B-splines are based on Theorem IV.3. Recall from the proof of Theorem IV.3 that the decomposition can be described using the tuple $\Phi_n = ((T_1, \alpha_1), (T_2, \mathrm{id}_{\mathrm{id}_\mathbb{R}}))$, where $\alpha_1 = \varrho_n \otimes \cdots \otimes \varrho_n$,

$$T_1(x) = (x-k)_{k=0}^{n+1}, \quad T_2(y) = \frac{1}{n!}\sum_{k=0}^{n+1}\binom{n+1}{k}(-1)^k y_k. \tag{V.2}$$

Thus the structure of the neural networks and the weights used for initialization are based on Φ_n. It should be clear that the realization of Φ_n is exactly the decomposition in Theorem IV.3. To implement Φ_n, the Python package `Keras` with `PyTorch` as the backend is used.

(iii) Comparison:

To restrict the analysis, the initial experiment is considered for degrees in $\{0, \ldots, 15\}$. The following are considered for each degree:

For every degree in $\{0, \ldots, 15\}$:
- Mean square error and maximum error over the support.
- Plot comparing the trained and untrained neural network to the B-spline.
- Plot of the difference between the B-spline and both neural networks.

Only a few specific plots are selected as illustrations in the report and can be found in Figure V.2 to Figure V.5.

Mean Square Error & Maximums Error

The mean square error and maximum error values can be found in Figure E.1 and are plotted in Figure V.2 with a logarithmic y-axis.

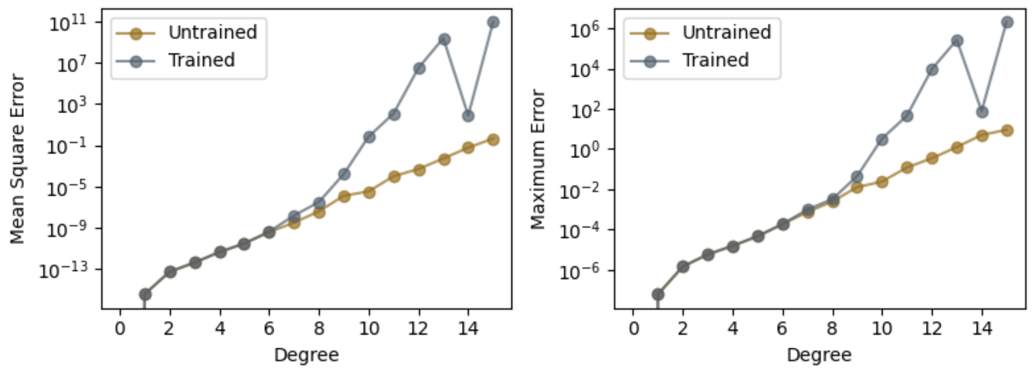

Figure V.2: *Plots of the mean square error and maximum error when approximating B-splines using untrained and trained neural networks that can realize the given B-spline. The error is only considered within the support of the B-spline.*

For degrees less than 7, it is observed that the neural networks approximate the B-splines within an acceptable boundary, with a mean square error less than $4 \cdot 10^{-10}$ and a maximum error less than $2 \cdot 10^{-4}$ for both the trained and untrained neural network. Afterward, the error increases exponentially. As seen in Figure V.2 the error of the trained and untrained neural network follows the same exponential growth until degree 7 for the mean square error, and degree 9 for the maximum error. From here the errors of the trained neural network escalate rapidly compared to the untrained neural network. As an exception, the error of the trained neural network decreases significantly at degree 14 compared to degrees 13 and 15.

Behavior of Trained & Untrained Neural Networks

When plotting the B-splines together with the trained and untrained neural networks for different degrees, it becomes clear that both neural networks have problems with oscillation. Additionally, the trained neural network has an increasing tendency to deviate from the desired behavior as the degree increases. Both principles are illustrated in Figure V.3. The oscillations are not a surprise, since the neural network is based on the decomposition in Theorem IV.3 which has the same problems, as seen in Figure V.1.

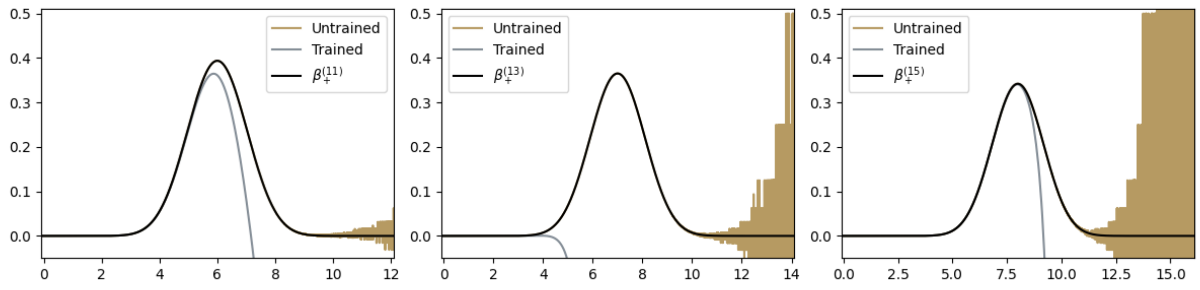

Figure V.3: *Plots of approximations of $\beta_+^{(11)}$, $\beta_+^{(13)}$, and $\beta_+^{(15)}$, using the untrained and trained neural network that can realize the given B-spline.*

The oscillation in the neural network, and thus the error, can arise from various computational challenges. Computational limitations like overflow, underflow, and rounding errors mostly cause these challenges. The scenarios where these challenges typically occur are a consequence of the bit-precision used for the computations. It should be noted that by default `SciPy` uses 64-bit and by default `Keras` uses 32-bit. The implications of this difference can be seen in Figure V.4.

Floating Point Numbers	**Float32**	**Float64**
Machine Epsilon	1.192e-07	2.220e-16
Real Min	1.175e-38	2.225e-308
Real Max	3.403e+38	1.798e+308

Figure V.4: *Overview of the relevant values for the data types `Float32` and `Float64`.*

This difference in bit-precision is likely why the B-splines realized by the neural networks begin to oscillate much sooner than the B-splines generated using the ***Irwin-Hall distribution*** even though it is two implementations of the same formula. This difference can be seen when comparing Figure V.1 and Figure V.3.

The computational limitations of finite precision arithmetic can lead to the following problems, [14].

- **Overflow**: Similar to underflow, but at exceptionally large values.
- **Rounding error**: As a consequence of finite precision arithmetic, rounding errors occur, and when handling iterative algorithms these errors typically accumulate.

The formula (V.2) includes binominal coefficients which can generate large values where rounding error is more likely to occur. Summing over these values with likely rounding errors accumulates the error which typically amplifies the oscillatory behavior, [3]. For the same reason, ϱ_n can also cause problems, when the input and n are too large. These are the likely causes of numerical instability seen in this experiment. Moreover, the formulas include dividing with $n!$, which can result in underflow for sufficiently large n. Overall, these factors contribute to why the oscillation in the neural network can happen and are all common errors in numerical approximation, [3].

Comparing the trained and untrained neural networks, the results for both the error and the individual plots indicate that the untrained ones perform better than the trained ones, as seen in Figure V.3. This might be attributed to the choices made regarding the training process. This includes the choice of learning rate, as this controls the step size in gradient descent during training. Another thing to notice is that the error creeps toward the center of the support as the degree increases. This means that oscillation affects the crucial parts of the B-splines for degrees above a certain threshold.

Validity of Neural Networks for Realizing B-splines

Based on the plots of the B-splines a visual inspection indicates that there is an interval in the support where the B-splines are close to zero. One could argue that a small error is still okay, as long as it is within such intervals or outside the support. Forcing the neural network to be zero within these intervals, can hopefully lower the error. Notice that this approach technically guarantees an error, since the B-splines are not equal to zero in this interval. However, this error is more predictable and could still be a decrease in the overall error, compared to the oscillations. One approach to force the neural network to be zero is by taking the product with an indicator function, which could be implemented as part of the neural network by Lemma III.9 and Lemma IV.9.

In the case where the error affects the crucial parts of the B-splines, it no longer makes sense to use the neural network. By considering the plots it is clear that the error becomes a problem between degrees 6 and 9. A zoomed view of the crucial part of the B-splines is highlighted in Figure V.5.

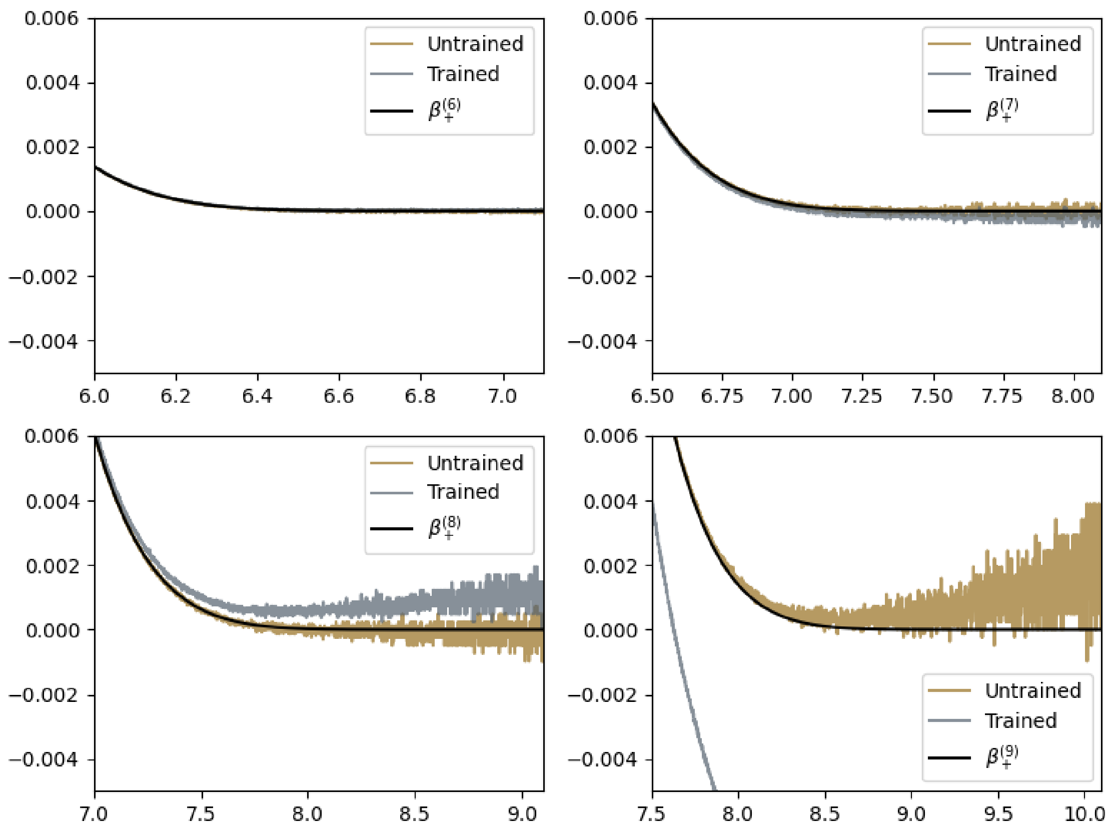

Figure V.5: *A zoomed view of approximations of $\beta_+^{(n)}$ with $n \in \{6,7,8,9\}$, using the untrained and trained neural network that can realize the given B-spline.*

Where exactly to put the threshold could depend on the context. In this report, the threshold for when the error becomes problematic is chosen to be at degree 7. Based on these considerations the neural networks in the main experiment will be limited to a maximum degree of 6.

V.2 Setup of the Main Experiment

The focus of this section is to explain the setup for the main experiment and the reasoning for the choices made. The main experiment consists of three approximations using three different methods. The first method is a traditional approximation using B-splines. This method's primary role is as a baseline for evaluating performance. The second and third methods use neural networks, where one has a structure equivalent to the B-spline approximation, and the other follows the same overall structure but with no restrictions to the connections. The purpose of the main experiment is to evaluate the performance of algorithms based on neural networks and compare them to the ones using B-splines. Some specific target function is generated since it is not feasible to cover all types of functions. This includes polynomials and continuous piecewise polynomials. Overall, the main experiment includes generating the target functions, approximating these functions using the three approximation methods, and comparing the results. The main experiment is outlined as follows:

Main Experiment:

(i) Generation of Target Functions:
- Polynomials.
- Continuous piecewise polynomials.

(ii) Approximation Methods:
- ***Spline***: B-spline approximation.
- ***Equiv***: Algorithm equivalent to ***Spline*** using neural networks.
- ***Fully***: Fully connected neural network with same structure as ***Equiv***.

(iii) Evaluation:
- Visual inspection of the approximation methods. Include plots of the target functions and the approximants.
- Average of the mean square error and maximum error.

To simplify the notation the approximation methods are denoted by ***Spline***, ***Equiv***, and ***Fully***, respectively.

A natural starting point is to implement and run the main experiment for univariate B-splines.

(i) Generation of Target Functions:

The first step in the main experiment is generating the functions to be approximated. The first class of functions that are considered are polynomials. This class is chosen since they are flexible and fit various data types. However, since data is very different depending on the context of applications, more types of functions are needed. Continuous piecewise polynomials are very flexible, offering a good balance between smoothness and adaptability. Real-world data often have different behavior over different intervals, [3]. To represent such data continuous piecewise polynomials are a good match since they can adjust to local variations by changing the polynomial in each interval.

Restrictions on the Target Functions

To establish a setup that is easier to compare and evaluate, the target functions f generated in the main experiment are constructed such that $(x, f(x)) \in [-1,1]^2$. This domain $[-1,1]^2$ is chosen because it includes both positive and negative values and because it is easy to control. The primary reason for this is to ensure that the randomly generated function does not get too big or deviate excessively. The restriction makes it easier to compare different solutions and ensures some control over the functions. The method for enforcing this restriction on the generated target functions depends on the function type. For polynomials normalizing the coefficients with respect to $\|\cdot\|_{L^1(\mathbb{R})}$ is one possible method. For the continuous piecewise polynomials the restriction can be achieved by shifting and scaling.

Within the interval, 500 random points are generated, which is the data used for the approximation. For the evaluation, 1001 equidistant points are generated where the error of the approximant is evaluated. The reason for distinguishing between the two cases is based on that applications typically have limited data available. Additionally, it is good practice not to evaluate the performance of approximations on the data it was fitted on, [3].

The polynomials are restricted to a maximum degree of 5, and analogously for the individual polynomials in the continuous piecewise polynomials. Moreover, the intervals in the continuous piecewise polynomials are restricted to a minimum size of $1/(2k)$, where $k \in \{2, \dots, 5\}$ is the number of intervals.

(ii) Approximation methods:

The approximation method used in ***Spline*** is well-known and the Python package `SciPy` has a ready-to-use implementation in terms of `scipy.interpolate.splrep`. Note according to [18] it only works for degrees in $\{1, \dots, 5\}$. ***Equiv*** and ***Fully*** uses the Python package `Keras` with `PyTorch` as the backend to implement neural networks. In contrast to the definition in the report, these packages use the commonly used definition. The difference between the definition used in this report and the one more commonly used is described in Remark A.1. Since ***Fully*** is a fully connected version of ***Equiv***, most considerations made while developing the main experiment are regarding ***Equiv***. Additionally, ***Spline*** is used as a guidance to construct ***Equiv***.

Considerations from B-spline Approximation

Spline uses B-splines translated and scaled based on the knots. Given knots $t_0, \dots, t_{k-1}$, where $k \in \mathbb{N}$, the B-splines are given by $\beta_j^{(0)} = \chi_{[t_j, t_{j+1})}$ and

$$\beta_j^{(n)}(x) = \frac{x - t_j}{t_{j+n} - t_j}\beta_j^{(n-1)}(x) + \frac{t_{j+n+1} - x}{t_{j+n+1} - t_{j+1}}\beta_{j-1}^{(n-1)}(x), \quad n \in \mathbb{N}. \tag{V.3}$$

The approximant f from using ***Spline*** is given by

$$f(x) = \sum_{j=0}^{J-1} c_j \beta_j^{(n)}(x), \tag{V.4}$$

with $n \in \{1, \dots, 5\}$, $x \in \mathbb{R}$, and where the number of summands $J = k - n - 1 \in \mathbb{N}$. Note that k is the number of knots. The knots divide the interval into subintervals, where each B-spline is the primary contributor within its subinterval.

Remark V.1:

Note that the recursive formula in (V.3) differs from the one in (V.1), but is equivalent when $t_j = j$ for $j \in \{0, \dots, k\}$, where $k \in \mathbb{N}$.

For the approximant f resulting from ***Spline*** it holds that $f \in \mathcal{C}^{n-1}(\mathbb{R}, \mathbb{R})$, where n is the degree of the B-splines. This is also the case for the ReLU-n-network as $\varrho_n \in \mathcal{C}^{n-1}(\mathbb{R}, \mathbb{R})$. Therefore the degree chosen for ***Spline*** and the power chosen for ***Equiv*** is equal. This also aligns with the decomposition in Theorem IV.3.

Construction of Neural Networks for Approximation

The neural network in ***Equiv*** is defined using the same formula in (V.4), where the B-splines are realized using the decomposition in Theorem IV.3, analogous with the initial experiment. An illustration of the neural network is highlighted in Figure V.6, where the addition of the grey connection yields the structure of ***Fully***. The second and fourth layer of the neural network has neurons equal to the number of summands, and the third layer has neurons equal to $(n + 2)J$, where $n \in \mathbb{N}_0$ is the degree.

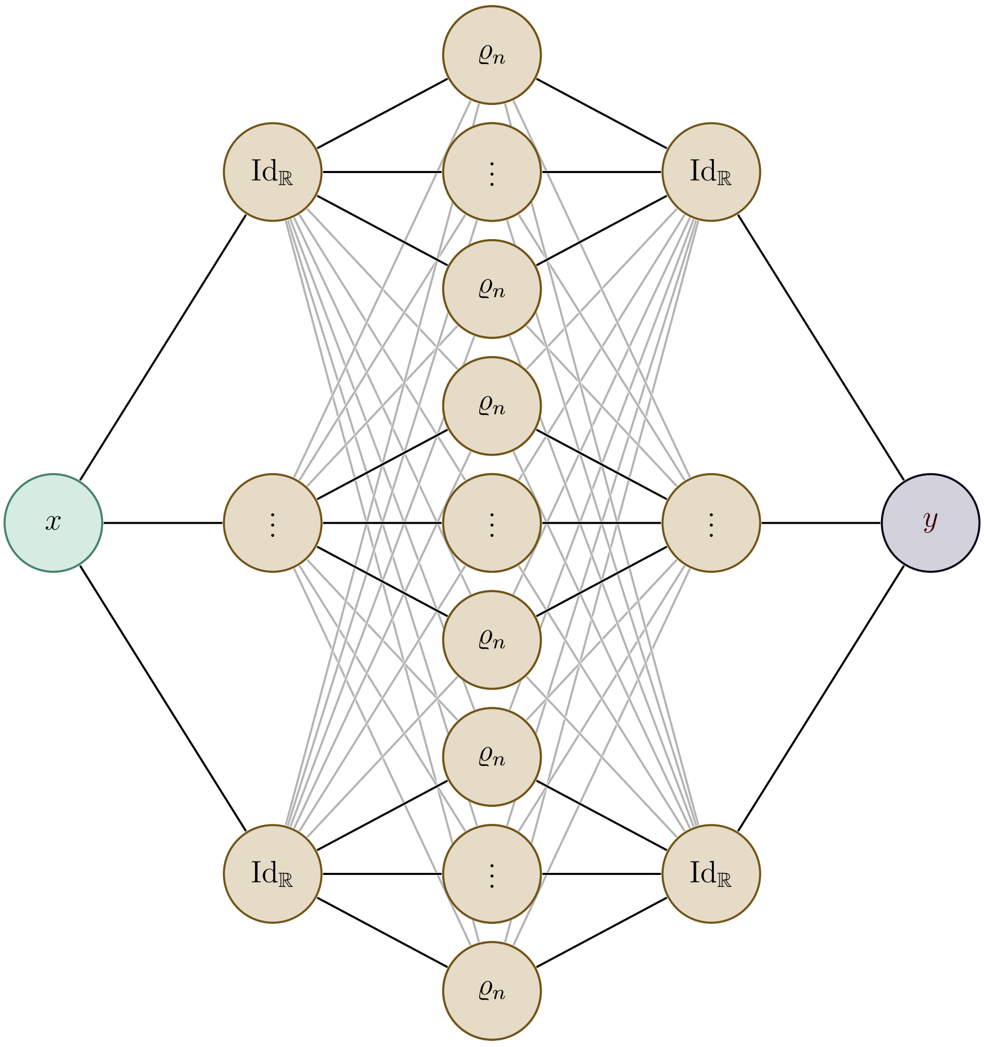

Figure V.6: *Illustration of the neural network in* ***Equiv*** *with the black connections and* ***Fully*** *with the additional grey connections, where* $n \in \mathbb{N}$.

Restrictions for Approximation Methods

Spline, ***Equiv***, and ***Fully*** are all restricted to the same number of summands to make them comparable. For ***Equiv*** and ***Fully*** this is done by restricting the structure to the one shown in Figure V.6. For ***Spline***, this is done by controlling the number of knots. Thus the same number of B-splines or equivalent structures are used in all approaches, but without necessarily being equal. The construction of ***Equiv*** does not have knots but has weights, which similarly translate the individual B-splines. These weights are randomly generated and trained afterward. Thus when choosing the location of the knots in ***Spline***, two approaches seem natural: choosing the knots randomly or distributing the knots evenly over the interval. To avoid the case where a knot would be placed at an endpoint of the interval, or very close by, knots are placed evenly over the interval.

Setting for the Approximation Methods

Given the wide array of parameters available in the approximation methods, a selection of appropriate parameters are chosen. This aims to cover a representative range of possible solutions from the approximation methods. Degrees 0 to 6 are chosen based on the initial experimental and since it contains every possible choice of degree of ***Spline***. The number of summands is chosen to be $2, 4, 6, 8$, and 10. For every one of the previous parameters, the approximation methods are considered for 100 iteration for each of the two function types. Overall, the different parameters make it possible to highlight strengths, weaknesses, and applicability to different scenarios.

As noted in the initial experiment, `SciPy` and `Keras` use different bit-precision. Just using the default precision and the chosen parameter, the main experiment takes around a week to run. It is expected that the main experiment will take even longer if it is done with 64-bit-precision. Due to time constraints, the default precision is chosen.

Evaluation

To evaluate ***Spline***, ***Equiv***, and ***Fully***, a plot of each function and the corresponding approximations are considered. This provides a visual representation of how well the approximation method captures the target function. Moreover, it allows for a direct comparison between the target function and ***Spline***, ***Equiv***, and ***Fully***, which can provide insight into the limitations of the approximation methods. Additionally, the mean square error and maximum error are evaluated. The mean square error measures the overall performance and the maximum error gives insight into the worst-case performance of an approximation. Together they offer a more complete picture of error from the approximation methods.

V.3 Results & Discussion

This section presents the findings of the main experiment, described in the previous section, and encompasses a discussion of the results and some general considerations. The individual approximation performance is examined, by visualisation and comparison. Firstly, it should be noted that the results are influenced by the approaches chosen for the main experiment. This includes choices made regarding areas such as the following:

- Types of target functions and implementation.
    - ∗ The restriction on the domain for the target functions.
    - ∗ The degree of the polynomials and the individual polynomials in the continuous piecewise polynomials.
    - ∗ The minimum size of the interval in the continuous piecewise polynomials.

- The choice of approximation methods and implementation.
    - ∗ Choice of learning rate and learning rate decay, when training the neural networks.
    - ∗ Early stopping criteria, when training the neural networks.

When comparing the number of plots with the relevant points to highlight, only a few specific plots are selected as illustrations in the report and can be found in Figure V.7 to Figure V.14. However, multiple similar figures exist, which all can be found together with the code. As noted, ***Spline*** only works for degrees 1 to 5, so for degrees 0 and 6, only ***Equiv*** and ***Fully*** are considered.

Visual Inspection of the Approximation Methods

This subsection is dedicated to describing the behavior of ***Spline***, ***Equiv***, and ***Fully*** seen when plotting the resulting approximants together with the target function. The analysis is grouped by degree.

Considerations Regarding Degree 0

First, the result for degree 0 is considered. The result from the main experiment resulted in no significant differences between the ***Equiv*** and ***Fully***. They are both similar in performance, compared to the true data. They both split the sum in the point zero no matter the number of summands given in the iteration. This principle is illustrated in Figure V.7. Moreover, roughly half of the time, there are jumps in the transition between the two subintervals, which also is illustrated in Figure V.7. However, it varies depending on whether it is ***Equiv***, ***Fully***, or both that exhibit these jumps. Overall, the performance for degree 0 is consistently poor for both ***Equiv*** and ***Fully***.

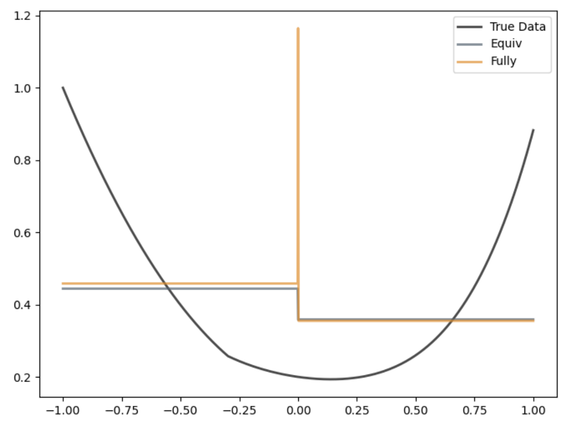

Figure V.7: *A plot from the main experiment with 4 summands, and degree 0. Example of a typical scenario for **Equiv** and **Fully** for degree 0.*

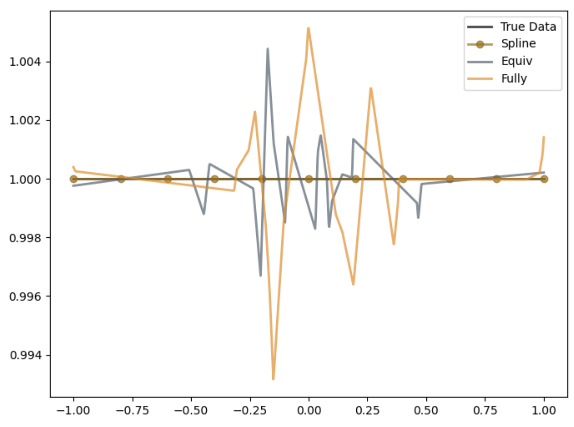

Figure V.8: *A plot from the main experiment with 10 summands, and degree 1. Example of **Equiv** and **Fully** failing to fit to a constant.*

There could be multiple reasons as to why the result always divides the interval at 0. For instance, it could be due to an error in the structure of the neural network. Another reason could be the training method used for the neural network. Most training methods use gradient descent, which does not mix well with an activation function with a constant zero derivative.

Considerations Regarding Degree 1

For degree 1, there is significant fluctuation in whether ***Equiv*** and ***Fully*** follow the true data. Sometimes they have acceptable performance, and other times they deviate strongly from the true data. An example where it deviates can be found in Figure V.8. However, they usually follow the true data and do not deviate too much. Figure V.9 is one of the better examples where this is the case. This figure also highlights a typical scenario for ***Spline*** for degree 1. ***Spline*** approximate the data significantly better than ***Equiv*** and ***Fully*** for higher summands. However, ***Spline*** and the true data do not match perfectly, and for low summands, ***Spline*** performed worse than ***Equiv*** and ***Fully***. This is highlighted in Figure V.10. Overall depending on the number of summands it varies whether ***Spline*** performs better than ***Equiv***, and ***Fully*** for degree 1.

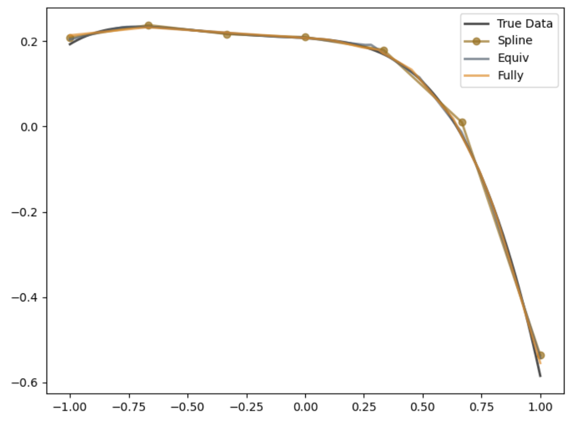

Figure V.9: *A plot from the main experiment with 6 summands, and degree 1. Example of* ***Spline***, ***Equiv***, *and* ***Fully*** *with acceptable performance for degree 1.*

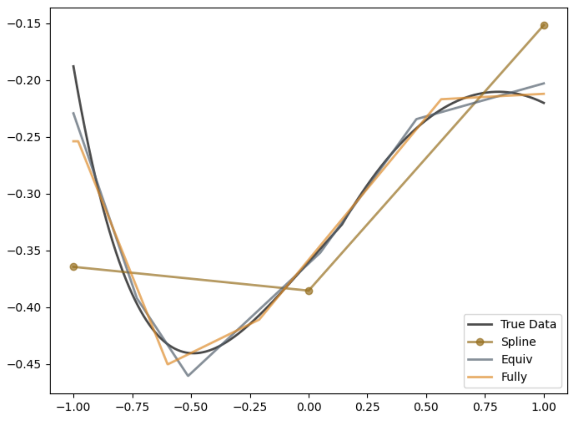

Figure V.10: *A plot from the main experiment with 2 summands, and degree 1. Example of* ***Spline*** *failing to fit the true data for 2 summands.*

Considerations Regarding Degree 2 to 5

For degrees 2 to 5, ***Spline*** performs better and better, which is illustrated in both Figure V.11 and Figure V.12. For ***Equiv*** and ***Fully***, they still vary in whether they perform okay or deviate from the data, and they still typically deviate when the true data follows a straight line, which is illustrated in Figure V.12. Given 2 summands, ***Spline*** performs better and better the higher the degree. For degree 2 and 2 summands, it varies in which of the three approximations method fits the true data best. For degrees, 3 and 4 with 2 summands, ***Spline*** is better than ***Equiv*** and ***Fully*** but does not typically fit the true data perfectly. However, for degree 5 and summands 2, ***Spline*** typically matches the true data. Overall, ***Spline*** performs much better than ***Equiv*** and ***Fully*** across the number of summands and degrees 2 to 5.

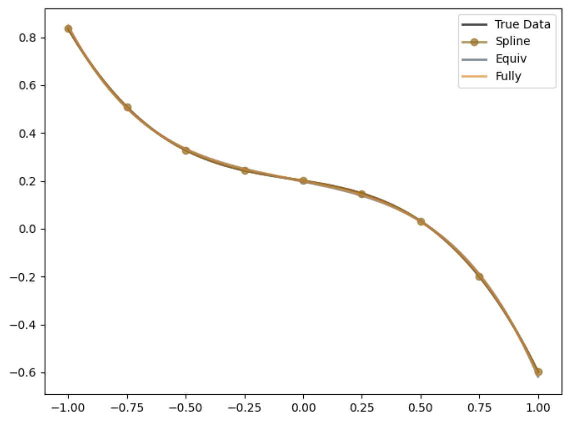

Figure ***V.11:*** *A plot from the main experiment with* 8 *summands, and degree* 4. *Example of* ***Spline****,* ***Equiv****, and* ***Fully*** *having comparable performance.*

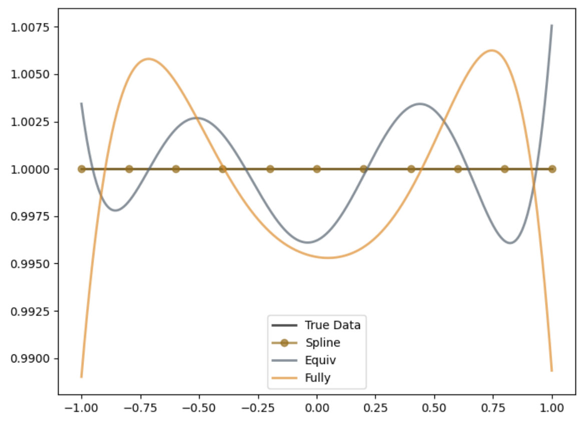

Figure ***V.12:*** *A plot from the main experiment with* 10 *summands, and degree* 5. *Example of* ***Equiv****, and* ***Fully*** *failing to fit a constant.*

Considerations Regarding Degree 6

As ***Spline*** only works up to degree 5, only ***Equiv*** and ***Fully*** are considered for degree 6. Similar to degrees 2 to 5, they both follow the true data. There are still a few outliers where one or both of them deviate from the data. An illustration of this can be found in Figure V.13.

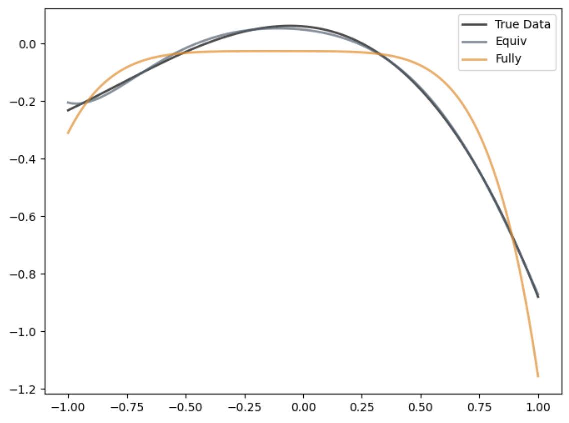

Figure ***V.13:*** *A plot from the main experiment with* 10 *summands, and degree* 6. *An example of* ***Equiv*** *outperforming* ***Fully*** *significantly.*

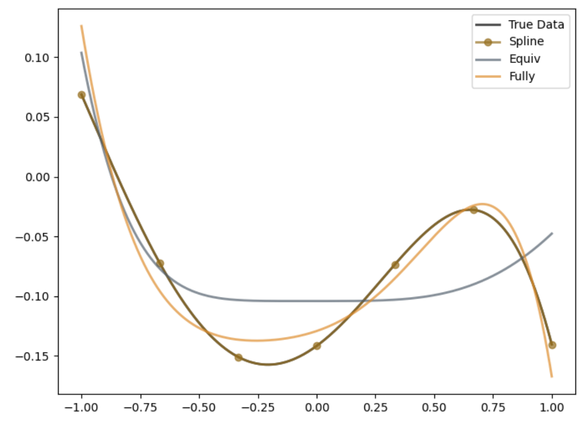

Figure ***V.14:*** *A plot from the main experiment with* 6 *summands, and degree* 5. *An example of* ***Fully*** *outperforming* ***Equiv*** *significantly.*

General Considerations

Figure V.13 together with Figure V.14, highlights that ***Equiv*** and ***Fully*** do not yield the same result. Figure V.13 is an example where ***Fully*** deviates and Figure V.14 is an example where ***Equiv*** deviates. Across the different degrees, the results indicate that ***Spline*** performs much better than ***Equiv*** and ***Fully***. However, ***Spline*** is poor when it comes to low summands and low degrees. ***Spline*** aside, the ***Equiv*** and ***Fully*** are behaving similarly with the occasional deviation. An exception to otherwise consistent performance occurs when the true data follows a straight line, examples of this can be seen in Figure V.8 and Figure V.12.

Average of Mean Square Error & Maximum Error

Next, the average of the mean square error and average maximum error across the 100 iterations, for each number of summands and degree as a parameter, are considered. Four of these 44 pictures are highlighted to illustrate the general principle and can be found in Figure V.15 to Figure V.18. At first, it should be noted that the result for degree 0 in general has much higher errors compared to the other degrees. Degree 0 is therefore excluded from the plots, as their presence would obscure the visualization of the remaining degrees due to their relatively minor magnitude. The errors clarify that ***Spline*** performs better when increasing the degree or the number of summands. Both errors improve exponentially and examples can be found in Figure V.15 and Figure V.16. The errors for 2 summands are generally higher than the other summands. However, it is only the case of 2 summands and degree 1 where ***Equiv*** and ***Fully*** are consistently better than ***Spline***. This is also clear from the illustration in Figure V.15.

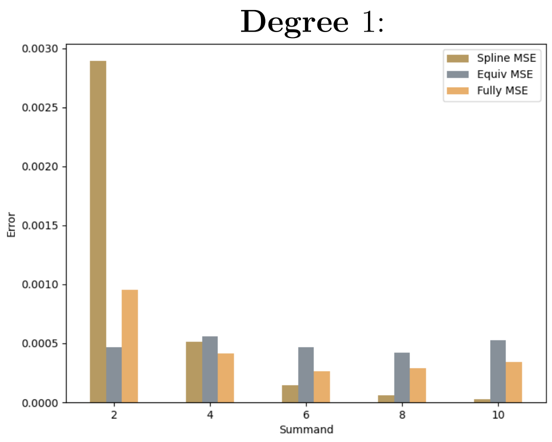

Figure V.15: *Mean square error of the approximation methods with degree* 1*, on the continuous piecewise polynomials.*

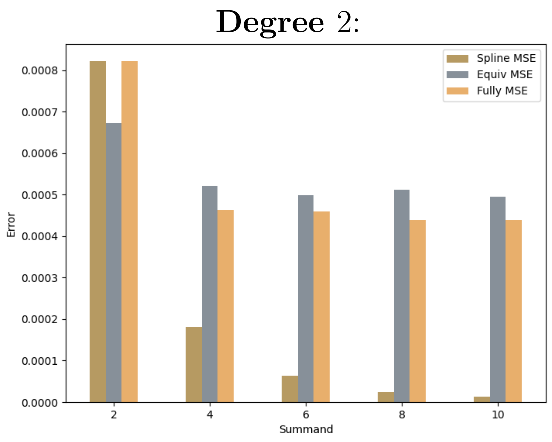

Figure V.16: *Mean square error of the approximation methods with degree* 2*, on the continuous piecewise polynomials.*

When comparing the performance of ***Equiv*** and ***Fully***, it becomes apparent that their performance depends on the degree. ***Fully*** typically perform better than ***Equiv*** for degrees 1 through 3 and ***Equiv*** performs best for degrees 5 and 6. This is most clearly seen in Figure V.17. Degree 4 is a gray area where ***Equiv*** sometimes performs best and other times ***Fully*** performs better, but they are relatively close. ***Equiv*** improves until degree 3 after which the performance plateaus. However, there is a slight variation after degree 3, but does not vary significantly. ***Fully*** improves until degree 3 after which the errors increase. Especially for degree 6 is there a high increase in error which results in a significant distance between ***Fully*** and ***Equiv***, which is illustrated in Figure V.18.

10 **Summands**:

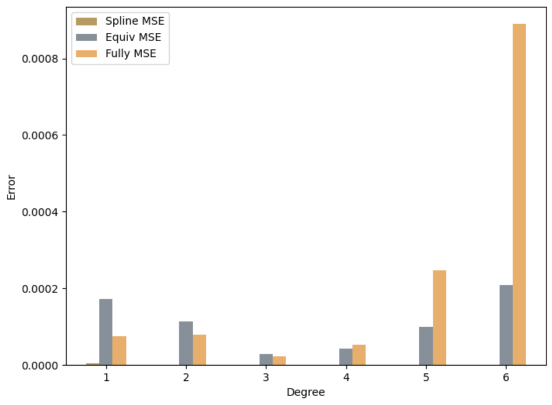

Figure V.17: *Mean square error of the approximation methods with* 10 *summands, on polynomials. Illustrates* ***Fully*** *worsens after degree* 4*.*

Degree 6:

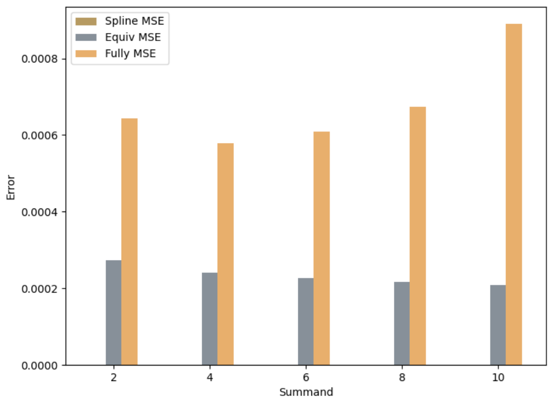

Figure V.18: *Mean square error of the approximation methods with degree* 6*, on polynomials. Illustrates the difference in error between* ***Equiv*** *and* ***Fully*** *for degree* 6*.*

Based on the comparison between all plots, the performance of ***Equiv*** and ***Fully*** are not significantly improved by additional summands. The performance still varies, but not in a way that signifies a strong correlation. The causes for these unexplained variations could be the initial weights when training the neural networks, which are randomly chosen.

The fact that ***Fully*** worsens significantly more than ***Equiv*** after degree 4, could be a testament to the notion that more weight is not always better. This aligns with the article [16], where better performance is achieved using sparsely connected neural networks compared to their fully connected counterparts. Additionally [19] showed that some neurons go dormant under training, which contributes to the potential advantage of sparsely connected neural networks over the fully connected counterparts.

Summary of Results:

- ***Spline*** performs better than ***Equiv*** and ***Fully*** except for 2 summands and degree 0.
- ***Equiv*** and ***Fully*** worsens after degree 4.
- ***Equiv*** outperforms ***Fully*** for degree 5 and 6.
- Both ***Equiv*** and ***Fully*** typically fail to approximate constants.

Final Comments on the Main Experiment

Due to time constraints, the main experiment is not generalized to multivariate B-splines. However, Theorem IV.7 and Theorem IV.10 allow for the generalization from univariate to multivariate B-splines. This generalization is a natural continuation of the main experiment. Moreover, it should be emphasized that the main experiment is the first attempt to implement this idea of a mixture between neural network and B-spline approximation. Both ***Equiv*** and ***Fully*** are constructed to emulate the well-known B-spline approximation method. It is expected that a first attempt may still have room for improvement, hence it is also reasonable to expect that the outcome may not surpass ***Spline***. Some improvements to the main experiment that could minimize the gap in performance between ***Spline*** and ***Equiv*** could be the following:

- Inject the weights from ***Spline*** into ***Equiv*** to insure that ***Equiv*** starts at equivalent performance.
- Insure that the same bit-precision is used throughout the main experiment.

It should be noted that ***Spline*** could be further improved by allowing for non-equidistant knots [20]. Thus it can most likely perform much better in a less restricted setting. For instance, the placement of the knots in the B-spline approximation does affect the performance. For lower summands having equidistant knots is likely, not optimal. Basing the knots' placement on the data's behavior will improve performance. Additionally in a real-world problem, the choice of the number of summands and degree would also depend on the given data and context of the application. As an example, the natural choice of degree might not have been 1 in Figure V.10, as the data behaves like a polynomial of higher degree. Overall, this concludes the experiments. However, it should be noted that the significance of the theoretical framework extends beyond the results of this experiment.

VI | Approximation Spaces

With the general setup for neural networks established, and the experiments concluded, the next step is to prepare some general tools from approximation theory. These tools are used to associate approximation spaces with a sequence of sets of realizations of neural networks. The goal is to use approximation theory as a framework to understand the approximation properties of the realization of neural networks. Since approximation theory is a well-studied area, the connection between approximation spaces and neural networks opens the possibility of adapting the analysis to already-known results in approximation theory. This chapter contains an introduction to these approximations spaces, and it is based on [12, p. 277-292], [15], [9], and [10, p. 216,234].

VI.1 Approximation Spaces

This section is devoted to introducing some general tools from approximation theory and defining the approximation spaces that later on are associated with the neural networks. In classical approximation theory, the notion of approximation spaces refers to quasi-normed spaces. The general idea is to define a function space by the approximation properties of the function. More specifically the *approximation spaces* consist of functions fulfilling a common upper bound or a common criteria for the decay of the approximation error. One way to measure the approximation error is by the *error of best approximation*.

Definition VI.1: Error of Best Approximation

Let $\mathcal{X}$ be a quasi-Banach space equipped with the quasi-norm $\|\cdot\|_{\mathcal{X}}$, let $f \in \mathcal{X}$ and let $\Sigma \subseteq \mathcal{X}$ be non-empty. The **error of best approximation** of f from Σ is given by

$$\mathscr{E}(f,\Sigma)_{\mathcal{X}} := \inf_{g\in\Sigma} \|f-g\|_{\mathcal{X}} \in [0,\infty).$$

Note that $\mathscr{E}(f,\Sigma)_{\mathcal{X}}$ gives the smallest error one can achieve using Σ.

Notation:

For the rest of this section let $\mathcal{X}$ be a quasi-Banach space equipped with the quasi-norm $\|\cdot\|_{\mathcal{X}}$, and let $\Sigma := \{\Sigma_n\}_{n\in\mathbb{N}_0}$ be an arbitrary family, where $\Sigma_n \subseteq \mathcal{X}$.

First, the approximation space of interest and its associated quasi-norm is introduced as an approximation class with an associated function. Later on, it is proven that the approximation class is indeed a quasi-Banach space with the associated function as its quasi-norm. In this case, the approximation class is an approximation space.

Definition VI.2: Approximation Classes

Let $f \in \mathcal{X}$, $\alpha \in (0,\infty)$ and $q \in (0,\infty]$. Define the approximation class

$$A_q^\alpha(\mathcal{X},\Sigma) \coloneqq \{f \in \mathcal{X} \mid \|f\|_{A_q^\alpha(\mathcal{X},\Sigma)} < \infty\},$$

and the associated function

$$\|f\|_{A_q^\alpha(\mathcal{X},\Sigma)} \coloneqq \begin{cases} \left(\sum_{n=1}^{\infty} \frac{1}{n}\left(n^\alpha \mathscr{E}(f,\Sigma_{n-1})_{\mathcal{X}}\right)^q\right)^{1/q}, & \text{if } 0 < q < \infty, \\ \sup_{n\geq 1} n^\alpha \mathscr{E}(f,\Sigma_{n-1})_{\mathcal{X}}, & \text{if } q = \infty. \end{cases}$$

The function $\|\cdot\|_{A_q^\alpha(\mathcal{X},\Sigma)}$ is also called the *Lorentz norm*. The family Σ is called an *approximation method*, and typically contains the approximants of interest. Moreover, the desired rate of decay of $\mathscr{E}(f,\Sigma_{n-1})_{\mathcal{X}}$ is prescribed by a discrete weighted ℓ^q-norm, where the weight depends on the parameter $\alpha > 0$. If $q = \infty$,

$$A_\infty^\alpha(\mathcal{X},\Sigma) = \left\{f \in \mathcal{X} \mid \mathscr{E}(f,\Sigma_n)_{\mathcal{X}} = \mathcal{O}(n^{-\alpha})\right\},$$

and if $q < \infty$, then $A_q^\alpha(\mathcal{X},\Sigma) \subseteq A_\infty^\alpha(\mathcal{X},\Sigma)$. Intuitively speaking, $A_q^\alpha(\mathcal{X},\Sigma)$ consists of those elements of $\mathcal{X}$ for which the error of best approximation by elements of Σ_n decays at least as $\mathcal{O}(n^{-\alpha})$ for $n \to \infty$.

Notation:

For the rest of the report, let $A_q^\alpha(\mathcal{X},\Sigma)$, and the associated function $\|\cdot\|_{A_q^\alpha(\mathcal{X},\Sigma)}$ be defined as in Definition VI.2.

It should be noted that definitions and notations for quasi-Banach spaces are used in relation to the approximation class before proving that it is a quasi-Banach space. This is done by replacing the quasi-Banach space with the approximation class and the quasi-norm with the function associated with the approximation class. The first definition that this is applied to is the following:

Definition VI.3: Continuous Embeddings

Let $\mathcal{X}_1, \mathcal{X}_2$ be quasi-Banach spaces, with quasi-norms $\|\cdot\|_{\mathcal{X}_1}$ and $\|\cdot\|_{\mathcal{X}_2}$ respectively. Then $\mathcal{X}_1 \hookrightarrow \mathcal{X}_2$ denotes a **continuous embedding** between $\mathcal{X}_1$ and $\mathcal{X}_2$, that is

$$\exists C \in \mathbb{R} : \mathcal{X}_1 \subseteq \mathcal{X}_2 \ \wedge \ C\|\cdot\|_{\mathcal{X}_1} \geq \|\cdot\|_{\mathcal{X}_2}.$$

Moreover, $\mathcal{X}_1$ is said to be **continuously embedded** in $\mathcal{X}_2$.

With these considerations, it is now possible to consider continuous embeddings between the approximation classes.

Proposition VI.4:

Assume $\Sigma_n \subseteq \Sigma_{n+1}$ for all $n \in \mathbb{N}_0$. If $\alpha = \beta$ and $q \leq s$ or if $\alpha > \beta$, the continuous embedding

$$A_q^\alpha(\mathcal{X}, \Sigma) \hookrightarrow A_s^\beta(\mathcal{X}, \Sigma)$$

holds.

Proof

Per assumptions $\Sigma_n \subseteq \Sigma_{n+1}$ for all $n \in \mathbb{N}_0$. Hence the error of best approximation satisfies that

$$\mathscr{E}(f, \Sigma_{n+1})_{\mathcal{X}} \leq \mathscr{E}(f, \Sigma_n)_{\mathcal{X}}, \quad \forall n \in \mathbb{N}_0. \tag{VI.1}$$

First assume $\alpha = \beta$ and $q \leq s$. Per construction of the quasi-norm and (VI.1), it holds that

$$\|f\|_{A_q^\alpha(\mathcal{X},\Sigma)} \geq \|f\|_{A_s^\alpha(\mathcal{X},\Sigma)}.$$

Therefore

$$A_q^\alpha(\mathcal{X}, \Sigma) \subseteq A_s^\alpha(\mathcal{X}, \Sigma),$$

which gives the desired.

Now assume $\alpha > \beta$ and $q, s \in (0, \infty]$. The setup is divided into three cases: $q = s$, $q < s$, and $q > s$. First, assume $q = s$. Then per construction of the quasi-norm and (VI.1), it holds that

$$\|f\|_{A_q^\alpha(\mathcal{X},\Sigma)} \geq \|f\|_{A_q^\beta(\mathcal{X},\Sigma)},$$

and therefore

$$A_q^\alpha(\mathcal{X}, \Sigma) \subseteq A_q^\beta(\mathcal{X}, \Sigma).$$

For $q < s$, similar steps as before gives that

$$\|f\|_{A_q^\alpha(\mathcal{X},\Sigma)} \geq \|f\|_{A_q^\beta(\mathcal{X},\Sigma)} \geq \|f\|_{A_s^\beta(\mathcal{X},\Sigma)}.$$

Lastly, assume $q > s$. First assume $q < \infty$, and define $p_1 := q/s > 1$, and p_2 such that

$$\frac{1}{p_1} + \frac{1}{p_2} = 1.$$

Then it holds that

$$\begin{aligned}
\|f\|_{A_s^\beta(\mathcal{X},\Sigma)} &= \left(\sum_{n=1}^{\infty} n^{-1} \Big(n^\beta \mathscr{E}(f, \Sigma_{n-1})_{\mathcal{X}} \Big)^s \right)^{1/s} \\
&= \left(\sum_{n=1}^{\infty} \Big(n^{\beta - 1/s - (\alpha - 1/q)} n^{\alpha - 1/q} \mathscr{E}(f, \Sigma_{n-1})_{\mathcal{X}} \Big)^s \right)^{1/s} \\
&\leq \left(\sum_{n=1}^{\infty} \Big(n^{\beta - 1/s - (\alpha - 1/q)} \Big)^{sp_2} \right)^{1/sp_2} \left(\sum_{n=1}^{\infty} \Big(n^{\alpha - 1/q} \mathscr{E}(f, \Sigma_{n-1})_{\mathcal{X}} \Big)^{sp_1} \right)^{1/sp_1} \\
&= \left(\sum_{n=1}^{\infty} \frac{1}{n^{(\alpha - \beta + 1/s - 1/q)sp_2}} \right)^{1/sp_2} \|f\|_{A_q^\alpha(\mathcal{X},\Sigma)},
\end{aligned}$$

where the inequality follows from Hölder's inequality for p_2 on the first term and p_1 on the last term in the sum. By defining $c := (\alpha - \beta + 1/s - 1/q)sp_2$, the first sum converges provided $c > 1$, per Theorem C.8. By considering

$$sp_2 = s\left(\frac{1}{1-\frac{1}{p_1}}\right) = \frac{s}{\frac{s}{s}-\frac{s}{q}} = \frac{1}{\frac{1}{s}-\frac{1}{q}},$$

it follows that

$$\begin{aligned} c &= \left(\alpha - \beta + \frac{1}{s} - \frac{1}{q}\right) sp_2 \\ &= \frac{\alpha - \beta + \frac{1}{s} - \frac{1}{q}}{\frac{1}{s}-\frac{1}{q}} \\ &= \frac{\alpha - \beta}{\frac{1}{s}-\frac{1}{q}} + 1, \end{aligned}$$

which shows that $c > 1$. Now assume $p = \infty$. By similar calculations, it holds that

$$\begin{aligned} \|f\|_{A_s^\beta(\mathcal{X},\Sigma)} &= \left(\sum_{n=1}^{\infty} n^{-1}\left(n^\beta \mathscr{E}(f,\Sigma_{n-1})_{\mathcal{X}}\right)^s\right)^{1/s} \\ &= \left(\sum_{n=1}^{\infty} \left(n^{\beta-1/s-\alpha} n^\alpha \mathscr{E}(f,\Sigma_{n-1})_{\mathcal{X}}\right)^s\right)^{1/s} \\ &\le \left(\sum_{n=1}^{\infty} \left(n^{\beta-1/s-\alpha} \sup_{n\ge 1} n^\alpha \mathscr{E}(f,\Sigma_{n-1})_{\mathcal{X}}\right)^s\right)^{1/s} \\ &= \left(\sum_{n=1}^{\infty} \left(n^{\beta-\alpha-1/s}\right)^s\right)^{1/s} \sup_{n\ge 1} n^\alpha \mathscr{E}(f,\Sigma_{n-1})_{\mathcal{X}}, \\ &= \left(\sum_{n=1}^{\infty} \frac{1}{n^{(\alpha-\beta+1/s)s}}\right)^{1/s} \|f\|_{A_q^\alpha(\mathcal{X},\Sigma)}. \end{aligned}$$

Similarly, it holds that

$$(\alpha - \beta + 1/s)s = (\alpha - \beta) + 1,$$

which with the use of Theorem C.8 gives that the first term converges.

For both $p < \infty$ and $p = \infty$, there exists a C such that

$$\|f\|_{A_s^\beta(\mathcal{X},\Sigma)} \le C\|f\|_{A_q^\alpha(\mathcal{X},\Sigma)},$$

so

$$A_q^\alpha(\mathcal{X},\Sigma) \subseteq A_s^\beta(\mathcal{X},\Sigma),$$

which gives the desired. ■

Note that this result shows that $A_q^\alpha(\mathcal{X},\Sigma)$ grows, when α decrease or q increase if α is fixed.

The next result similarly considers the continuous embeddings between approximation classes but with different approximation methods.

Lemma VI.5:

Assume that the following properties hold:

(i) $\Sigma_0 = \tilde{\Sigma}_0 = \{0\}$.

(ii) $\Sigma_n \subseteq \Sigma_{n+1} \wedge \tilde{\Sigma}_n \subseteq \tilde{\Sigma}_{n+1} \ \forall n \in \mathbb{N}_0$.

(iii) $\exists\, c \in \mathbb{N},\ \tilde{C} > 0\ \forall f \in \mathcal{X},\ m \in \mathbb{N} : \mathscr{E}(f, \Sigma_{cm})_{\mathcal{X}} \leq \tilde{C}\mathscr{E}\left(f, \tilde{\Sigma}_m\right)_{\mathcal{X}}$.

Then for arbitrary $q \in (0, \infty]$ and $\alpha > 0$ it holds that $A_q^\alpha\left(\mathcal{X}, \tilde{\Sigma}\right) \hookrightarrow A_q^\alpha(\mathcal{X}, \Sigma)$, that is

$$\exists\, C > 0 : \|f\|_{A_q^\alpha(\mathcal{X},\Sigma)} \leq C\|f\|_{A_q^\alpha\left(\mathcal{X},\tilde{\Sigma}\right)} \quad \forall f \in A_q^\alpha\left(\mathcal{X}, \tilde{\Sigma}\right), \tag{VI.2}$$

where C depends on α, q, c and $\tilde{C}$.

Proof

Assume $f \in A_q^\alpha\left(\mathcal{X}, \tilde{\Sigma}\right)$. To simplify the notation define

$$\varepsilon_n := \mathscr{E}(f, \Sigma_n)_{\mathcal{X}}, \quad \delta_n := \mathscr{E}\left(f, \tilde{\Sigma}_n\right)_{\mathcal{X}} \quad \forall n \in \mathbb{N}_0.$$

Since $\tilde{\Sigma}_0 = \{0\}$, see (i),

$$\|f\|_{\mathcal{X}} = \delta_0.$$

Moreover, as $0 \in \Sigma_n$ per (i)-(ii), it holds that

$$\varepsilon_n \leq \|f\|_{\mathcal{X}} = \delta_0 \quad \forall n \in \mathbb{N}_0.$$

Thus per (iii) there exist $c \in \mathbb{N}$ and $C_1 > 0$ such that

$$\varepsilon_{cm} \leq C_1\delta_m \quad \forall m \in \mathbb{N}.$$

Therefore by defining

$$m_n := \left\lfloor \frac{n-1}{c} \right\rfloor \in \mathbb{N}$$

for $n \in \mathbb{N}_{\geq c+1}$ it follows that

$$\varepsilon_{n-1} \leq \varepsilon_{cm_n} \leq C_1\delta_{m_n},$$

as $n - 1 \geq cm_n$. Overall it holds that

$$\varepsilon_{n-1} \leq \begin{cases} \delta_0, & \text{if } 1 \leq n \leq c, \\ C_1\delta_{m_n}, & \text{if } n \geq c+1. \end{cases}$$

For $n \in \mathbb{N}_{\geq c+1}$ it holds that

$$n \leq cm_n + c + 1 \leq (2c+1)m_n,$$

since $m_n \geq 1$ and $m_n \geq (n-1)/c - 1$. Hence

$$n^\alpha \leq (2c+1)^\alpha m_n^\alpha.$$

Similarly,

$$\frac{1}{n} \leq \frac{1}{m_n} \quad \forall n \in \mathbb{N}_{\geq c+1},$$

since $n \geq m_n$. The setup is now divided into two cases: $q < \infty$ and $q = \infty$.

Assume first that $q < \infty$. Using the previous equalities it holds that

$$\begin{aligned}
\|f\|^q_{A^\alpha_q(\mathcal{X},\Sigma)} &= \sum_{n=1}^{\infty} \frac{1}{n} \left(n^\alpha \varepsilon_{n-1}\right)^q \\
&= \sum_{n=1}^{c} \frac{1}{n} \left(n^\alpha \varepsilon_{n-1}\right)^q + \sum_{n=c+1}^{\infty} \frac{1}{n} \left(n^\alpha \varepsilon_{n-1}\right)^q \\
&\leq \delta_0^q \left(\sum_{n=1}^{c} n^{\alpha q - 1} \right) + C_1^q \sum_{n=c+1}^{\infty} \frac{1}{n} (n^\alpha \delta_{m_n})^q \\
&\leq C_2 \delta_0^q + C_1^q (2c+1)^{\alpha q} \sum_{n=c+1}^{\infty} \frac{1}{m_n} (m_n^\alpha \delta_{m_n})^q,
\end{aligned}$$

where

$$C_2 \coloneqq \sum_{n=1}^{c} n^{\alpha q - 1}.$$

To consider the previous inequalities further it is desired to use the size of the set

$$\{n \in \mathbb{N}_{\geq c+1} \mid m_n = m\}.$$

This set makes sense, as for $n \in \mathbb{N}_{\geq c+1}$ such that $m_n = m$ for some $m \in \mathbb{N}$, it holds that

$$m \leq \frac{n-1}{c} < m + 1.$$

This is equivalent with

$$cm + 1 \leq n < cm + c + 1,$$

so it holds that

$$|\{n \in \mathbb{N}_{\geq c+1} \mid m_n = m\}| \leq |\{n \in \mathbb{N}_{\geq c+1} \mid cm + 1 \leq n < cm + c + 1\}| = c.$$

Using this it holds that

$$\begin{aligned}
\sum_{n=c+1}^{\infty} \frac{1}{m_n} (m_n^\alpha \delta_{m_n})^q &= \sum_{m=1}^{\infty} \frac{1}{m} (m^\alpha \delta_m)^q |\{n \in \mathbb{N}_{\geq c+1} \mid m_n = m\}| \\
&\leq c \sum_{m=1}^{\infty} \frac{1}{m} (m^\alpha \delta_m)^q \\
&\leq c \sum_{m=1}^{\infty} \frac{1}{m} (m^\alpha \delta_{m-1})^q \\
&= c \|f\|^q_{A^\alpha_q(\mathcal{X},\tilde{\Sigma})}.
\end{aligned}$$

Overall this gives that

$$\begin{aligned}\|f\|^q_{A^\alpha_q(\mathcal{X},\Sigma)} &\leq \Big(C_2 + C_1^q(2c+1)^{\alpha q}c\Big)\|f\|^q_{A^\alpha_q\left(\mathcal{X},\tilde{\Sigma}\right)}\\ &= C\|f\|^q_{A^\alpha_q\left(\mathcal{X},\tilde{\Sigma}\right)},\end{aligned}$$

where

$$C := C_2 + C_1^q(2c+1)^{\alpha q}c.$$

Now assume $q = \infty$. Similarly as for $q < \infty$ it holds that

$$\begin{aligned}\|f\|_{A^\alpha_q(\mathcal{X},\Sigma)} &= \sup_{n\geq 1} n^\alpha \varepsilon_{n-1}\\ &\leq \sup_{c\geq n\geq 1} n^\alpha\varepsilon_{n-1} + \sup_{n\geq c+1} n^\alpha\varepsilon_{n-1}\\ &\leq \delta_0 \sup_{c\geq n\geq 1} n^\alpha + C_1 \sup_{n\geq c+1} n^\alpha \delta_{m_n}\\ &\leq C_2\delta_0 + C_1(2c+1)^\alpha \sup_{n\geq c+1} m_n^\alpha \delta_{m_n}.\end{aligned}$$

By making the same sets it holds that

$$\begin{aligned}\sup_{n\geq c+1} m_n^\alpha\delta_{m_n} &= \sup_{m\geq 1} m^\alpha\delta_m |\{n\in\mathbb{N}_{\geq c+1} \mid m_n = m\}|\\ &\leq c\sup_{m\geq 1} m^\alpha\delta_m\\ &\leq c\sup_{m\geq 1} m^\alpha\delta_{m-1}\\ &= c\|f\|_{A^\alpha_q\left(\mathcal{X},\tilde{\Sigma}\right)},\end{aligned}$$

which overall gives that

$$\begin{aligned}\|f\|_{A^\alpha_q(\mathcal{X},\Sigma)} &\leq \Big(C_2\delta_0 + C_1(2c+1)^\alpha c\Big)\|f\|_{A^\alpha_q\left(\mathcal{X},\tilde{\Sigma}\right)}\\ &= C\|f\|_{A^\alpha_q\left(\mathcal{X},\tilde{\Sigma}\right)},\end{aligned}$$

where

$$C := C_2\delta_0 + C_1(2c+1)^\alpha c.$$

In both cases, C only depends on α, q, c, and C_1, which completes the proof. ■

Remark VI.6:

The lemma can be rewritten with the third property replaced by the following:

(iii) Assume that

$$\mathscr{E}\left(f,\Sigma_{cm}\right)_{\mathcal{X}} \leq \mathscr{E}\left(f,\tilde{\Sigma}_m\right)_{\mathcal{X}} \quad \forall m\geq m_0\in\mathbb{N}.$$

This holds because of the following: By defining $\tilde{c} := n_0 c$, then $m_0 m \leq m_0$ for $m\in\mathbb{N}$. Hence $\tilde{\Sigma}_m \subseteq \tilde{\Sigma}_{m_0 m}$. Therefore

$$\begin{aligned}\mathscr{E}\left(f,\Sigma_{\tilde{c}m}\right)_{\mathcal{X}} &= \mathscr{E}\left(f,\tilde{\Sigma}_{cm_0m}\right)_{\mathcal{X}}\\ &\leq C\mathscr{E}\left(f,\tilde{\Sigma}_{m_0m}\right)_{\mathcal{X}}\\ &\leq C\mathscr{E}\left(f,\tilde{\Sigma}_m\right)_{\mathcal{X}}.\end{aligned}$$

A list of axioms is needed to verify some properties of the approximation classes. To define these axioms, define

$$\Sigma_\infty \coloneqq \bigcup_{n\in\mathbb{N}_0} \Sigma_n.$$

The axioms are then given by:

(P1) $\Sigma_0 = \{0\}$.

(P2) $\Sigma_n \subseteq \Sigma_{n+1} \quad \forall n \in \mathbb{N}_0$.

(P3) $\Sigma_n = C \cdot \Sigma_n \quad \forall C \in \mathbb{R} \setminus \{0\}, n \in \mathbb{N}_0$.

(P4) $\exists\, c \in \mathbb{N} : \Sigma_n + \Sigma_n \subseteq \Sigma_{cn} \quad \forall n \in \mathbb{N}_0$.

(P5) Σ_∞ is dense in $\mathcal{X}$.

For readers familiar with approximation classes the list of properties might seem incomplete. A discussion of this can be found in Remark A.2.

With the previous in place it is now desired to prove that the approximation class $A^\alpha_q(\mathcal{X}, \Sigma)$ is a well-defined approximation space.

Proposition VI.7: $A^\alpha_q(\mathcal{X}, \Sigma)$ is an Approximation Spaces

Assume properties (P1)-(P5) hold. Then the classes $(A^\alpha_q(\mathcal{X}, \Sigma), \|\cdot\|_{A^\alpha_q(\mathcal{X},\Sigma)})$ are quasi-Banach spaces satisfying that $A^\alpha_q(\mathcal{X}, \Sigma) \hookrightarrow \mathcal{X}$. Moreover if $\alpha = \beta$ and $q \le s$ or if $\alpha > \beta$, then it holds that

$$A^\alpha_q(\mathcal{X}, \Sigma) \hookrightarrow A^\beta_s(\mathcal{X}, \Sigma).$$

Proof

First it is desired to show that $(A^\alpha_q(\mathcal{X}, \Sigma), \|\cdot\|_{A^\alpha_q(\mathcal{X},\Sigma)})$ is a quasi-Banach space. For this both completeness and that $A^\alpha_q(\mathcal{X}, \Sigma)$ is a quasi-normed space with quasi-norm $\|\cdot\|_{A^\alpha_q(\mathcal{X},\Sigma)}$ must be proven. Moreover it is desired to show that $A^\alpha_q(\mathcal{X}, \Sigma) \hookrightarrow \mathcal{X}$. To do so it is desired to use Theorem C.12. Therefore it must be proven that the definition of $A^\alpha_q(\mathcal{X}, \Sigma)$ matches their definition of generalized approximation space, see Definition C.9. Per assumption $\mathcal{X}$ is a quasi-normed space, and Σ is defined such that (P1)-(P4) are satisfied, which leaves S being an admissible sequence space. Note the exact definition of an admissible sequence space does not matter in this context. In the current setup the sequence space S is implicitly given by $S \coloneqq \{\{a_n\}_{n=1}^\infty \subset \mathbb{R} \mid \|a_n\|_S < \infty\}$, where

$$\|\{a_n\}_{n=1}^\infty\|_S \coloneqq \left\| \left\{ n^{\alpha - \frac{1}{q}} a_n \right\}_{n=1}^\infty \right\|_{\ell^p(\mathbb{R})}$$

and $\alpha > 0$. Hence Remark C.10 gives that S is an admissible sequence space. Therefore is Proposition C.11 applicable, so $A^\alpha_q(\mathcal{X}, \Sigma)$ is a quasi-normed space, and

$$A^\alpha_q(\mathcal{X}, \Sigma) \hookrightarrow \mathcal{X}.$$

For the completeness Theorem C.12 is used. Per the definition of S, it holds that

$$\lim_{k\to\infty} \|\{a_n\}_{n=1}^k\|_S = \|\{a_n\}_{n=1}^\infty\|_S,$$

so Theorem C.12(ii) is satisfied and thus $A^\alpha_q(\mathcal{X}, \Sigma)$ is complete. The last part of the proposition follows directly from Proposition VI.4, because of (P2). ■

Under certain conditions $\|\cdot\|_{A^\alpha_q(\mathcal{X},\Sigma)}$ is a proper norm. A comment on these conditions can be found in Remark A.3.

VI.2 Approximation Classes of Neural Networks

Now the focus is restricted to the approximation classes associated with neural networks. Both generalized and strict neural networks will be studied in this setup. For this section, a list of assumptions is made. This is stated in the following:

Notation:

For the rest of this chapter let $d, k \in \mathbb{N}$, $\varrho : \mathbb{R} \to \mathbb{R}$ be an activation function, $\mathscr{L}$ a depth growth function, $\Omega \subseteq \mathbb{R}^d$ a subset, and $\mathcal{X}$ a quasi-Banach space, equipped with the quasi-norm $\|\cdot\|_{\mathcal{X}}$, with elements $f : \Omega \to \mathbb{R}^k$. Moreover, let $\Sigma := \{\Sigma_n\}_{n\in\mathbb{N}_0}$ be a family, where $\Sigma_n \subseteq \mathcal{X}$ is arbitrary unless otherwise stated.

The input and output dimensions d and k, as well as the set Ω are implicitly described by the space $\mathcal{X}$, when used.

This section is devoted to showing that the approximation theoretic properties of the resulting function classes gives that it does not matter whether the neural networks are generalized or strict. This holds when assuming that $\Omega \subset \mathbb{R}^d$ is bounded and the activation ϱ satisfies mild conditions. An equivalent result on an unbounded domain where the activation function ϱ can represent the identity is also stated. For this, the approximation spaces associated with neural networks are presented. However, as for the general approximation spaces, these are later proven to be well-defined. Therefore the use of *approximation space* and *quasi-norm*, is misleading, as it only is an approximation class with an associated function for now. Similarly, continuous embeddings are used for approximation classes in the same way as in the previous section.

Definition VI.8: Approximation Families for Generalized Neural Networks

For $n \in \mathbb{N}$ define the approximation families

$$\begin{aligned}
\mathtt{W}_0(\mathcal{X}, \varrho, \mathscr{L}) &:= \{0\} =: \mathtt{N}_0(\mathcal{X}, \varrho, \mathscr{L}),\\
\mathtt{W}_n(\mathcal{X}, \varrho, \mathscr{L}) &:= \mathtt{NN}^{\varrho, d, k}_{n, \mathscr{L}(n), \infty}(\Omega) \cap \mathcal{X},\\
\mathtt{N}_n(\mathcal{X}, \varrho, \mathscr{L}) &:= \mathtt{NN}^{\varrho, d, k}_{\infty, \mathscr{L}(n), n}(\Omega) \cap \mathcal{X}.
\end{aligned}$$

Remark VI.9:

Note that for $n = 0$, the introduced approximation families are defined to be $\{0\}$, even though $\mathtt{NN}^{\varrho, d, k}_{0, \mathscr{L}(n), \infty}(\Omega)$ and $\mathtt{NN}^{\varrho, d, k}_{\infty, \mathscr{L}(n), 0}(\Omega)$ not necessarily are equal to $\{0\}$. For instance $\mathtt{NN}^{\varrho, d, k}_{0, L, \infty}$ is the set of constant functions, see Lemma II.12, and $\mathtt{NN}^{\varrho, d, k}_{\infty, L, 0}$ is the set of affine linear functions.

In the case where the depth growth function is constant, that is $\mathscr{L} = L$ for some $L \in \mathbb{N}$, $\mathscr{L}$ is replaced with L in the notation, for instance $\mathtt{N}_n(\mathcal{X}, \varrho, L)$.

Definition VI.10: Approximation Space for Generalized Neural Networks

Define the approximation space and the associated quasi-norms

$$W_q^\alpha(\mathcal{X}, \varrho, \mathscr{L}) \coloneqq A_q^\alpha(\mathcal{X}, \Sigma),$$
$$\| \cdot \|_{W_q^\alpha(\mathcal{X}, \varrho, \mathscr{L})} \coloneqq \| \cdot \|_{A_q^\alpha(\mathcal{X}, \Sigma)},$$

with $\Sigma_n \coloneqq \mathtt{W}_n(\mathcal{X}, \varrho, \mathscr{L})$ for $n \in \mathbb{N}_0$. Similarly define

$$N_q^\alpha(\mathcal{X}, \varrho, \mathscr{L}) \coloneqq A_q^\alpha(\mathcal{X}, \Sigma),$$
$$\| \cdot \|_{N_q^\alpha(\mathcal{X}, \varrho, \mathscr{L})} \coloneqq \| \cdot \|_{A_q^\alpha(\mathcal{X}, \Sigma)},$$

with $\Sigma_n \coloneqq \mathtt{N}_n(\mathcal{X}, \varrho, \mathscr{L})$ for $n \in \mathbb{N}_0$.

The space $W_q^\alpha(\mathcal{X}, \varrho, \mathscr{L})$ is referred to as the *approximation spaces associated with connection complexity*. Similarly $N_q^\alpha(\mathcal{X}, \varrho, \mathscr{L})$ is referred to as the *approximation space associated with neuron complexity*. This notation highlights the activation function ϱ and the depth growth function $\mathscr{L}$, which is chosen because of the role in the approximation spaces. If the depth growth function is constant similar notation as for the approximation families is used.

Analogously approximation families and spaces are defined for strict neural networks:

Definition VI.11: Approximation Families for Strict Neural Networks

For $n \in \mathbb{N}$ define the approximation families

$$\mathtt{SW}_0(\mathcal{X}, \varrho, \mathscr{L}) \coloneqq \{0\} \eqqcolon \mathtt{SN}_0(\mathcal{X}, \varrho, \mathscr{L}),$$
$$\mathtt{SW}_n(\mathcal{X}, \varrho, \mathscr{L}) \coloneqq \mathtt{SNN}^{\varrho, d, k}_{n, \mathscr{L}(n), \infty}(\Omega) \cap \mathcal{X},$$
$$\mathtt{SN}_n(\mathcal{X}, \varrho, \mathscr{L}) \coloneqq \mathtt{SNN}^{\varrho, d, k}_{\infty, \mathscr{L}(n), n}(\Omega) \cap \mathcal{X}.$$

Similarly as for the generalized neural network, a special notation is used for the case where the depth growth function is constant.

Definition VI.12: Approximation Space for Strict Neural Networks

Define the approximation spaces and the associated quasi-norms

$$SW_q^\alpha(\mathcal{X}, \varrho, \mathscr{L}) \coloneqq A_q^\alpha(\mathcal{X}, \Sigma),$$
$$\| \cdot \|_{SW_q^\alpha(\mathcal{X}, \varrho, \mathscr{L})} \coloneqq \| \cdot \|_{A_q^\alpha(\mathcal{X}, \Sigma)},$$

with $\Sigma_n \coloneqq \mathtt{SW}_n(\mathcal{X}, \varrho, \mathscr{L})$ for $n \in \mathbb{N}_0$. Similarly define

$$SN_q^\alpha(\mathcal{X}, \varrho, \mathscr{L}) \coloneqq A_q^\alpha(\mathcal{X}, \Sigma),$$
$$\| \cdot \|_{SN_q^\alpha(\mathcal{X}, \varrho, \mathscr{L})} \coloneqq \| \cdot \|_{A_q^\alpha(\mathcal{X}, \Sigma)},$$

with $\Sigma_n \coloneqq \mathtt{SN}_n(\mathcal{X}, \varrho, \mathscr{L})$ for $n \in \mathbb{N}_0$.

These families are used to compare the relation between the connectivity and the number of neurons. It should be noted that Σ is the one related to neural networks of fixed or varying depth $L \in \mathbb{N} \cup \{\infty\}$.

Notation:

For the rest of the report define

$$\mathcal{L} := \sup_{n\in\mathbb{N}} \mathscr{L}(n) \in \mathbb{N} \cup \{\infty\},$$

given a depth growth function $\mathscr{L}$.

In the case where $\mathcal{L} = 2$ the neural network is called *shallow*. The depth growth function $\mathscr{L}$ and $\mathcal{L}$ are equivalent, see Proposition C.1. A special case of the approximation spaces is when the upper bound for the number of connections and the number of neurons is ∞. These are defined as follows:

Definition VI.13:

Define the space

$$\Sigma_\infty(\mathcal{X}, \varrho, \mathscr{L}) := \mathtt{NN}^{\varrho,d,k}_{\infty,\mathcal{L},\infty}(\Omega) \cap \mathcal{X} = \bigcup_{n\in\mathbb{N}_0} \mathtt{W}_n(\mathcal{X}, \varrho, \mathscr{L}) = \bigcup_{n\in\mathbb{N}_0} \mathtt{N}_n(\mathcal{X}, \varrho, \mathscr{L}).$$

Notation:

For the rest of the report, the families and spaces defined in Definition VI.8 to Definition VI.13 are used, without referring to the definition.

Per construction

$$\mathtt{SW}_n(\mathcal{X}, \varrho, \mathscr{L}) \subseteq \mathtt{W}_n(\mathcal{X}, \varrho, \mathscr{L}), \quad \forall n \in \mathbb{N}_0,$$

and similarly for $\mathtt{N}_n(\mathcal{X}, \varrho, \mathscr{L})$. So the generalized neural networks are at least as expressive as strict ones. Therefore it makes sense that there is a connection between the corresponding approximation classes.

Proposition VI.14:

Let $\alpha > 0$ and $q \in (0, \infty]$. Then it holds that

$$\begin{aligned}\|\cdot\|_{W^\alpha_q(\mathcal{X},\varrho,\mathscr{L})} &\leq \|\cdot\|_{SW^\alpha_q(\mathcal{X},\varrho,\mathscr{L})},\\ \|\cdot\|_{N^\alpha_q(\mathcal{X},\varrho,\mathscr{L})} &\leq \|\cdot\|_{SN^\alpha_q(\mathcal{X},\varrho,\mathscr{L})}.\end{aligned}$$

Hence

$$\begin{aligned}SW^\alpha_q(\mathcal{X}, \varrho, \mathscr{L}) &\hookrightarrow W^\alpha_q(\mathcal{X}, \varrho, \mathscr{L}),\\ SN^\alpha_q(\mathcal{X}, \varrho, \mathscr{L}) &\hookrightarrow N^\alpha_q(\mathcal{X}, \varrho, \mathscr{L}).\end{aligned}$$

Proof, see Appendix B.

The converse result holds under mild assumptions on the activation function ϱ when $\Omega \subset \mathbb{R}^d$ is bounded, where the approximation is in $L^p(\Omega, \mathbb{R}^k)$. Moreover, it holds on unbounded domains for ϱ that can represent the identity.

Theorem VI.15: Approximation Classes of Generalized & Strict ϱ-networks
Let $\Omega \subseteq \mathbb{R}^d$ be a measurable set with non-zero measure. Assume either that

(i) ϱ can represent the identity $\mathrm{id}_{\mathbb{R}}$, with m terms for some $m \in \mathbb{N}$,

(ii) Ω is bounded, ϱ is continuous, and ϱ is differentiable at some $x_0 \in \mathbb{R}$ with $\varrho'(x_0) \neq 0$.

Then for any $\mathscr{L}$, $k \in \mathbb{N}$, $\alpha > 0$, $p, q \in (0, \infty]$, with $\mathcal{X} := X_p^k(\Omega)$, it holds that

$$SW_q^\alpha(\mathcal{X}, \varrho, \mathscr{L}) = W_q^\alpha(\mathcal{X}, \varrho, \mathscr{L}),$$
$$SN_q^\alpha(\mathcal{X}, \varrho, \mathscr{L}) = N_q^\alpha(\mathcal{X}, \varrho, \mathscr{L}).$$

Furthermore, there exist $C_W, C_N < \infty$ such that

$$\|\cdot\|_{W_q^\alpha(\mathcal{X}, \varrho, \mathscr{L})} \leq \|\cdot\|_{SW_q^\alpha(\mathcal{X}, \varrho, \mathscr{L})} \leq C_W \|\cdot\|_{W_q^\alpha(\mathcal{X}, \varrho, \mathscr{L})},$$
$$\|\cdot\|_{N_q^\alpha(\mathcal{X}, \varrho, \mathscr{L})} \leq \|\cdot\|_{SN_q^\alpha(\mathcal{X}, \varrho, \mathscr{L})} \leq C_N \|\cdot\|_{N_q^\alpha(\mathcal{X}, \varrho, \mathscr{L})}.$$

Proof
The proof is stated for the approximation spaces associated with connection complexity since the proof with neuron complexity is analogous. It will be noted where the proofs differ.

Let $\mathcal{X} := X_p^k(\Omega)$. The setup is divided into two cases, depending on which assumption is made:

(i) Assume (i). Then Lemma III.8 gives that

$$\begin{aligned} \mathtt{W}_n(\mathcal{X}, \varrho, \mathscr{L}) &= \mathtt{NN}^{\varrho, d, k}_{n, \mathscr{L}(n), \infty}(\Omega) \cap \mathcal{X} \\ &\subseteq \mathtt{SNN}^{\varrho, d, k}_{m^2 n, \mathscr{L}(m^2 n), \infty}(\Omega) \cap \mathcal{X} \\ &= \mathtt{SW}_{m^2 n}(\mathcal{X}, \varrho, \mathscr{L}), \end{aligned}$$

since

$$\mathscr{L}(n) \leq \mathscr{L}(m^2 n), \quad \forall n \in \mathbb{N}.$$

Therefore

$$\mathscr{E}(f, \mathtt{SW}_{m^2 n}(\mathcal{X}, \varrho, \mathscr{L}))_{\mathcal{X}} \leq \mathscr{E}(f, \mathtt{W}_n(\mathcal{X}, \varrho, \mathscr{L}))_{\mathcal{X}}, \quad \forall n \in \mathbb{N}_0.$$

(ii) Assume (ii). Since ϱ is continuous it holds that all functions $f \in \mathtt{NN}^{\varrho, d, k}_{\infty, \infty, \infty}$ are continuous functions $f : \mathbb{R}^d \to \mathbb{R}^k$. Moreover $\overline{\Omega}$ is compact, as Ω is bounded. Therefore

$f\restriction_{\overline{\Omega}}$ is uniformly continuous and bounded. Hence it also holds for $f\restriction_{\Omega}$. Overall the previous implies that

$$\begin{aligned}\mathtt{SW}_n(\mathcal{X},\varrho,\mathscr{L}) &= \mathtt{SNN}^{\varrho,d,k}_{n,\mathscr{L}(n),\infty}(\Omega)\cap\mathcal{X}\\ &= \mathtt{SNN}^{\varrho,d,k}_{n,\mathscr{L}(n),\infty}(\Omega).\end{aligned}$$

Similarly as in (i)

$$\mathscr{L}(n)\le\mathscr{L}(4n),\quad \forall n\in\mathbb{N},$$

and Lemma III.5 gives that

$$\begin{aligned}\mathtt{W}_n(\mathcal{X},\varrho,\mathscr{L}) &= \mathtt{NN}^{\varrho,d,k}_{n,\mathscr{L}(n),\infty}(\Omega)\cap\mathcal{X}\\ &\subseteq \overline{\mathtt{SNN}^{\varrho,d,k}_{4n,\mathscr{L}(4n),\infty}(\Omega)}^{\mathcal{X}}\\ &\subseteq \overline{\mathtt{SW}_{4n}(\mathcal{X},\varrho,\mathscr{L})}^{\mathcal{X}},\end{aligned}$$

where the closure is taken with respect to the topology induced by $\|\cdot\|_{\mathcal{X}}$. Since $\Omega\subset\mathbb{R}^d$ is bounded, the locally uniform convergence on $\mathbb{R}^d$ implies convergence in $\mathcal{X}$, which allows the closure to be replaced in Lemma III.5. Let $X\subseteq\mathcal{X}$ be arbitrary. Then the continuity of $\|\cdot\|_{\mathcal{X}}$ gives that

$$\inf_{g\in X}\|f-g\|_{\mathcal{X}} = \inf_{g\in\overline{X}}\|f-g\|_{\mathcal{X}}.$$

So it does not change the distance of a function f to the set X, by switching from X to $\overline{X}$. This gives that

$$\mathscr{E}(f,\mathtt{SW}_{4n}(\mathcal{X},\varrho,\mathscr{L}))_{\mathcal{X}}\le\mathscr{E}(f,\mathtt{W}_n(\mathcal{X},\varrho,\mathscr{L}))_{\mathcal{X}},\quad \forall n\in\mathbb{N}_0.$$

In the proof of the approximation spaces with neuron complexity $4n$ are replaced by $2n$.

Now Lemma VI.5 gives that there exists a C_W such that

$$\|\cdot\|_{SW^{\alpha}_q(\mathcal{X},\varrho,\mathscr{L})}\le C_W\|\cdot\|_{W^{\alpha}_q(\mathcal{X},\varrho,\mathscr{L})}.$$

Therefore

$$\|\cdot\|_{W^{\alpha}_q(\mathcal{X},\varrho,\mathscr{L})}\le\|\cdot\|_{SW^{\alpha}_q(\mathcal{X},\varrho,\mathscr{L})}\le C_W\|\cdot\|_{W^{\alpha}_q(\mathcal{X},\varrho,\mathscr{L})}$$

according to Proposition VI.14, which was the desired. ■

Remark VI.16:

From a practical point of view, it is more convenient to work with strict neural networks instead of generalized neural networks. The problem with this approach is theoretical since strict neural networks do not possess the same closure properties as generalized neural networks. However, Theorem VI.15 shows that generalized and strict neural networks have identical properties, under specific conditions. Under the assumption that the domain is bounded, they are equal, and in the case of unbounded domains, it holds for ReLU-networks. So in these cases, it is irrelevant that the construction is based on the generalized neural networks.

VI.3 Connection Complexity & Neuron Complexity

In this section, the relation between the connectivity and the number of neurons is explored, as well as the relation to different depth growth functions.

Lemma VI.17:

Let $\Omega \subseteq \mathbb{R}^d$ be a measurable set with non-zero measure, and $\mathcal{X} := X_p^k(\Omega)$ for $p \in (0, \infty]$. Then for any $\alpha > 0$ and $q \in (0, \infty]$ it holds that

$$W_q^\alpha(\mathcal{X}, \varrho, \mathscr{L}) \hookrightarrow N_q^\alpha(\mathcal{X}, \varrho, \mathscr{L}) \hookrightarrow W_q^{\alpha/2}(\mathcal{X}, \varrho, \mathscr{L})$$
$$SW_q^\alpha(\mathcal{X}, \varrho, \mathscr{L}) \hookrightarrow SN_q^\alpha(\mathcal{X}, \varrho, \mathscr{L}) \hookrightarrow SW_q^{\alpha/2}(\mathcal{X}, \varrho, \mathscr{L}).$$

Furthermore, there exist $C, C_S > 0$ such that

$$\|\cdot\|_{W_q^\alpha(\mathcal{X}, \varrho, \mathscr{L})} \geq \|\cdot\|_{N_q^\alpha(\mathcal{X}, \varrho, \mathscr{L})} \geq C\|\cdot\|_{W_q^{\alpha/2}(\mathcal{X}, \varrho, \mathscr{L})},$$
$$\|\cdot\|_{SW_q^\alpha(\mathcal{X}, \varrho, \mathscr{L})} \geq \|\cdot\|_{SN_q^\alpha(\mathcal{X}, \varrho, \mathscr{L})} \geq C_S\|\cdot\|_{SW_q^{\alpha/2}(\mathcal{X}, \varrho, \mathscr{L})}.$$

If $\mathcal{L} = 2$, the exponent $\alpha/2$ can be replaced by α, which implies that

$$W_q^\alpha(\mathcal{X}, \varrho, \mathscr{L}) = N_q^\alpha(\mathcal{X}, \varrho, \mathscr{L}),$$

with equivalent norms.

Proof

The proof is stated for the approximation spaces associated with generalized neural networks since the proof with strict neural networks is analogous. According to Lemma II.13 and (II.3), it holds that

$$\begin{aligned}\mathtt{NN}^{\varrho,\,d,\,k}_{n,\,\mathscr{L}(n),\,\infty} &\subseteq \mathtt{NN}^{\varrho,\,d,\,k}_{n,\,\mathscr{L}(n),\,n}\\ &\subseteq \mathtt{NN}^{\varrho,\,d,\,k}_{\infty,\,\mathscr{L}(n),\,n}\\ &\subseteq \mathtt{NN}^{\varrho,\,d,\,k}_{n^2+(d+k)n+dk,\,\mathscr{L}(n),\,n}\\ &\subseteq \mathtt{NN}^{\varrho,\,d,\,k}_{n^2+(d+k)n+dk,\,\mathscr{L}(n),\,\infty}\end{aligned}$$

for all $n \in \mathbb{N}_0$. Therefore the error of best approximation satisfies

$$\begin{aligned}\mathscr{E}(f, \mathtt{W}_n(\mathcal{X}, \varrho, \mathscr{L}))_{\mathcal{X}} &\geq \mathscr{E}(f, \mathtt{N}_n(\mathcal{X}, \varrho, \mathscr{L}))_{\mathcal{X}} &\text{(VI.3)}\\ &\geq \mathscr{E}(f, \mathtt{W}_{n^2+(d+k)n+dk}(\mathcal{X}, \varrho, \mathscr{L}))_{\mathcal{X}} &\text{(VI.4)}\end{aligned}$$

for all $n \in \mathbb{N}_0$. Now (VI.3) gives that

$$\|\cdot\|_{W_q^\alpha(\mathcal{X}, \varrho, \mathscr{L})} \geq \|\cdot\|_{N_q^\alpha(\mathcal{X}, \varrho, \mathscr{L})},$$
$$W_q^\alpha(\mathcal{X}, \varrho, \mathscr{L}) \subseteq N_q^\alpha(\mathcal{X}, \varrho, \mathscr{L}).$$

It is now desired to prove that there exists a $C > 0$ such that

$$\|\cdot\|_{N_q^\alpha(\mathcal{X}, \varrho, \mathscr{L})} \geq C\|\cdot\|_{W_q^{\alpha/2}(\mathcal{X}, \varrho, \mathscr{L})}.$$

The proof is stated for $q < \infty$, since $q = \infty$ is similar. Now define $j := \max\{d, k\}$. To get the desired inequality, the sum in the norm is divided in two as follows:

$$\begin{aligned}\|f\|^q_{W^\alpha_q(\mathcal{X},\varrho,\mathscr{L})} &= \sum_{m=1}^{\infty} \frac{1}{m}\left(m^\alpha \mathscr{E}(f, \mathtt{W}_{m-1}(\mathcal{X},\varrho,\mathscr{L}))_{\mathcal{X}}\right)^q \\ &= \sum_{m=1}^{(j+1)^2} \frac{1}{m}\left(m^\alpha \mathscr{E}(f, \mathtt{W}_{m-1}(\mathcal{X},\varrho,\mathscr{L}))_{\mathcal{X}}\right)^q \\ &\quad+ \sum_{m\geq (j+1)^2+1} \frac{1}{m}\left(m^\alpha \mathscr{E}(f, \mathtt{W}_{m-1}(\mathcal{X},\varrho,\mathscr{L}))_{\mathcal{X}}\right)^q .\end{aligned}$$

For the first term, it holds that

$$\begin{aligned}\sum_{m=1}^{(j+1)^2} \frac{1}{m}\left(m^\alpha \mathscr{E}(f, \mathtt{W}_{m-1}(\mathcal{X},\varrho,\mathscr{L}))_{\mathcal{X}}\right)^q &\leq \left(\mathscr{E}(f, \mathtt{W}_0(\mathcal{X},\varrho,\mathscr{L}))_{\mathcal{X}}\right)^q \sum_{m=1}^{(j+1)^2} m^{\alpha q-1} \\ &= C_1\|f\|^q_{\mathcal{X}} \\ &\leq C_1\|f\|^q_{N^{2\alpha}_q(\mathcal{X},\varrho,\mathscr{L})},\end{aligned}$$

where

$$C_1 := \sum_{m=1}^{(j+1)^2} m^{\alpha q-1}.$$

To consider the last term a connection on the summands in the sum is considered. By definition of j it holds that

$$n^2 + (d+k)n + dk \leq (n+j)^2. \tag{VI.5}$$

Assume m satisfies that

$$(n+j)^2 + 1 \leq m \leq (n+j+1)^2.$$

By rewriting the upper and lower bound by

$$\begin{aligned}(n+j)^2 + 1 &= n^2 + 2nj + j^2 + 1, \\ (n+j+1)^2 &= n^2 + j^2 + 1 + 2nj + 2n,\end{aligned}$$

it holds that

$$\begin{aligned}m &\leq n^2\left(1 + \frac{j^2+1}{n^2} + \frac{2j+2}{n}\right) \\ &\leq n^2(j^2 + 2j + 4) \\ &\leq n^2(4j^2 + 8j + 4) \\ &= n^2(2j+2)^2.\end{aligned}$$

Thus $m \lesssim n^2$. Using this it holds that

$$m^{\alpha q-1} \lesssim n^{2\alpha q-2}.$$

The sum

$$\sum_{m=(n+j)^2+1}^{(n+j+1)^2} m^{\alpha q-1}$$

has

$$((n+j)+1)^2 - (n+j)^2 = 2n + 2j + 1$$

summands. Therefore

$$\begin{aligned}\sum_{m=(n+j)^2+1}^{(n+j+1)^2} m^{\alpha q-1} &\leq \sum_{m=(n+j)^2+1}^{(n+j+1)^2} C_2 n^{2\alpha q-2} \\ &= C_2(2n+2j+1)n^{2\alpha q-2} \\ &\leq C_2(2n+2jn+n)n^{2\alpha q-2} \\ &= C_2(2+2j+1)n^{2\alpha q-1}\end{aligned}$$

for all $n \in \mathbb{N}$, where C_2 comes from the equivalent relation. By defining $C_3 \coloneqq C_2(2+2j+1)$, which only depends on α, q, and j, and is finite, it holds that

$$\sum_{m=(n+j)^2+1}^{(n+j+1)^2} m^{\alpha q-1} \leq C_3 n^{2\alpha q-1} \quad \forall n \in \mathbb{N}.$$

Now (VI.4) and (VI.5) gives that for all $n \in \mathbb{N}$

$$\begin{aligned}&\sum_{m=(n+j)^2+1}^{(n+j+1)^2} \frac{1}{m}\left(m^{\alpha}\mathscr{E}(f, \mathtt{W}_{m-1}(\mathcal{X}, \varrho, \mathscr{L}))_{\mathcal{X}}\right)^q \\ &\qquad\leq \left(\mathscr{E}(f, \mathtt{W}_{(n+j)^2}(\mathcal{X}, \varrho, \mathscr{L}))_{\mathcal{X}}\right)^q \sum_{m=(n+j)^2+1}^{(n+j+1)^2} m^{\alpha q-1} \\ &\qquad\leq \left(\mathscr{E}(f, \mathtt{W}_{n^2+(d+k)n+dk}(\mathcal{X}, \varrho, \mathscr{L}))_{\mathcal{X}}\right)^q C n^{2\alpha q-1} \\ &\qquad\leq \left(\mathscr{E}(f, \mathtt{N}_n(\mathcal{X}, \varrho, \mathscr{L}))_{\mathcal{X}}\right)^q C n^{2\alpha q-1}.\end{aligned}$$

Using this by rewriting the following sum to align with the inequality it holds that

$$\begin{aligned}&\sum_{m\geq(j+1)^2+1} \frac{1}{m}\left(m^{\alpha}\mathscr{E}(f, \mathtt{W}_{m-1}(\mathcal{X}, \varrho, \mathscr{L}))_{\mathcal{X}}\right)^q \\ &\qquad= \sum_{n\in\mathbb{N}} \sum_{m=(n+j)^2+1}^{(n+j+1)^2} \frac{1}{m}\left(m^{\alpha}\mathscr{E}(f, \mathtt{W}_{m-1}(\mathcal{X}, \varrho, \mathscr{L}))_{\mathcal{X}}\right)^q \\ &\qquad\leq C_2 \sum_{n\in\mathbb{N}} n^{2\alpha q-1}\left(\mathscr{E}(f, \mathtt{W}_n(\mathcal{X}, \varrho, \mathscr{L}))_{\mathcal{X}}\right)^q \\ &\qquad\leq C_2\|f\|^q_{N_q^{2\alpha}(\mathcal{X}, \varrho, \mathscr{L})}.\end{aligned}$$

Combining the two terms it holds that

$$\|f\|^q_{W_q^{\alpha}(\mathcal{X}, \varrho, \mathscr{L})} \leq C_1\|f\|^q_{N_q^{2\alpha}(\mathcal{X}, \varrho, \mathscr{L})} + C_2\|f\|^q_{N_q^{2\alpha}(\mathcal{X}, \varrho, \mathscr{L})}$$

By rewriting this it holds that

$$\|\cdot\|_{N_q^{\alpha}(\mathcal{X}, \varrho, \mathscr{L})} \geq C\|\cdot\|_{W_q^{\alpha/2}(\mathcal{X}, \varrho, \mathscr{L})}.$$

To prove the last part of the lemma, assume that $\mathcal{L} = 2$. Then by (II.4) it holds that

$$\mathtt{NN}^{\varrho,\,d,\,k}_{\infty,\,\mathscr{L}(n),\,n} \subseteq \mathtt{NN}^{\varrho,\,d,\,k}_{(d+k)n,\,\mathscr{L}(n),\,n} \subseteq \mathtt{NN}^{\varrho,\,d,\,k}_{(d+k)n,\,\mathscr{L}(n),\,\infty},$$

which gives that

$$\mathscr{E}(f, \mathtt{N}_n(\mathcal{X}, \varrho, \mathscr{L}))_{\mathcal{X}} \leq \mathscr{E}(f, \mathtt{W}_{(d+k)n}(\mathcal{X}, \varrho, \mathscr{L}))_{\mathcal{X}} \quad \forall n \in \mathbb{N}_0.$$

Therefore assumption (iii) is satisfied in Lemma VI.5 and (i)-(ii) holds per construction of the approximation families. Hence Lemma VI.5 gives that

$$W_q^\alpha(\mathcal{X}, \varrho, \mathscr{L}) \supseteq N_q^\alpha(\mathcal{X}, \varrho, \mathscr{L})$$

together with a quasi-norm estimate. Combined with the general setup the spaces coincide with equivalent quasi-norm. ■

With this statement in place, the relation between approximation classes associated with different depth growth functions can be considered.

Lemma VI.18:

Assume $\mathscr{L}_1 \preceq \mathscr{L}_2$, and let c, n_0 be defined as in Definition II.6. Then for all $\alpha > 0$ and $q \in (0, \infty]$, there exist constants $C_W, C_N \in [1, \infty)$ depending on $\mathscr{L}_1, \mathscr{L}_2, \alpha$, and q such that

$$\begin{aligned} W_q^\alpha(\mathcal{X}, \varrho, \mathscr{L}_1) &\hookrightarrow W_q^\alpha(\mathcal{X}, \varrho, \mathscr{L}_2), \\ C_W \|\cdot\|_{W_q^\alpha(\mathcal{X}, \varrho, \mathscr{L}_1)} &\geq \|\cdot\|_{W_q^\alpha(\mathcal{X}, \varrho, \mathscr{L}_2)}, \end{aligned}$$

and

$$\begin{aligned} N_q^\alpha(\mathcal{X}, \varrho, \mathscr{L}_1) &\hookrightarrow N_q^\alpha\left(\mathcal{X}, \varrho, \mathscr{L}_2\right), \\ C_N \|\cdot\|_{N_q^\alpha(\mathcal{X}, \varrho, \mathscr{L}_1)} &\geq \|\cdot\|_{N_q^\alpha(\mathcal{X}, \varrho, \mathscr{L}_2)}. \end{aligned}$$

An analogous statement holds for strict neural networks. Moreover, the constants C_W and C_N only depends on the constants c, n_0, α and q.

Proof

The proof is stated for the approximation spaces associated with connection complexity since the proof with neuron complexity is analogous. It is desired to use Lemma VI.5. To do so properties (i)-(iii) are checked. Assumption (iii) is replaced by the equivalent assumption from Remark VI.6. Per construction of the spaces $\mathtt{W}_n\left(\mathcal{X}, \varrho, \mathscr{L}_2\right)$ both (i) and (ii) are satisfied. With the focus on properties (iii) it holds that for $n \geq n_0$

$$\mathscr{L}_1(n) \leq \mathscr{L}_2(cn),$$

per $\mathscr{L}_1 \preceq \mathscr{L}_2$. This implies that

$$\mathtt{NN}^{\varrho,\, d,\, k}_{n,\, \mathscr{L}_1(n),\, \infty} \subseteq \mathtt{NN}^{\varrho,\, d,\, k}_{n,\, \mathscr{L}_2(cn),\, \infty} \subseteq \mathtt{NN}^{\varrho,\, d,\, k}_{cn,\, \mathscr{L}_2(cn),\, \infty},$$

for all $n \geq n_0$. Therefore the error of best approximation satisfies

$$\mathscr{E}\left(f, \mathtt{W}_{cn}\left(\mathcal{X}, \varrho, \mathscr{L}_2\right)\right)_{\mathcal{X}} \leq \mathscr{E}(f, \mathtt{W}_n(\mathcal{X}, \varrho, \mathscr{L}_1))_{\mathcal{X}} \quad \forall n \geq n_0,$$

which gives that (iii) in Remark VI.6 is satisfied. Hence Lemma VI.5 gives the desired result. Note that the lemma also gives that the constant in (VI.2) only depends on the constants c, n_0, α and q. The proof for strict neural networks follows by replacing $\mathtt{NN}$ with $\mathtt{SNN}$. ■

A similar result in the case of equivalent depth growth functions is stated in the following theorem.

Theorem VI.19:
Assume $\mathscr{L}_1 \sim \mathscr{L}_2$, and let c, n_0 be defined as in Definition II.6. Then for all $\alpha > 0$ and $q \in (0, \infty]$, there exist constants $C_W, C_N \in [1, \infty)$ such that

$$\begin{aligned} W_q^\alpha(\mathcal{X}, \varrho, \mathscr{L}_1) &= W_q^\alpha(\mathcal{X}, \varrho, \mathscr{L}_2), \\ \frac{1}{C_W}\|\cdot\|_{W_q^\alpha(\mathcal{X},\varrho,\mathscr{L}_1)} \leq \|\cdot\|_{W_q^\alpha(\mathcal{X},\varrho,\mathscr{L}_2)} &\leq C_W\|\cdot\|_{W_q^\alpha(\mathcal{X},\varrho,\mathscr{L}_1)}, \end{aligned}$$

and

$$\begin{aligned} N_q^\alpha(\mathcal{X}, \varrho, \mathscr{L}_1) &= N_q^\alpha(\mathcal{X}, \varrho, \mathscr{L}_2), \\ \frac{1}{C_N}\|\cdot\|_{N_q^\alpha(\mathcal{X},\varrho,\mathscr{L}_1)} \leq \|\cdot\|_{N_q^\alpha(\mathcal{X},\varrho,\mathscr{L}_2)} &\leq C_N\|\cdot\|_{N_q^\alpha(\mathcal{X},\varrho,\mathscr{L}_1)}. \end{aligned}$$

An analogous statement holds for strict neural networks. Moreover, the constants C_W and C_N only depends on the constants c, n_0, α and q.

Proof
The proof is stated for the approximation spaces associated with generalized neural networks since the proof with strict neural networks is analogous. Per assumption, there exist $c, n_0 \in \mathbb{N}_0$ such that

$$\begin{aligned} \mathscr{L}_1(n) &\leq \mathscr{L}_2(cn), \\ \mathscr{L}_1(cn) &\geq \mathscr{L}_2(n). \end{aligned}$$

Using Lemma VI.18 in both cases it holds that

$$\begin{aligned} W_q^\alpha(\mathcal{X}, \varrho, \mathscr{L}_1) &\hookrightarrow W_q^\alpha(\mathcal{X}, \varrho, \mathscr{L}_2), & W_q^\alpha(\mathcal{X}, \varrho, \mathscr{L}_1) &\hookleftarrow W_q^\alpha(\mathcal{X}, \varrho, \mathscr{L}_2), \\ \|\cdot\|_{W_q^\alpha(\mathcal{X},\varrho,\mathscr{L}_2)} &\leq C_W\|\cdot\|_{W_q^\alpha(\mathcal{X},\varrho,\mathscr{L}_1)}, & C_W\|\cdot\|_{W_q^\alpha(\mathcal{X},\varrho,\mathscr{L}_2)} &\geq \|\cdot\|_{W_q^\alpha(\mathcal{X},\varrho,\mathscr{L}_1)}, \end{aligned}$$

and similarly for the approximation classes $N_q^\alpha(\mathcal{X}, \varrho, \mathscr{L}_1)$ and $N_q^\alpha(\mathcal{X}, \varrho, \mathscr{L}_2)$. Combining these gives the desired result. ■

If $\mathcal{L} < \infty$ then Proposition C.1 gives that $\mathscr{L} \sim \mathcal{L}$, so per Theorem VI.19 it holds that

$$W_q^\alpha(\mathcal{X}, \varrho, \mathscr{L}) = W_q^\alpha(\mathcal{X}, \varrho, \mathcal{L})$$

with the norms listed in the theorem. A similar result holds for $N_q^\alpha(\mathcal{X}, \varrho, \mathscr{L})$ and strict neural networks as well.

VI.4 Approximation Spaces of Neural Networks

The goal is to verify that the approximation classes associated with the neural networks are actual approximation spaces. To do so Proposition VI.7 is used. Therefore it is desired to prove that properties (P1)-(P5) are satisfied. Domains with certain properties will be necessary for this section. These domains will be defined in the following:

Definition VI.20: Admissible Domains

Let $\Omega \subseteq \mathbb{R}^d$. Then Ω is said to be an **admissible domain** if it is Borel measurable with nonzero measure.

Properties (P1)-(P4) hold in a more general setup and are proven in the next lemma. However, (P5) can fail to hold for certain activation functions, and needs a bit more consideration.

Lemma VI.21:

Let $\Sigma_n := \mathtt{W}_n(\mathcal{X}, \varrho, \mathscr{L})$ or let $\Sigma_n := \mathtt{N}_n(\mathcal{X}, \varrho, \mathscr{L})$. Then Σ_n satisfies (P1)-(P4), with $c = 2 + \min\{d, k\}$ in (P4).

Proof

The proof is stated for the approximation spaces associated with connection complexity since the proof with neuron complexity is analogous. The places where the proof varies are noted.

(P1) Per definition it holds that $\mathtt{W}_0(\mathcal{X}, \varrho, \mathscr{L}) = \{0\}$, which is property (P1).

(P2) It is desired to prove that $\mathtt{W}_n(\mathcal{X}, \varrho, \mathscr{L}) \subseteq \mathtt{W}_{n+1}(\mathcal{X}, \varrho, \mathscr{L})$ for all $n \in \mathbb{N}_0$. The setup is divided into $n = 0$ and $n \in \mathbb{N}$.

For $n = 0$ it is desired to prove that $0 \in \mathtt{W}_1(\mathcal{X}, \varrho, \mathscr{L})$. Lemma II.12 with $N = 0$ and $L = 1$ gives that $0 \in \mathtt{NN}^{\varrho, d, k}_{0, 1, 0}$. Thus per the construction of the families it holds that

$$0 \in \mathtt{NN}^{\varrho, d, k}_{0, 1, 0} \subseteq \mathtt{NN}^{\varrho, d, k}_{W, L, N} \quad \forall W, L, N \in \mathbb{N} \cup \{\infty\}.$$

By choosing $N = \infty$ and $W = 1$, it holds that $0 \in \mathtt{W}_1(\mathcal{X}, \varrho, \mathscr{L})$, as desired.

Now consider $n \in \mathbb{N}$. Let $W, N \in \mathbb{N}_0$, and assume $L \leq \tilde{L}$, for $L, \tilde{L} \in \mathbb{N} \cup \{\infty\}$. Per the construction of the families, it holds that

$$\mathtt{NN}^{\varrho, d, k}_{W, L, \infty} \subseteq \mathtt{NN}^{\varrho, d, k}_{W+1, \tilde{L}, \infty}.$$

Since $\mathscr{L}$ is non-decreasing, that is $\mathscr{L}(n) \leq \mathscr{L}(n+1)$, it holds that

$$\mathtt{W}_n(\mathcal{X}, \varrho, \mathscr{L}) \subseteq \mathtt{W}_{n+1}(\mathcal{X}, \varrho, \mathscr{L}) \quad \forall n \in \mathbb{N},$$

which proves property (P2).

(P3) It is desired to prove that $c \cdot \mathtt{W}_n(\mathcal{X}, \varrho, \mathscr{L}) = \mathtt{W}_n(\mathcal{X}, \varrho, \mathscr{L})$ for all $c \in \mathbb{R} \setminus \{0\}$ and $n \in \mathbb{N}_0$. For $n = 0$ the setup holds per construction of the families. For $n \in \mathbb{N}$, the setup is divided into two cases:

$$c \cdot \mathtt{W}_n(\mathcal{X}, \varrho, \mathscr{L}) \subseteq \mathtt{W}_n(\mathcal{X}, \varrho, \mathscr{L}),$$
$$c \cdot \mathtt{W}_n(\mathcal{X}, \varrho, \mathscr{L}) \supseteq \mathtt{W}_n(\mathcal{X}, \varrho, \mathscr{L}).$$

For the first part assume $f \in \mathtt{NN}^{\varrho, d, k}_{W, L, N}$. Then Lemma II.15(i) gives that $cf \in \mathtt{NN}^{\varrho, d, k}_{W, L, N}$ for $c \in \mathbb{R}$. Therefore it holds that

$$c \cdot \mathtt{W}_n(\mathcal{X}, \varrho, \mathscr{L}) \subseteq \mathtt{W}_n(\mathcal{X}, \varrho, \mathscr{L}) \quad \forall c \in \mathbb{R},\ n \in \mathbb{N}.$$

The second case is proved using the same procedure but with $1/c$, where $c \in \mathbb{R} \setminus \{0\}$. Thus the result holds.

(P4) It is desired to prove that there exists a $c \in \mathbb{N}$ such that

$$\mathtt{W}_n(\mathcal{X}, \varrho, \mathscr{L}) + \mathtt{W}_n(\mathcal{X}, \varrho, \mathscr{L}) \subseteq \mathtt{W}_{cn}(\mathcal{X}, \varrho, \mathscr{L}) \quad \forall n \in \mathbb{N}_0.$$

For $n = 0$, the setup holds per construction of the families. For $n \in \mathbb{N}$, assume $f_1, f_2 \in \mathtt{W}_n(\mathcal{X}, \varrho, \mathscr{L})$. Let $g_1, g_2 \in \mathtt{NN}^{\varrho,\, d,\, k}_{n,\, \mathscr{L}(n),\, \infty}$ such that

$$f_m = g_m\restriction_\Omega, \quad m \in \{1, 2\}.$$

By defining $L := \min\{\mathscr{L}(n), n\}$, Lemma II.13 gives that $g_m \in \mathtt{NN}^{\varrho,\, d,\, k}_{n,\, L,\, \infty}$. Now define $c_0 := \min\{d, k\}$, and

$$W_1 := 2n + c_0\,(L - 1) \leq (2 + c_0)n =: W_2.$$

Using Lemma II.15(ii) with $W = n$, $L = L$, $N = \infty$, and $n = 2$, it holds that

$$g_1 + g_2 \in \mathtt{NN}^{\varrho,\, d,\, k}_{W_1,\, L,\, \infty}.$$

Since $L \leq \mathscr{L}(n)$, $\mathscr{L}$ is non-decreasing, and $n \leq W_2$, it holds that

$$\mathtt{NN}^{\varrho,\, d,\, k}_{W_1,\, L,\, \infty} \subseteq \mathtt{NN}^{\varrho,\, d,\, k}_{W_1,\, \mathscr{L}(W_2),\, \infty}.$$

Therefore

$$f_1 + f_2 \in \mathtt{W}_{cn}(\mathcal{X}, \varrho, \mathscr{L})$$

for $c = 2 + c_0$. In the case of neuron complexity Proposition II.9 instead of Lemma II.13 is used where $L := \min\{\mathscr{L}(n), n + 1\}$. ■

$\Sigma_\infty(\mathcal{X}, \varrho, \mathscr{L})$ is a linear space since Lemma VI.21 gives that (P3) and (P4) is satisfied.

With (P1)-(P4) in place, it remains to prove (P5), which is that Σ_∞ is dense in $\mathcal{X}$. As mentioned, property (P5) can fail to hold for certain activation functions. An example of such activation functions can be found in Example A.5. Thus (P5) is proven under specific assumptions. To do so, the density of $\Sigma_\infty(\mathcal{X}, \varrho, \mathscr{L})$ in $\mathcal{X} := X^k_p(\Omega)$ is explored. For this, the concept of a *non-degenerate* activation function is defined.

Definition VI.22: Non-degenerated

The activation function $\varrho : \mathbb{R} \to \mathbb{R}$ is called **non-degenerate** if it satisfies the following:

(i) ϱ is Borel measurable.

(ii) ϱ is locally bounded.

(iii) There exists a closed null-set $X \subseteq \mathbb{R}$ such that ϱ is continuous at every point $x_0 \in \mathbb{R} \setminus X$.

(iv) There does not exists a polynomial $p : \mathbb{R} \to \mathbb{R}$ such that $\varrho(x) = p(x)$ for almost all $x \in \mathbb{R}$.

Most of these conditions are fulfilled when the activation function is continuous, see Proposition C.13. The density of $\Sigma_\infty(\mathcal{X}, \varrho, \mathscr{L})$ in $\mathcal{X} := X_p^k(\Omega)$ is first proven for bounded domains:

Theorem VI.23:

Let the activation function ϱ be Borel measurable and locally bounded. Let $\Omega \subset \mathbb{R}^d$ be a bounded admissible domain, assume $\mathcal{L} \geq 2$, and let $\mathcal{X} := X_p^k(\Omega)$, with $p \in (0, \infty)$. Then the following holds:

(i) $\mathtt{NN}^{\varrho, d, k}_{\infty, \infty, \infty}(\Omega) \subseteq \mathcal{X}$.

(ii) If ϱ is non-degenerate, then $\Sigma_\infty(\mathcal{X}, \varrho, \mathscr{L})$ is dense in $\mathcal{X}$.

If additionally ϱ is continuous then (i) and (ii) hold for $p = \infty$.

Proof

(i) It is desired to prove that $\mathtt{NN}^{\varrho, d, k}_{\infty, \infty, \infty}(\Omega) \subseteq X_p^k(\Omega)$. The setup is divided into two cases: $p < \infty$ and $p = \infty$. First consider $p < \infty$: Per assumption, ϱ is Borel measurable and locally bounded. Therefore Lemma C.14, with assumption (i)-(ii), gives that all $f \in \mathtt{NN}^{\varrho, d, k}_{\infty, \infty, \infty}$ is locally bounded and measurable. Since the domain Ω is bounded it holds that $f{\restriction}_\Omega \in L^p(\Omega, \mathbb{R}^k)$ for all $p \in (0, \infty]$. Therefore it holds that $f{\restriction}_\Omega \in X_p^k(\Omega)$ if $p < \infty$, which was the desired.

To prove (i) for $p = \infty$, ϱ is assumed to be continuous. Therefore Lemma C.14, with assumption (iii), gives that f is continuous. Thus f is uniformly continuous and bounded on the compact set $\overline{\Omega}$. Therefore $f{\restriction}_\Omega$ is uniformly continuous and bounded, so $f{\restriction}_\Omega \in X_\infty^k(\Omega)$.

(ii) Similar to the proof of (i), the setup is divided into two cases: $p < \infty$ and $p = \infty$. For $p < \infty$ let $f \in X_p^k(\Omega)$, be given by $f := (f_1, \dots, f_k)$, and assume $\varepsilon > 0$. For all $m \in \{1, \dots, k\}$ define $\tilde{f}_m \in L^p(\mathbb{R}^d, \mathbb{R})$, to be f_m in Ω and zero elsewhere in $\mathbb{R}^d$. Per Theorem C.15 it holds that $\mathcal{C}_c^\infty(\mathbb{R}^d, \mathbb{R})$ is dense in $L^p(\mathbb{R}^d, \mathbb{R})$. Therefore there exists $g_m \in \mathcal{C}_c^\infty(\mathbb{R}^d, \mathbb{R})$ such that

$$\left\| \tilde{f}_m - g_m \right\|_{L^p(\mathbb{R}^d, \mathbb{R})} \leq \varepsilon.$$

Note $\mathtt{NN}^{\varrho, d, 1}_{\infty, 2, \infty} \subseteq \mathtt{NN}^{\varrho, d, 1}_{\infty, \mathcal{L}, \infty}$ holds per assumptions. Now choose $R > 0$ such that

$$\operatorname{supp}(g_m), \Omega \subseteq [-R, R]^d.$$

Then The Universal Approximation Theorem, Theorem C.16, with the compact set $K = [-R, R]^d$ and the continuous functions $f = g_m$, gives that there exists

$$h_m \in \mathtt{NN}^{\varrho, d, 1}_{\infty, 2, \infty} \subseteq \mathtt{NN}^{\varrho, d, 1}_{\infty, \mathcal{L}, \infty}$$

such that

$$\|g_m - h_m\|_{L^\infty([-R,R]^d, \mathbb{R})} \leq \frac{\varepsilon}{(2R)^{d/p}}.$$

Now Lemma II.16 gives that

$$h := (h_1, \dots, h_k) \in \mathtt{NN}^{\varrho, d, k}_{\infty, \mathcal{L}, \infty}$$

and per Theorem VI.23(i) $h\restriction_\Omega \in X_p^k(\Omega)$. Thus $h\restriction_\Omega \in \Sigma_\infty(X_p^k(\Omega), \varrho, \mathscr{L})$. In general for $c_1, c_2 \in \mathbb{R}$ it holds that

$$(c_1 + c_2)^p \leq (2\max\{c_1, c_2\})^p \leq 2^p\left(c_1^p + c_2^p\right).$$

Using this it holds that

$$\begin{aligned}\left|h_m(x) - \tilde{f}_m(x)\right|^p &\leq \left(|h_m(x) - g_m(x)| + \left|g_m(x) - \tilde{f}_m(x)\right|\right)^p \\ &\leq 2^p\left(\frac{\varepsilon^p}{(2R)^d} + \left|g_m(x) - \tilde{f}_m(x)\right|^p\right)\end{aligned}$$

for all $x \in [-R, R]^d$. Therefore it holds that

$$\begin{aligned}\left\|h_m - \tilde{f}_m\right\|^p_{L^p([-R,R]^d,\mathbb{R})} &= \int_{[-R,R]^d} \left|h_m(x) - \tilde{f}_m(x)\right|^p dx \\ &\leq \int_{[-R,R]^d} 2^p\left(\frac{\varepsilon^p}{(2R)^d} + \left|g_m(x) - \tilde{f}_m(x)\right|^p\right) dx \\ &\leq 2^p\left(\frac{\varepsilon^p}{(2R)^d}\int_{[-R,R]^d} 1\, dx + \left\|g_m - \tilde{f}_m\right\|^p_{L^p([-R,R]^d,\mathbb{R})}\right) \\ &\leq 2^p\left(\varepsilon^p + \varepsilon^p\right) \\ &\leq 2^{p+1}\varepsilon^p.\end{aligned}$$

Since $\tilde{f}_m\restriction_\Omega = f_m$ it holds that

$$\begin{aligned}\|f - h\restriction_\Omega\|^p_{L^p(\Omega,\mathbb{R}^k)} &\leq \sum_{m=1}^k \|f_m - h_m\restriction_\Omega\|^p_{L^p(\Omega,\mathbb{R})} \\ &\leq \sum_{m=1}^k \left\|\tilde{f}_m - h_m\right\|^p_{L^p([-R,R]^d,\mathbb{R})} \\ &\leq 2^{1+p}k\varepsilon^p,\end{aligned}$$

and since $\varepsilon > 0$ was arbitrary, this proves the desired density for $p < \infty$.

For $p = \infty$, let $f \in X_\infty^k(\Omega)$, be given by $f := (f_1, \dots, f_k)$, and assume $\varepsilon > 0$. Lemma C.17 gives that there exists a continuous function $\tilde{f} : \mathbb{R}^d \to \mathbb{R}^k$ such that $f = \tilde{f}\restriction_\Omega$. Let $\tilde{f}_m$ be the component function in $\tilde{f}$ such that

$$\tilde{f} := \left(\tilde{f}_1, \dots, \tilde{f}_k\right),$$

and choose $R > 0$ such that $\Omega \subseteq [-R, R]^d$. Then for all $m \in \{1, \dots, k\}$ there exist $h_m \in \mathtt{NN}^{\varrho, d, 1}_{\infty, 2, \infty} \subseteq \mathtt{NN}^{\varrho, d, 1}_{\infty, \mathcal{L}, \infty}$ such that

$$\left\|h_m - \tilde{f}_m\right\|_{L^\infty([-R,R]^d,\mathbb{R})} \leq \varepsilon$$

according to The Universal Approximation Theorem, Theorem C.16, with $K = [-R, R]^d$ and $f = \tilde{f}_m$.

Now Lemma II.16 gives that

$$h := (h_1, \dots, h_k) \in \mathtt{NN}^{\varrho, d, k}_{\infty, \mathcal{L}, \infty}.$$

Since ϱ is continuous $h\restriction_\Omega \in X_\infty^k(\Omega)$ per (i). Thus $h\restriction_\Omega \in \Sigma_\infty(X_\infty^k(\Omega), \varrho, \mathscr{L})$. Therefore as $\tilde{f}_m\restriction_\Omega = f_m$ it holds that

$$\begin{aligned}\|f(x) - h\restriction_\Omega(x)\|_\infty &= \max_{m\in\{1,\dots,k\}} \|f_m(x) - h_m\restriction_\Omega(x)\|_{L^\infty(\Omega,\mathbb{R})}\\ &\leq \max_{m\in\{1,\dots,k\}} \left\|\tilde{f}_m(x) - h_m(x)\right\|_{L^\infty([-R,R]^d,\mathbb{R})}\\ &\leq \varepsilon.\end{aligned}$$

Since $\varepsilon > 0$ was arbitrary, this proves the desired density for $p = \infty$. ■

Given specific assumptions on the activation function, Theorem VI.23 directly gives that (P5) is satisfied for bounded admissible domains:

Corollary VI.24:

If $\mathcal{L} \geq 2$ and ϱ is continuous and not a polynomial, then property (P5) holds for any bounded admissible domain $\Omega \subset \mathbb{R}^d$ and $p \in (0, \infty]$.

Proof

Theorem VI.23(ii) will be used, so it is only needed to check whether ϱ is non-degenerate or not since the rest of the conditions are satisfied by assumption. This holds per Proposition C.18 since it is continuous and not a polynomial. Thus $\Sigma_\infty(X_p^k(\Omega), \varrho, \mathscr{L})$ is dense in $X_p^k(\Omega)$, which is properties (P5). ■

With this in place, it is desired to prove that (P5) is satisfied for unbounded admissible domain as well. To do so, some notation is needed:

- The *periodization* of a function, see Definition C.19.
- The *radially decreasing* L^1*-majoriant* of a function, see Definition C.20.
- *Radon measures*, see Definition C.21.
- The dual space for $\mathcal{C}_0(\mathbb{R}^d)$ is denoted by $(\mathcal{C}_0(\mathbb{R}^d))^*$.

Theorem VI.25:

Let $p \in (0, \infty]$, and ϱ be Borel measurable and locally bounded, and assume that there exists some

$$f \in \overline{\mathtt{NN}^{\varrho,d,1}_{\infty,L,\infty}(\Omega) \cap X_p^1(\Omega)}$$

where the closure is taken in $L^p(\mathbb{R}^d, \mathbb{R})$, such that $f : \mathbb{R}^d \to \mathbb{R}$ and the following holds:

(i) There is a non-increasing function $g : [0, \infty) \to [0, \infty)$ which satisfy that

$$\int_{\mathbb{R}^d} g(\|x\|_{\mathbb{R}^d})\, dx < \infty$$

and that

$$|f(x)| \leq g(\|x\|_{\mathbb{R}^d}) \quad \forall x \in \mathbb{R}^d.$$

(ii) The integral

$$\int_{\mathbb{R}^d} f(x)\, dx \neq 0.$$

Then $\Sigma_\infty(X_p^k(\Omega), \varrho, \mathscr{L})$ is dense in $X_p^k(\Omega)$ for every admissible domain $\Omega \subseteq \mathbb{R}^d$ and every $k \in \mathbb{N}$.

Proof
By defining

$$\mathcal{V} := \mathtt{NN}^{\varrho,\,d,\,1}_{\infty,\,L,\,\infty} \cap X^1_p(\mathbb{R}^d)$$

Lemma II.15 gives that $\mathcal{V}$ is a vector space. As a preliminary consider $A \in \mathrm{GL}(\mathbb{R}^d)$, and $b \in \mathbb{R}^d$. Then Lemma II.17(i), with $P = A \cdot + b$, and $Q = I$ gives that if $h \in \mathcal{V}$, then $h(A \cdot + b) \in \mathcal{V}$. Taking the closure in $X^1_p(\mathbb{R}^d)$, the same properties holds for $\overline{\mathcal{V}}$.

To prove the desired first $\mathcal{V}$ is proven to be dense in $X^1_p(\mathbb{R}^d)$, where the setup is divided into three cases: (i): $p \in [1, \infty)$, (ii): $p \in (0, 1)$, and (iii): $p = \infty$.

(i) Let $\tilde{g}$ be any radially decreasing L^1-majorant for 0, see Definition C.20. Then by assumption (i) it follows that $g + \tilde{g}$ is a radially decreasing L^1-majorant for h. By defining a periodization P of $|f|$, see Definition C.19, Lemma C.22 shows that

$$P|f| \in L^\infty(\mathbb{R}^d, \mathbb{R}) \subseteq L^p_{\mathrm{loc}}(\mathbb{R}^d, \mathbb{R}),$$

where the inclusion holds per construction of the spaces. Now since $f \in L^p(\mathbb{R}^d, \mathbb{R})$ per assumption,

$$\int_{\mathbb{R}^d} f(x)\, dx \neq 0,$$

per (ii), and $P|f| \in L^p_{\mathrm{loc}}(\mathbb{R}^d, \mathbb{R})$, Corollary C.23 gives that the space

$$\mathcal{V}_0 := \left\{ f_{n,m} \;\middle|\; dn > 0,\, m \in \mathbb{Z}^d \right\}$$

is dense in $L^p(\mathbb{R}^d, \mathbb{R})$, where $f_{n,m}(x) := 2^{nd/p} f(2^n x - m)$. Since $f \in \overline{\mathcal{V}}$, it holds that $\mathcal{V}_0 \subseteq \overline{\mathcal{V}}$, according to the preliminary part of the proof. Hence $\mathcal{V} \subseteq L^p(\mathbb{R}^d, \mathbb{R}) = X^1_p(\mathbb{R}^d)$ is dense, which completes this part of the proof.

(ii) For $p \in (0, 1)$, it holds that $f \in L^1(\mathbb{R}^d, \mathbb{R}) \cap L^p(\mathbb{R}^d, \mathbb{R})$, and

$$\int_{\mathbb{R}^d} f(x)\, dx \neq 0.$$

Therefore Proposition C.24 gives that there exists a $\lambda \in \mathbb{R}$ such that

$$\|\lambda P f(B \bullet) - 1\|_{L^p(\Omega, \mathbb{R})} < 1,$$

where P is a periodization of f with translation matrix B, and $\Omega = [0, 1)^d$. By defining $\mathcal{V}_0$ and $f_{n,m}$ the same as in (i) Theorem C.25 shows that $\mathcal{V}_0$ is dense in $L^p(\mathbb{R}^d, \mathbb{R})$ since

$$Sc := \sum_{n \in \mathbb{N}} \sum_{m \in \mathbb{Z}^d} c_{n,m} f_{n,m}, \quad \forall c \in \ell^p(\mathbb{N} \times \mathbb{Z}^d)$$

is subjective. By following the same step as in (i), it holds that $\mathcal{V}$ is dense in $L^p(\mathbb{R}^d, \mathbb{R})$.

(iii) For $p = \infty$, it should be noted that $X^1_\infty(\mathbb{R}^d) = \mathcal{C}_0(\mathbb{R}^d, \mathbb{R})$. Assume for contradiction that $\mathcal{V}$ is not dense in $\mathcal{C}_0(\mathbb{R}^d, \mathbb{R})$. Then there exists a function $w \in \mathcal{C}_0(\mathbb{R}^d, \mathbb{R}) \setminus \overline{\mathcal{V}}$ such that $w \neq 0$. It is now desired to use a standard trick for Hahn-Banach Theorem, see Theorem C.26. To do so consider

$$\inf_{v \in \overline{\mathcal{V}}} \|w + v\|_\infty > 0$$

since $\overline{\mathcal{V}}$ is closed. By defining a bigger subspace

$$\tilde{\mathcal{V}} := \left\{ \overline{\mathcal{V}} + cw \mid c \in \mathbb{R} \right\}$$

every element $h := u + cw \in \tilde{\mathcal{V}}$ has a unique representation, where $u \in \overline{\mathcal{V}}$. Now define

$$\psi(h) := c \inf_{v \in \overline{\mathcal{V}}} \|w + v\|_\infty,$$

which is a linear function, and is not zero for h, where $c \neq 0$. However for h, where $c = 0$, $h \in \overline{\mathcal{V}}$, and $\psi(h) = 0$. Moreover, let P be the operator given by $P(h) := \|h\|_\infty$. Then

$$\psi(h) = c \inf_{v \in \overline{\mathcal{V}}} \|w + v\|_\infty < |c| \left\| \frac{1}{c} u + w \right\|_\infty = \|u + cw\|_\infty = \|h\|_\infty.$$

Now Hahn-Banach Theorem with $\mathcal{X} = \mathcal{C}_0(\mathbb{R}^d, \mathbb{R})$, $V = \tilde{\mathcal{V}}$, and ψ, P given as above implies that there exists a bounded linear function $\varphi \in (\mathcal{C}_0(\mathbb{R}^d, \mathbb{R}))^*$ such that $\varphi\restriction_{\tilde{\mathcal{V}}} = \psi$. Moreover $\varphi \not\equiv 0$ and $\varphi \equiv 0$ on $\overline{\mathcal{V}}$. To get a contradiction it is desired to prove that $\varphi \equiv 0$.

Now Riesz Representation Theorem, see Theorem C.27, gives that there exists a finite real-valued signed Borel-measure μ on $\mathbb{R}^d$ such that

$$\varphi(h) = \int_{\mathbb{R}^d} h(x)\, d\mu(x) \quad \forall h \in \mathcal{C}_0(\mathbb{R}^d, \mathbb{R}).$$

Note the assumption holds since $\mathbb{R}^d$ is a locally compact Haussdorf space. Moreover, since μ is signed, Jordan Decomposition, see Theorem C.28, gives that there exist finite positive Borel measures μ_+ and μ_- such that

$$\mu = \mu_+ - \mu_-.$$

Now let $h \in \mathcal{C}_0(\mathbb{R}^d, \mathbb{R})$ be arbitrary, and define $f_c : \mathbb{R}^d \to \mathbb{R}$, for $c > 0$ given by

$$x \mapsto c^d f(cx).$$

Let T_x be the translation operator given by $T_x f(\dot{}) := f(\cdot - x)$. Then it holds that $T_x f_c(y) = f_c(y - x)$. Thus $T_x f_c \in \overline{\mathcal{V}}$, for all $x \in \mathbb{R}^d$, per the preliminary part of the proof. Therefore $\varphi(T_x f_c) = 0$ for all $x \in \mathbb{R}^d$. It is desired to use Fubini's theorem on an integral

$$\int_{\mathbb{R}^d} (h * f_c)(x)\, d\mu(x).$$

To do so, it is applied on each of the integrals

$$\int_{\mathbb{R}^d} (h * f_c)(x)\, d\mu_\pm(x).$$

This is justified, since

$$\begin{aligned}\int_{\mathbb{R}^d} \int_{\mathbb{R}^d} |h(z) f_c(x - z)|\, dz\, d\mu_\pm(x) &\leq \mu_\pm(\mathbb{R}^d) \|h\|_{L^\infty(\mathbb{R}^d, \mathbb{R})} \|T_z f_c\|_{L^1(\mathbb{R}^d, \mathbb{R})} \\ &= \mu_\pm(\mathbb{R}^d) \|h\|_{L^\infty(\mathbb{R}^d, \mathbb{R})} \|f_c\|_{L^1(\mathbb{R}^d, \mathbb{R})} \\ &< \infty.\end{aligned}$$

Therefore Fubini's theorem and a change of variables with $y = -z$ gives that

$$\begin{aligned}\int_{\mathbb{R}^d} (h * f_c)(x)\, d\mu(x) &= \int_{\mathbb{R}^d}\int_{\mathbb{R}^d} h(z) f_c(x-z)\, dz\, d\mu(x) \\ &= \int_{\mathbb{R}^d} h(-y) \int_{\mathbb{R}^d} f_c(y+x)\, d\mu(x)\, dy \\ &= \int_{\mathbb{R}^d} h(-y)\varphi(T_{-y} f_c)\, dy \\ &= 0\end{aligned}$$

for all $c > 0$.

Per assumption (i) there exists a $C < \infty$ such that

$$\int_{\mathbb{R}^d} f(x)\, dx = C.$$

Since $h \in \mathcal{C}_0(\mathbb{R}^d, \mathbb{R})$ is bounded and uniformly continuous, Theorem C.29 gives that $h * f_c \to Ch$ as $c \to \infty$. Since μ is a finite measure it holds that

$$\begin{aligned}\varphi(h) &= \int_{\mathbb{R}^d} h(x)\, d\mu(x) \\ &= \lim_{c\to\infty} \frac{1}{C} \int_{\mathbb{R}^d} (h * f_c)(x)\, d\mu(x) \\ &= 0.\end{aligned}$$

Therefore $\varphi \equiv 0$ on $\mathcal{C}_0(\mathbb{R}^d, \mathbb{R})$, which is a contradiction.

Now let $h \in X_p^k(\mathbb{R}^d)$, and define each component function of h by h_n for $n \in \{1, \dots, k\}$, such that $h = (h_1, \dots, h_k)$. Then for each h_n Lemma C.17 gives that there exists a $\tilde{h}_n$ such that

$$h_n = \tilde{h}_n\restriction_{\Omega}.$$

Let $\varepsilon > 0$ be arbitrary and define $p_0 := \min\{1, p\}$. Since $\mathcal{V}$ is dense in $X_p^1(\mathbb{R}^d)$, there exist functions $\tilde{f}_n$ for each $n \in \{1, \dots, k\}$ such that

$$\left\|\tilde{h}_n - \tilde{f}_n\right\|_{X_p^1(\Omega)}^{p_0} \le \frac{\varepsilon^{p_0}}{k}.$$

By defining

$$\tilde{f} := \left(\tilde{f}_1\restriction_{\Omega}, \dots, \tilde{f}_k\restriction_{\Omega}\right)$$

it holds that

$$\tilde{f} \in \mathtt{NN}_{\infty, L, \infty}^{\varrho, d, k}(\Omega) \cap X_p^k(\mathbb{R}^d) = \Sigma_\infty\left(X_p^k(\Omega), \varrho, \mathscr{L}\right)$$

according to Lemma II.16, where

$$\left\|h - \tilde{f}\right\|_{X_p^k(\Omega)}^{p_0} = \sum_{n=1}^{k} \left\|\tilde{h}_n - \tilde{f}_n\right\|_{X_p^1(\Omega)}^{p_0} \le \varepsilon^{p_0}.$$

Hence $\left\|h - \tilde{f}\right\|_{X_p^k(\Omega)} \le \varepsilon$. Therefore $\Sigma_\infty\left(X_p^k(\Omega), \varrho, \mathscr{L}\right)$ is dense in $X_p^k(\Omega)$, since $\varepsilon > 0$ and $h \in X_p^k(\mathbb{R}^d)$ was arbitrary. ■

Note the integral in (ii) in Theorem VI.25 is well-defined, since

$$\int_{\mathbb{R}^d} |f(x)|\, dx \leq \int_{\mathbb{R}^d} g(\|x\|_{\mathbb{R}^d})\, dx < \infty.$$

With (P1)-(P5) satisfied, Proposition VI.7 is now used to prove that the approximation classes associated with the neural networks are actual approximation spaces:

Theorem VI.26: Approximation Spaces of Generalized Neural Networks

Let $d \in \mathbb{N}$, $p \in (0, \infty]$ and let $\Omega \subseteq \mathbb{R}^d$ be an admissible domain. Assume that at least one of the following properties holds:

(i) The activation function is continuous and not a polynomial, $\mathcal{L} \geq 2$, and Ω is bounded.

(ii) The space

$$\mathtt{NN}^{\varrho, d, 1}_{\infty, \mathcal{L}, \infty}(\Omega) \cap X^1_p(\Omega)$$

contains some compactly supported, bounded, and positive f.

Then, for every $k \in \mathbb{N}$, $\alpha > 0$, $q \in (0, \infty]$, and with $\mathcal{X} := X^k_p(\Omega)$ it hold that properties (P1)-(P5) are satisfied for

$$\Sigma_n := \mathtt{W}_n(\mathcal{X}, \varrho, \mathscr{L}) \quad \wedge \quad \Sigma_n := \mathtt{N}_n(\mathcal{X}, \varrho, \mathscr{L}),$$

and the classes

$$\left(W^\alpha_q(\mathcal{X}, \varrho, \mathscr{L}), \|\cdot\|_{W^\alpha_q(\mathcal{X}, \varrho, \mathscr{L})}\right),$$
$$\left(N^\alpha_q(\mathcal{X}, \varrho, \mathscr{L}), \|\cdot\|_{N^\alpha_q(\mathcal{X}, \varrho, \mathscr{L})}\right)$$

are quasi-Banach spaces.

Proof

No matter which of the assumptions there is made between (i) and (ii) Lemma VI.21 gives that (P1)-(P4) is satisfied. The setup is divided into two cases: When (i) is assumed and when (ii) is assumed. For (i) Corollary VI.24 gives that (P5) is satisfied. For (ii) Lemma C.30 gives that there exists a μ such that the assumptions in Theorem VI.25 are satisfied. Therefore (P5) is satisfied. Since (P1)-(P5) are satisfied, Proposition VI.7 gives that the spaces are quasi-Banach spaces. ■

The theorem gives that for $\mathcal{X} = X^k_p(\Omega)$ and for an appropriately defined family Σ properties (P1)-(P5) holds, and the constructed approximation classes for neural networks are approximation spaces.

VI.5 Approximation Spaces of ReLU-networks

The approximation spaces presented in the previous section are now considered in the setup with specific activation functions, namely ReLU and its powers. Similar to the previous section it should be verified that the approximation spaces constructed using ReLU and its powers are well-defined. The approximation classes, for both generalized and strict ReLU-networks are explored in the following:

Theorem VI.27: Approximation Spaces of Generalized & Strict ϱ_r-networks
Let $d, k \in \mathbb{N}$, $p \in (0, \infty]$, let $\Omega \subseteq \mathbb{R}^d$ be an admissible domain, and let $\mathcal{X} := X_p^k(\Omega)$. Then the following holds:

(i) For all $\alpha > 0$, $q \in (0, \infty]$, and $r \in \mathbb{N}$, it holds that

$$W_q^\alpha(\mathcal{X}, \varrho_r, \mathscr{L}) = SW_q^\alpha(\mathcal{X}, \varrho_r, \mathscr{L}),$$
$$N_q^\alpha(\mathcal{X}, \varrho_r, \mathscr{L}) = SN_q^\alpha(\mathcal{X}, \varrho_r, \mathscr{L}),$$

and there exist $C_W, C_N < \infty$ such that

$$\|\cdot\|_{W_q^\alpha(\mathcal{X}, \varrho_r, \mathscr{L})} \leq \|\cdot\|_{SW_q^\alpha(\mathcal{X}, \varrho_r, \mathscr{L})} \leq C_W \|\cdot\|_{W_q^\alpha(\mathcal{X}, \varrho_r, \mathscr{L})},$$
$$\|\cdot\|_{N_q^\alpha(\mathcal{X}, \varrho_r, \mathscr{L})} \leq \|\cdot\|_{SN_q^\alpha(\mathcal{X}, \varrho_r, \mathscr{L})} \leq C_N \|\cdot\|_{N_q^\alpha(\mathcal{X}, \varrho_r, \mathscr{L})}.$$

(ii) If $\mathscr{L}$ satisfies that

$$\mathcal{L} \geq \begin{cases} 2 & \text{if } \Omega \text{ is bounded,} \\ 2 & \text{if } d = 1, \\ 3 & \text{otherwise,} \end{cases}$$

then for all $\alpha > 0$, $q \in (0, \infty]$, and $r \in \mathbb{N}$, it holds that properties (P1)-(P5) are satisfied for

$$\Sigma_n := \mathtt{W}_n(\mathcal{X}, \varrho_r, \mathscr{L}) \quad \wedge \quad \Sigma_n := \mathtt{N}_n(\mathcal{X}, \varrho_r, \mathscr{L}),$$

and the classes

$$\left(W_q^\alpha(\mathcal{X}, \varrho_r, \mathscr{L}), \|\cdot\|_{W_q^\alpha(\mathcal{X}, \varrho_r, \mathscr{L})}\right),$$
$$\left(N_q^\alpha(\mathcal{X}, \varrho_r, \mathscr{L}), \|\cdot\|_{N_q^\alpha(\mathcal{X}, \varrho_r, \mathscr{L})}\right)$$

are quasi-Banach spaces.

Proof

(i) Firstly note that ϱ_r can represent the identity using $2r + 2$ terms, according to Lemma III.7. Thus assumption (i) is satisfied in Theorem VI.15, which implies the claim.

(ii) Assume Ω is bounded. Per the assumption in (ii) it holds that $\mathcal{L} \geq 2$. Thus assumption Theorem VI.26(i) is satisfied, which implies that the desired holds.

Now, assume $d = 1$. Since ϱ_r is continuous and not a polynomial, Proposition C.18 gives that ϱ_r is non-degenerated. Then Lemma IV.9 and Lemma IV.8 imply that there exists a compactly supported, continuous, non-negative function $g \neq 0$ such that

$$g \in \mathtt{NN}^{\varrho_r, d, 1}_{\infty, 2, \infty} \cap X^1_p(\mathbb{R}^d).$$

Next, assume Ω is unbounded and $d > 1$. In this case the existence of a

$$g \in \mathtt{NN}^{\varrho_r, d, 1}_{\infty, 3, \infty} \cap X^1_p(\mathbb{R}^d),$$

with the same properties follows from analog results. In both cases using Theorem VI.26 gives the desired result. ■

The result states that strict and generalized ReLU-networks yield identical approximation classes for any admissible domain $\Omega \subseteq \mathbb{R}^d$. This holds even for unbounded domains. Moreover, the theorem gives that the approximation classes of ReLU-networks are well-defined quasi-Banach spaces, as soon as $\mathscr{L}$ satisfies the assumption in (ii).

VII | Additional Discussion Points

The main focus of this report is to give an introduction to a general framework for studying approximation properties of deep neural networks from an approximation space viewpoint. This framework enables the possibility of transferring various results from approximation theory to deep neural networks. Since this approach is still a rather new field of research, there are a lot of unexplored ideas. This both concerns theoretical areas that have not yet been covered and the expected impact on other areas. This chapter provides insight into some of these points, as well as already covered problems related to the main focus.

VII.1 ReLU Activation Function

The choice of activation function influences the approximation properties of a neural network, and therefore the approximation spaces. However, as described in [12, p. 292 around Theorem 4.1] the activation function can be too powerful. This is highlighted by considering an activation function where the corresponding approximation space collapses to the whole space, generating an uninteresting problem from a mathematical perspective. Moreover, from an applied point of view, the described activation function renders it hopeless to compute a near-best approximation. Such an activation function is too complex to implement in practice, which again would constitute an approximation problem. Thus, the activation function should be chosen carefully. One approach is to choose an activation function that is commonly used in applications, such as ReLU and its power.

As proven in Theorem VI.27, the approximation classes for ReLU-networks are well-defined, but compared to the example above it is important that the approximation space does not collapse to the whole space. Fortunately, the space fulfills this, since [12, Theorem 4.16, p. 299] proves that the space is non-trivial.

The expressivity of neural networks with more general activation functions can be related to the approximation space constructed for ReLU activation functions. Different examples of activation functions were introduced earlier, but multiple other examples exist. Activation function similar to ReLU is illustrated in Figure VII.1.

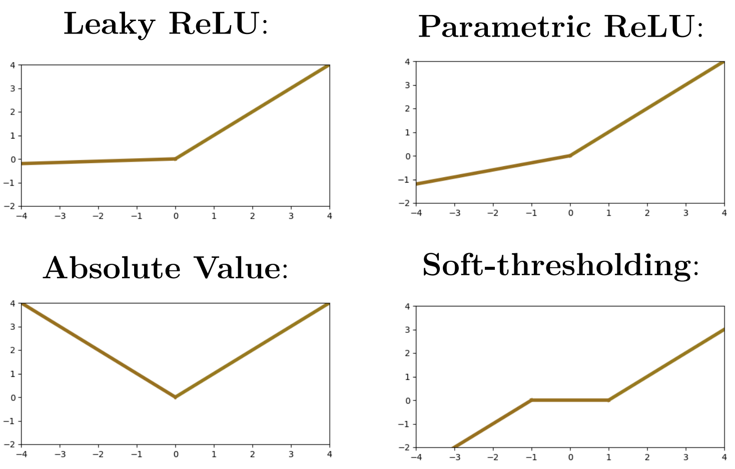

Figure VII.1: *Illustration of different activation functions.*

These examples in Figure VII.1 all have in common that the approximation space with the specific activation function is identical to the space associated with ReLU activation functions, [12, Theorem 4.7, p. 295]. Furthermore, this result proves that if ϱ is continuous and piecewise polynomial of degree at most r, then its approximation spaces are contained in the approximation spaces associated with ϱ_r. Similar by comparing different ReLU activation functions, [12, Corollary 4.14, p.298] yields the connection illustrated in Figure VII.2, for $r \geq 2$.

$$\begin{array}{ccccc} W_q^\alpha(\mathcal{X}, \varrho_1, \mathscr{L}) & \subseteq & W_q^\alpha(\mathcal{X}, \varrho_2, \mathscr{L}) & = & W_q^\alpha(\mathcal{X}, \varrho_r, \mathscr{L}) \\ \rotatebox{90}{$\supseteq$} & & \rotatebox{90}{$\supseteq$} & & \\ N_q^\alpha(\mathcal{X}, \varrho_1, \mathscr{L}) & \subseteq & N_q^\alpha(\mathcal{X}, \varrho_2, \mathscr{L}) & = & W_q^\alpha(\mathcal{X}, \varrho_r, \mathscr{L}) \end{array}$$

Figure VII.2: *Illustration of the relations between the approximation classes, with $r \geq 2$.*

Most of the results considered are for ReLU or activation functions related to ReLU. However, approximation spaces using other activation functions could also be interesting to study.

VII.2 Embeddings with Besov Spaces

To give some insight into the impact of depth on the expressivity of neural networks, [12] connects the approximation spaces associated with ReLU to classical function spaces, called Besov spaces. Besov spaces are denoted by $\mathcal{B}^s_{p,q}(\Omega) := \mathcal{B}^s_q(L_p(\Omega, \mathbb{R}))$, where $s \in (0, \infty)$, $p, q \in (0, \infty]$, and where $\Omega \subseteq \mathbb{R}^d$ is a so-called Lipschitz domain. Functions in these spaces are said to have smoothness of order s in $L^p(\Omega, \mathbb{R})$ with q indicating a finer gradation of this smoothness, [8, p. 371]. General Besov spaces intersect with $\mathcal{C}^3_c(\Omega, \mathbb{R})$ non-trivially for open admissible domains $\Omega \subseteq \mathbb{R}^d$, [12]. Thus it is interesting to consider the intersection of $\mathcal{C}^3_c(\Omega, \mathbb{R})$ and the approximation spaces associated with ReLU. With this focus [12, Theorem 4.17, p. 299] shows that

$$N_q^\alpha(\mathcal{X}, \varrho_1, \mathscr{L}) \cap \mathcal{C}^3_c(\Omega, \mathbb{R}) = \{0\}$$

if $\alpha > 2(L-1)$, and

$$W_q^\alpha(\mathcal{X}, \varrho_1, \mathscr{L}) \cap \mathcal{C}^3_c(\Omega, \mathbb{R}) = \{0\}$$

if $\alpha > 2\lfloor L/2 \rfloor$. To have any hope that the intersection is non-trivial, it is therefore crucial to assume that $\alpha \leq 2(L-1)$, $\alpha \leq 2\lfloor L/2 \rfloor$ for $N_q^\alpha(\mathcal{X}, \varrho_1, \mathscr{L})$ and $W_q^\alpha(\mathcal{X}, \varrho_1, \mathscr{L})$ respectively. These requirements give some insight into the impact of the depth on the expressivity of the neural network.

Focusing on the relation to Besov spaces, [12] consider embeddings from Besov spaces into the approximation spaces associated with ReLU. Likewise, they consider embeddings from the approximation spaces into Besov spaces. The established embeddings imply that functions with very low Besov smoothness and specific properties can be well approximated by neural networks, given the neural networks are sufficiently deep. In particular, given specific assumptions on the domain $\Omega \subseteq \mathbb{R}^d$, the values $r \geq 2$, and $\mathcal{L} \geq 2 + 2\lceil \log_2(d) \rceil$, [12, Theorem 5.5, p. 305-306] gives that

$$\mathcal{B}^{d\alpha}_{p,q}(\Omega) \hookrightarrow W_q^\alpha(\mathcal{X}_p(\Omega), \varrho_r, \mathscr{L})$$

for all $p, q \in (0, \infty]$ and α such that

$$0 < \alpha < \frac{r + \min\{1, p^{-1}\}}{d}.$$

Considering these restrictions it holds that for large input dimension d, the condition for $\mathcal{L}$ is only satisfied for quite deep neural networks. Moreover, the Besov smoothness α becomes quite small as d increases.

[12, Theorem 5.7, p. 306] contains limits on the embeddings from the approximation space into Besov space as follows: If $\alpha < (L-1)\min\{s, 2\}$, then

$$N_q^\alpha(\mathcal{X}, \varrho_1, \mathscr{L}) \not\hookrightarrow \mathcal{B}_{\sigma,\tau}^s(\Omega).$$

Similar if $\alpha < 2\lfloor L/2 \rfloor \min\{s, 2\}$, then

$$W_q^\alpha(\mathcal{X}, \varrho_1, \mathscr{L}) \not\hookrightarrow \mathcal{B}_{\sigma,\tau}^s(\Omega).$$

A consequence of this result is that for unbounded depth, none of the spaces $N_q^\alpha(\mathcal{X}, \varrho_1, \mathscr{L})$, $W_q^\alpha(\mathcal{X}, \varrho_1, \mathscr{L})$ can be embedded into any Besov spaces of strictly positive smoothness $s > 0$, [12, p. 306].

With a focus on input dimension 1 and $\mathcal{X} = L^p((0,1), \mathbb{R})$, an embedding from the approximation space into a Besov space is obtained in [12, Theorem 5.13, p. 308].

VII.3 B-spline Approximation

As proven in Chapter IV ReLU-networks can approximate B-splines arbitrarily well under specific assumptions. Therefore it is convenient to further look at functions built on B-splines. By dilating and translating multivariate B-splines one can consider

$$\beta_{j,k,d}^n := 2^{jd/2} \beta_d^{(n)}(2^j x - k),$$

for $j \in \mathbb{Z}, k \in \mathbb{Z}^d$ which together define well-studied wavelet-type system. Per Lemma II.15 and Lemma III.9, the constructed families for the realization of neural networks are closed under affine linear maps and multiplication, when $n \geq 2$. Thus these types of functions can be realized from a neural network as well. This result with all its conditions can be found in [12, Lemma 5.3, p. 304]. Using the approximation spaces constructed in Section VI.1 for these wavelet-types systems and comparing them with the approximation spaces associated with ReLU constructed in Section VI.2, embeddings are constructed between them, in [12, Corollary 5.4, p. 305]. These embeddings bridge the gap to the Besov spaces mentioned in the previous section.

Even though the traditional focus in approximation theory is the approximation of target functions, this thesis considers the approximation of approximants, such as B-splines. This approach allows for the use of knowledge regarding the approximation space of B-splines in connection with neural networks. The result is a theoretical bridge that ensures the inclusion of the approximation space of B-splines in the approximation space of neural networks. Thus the neural networks can theoretically approximate just as well or better than B-splines. Notice, however, that this neither proves nor disproves that neural networks can do better than B-splines. Getting this clarification will be of significant interest. The goal is to prove that the approximation space of neural networks covers much more than other approximation spaces of other classical approximants, such as B-splines, and that the approximation is more efficient.

VII.4 Novel Approximation Classes

Focusing on the spaces $N_q^\alpha(\mathcal{X}, \varrho_n, \mathscr{L})$, and $W_q^\alpha(\mathcal{X}, \varrho_n, \mathscr{L})$ they are constructed in a way that makes them easy to adapt to another setup. This gives some flexibility when working with them. If the focus is on convolutional neural networks, the weights can be restricted to convolutional weights. However, a full understanding of the classes is not yet reached, [8, p. 389]. A finer characterization of the classes is still of significant interest, such as $N_q^\alpha(\mathcal{X}, \varrho_n, \mathscr{L})$, and $W_q^\alpha(\mathcal{X}, \varrho_n, \mathscr{L})$, where $n \in \{1, 2\}$ and with some assumption on $\mathscr{L}$. Another point of interest could be to construct more precise upper bounds for the parameters and possibly optimal upper bounds. Similarly, the results are derived with no constraint on the neural network parameters. A study of neural networks with restrictions and, or relations applied to the weights using the approximation space framework could be of interest. This could for instance be the above example with convolutional weights. Another example is the so-called ResNet architecture, which has done remarkable work and has empirically shown that it is easier to train deep neural networks when using the ResNet structure.

Overall, it makes sense to get a better understanding of the corresponding approximation class. Even though, the understanding of these classes is far from complete, the result in this report provides some useful insight on the structure. Additionally, understanding the role of depth growth function, in general, is still an open question. In this report, the primary result considers the depth L in terms of the associated approximation spaces for ReLU-networks, for both finite and infinite L.

Generally, methods of approximation are divided into linear and nonlinear approximation. They are defined in the same way as for neural networks, by constructing sets $\Sigma = \{\Sigma_n\}$. They are called *linear* if the sets Σ_n is a linear space of dimension n, and *nonlinear* otherwise. Examples of classical linear approximation are polynomials, splines, and wavelets, and examples of nonlinear approximation are piecewise polynomials, n-term approximation, and neural network approximation, [8, p. 375-376]. Compared to the constructed family Σ in this report, this also makes sense, due to (P4), which shows that they are not linear spaces.

Approximation classes for many approximation methods in general, both linear and nonlinear, are characterized. For instance, the approximation spaces for all classical linear methods of approximation are characterized for all $\alpha \in (0, \infty)$, when $\mathcal{X} = L^p([0,1], \mathbb{R})$ with $p \in [0, \infty)$ or when $\mathcal{X} = \mathcal{C}([0,1], \mathbb{R})$, [10]. A general observation is that the nonlinear approximation spaces tend to be significantly larger compared to those in the linear case. However, a precise characterization of these approximation classes is beyond the current understanding of approximation using neural networks. As noted, the current results, including the ones presented in this report, only capture some structural elements of these classes.

VII.5 Model Classes

The approach of this report considers how well the neural network can approximate a target function. Another important approach is to consider how well the target functions can be approximated by neural networks. This is in focus in [8] by considering model classes instead of approximation classes. Similar to the argumentation in this report [8] restricts the focus on ReLU-1-networks. They show that by increasing the depth the class of realization increases. Moreover, they compare the approximation efficiency of neural networks to other classical methods in approximation theory. From an applied point of view, this is of interest, as in Chapter V. They introduce tools to evaluate a method of approximation, making it easier to compare. This includes among other things approximation rates on model classes and metric entropy. Additionally, they comment that approximation classes are another tool to compare with, as done in this report, [8, p. 389]. Similarly, [6] takes the same approach by comparing approximation using ReLU to other classical approximation methods but using the same approach as in this report namely approximation classes. Similarly to [8], [6] also focuses on the impact of the depth. For instance, they present the advantages of the deep neural network versus more shallow neural networks.

The model classes denoted by K contain information about the desired functions f. When focusing on numerical solutions to partial differential equations K is typically provided by a regularity theorem. Similarly, if the focus is signal processing, K consists of the information about the underlying signal, such as band limits or sparsity, [8, p. 377]. In contrast to the error of best approximation, [8] use the *class error* given by

$$\mathscr{E}(K, \Sigma)_{\mathcal{X}} \coloneqq \sup_{f \in K} \mathscr{E}(f, \Sigma)_{\mathcal{X}},$$

which is the worst-case error on K. One could also define the error by averaging. This is useful when considering stochastic processes with some underlying probability measure. How good the approximation methods are, dependent on how fast the error $\mathscr{E}(K, \Sigma)_{\mathcal{X}}$ tends to zero. This concept is also used to compare two approximation methods. Note that the error of best approximation gives the smallest error one can achieve using Σ_n. However, it does not address the question of how to find such a best approximation.

VII.6 Approximation using Data

As noted in Chapter I there is typically only training data available when approximating a target function in applications. A classical example of this is within supervised learning. From this, the theoretical point of view does not match the task in application. Such training data alone do not allow any information about how well the target function can be recovered or how well an approximant can approximate the target function. What is needed for the latter is additional information about the target function. This information is denoted *model class information* and is an assumption typically based on the research field, [8, p. 426]. Moreover, the performance guarantees are necessarily worse than for approximation where there is full access to the target function. Overall, the application of approximation using data is a vast area, and examples of these are within image processing, statistical estimation, regularity for PDEs, and adaptive algorithms, [9, p. 51]. With the focus on the connection between the theoretical and application point of view, [8] highlights open issues. For instance, how to numerically impose stability in parameter selection, and how the imposition of stability limits the performance of the neural network.

Conclusion

As shown in this thesis the expressivity and the structure of deep neural networks can be explored under the mathematical framework of approximation theory. Sets of realizations of neural networks were presented as a tool to construct the associated approximation spaces. With this goal, several results describing neural networks and the sets of realizations were established. This included the complexity of neural networks and elementary properties, such as linear combination and composition, of the sets of realizations. Additionally, the activation function ReLU and its power were presented, along with some of their properties.

Through decompositions using ReLU activation functions, it was shown that ReLU-networks can realize B-splines. This allowed for the construction of an experiment meant to demonstrate these results in practice. The idea was to create an approximation method using neural networks which were equivalent to approximation methods using B-splines. Both methods were then compared on generated target functions using different parameters in the methods.

The result showed similar performance between the approximation methods using the constructed neural network and the fully connected neural network, but B-spline approximation outperformed both of them. The experiment discussion does, however, mention a few ways to improve the neural network approach, which might make the neural network approach viable as an alternative to the traditional B-spline approximation. Overall, it is not evident that the constructed approximation method using neural networks, at this point in time, is a better fit than the commonly used B-spline approximation. The experiment served as an indication that the introduced methods could be applicable as an approximation method, based on the result that neural networks can realize B-splines.

The introduced tools, such as the elementary properties of the sets of realization were used in conjunction with approximation theory seeking to characterize approximation spaces of deep neural networks. At first, the spaces were proven to be well-defined. With a focus on ReLU and its power, some characterization of these spaces was explored. Overall, the results reveal how certain choices such as a specific type of activation function or skipping connections, can influence the expressive power of a neural network. Hence, the results provide insight into the potential adaption of neural network complexity without compromising its expressivity, which gives a degree of freedom.

Motivated by the focus area, an introduction to further work was discussed using related articles and open problems. This includes the need for a finer characterization of the approximation spaces than the one presented in the report.

Appendix

A | Remarks & Notations

Remark A.1:

Note that the definition of the neural network commonly used in applications differs from the one presented in this report. Typically the neural network is defined as the realization. However, given a function f, there is no guarantee that a neural network Φ with $\mathtt{R}(\Phi) = f$ is unique. This is the reason for separating the network from the function, as the number of layers, neurons, or weights of the function f is not well-defined. Defining

$$L(f) \coloneqq \min\{L(\Phi) \mid \Phi \text{ neural network with } \mathtt{R}(\Phi) = f\}$$

and similarly for $N(f)$ and $W(f)$ could be an attempt to rectify this problem. However, this makes the setup less transparent and conceals whether the three values originate from the same neural network.

Additionally, the definition used in most applications assumes the same activation function for all neurons in a common layer. By further assuming the activation function is the same across all layers, except for the last one, this corresponds to the strict neural networks. Therefore, as the name indicates, the generalized neural network is a more generalized definition than the one normally used.

The reason for generalizing the definition compared to most applications is founded on essential properties. In Section II.2, it is proven that classes constructed from neural networks admit appealing closure properties under linear combinations and compositions of functions, resulting in a convenient calculus of neural networks. By using the definition used in most applications or the definition for strict neural networks, these do not hold.

Instead of using the identity in the last entry in the tuple, one could define a α_ℓ for $\ell = L$ by the identity, as in [12]. Nevertheless, for the sake of a clearer distinction from the other instances of α_ℓ, this particular choice is not adopted in this report. Moreover, the definition of the neural network mostly used in applications ends with an affine-linear map. Preserving this feature, could either be achieved by letting the last layer differ from the rest by only consisting of the affine linear map or by defining $\alpha_L = \mathrm{id}_{\mathbb{R}^{N_L}}$. The last option is chosen since it ensures that all layers have the same structure, which contributes to the closure properties for the classes constructed from neural networks.

The definition of the realization of a neural network again argues for the choices made regarding the definition of neural networks, as it contributes to the closure properties for the classes constructed in Definition II.8.

Remark A.2:

The properties (P1)-(P5) plus an additional (P6) property are used in general theory regarding the approximation classes of the type presented in Lemma VI.5. However, (P1)-(P5) is enough in this report, so (P6) is left out. The properties as well as some argumentation for the properties can be found in [12, p. 278] and [10, p. 216].

Remark A.3:

Note that $\| \cdot \|_{A_q^\alpha(\mathcal{X},\Sigma)}$ is not a proper norm. Even if $\mathcal{X}$ is a Banach space and $q \in [1, \infty]$ it is still only a quasi-norm. Only if additionally all the sets Σ_n are vector spaces, $\| \cdot \|_{A_q^\alpha(\mathcal{X},\Sigma)}$ is a norm, see [12, p. 279]. This is the case if $c = 1$ in (P4).

Definition A.4:

Let $d, k \in \mathbb{N}$. Let $A \in \mathbb{R}^{d\times k}$, $n \in \{1, \dots, d\}$ and $m \in \{1, \dots, k\}$. Then

(i) $A^T \in \mathbb{R}^{k\times d}$ is the transpose of A.

(ii) $A_{n,-} \in \mathbb{R}^{1\times k}$, is the n-th row of A.

(iii) $A_{-,m} \in \mathbb{R}^{d\times 1}$, is the m-th column of A.

(iv) $A_{(n)} \in \mathbb{R}^{(d-1)\times k}$ is A with the n-th row deleted.

(v) $A_{[m]} \in \mathbb{R}^{d\times (k-1)}$ is A with the m-th column deleted.

Example A.5:

Assume ϱ is a polynomial and let $\mathscr{L}$ be bounded. Consider the setup for $\mathcal{X} := X_p^k(\Omega)$ with $p \in (0, \infty]$. Per construction of the approximation classes $\mathtt{W}_n(\mathcal{X}, \varrho, \mathscr{L})$ and $\mathtt{N}_n(\mathcal{X}, \varrho, \mathscr{L})$, it holds that element in the classes is a realization of a ϱ-networks. Since the composition of a polynomial with an affine linear map still is a polynomial with a similarly bounded degree, the realizations can be considered as a composition only of polynomials. When taking the composition of two polynomials with degree m the result is a polynomial where the degree m^2. Since $\mathscr{L}$ is bounded the realizations are the composition of finitely many polynomials of bounded degree, which is a polynomial of bounded degree. Therefore $\Sigma_\infty(\mathcal{X}, \varrho, \mathscr{L})$ only contains polynomials of bounded degree, and the space is finite-dimensional. However, the space $\mathcal{X}$ is infinite-dimensional, so $\Sigma_\infty(\mathcal{X}, \varrho, \mathscr{L})$ is not dense in $\mathcal{X}$ for non-trivial Ω.

B | Additional Proofs

This appendix is a supplement with the proofs omitted in the report.

Proposition II.9
Let Φ be a neural network. Then

$$L(\Phi) \le 1 + N(\Phi), \tag{II.1}$$
$$W(\Phi) \le \big(N(\Phi) + d_{in}(\Phi)\big)\big(N(\Phi) + d_{out}(\Phi)\big). \tag{II.2}$$

If $L(\Phi) = 2$, then $W(\Phi) \le (d_{in}(\Phi) + d_{out}(\Phi))N(\Phi)$.

Proof
By definition $N_\ell \ge 1$ for all $1 \le \ell \le L-1$, which implies

$$L(\Phi) = 1 + \sum_{\ell=1}^{L-1} 1 \le 1 + \sum_{\ell=1}^{L-1} N_\ell = 1 + N(\Phi),$$

proving (II.1). Since $A^{(\ell)} \in \mathbb{R}^{N_{\ell-1}\times N_\ell}$, it holds that $\|A^{(\ell)}\|_{\ell^0} \le N_{\ell-1}N_\ell$ for all $1 \le \ell < L$. Therefore

$$\begin{aligned}
W(\Phi) &= \sum_{\ell=1}^{L} \left\|A^{(\ell)}\right\|_{\ell^0} \\
&\le \sum_{\ell=1}^{L} N_{\ell-1}N_\ell \\
&\le \sum_{\ell_1=0}^{L-1}\sum_{\ell_2=1}^{L} N_{\ell_1}N_{\ell_2} \\
&= \left(\sum_{\ell_1=0}^{L-1} N_{\ell_1}\right)\left(\sum_{\ell_2=1}^{L} N_{\ell_2}\right) \\
&= \left(N_0 + \sum_{\ell_1=1}^{L-1} N_{\ell_1}\right)\left(N_L + \sum_{\ell_2=1}^{L-1} N_{\ell_2}\right) \\
&= \big(N(\Phi) + d_{in}(\Phi)\big)\big(N(\Phi) + d_{out}(\Phi)\big),
\end{aligned}$$

which proves (II.2). Now assume $L = 2$. Then it holds that

$$\begin{aligned}
W(\Phi) &= \|T_1\|_{\ell^0} + \|T_2\|_{\ell^0} \\
&\le N_0N_1 + N_1N_2 \\
&= (N_0 + N_2)N_1 \\
&= (d_{in}(\Phi) + d_{out}(\Phi))N(\Phi),
\end{aligned}$$

which completes the proof. ■

Proposition II.10
Let Φ be a neural network with $d_{out}(\Phi) = k$. If $W(\Phi) < L(\Phi)$, then $\mathtt{R}(\Phi) = c \in \mathbb{R}^k$.

Proof
First notice that

$$\sum_{\ell=1}^{L} \|T_\ell\|_{\ell^0} = W(\Phi) < L(\Phi) = L.$$

Since $\|T_\ell\|_{\ell^0} \geq 0$ and the sum has L terms there exists some $1 \leq \ell \leq L$ such that $\|T_\ell\|_{\ell^0} = 0$, which implies that T_ℓ is a constant map. Therefore $\mathtt{R}(\Phi)$ is constant as well, since

$$\mathtt{R}(\Phi) = T_L \circ \alpha_{L-1} \circ T_{L-1} \circ \cdots \circ \alpha_\ell \circ T_\ell \circ \cdots \circ \alpha_1 \circ T_1.$$

■

Proposition II.11
Let $\varrho : \mathbb{R} \to \mathbb{R}$ be an activation function, let $d, k \in \mathbb{N}$, $W, N \in \mathbb{N}_0 \cup \{\infty\}$, and let $L \in \mathbb{N} \cup \{\infty\}$. Then

$$\left\{ f : \mathbb{R}^d \to \mathbb{R}^k \;\middle|\; \exists c \in \mathbb{R}^k : f = c \right\} \subseteq \mathtt{SNN}^{\varrho,\,d,\,k}_{W,\,L,\,N} \subseteq \mathtt{NN}^{\varrho,\,d,\,k}_{W,\,L,\,N}.$$

Proof
Assume $f = c \in \mathbb{R}^k$. Then any neural network Φ with $T_L = c$ satisfies $\mathtt{R}(\Phi) = c = f$. This is the case both if Φ is strict or not. The inclusion $\mathtt{SNN}^{\varrho,\,d,\,k}_{W,\,L,\,N} \subseteq \mathtt{NN}^{\varrho,\,d,\,k}_{W,\,L,\,N}$ follow directly from the definition of the sets. ■

Lemma II.12
Let $\varrho : \mathbb{R} \to \mathbb{R}$ be an activation function, let $d, k \in \mathbb{N}$, $N \in \mathbb{N}_0 \cup \{\infty\}$, and let $L \in \mathbb{N} \cup \{\infty\}$. Then

$$\mathtt{NN}^{\varrho,\,d,\,k}_{0,\,L,\,N} = \mathtt{SNN}^{\varrho,\,d,\,k}_{0,\,L,\,N} = \left\{ f : \mathbb{R}^d \to \mathbb{R}^k \;\middle|\; \exists c \in \mathbb{R}^k : f = c \right\}.$$

Proof
Because of Proposition II.11 with $W = 0$ it is enough to show that

$$\mathtt{NN}^{\varrho,\,d,\,k}_{0,\,L,\,N} \subseteq \left\{ f : \mathbb{R}^d \to \mathbb{R}^k \;\middle|\; \exists c \in \mathbb{R}^k : f = c \right\}.$$

Let $f \in \mathtt{NN}^{\varrho,\,d,\,k}_{0,\,L,\,N}$. Then there exists a $\Phi \in \mathcal{NN}^{\varrho,\,d,\,k}_{0,\,L,\,N}$ such that $\mathtt{R}(\Phi) = f$. Therefore $W(\Phi) = 0 < 1 \leq L(\Phi)$ implying that $f = c$ according to Proposition II.10. ■

Lemma III.8
Assume that $\varrho : \mathbb{R} \to \mathbb{R}$ can represent the identity with n terms, let $d, k \in \mathbb{N}$, $W, N \in \mathbb{N}_0$ and let $L \in \mathbb{N} \cup \{\infty\}$ be arbitrary. Then

$$\mathtt{NN}^{\varrho,\,d,\,k}_{W,\,L,\,N} \subseteq \mathtt{SNN}^{\varrho,\,d,\,k}_{n^2W,\,L,\,nN}.$$

Proof
As in Lemma III.5 it is desired to use Proposition C.6. For $d, k \in \mathbb{N}$, define

$$\mathcal{G}_{d,k} := \{f : \mathbb{R}^d \to \mathbb{R}^k\},$$

and let $\mathcal{T}_{d,k}$ be the discrete topology on $\mathcal{G}_{d,k}$. Then $(\mathcal{G}, \mathcal{T})$, where $\mathcal{G} := \{\mathcal{G}_{d,k}\}_{d,k\in\mathbb{N}}$, and $\mathcal{T} := \{\mathcal{T}_{d,k}\}_{d,k\in\mathbb{N}}$ is a network compatible topology family, see Definition C.5.

By assumption there exist $a_i, b_i, c_i \in \mathbb{R}$ for $i \in \{i, \dots, n\}$ and some $c \in \mathbb{R}$ such that

$$x = c + \sum_{i=1}^{n} a_i \varrho(b_i x + c_i), \quad x \in \mathbb{R}.$$

By defining

$$\begin{aligned} E_m &: \mathbb{R} \to \mathbb{R}^n, && x \mapsto (b_1 x + c_1, \dots, b_n x + c_n), \\ D_m &: \mathbb{R}^n \to \mathbb{R}, && x \mapsto c + \sum_{i=1}^{n} a_i x_i, \end{aligned}$$

it follows by the assumption that

$$D_m \circ (\varrho \otimes \cdots \otimes \varrho) \circ E_m = \mathrm{id}_{\mathbb{R}} \quad \forall\, m \in \mathbb{N}.$$

Thus, all the assumptions of Proposition C.6 are satisfied, which gives for all $d, k \in \mathbb{N}$, $W, N \in \mathbb{N}_0 \cup \{\infty\}$, and $L \in \mathbb{N} \cup \{\infty\}$ it holds that

$$\mathtt{NN}^{\varrho,\,d,\,k}_{W,\,L,\,N} \subseteq \overline{\mathtt{SNN}^{\varrho,\,d,\,k}_{n^2W,\,L,\,nN}} = \mathtt{SNN}^{\varrho,\,d,\,k}_{n^2W,\,L,\,nN},$$

where the closure is taken with respect to the discrete topology. Note that the last equality holds, since the closure with respect to the discrete topology does nothing, as all sets are both open and closed in the discrete topology. ■

Proposition IV.2
Let $n \in \mathbb{N}_0$, and let $\beta_+^{(n)}$ be a B-spline of degree n. Then $\beta_+^{(n)}$ is non-negative and $\operatorname{supp}\left(\beta_+^{(n)}\right) \subseteq [0, n+1]$.

Proof
Per definition $\beta_+^{(0)}$ is non-negative and convolutions of non-negative functions are still non-negative.

Proving that supp $\left(\beta_+^{(n)}\right) \subseteq [0, n+1]$ is done by induction. It follows directly from the definition of $\beta_+^{(0)}$ that supp $\left(\beta_+^{(0)}\right) \subseteq [0,1]$. For the induction step assume that it holds for $n = k$. Since $\beta_+^{(k+1)} = \beta_+^{(k)} * \beta_+^{(0)}$, and

$$\operatorname{supp}\left(\beta_+^{(k)}\right) \subseteq [0, k+1], \quad \operatorname{supp}\left(\beta_+^{(0)}\right) \subseteq [0,1]$$

it follows that

$$\beta_+^{(k+1)}(x) = \int_{[0,k+1]\cap(x-[0,1])} \beta_+^{(k)}(y)\beta_+^{(0)}(y-x)\,dy.$$

Now consider any $x_k \notin [0, k+2]$. In this case it holds that $[0, k+1] \cap (x_k - [0,1]) = \emptyset$, which imply that $\beta_+^{(k+1)}(x_k) = 0$. This proves that $\operatorname{supp}(\beta_+^{(k+1)}) \subseteq [0, k+2]$. ■

Proposition VI.14
Let $\alpha > 0$ and $q \in (0, \infty]$. Then it holds that

$$\begin{aligned} \|\cdot\|_{W_q^\alpha(\mathcal{X},\varrho,\mathscr{L})} &\le \|\cdot\|_{SW_q^\alpha(\mathcal{X},\varrho,\mathscr{L})}, \\ \|\cdot\|_{N_q^\alpha(\mathcal{X},\varrho,\mathscr{L})} &\le \|\cdot\|_{SN_q^\alpha(\mathcal{X},\varrho,\mathscr{L})}. \end{aligned}$$

Hence

$$\begin{aligned} SW_q^\alpha(\mathcal{X},\varrho,\mathscr{L}) &\hookrightarrow W_q^\alpha(\mathcal{X},\varrho,\mathscr{L}), \\ SN_q^\alpha(\mathcal{X},\varrho,\mathscr{L}) &\hookrightarrow N_q^\alpha(\mathcal{X},\varrho,\mathscr{L}). \end{aligned}$$

Proof
The proof is stated for the approximation spaces associated with connection complexity since the proof with neuron complexity is analogous. Per construction

$$\mathtt{SW}_n(\mathcal{X},\varrho,\mathscr{L}) \subseteq \mathtt{W}_n(\mathcal{X},\varrho,\mathscr{L}), \quad \forall n \in \mathbb{N}_0.$$

Hence the error of best approximation satisfies that

$$\mathscr{E}(f, \mathtt{W}_n(\mathcal{X},\varrho,\mathscr{L}))_{\mathcal{X}} \le \mathscr{E}(f, \mathtt{SW}_n(\mathcal{X},\varrho,\mathscr{L}))_{\mathcal{X}}, \quad \forall n \in \mathbb{N}_0.$$

This and per construction of the quasi-norm, it holds that

$$\|\cdot\|_{W_q^\alpha(\mathcal{X},\varrho,\mathscr{L})} \le \|\cdot\|_{SW_q^\alpha(\mathcal{X},\varrho,\mathscr{L})}.$$

Therefore

$$SW_q^\alpha(\mathcal{X},\varrho,\mathscr{L}) \subseteq W_q^\alpha(\mathcal{X},\varrho,\mathscr{L}),$$

which shows that $SW_q^\alpha(\mathcal{X},\varrho,\mathscr{L}) \hookrightarrow W_q^\alpha(\mathcal{X},\varrho,\mathscr{L})$. ■

C | Lexicon

This appendix contains definitions and results that are referred to throughout this report. These are stated without proof. The purpose of this is to let the reader view the referenced theory without having to track down the source.

Proposition C.1: [12, p. 286]

Let $\mathcal{L} < \infty$, then $\mathscr{L} \sim \mathcal{L}$.

Lemma C.2: [12, Lemma A.3(i), p. 320]

Let $p, q, r \in \mathbb{N}$. Let $T : \mathbb{R}^p \to \mathbb{R}^q$ and $S : \mathbb{R}^q \to \mathbb{R}^r$ be linear maps. Then

$$\|ST\|_{\ell^0} \leq \|S\|_{\ell^{0,\infty}} \|T\|_{\ell^0},$$
$$\|ST\|_{\ell^0} \leq \|S\|_{\ell^0} \|T\|_{\ell^{0,\infty}_*}.$$

Lemma C.3: [12, Lemma A.4, p. 321-322]

Let $\varrho, \sigma : \mathbb{R} \to \mathbb{R}$ be activation functions where σ is non-constant. Assume that there exists a $\Phi_\sigma \in \mathcal{NN}^{\varrho,1,1}_{w,\ell,n}$ such that $\sigma = \mathtt{R}(\Phi_\sigma)$ with $L(\Phi_\sigma) = \ell \in \mathbb{N}$, $w \in \mathbb{N}_0$, and $n \in \mathbb{N}$. Then, for any $d \in \mathbb{N}$ and $\alpha_i \in \{\sigma, \mathrm{id}_\mathbb{R}\}$, for $1 \leq i \leq d$, there exists a ϱ-network

$$\Phi := \Big((T_1, \beta_1), \dots, (T_{\ell-1}, \beta_{\ell-1}), (T_\ell, \mathrm{id}_{\mathbb{R}^d})\Big) \in \mathcal{NN}^{\varrho,d,d}_{dw,\ell,dn},$$

for which it holds that $\mathtt{R}(\Phi) = \alpha_1 \otimes \cdots \otimes \alpha_d$, and

$$\begin{aligned} \|T_1\|_{\ell^{0,\infty}} &\leq m, & \|T_1\|_{\ell^{0,\infty}_*} &\leq 1, \\ \|T_\ell\|_{\ell^{0,\infty}} &\leq 1, & \|T_\ell\|_{\ell^{0,\infty}_*} &\leq m. \end{aligned}$$

Additionally, if Φ_σ is a strict neural network and, $\alpha_i = \sigma$ for all $i \in \{1, \dots, d\}$, then Φ can be chosen to be strict.

Lemma C.4: [12, Lemma A.7, p. 330]

Let $\{f_n\}_{n=0}^\infty$ and $\{g_n\}_{n=0}^\infty$ be sequences of functions $f_n : \mathbb{R}^d \to \mathbb{R}^k$, and $g_n : \mathbb{R}^k \to \mathbb{R}^\ell$. Assume that f_0, g_0 are continuous and that

$$f_n \xrightarrow[n\to\infty]{} f_0, \quad g_n \xrightarrow[n\to\infty]{} g_0$$

with locally uniform convergence. Then $g_0 \circ f_0$ is continuous, and $g_n \circ f_n \xrightarrow[n\to\infty]{} g_0 \circ f_0$ with locally uniform convergence.

Definition C.5: [12, Definition A.5, p. 326]

Let $d, k \in \mathbb{N}$, and let $X := \{f : \mathbb{R}^d \to R^k\}$. Let $\mathcal{G}_{d,k} \subseteq X$, and a topology $\mathcal{T}_{d,k}$ on X be fixed. Define $\mathcal{G} := \{\mathcal{G}_{d,k}\}_{d,k\in\mathbb{N}}$ and $\mathcal{T} := \{\mathcal{T}_{d,k}\}_{d,k\in\mathbb{N}}$. Then the tuple $(\mathcal{G}, \mathcal{T})$ is called a **network compatible topology family**, if it satisfied the following conditions:

(i) $\{T : \mathbb{R}^d \to \mathbb{R}^k \mid T \text{ affine-linear }\} \subseteq \mathcal{G}_{d,k} \quad \forall\, d, k \in \mathbb{N}$.

(ii) Given a sequence $\{f_m^{(n)}\}_{n\in\mathbb{N}_0}$ of functions $f_m^{(n)} : \mathbb{R} \to \mathbb{R}$ for $m \in \{1, \dots, p\}$, where $p \in \mathbb{N}$, that satisfies $f_m^{(0)} \in \mathcal{G}_{1,1}$ and

$$f_m^{(n)} \xrightarrow[n\to\infty]{\mathcal{T}_{1,1}} f_m^{(0)},$$

it holds that

$$f_1^{(n)} \otimes \cdots \otimes f_p^{(n)} \xrightarrow[n\to\infty]{\mathcal{T}_{1,1}} f_1^{(0)} \otimes \cdots \otimes f_p^{(0)},$$

and $f_1^{(0)} \otimes \cdots \otimes f_p^{(0)} \in \mathcal{G}_{p,p}$.

(iii) If $f_n : \mathbb{R}^d \to \mathbb{R}^k$ and $g_n : \mathbb{R}^k \to \mathbb{R}^\ell$ for all $n \in \mathbb{N}_0$, and if

$$f_0 \in \mathcal{G}_{d,k}, \quad f_n \xrightarrow[n\to\infty]{\mathcal{T}_{d,k}} f_0,$$
$$g_0 \in \mathcal{G}_{k,\ell}, \quad g_n \xrightarrow[n\to\infty]{\mathcal{T}_{k,\ell}} g_0,$$

then $g_0 \circ f_0 \in \mathcal{G}_{d,\ell}$ and

$$g_n \circ f_n \xrightarrow[n\to\infty]{\mathcal{T}_{d,\ell}} g_0 \circ f_0.$$

Proposition C.6: [12, Proposition A.6, p. 326]

Let $\varrho : \mathbb{R} \to \mathbb{R}$ and let $(\mathcal{G}, \mathcal{T})$ be a network compatible topology family satisfying the following:

(i) $\varrho \in \mathcal{G}_{1,1}$.

(ii) There exists a $n \in \mathbb{N}$ such that for all $m \in \mathbb{N}$ there exist affine linear maps $F_m : \mathbb{R} \to \mathbb{R}$, $D_m : \mathbb{R}^n \to \mathbb{R}$, and $E_m : \mathbb{R} \to \mathbb{R}^n$ such that

$$F_m := D_m \circ (\varrho \otimes \cdots \otimes \varrho) \circ E_m$$

satisfy $F_m \xrightarrow[m\to\infty]{\mathcal{T}_{1,1}} \mathrm{id}_{\mathbb{R}}$.

Then for all $d, k \in \mathbb{N}$, $W, N \in \mathbb{N}_0 \cup \{\infty\}$ and $L \in \mathbb{N} \cup \{\infty\}$ it holds that

$$\mathtt{NN}^{\varrho,\,d,\,k}_{W,\,L,\,W} \subseteq \overline{\mathtt{SNN}^{\varrho,\,d,\,k}_{n^2W,\,L,\,nN}},$$

where the closure is a sequential closure which is taken with respect to the topology $\mathcal{T}_{d,k}$.

Lemma C.7: [12, Lemma A.8, p. 331]

Define $T_y f : \mathbb{R} \to \mathbb{R}$ given by $x \mapsto T_y f(x) = f(x-y)$ for $f : \mathbb{R} \to \mathbb{R}$ and $y \in \mathbb{R}$, and let $X^n : \mathbb{R} \to \mathbb{R}$ be given by $x \mapsto x^n$ for $n \in \mathbb{N}_0$, with convention $X^0 \equiv 1$. By defining

$$\mathcal{V}_n :=\{T_y X^n \mid y \in \mathbb{R}\}$$

it holds that $\mathcal{V}_n = \mathbb{R}_{\deg \leq n}[x]$, where $\mathbb{R}_{\deg \leq n}[x]$ has dimension $n+1$.

Theorem C.8: [17, p. 64, Theorem 4.40]

The series

$$\sum_{n=1}^{\infty} \frac{1}{n^p},$$

where $p \in \mathbb{R}_+$ is convergent when $p > 1$, and divergent when $p \leq 1$.

Definition C.9: Approximation Space [1, Definition 3.4, p. 9]

Let $\mathcal{X}$ be a quasi-normed space, let $\Sigma := \{\Sigma_n\}_{n=0}^{\infty}$, where $\Sigma_n \subseteq \mathcal{X}$, such that (P1)-(P4) are satisfied, and let S be an admissible sequence space. Then the space

$$A(\mathcal{X}, S) := \{f \in \mathcal{X} \mid \{\mathscr{E}(f, \Sigma_n)\}_{n=0}^{\infty} \in S\}$$

endowed with

$$\|f\|_{A(\mathcal{X},S)} := \|\{\mathscr{E}(f, \Sigma_n)\}_{n=0}^{\infty}\|_S\,,$$

is called an **approximation space** or a **generalized approximation space**.

Remark C.10: [1, Remark 3.5, p. 9]

Let $b_n > 0$ be fixed for $n \in \mathbb{N}_0$ and let $p \in (0, \infty]$. If $K(n) < K(n+1)$ for all $n \in \mathbb{N}_0$ and S is defined by the quasi-norm

$$\|\{a_n\}_{n=1}^{\infty}\|_S := \|\{b_n a_n\}_{n=1}^{\infty}\|_{\ell^p(\mathbb{R})}, \tag{C.1}$$

then S is admissible if there exists some constant $C > 0$ such that, for all $n \in \mathbb{N}_0$

$$\begin{gathered} \max\{b_{K(n)}, \dots, b_{K(n+1)-1}\} \leq C b_n \\ (K(n) - K(n+1))^{1/q} \leq C. \end{gathered} \tag{C.2}$$

Particularly, if $K(n) = Kn$ with some $K \in \mathbb{N}$ and $b_n = (n+1)^{\alpha - 1/q}$ with some $\alpha > 0$, then (C.2) is satisfied and (C.1) defines an admissible sequence space.

Proposition C.11: [1, Proposition 3.8, p. 9]

Let $A(\mathcal{X}, \Sigma)$ be defined as in Definition VI.2. Then $A(\mathcal{X}, \Sigma)$ is a quasi-normed space, which is continuously embedded into $\mathcal{X}$. If $\mathcal{X}$ and S are normed spaces and all Σ_n are linear subspaces of $\mathcal{X}$, then $A(\mathcal{X}, \Sigma)$ is a normed space.

Theorem C.12: Completeness of $A_q^\alpha(\mathcal{X}, \Sigma)$ [1, Theorem 3.12(ii), p. 11]

Suppose that S has the property

$$\|\{a_n\}_{n=0}^\infty\|_S \le C \lim_{k\to\infty} \|\{a_n\}_{n=0}^k\|_S \quad \forall \{a_n\}_{n=0}^\infty \in S, \quad a_0 \ge a_1 \ge \cdots \ge 0,$$

where $C > 0$ is a constant depending only on S. Let $\mathcal{X}$ be complete, and assume that a decreasing sequence $\{a_n\}_{n=0}^\infty \subseteq [0, \infty)$ belongs to S if and only if

$$\lim_{k\to\infty} \|\{a_n\}_{n=0}^k\|_S = \sup_k \|\{a_n\}_{n=0}^k\|_S < \infty.$$

Then $A(\mathcal{X}, S)$ is complete.

Proposition C.13: [12, p. 336]

Let ϱ be a continuous activation function. Then it is locally bounded and Borel measurable.

Lemma C.14: [12, Lemma B.1, p. 335]

Let $\mathcal{X}$ be the one of the following classes of functions:

(i) Locally bounded functions.

(ii) Borel-measurable functions.

(iii) Continuous functions.

(iv) Lipschits continuous functions.

(v) Locally Lipschitz continuous functions.

Then if the activation function ϱ belongs to $\mathcal{X}$, it holds that any $f \in \mathtt{NN}_{\infty,\infty,\infty}^{\varrho,d,k}$ also belongs to $\mathcal{X}$.

Theorem C.15: [2, Problem 1(b), p. 1]

The space $\mathcal{C}_c^\infty(\mathbb{R}^d, \mathbb{R})$ is dense in $L^p(\mathbb{R}^d, \mathbb{R})$.

Theorem C.16: The Universal Approximation Theorem [12, Theorem 3.22, p. 289]

Let $\varrho : \mathbb{R} \to \mathbb{R}$ be a non-degenerated activation function, let $K \subseteq \mathbb{R}^d$ be compact, let $f : \mathbb{R}^d \to \mathbb{R}$ be a continuous function, and assume $\varepsilon > 0$. Then, there exist a $N \in \mathbb{N}$ and suitable b_j, $c_j \in \mathbb{R}$, $w_j \in \mathbb{R}^d$, for $1 \leq j \leq N$, such that the function $g : \mathbb{R}^d \to \mathbb{R}$ given by

$$x \mapsto \sum_{j=1}^{N} c_j \varrho\left(\langle w_j, x\rangle + b_j\right)$$

satisfies

$$\|f - g\|_{L^\infty(K)} \leq \varepsilon.$$

Lemma C.17: [12, Lemma 3.20, p. 289]

Let $\Omega \subseteq \mathbb{R}^d$ be an admissible domain, let $k \in \mathbb{N}$, and let $0 < p < \infty$. Then

$$X_p^k(\Omega) = \left\{ f\restriction_\Omega \;\middle|\; f \in X_p^k(\mathbb{R}^d) \right\},$$
$$X_\infty^k(\Omega) = \left\{ f\restriction_\Omega \;\middle|\; f \in \mathcal{C}_0(\mathbb{R}^d, \mathbb{R}^k) \right\}.$$

Additionally, $X_p^k(\Omega)$ is a quasi-Banach space.

Proposition C.18: [12, p. 336]

If the activation function is continuous, it is non-degenerate if and only if it is not a polynomial.

Definition C.19: Periodization [4, p. 553, 551]

Let $f : \mathbb{R}^d \to [0, \infty]$ be a function. Then the **periodization** of f, denoted by Pf, is given by

$$Pf(x) \coloneqq |\det(B)| \sum_{y \in \mathbb{Z}^d} f(x - By) \quad x \in \mathbb{R}^d,$$

where $B \in \mathbb{R}^{d \times d}$ is a translation matrix, which is invertible.

Definition C.20: [4, p. 551]

Let $f : \mathbb{R}^d \to [0, \infty]$ be a function. Then f has a **radially decreasing L^1-majoriant** if

$$|f| \leq \mu$$

almost everywhere for some radial function $\mu(\|x\|_{\mathbb{R}^d}) \in L^1(\mathbb{R}^d, \mathbb{R})$ such that $\mu(\|x\|_{\mathbb{R}^d})$ decreases as a function of $\|x\|_{\mathbb{R}^d}$.

Definition C.21: Complex Radon Measure [11, p. 212, 222]

Let $\mathcal{X}$ be a locally compact Hausdorff space. A **Radon measure** is a Borel measure that is finite on all compact sets, outer regular on all Borel sets, and inner regular on all open sets. A **Complex Radon measures** is a Borel measure whose real and imaginary parts are signed Borel measures.

Lemma C.22: [4, Lemma A.2, p. 551]

Let $f \in L^\infty(\mathbb{R}^d, \mathbb{R})$ have a radially decreasing L^1-majorant, and assume $P|f|$ is a periodization of $|f|$. Then $P|f| \in L^\infty(\mathbb{R}^d, \mathbb{R})$

Corollary C.23: [4, Corollary 1, p. 539]

Let $g \in L^p(\mathbb{R}^d, \mathbb{R})$, for $1 \leq p < \infty$, and let $P|g|$ be a periodization of $|g|$, such that $P|g| \in L^p_{\text{loc}}(\mathbb{R}^d, \mathbb{R})$ for $1 < p < \infty$. If

$$\int_{\mathbb{R}^d} g(x)\, dx \neq 0,$$

then the space

$$\mathcal{V}_+(g) := \{g_{j,k} \mid j > 0,\ k \in \mathbb{Z}^d\}$$

spans $L^p(\mathbb{R}^d, \mathbb{R})$, where $g_{j,k}(x) := 2^{jd/p} g(2^j x - k)$.

Proposition C.24: Sufficient Conditions [13, Proposition 2(a), p. 241]

Let $f \in L^p(\mathbb{R}^d, \mathbb{R})$, for $p \in (0, 1]$, and let Pf be a periodization of f with translation matrix B, and assume $\Omega = [0, 1)^d$. Moreover, assume $f \in L^1(\mathbb{R}^d, \mathbb{R})$ with

$$\int_{\mathbb{R}^d} f(x)\, dx \neq 0.$$

Then

$$\|\lambda P f(B\bullet) - 1\|_{L^p(\Omega, \mathbb{R})} < 1,$$

for some $\lambda \in \mathbb{R}$.

Theorem C.25: Synthesis onto L^p [13, Theorem 3, p. 243]

Let $f \in L^p(\mathbb{R}^d, \mathbb{R})$, for $p \in (0,1]$, and assume that

$$Q := \|\lambda P f(B\bullet) - 1\|_{L^p(\Omega,\mathbb{R})} < 1,$$

for some $\lambda \in \mathbb{R}$, where $\Omega := [0,1)^d$. Then $S : \ell^p(\mathbb{N} \times \mathbb{Z}^d) \to L^p(\mathbb{R}^d, \mathbb{R})$ given by

$$Sc := \sum_{j\in\mathbb{N}} \sum_{k\in\mathbb{Z}^d} c_{j,k} f_{j,k},$$

is open, and subjective, where $c := \{c_{j,k}\}_{j\in\mathbb{N},k\in\mathbb{Z}^d}$ and

$$f_{j,k} := |\det(a_j)|^{\frac{1}{p}} f(a_j x - Bk) \quad x \in \mathbb{R}^d.$$

Indeed, if $f \in L^p(\mathbb{R}^d, \mathbb{R})$ and $\tilde{Q} \in (Q,1)$ then there exists a sequence $c \in \ell^p(\mathbb{N} \times \mathbb{Z}^d)$ such that $Sc = h$ and

$$\|c\|_{\ell^p(\mathbb{N}\times\mathbb{Z}^d)} \leq (1-\tilde{Q})^{-\frac{1}{p}} |\lambda| |\det(B)| \|h\|_{L^p(\mathbb{R}^d,\mathbb{R})}.$$

Theorem C.26: Hahn-Banach Theorem [11, Theorem 5.8, p. 158]

Let $\mathcal{X}$ be a real vector space, let P a sublinear functional on $\mathcal{X}$, $V \subseteq \mathcal{X}$ a subspace, and ψ a linear functional on V such that $\psi(x) \leq P(x)$ for all $x \in V$. Then there exists a linear functional φ on $\mathcal{X}$ such that $\varphi(x) \leq P(x)$ for all $x \in \mathcal{X}$ and $\varphi\restriction_V = \psi$.

Theorem C.27: Riesz Representation Theorem [11, p. 223, Theorem 7.17]

Let $\mathcal{X}$ be a locally compact Hausdorff space, and denote $M(\mathcal{X})$ to be the space of complex Radon measures on $\mathcal{X}$, and define

$$I_\mu(f) := \int_{\mathcal{X}} f(x)\, d\mu(x),$$

for $\mu \in M(\mathcal{X})$ and $f \in \mathcal{C}_c(\mathcal{X})$. Then the map $\mu \mapsto I_\mu$ is an isometric isomorphism from $M(\mathcal{X})$ to $(\mathcal{C}_c(\mathcal{X}))^*$.

Theorem C.28: Jordan Decomposition [11, p. 87, Theorem 3.4]

If μ is a signed measure, there exist unique positive measures μ^+, and μ^- such that $\mu = \mu^+ - \mu^-$ and $\mu^+ \perp \mu^-$.

Theorem C.29: [11, p. 242-243, Theorem 8.14(b)]

Let $f \in L^1(\mathbb{R}^d, \mathbb{R})$ and assume

$$\int_{\mathbb{R}^d} f(x)\,dx = C \in \mathbb{R}.$$

Define $f_c(x) := c^{-d} f(c^{-1}x)$. If h is bounded and uniformly continuous, then $h * f_c \to Cf$ uniformly as $c \to 0$.

Lemma C.30: [12, Remark on p. 290]

Let g be bounded and with compact support. Then there exists a non-increasing function $\mu : [0,\infty) \to [0,\infty)$ such that

$$\int_{\mathbb{R}^d} \mu(\|x\|_{\mathbb{R}^d})\,dx < \infty$$

and

$$|g(x)| \leq \mu(\|x\|_{\mathbb{R}^d}) \quad \forall x \in \mathbb{R}^d.$$

D | Errata to [Gribonval et al., 2021]

Some errata to "Approximations Spaces of Deep Neural Networks", [12].

- Page 286 line $31-31$: "$\leq$", should be "$\preceq$".
- Page 286 line 31: "b_+", should be "b".
- Page 315 align 1: "ℓ^0", should be "ℓ_0".
- Page 318 align 14: "$\mathbb{R}^d$", should be "$\mathbb{R}^k$".
- Page 324 line 24: "$\alpha_\ell^{(1)} \otimes \ldots \otimes \alpha_\ell^{(N_\ell)}$", should be "$\alpha_\ell^{(1)} \otimes \cdots \otimes \alpha_\ell^{(N_\ell)}$".
- Page 324 line 24: "L", should be "K".
- Page 324 line 24: "K_ℓ", should be "N_ℓ".
- Page 326 line 2: "$\mathtt{NN}^{\varrho,\,d,\,k}_{W+Nw,\,1+(L'-1)(\ell+1),\,N(2+m)}$", should be "$\mathtt{NN}^{\varrho,\,d,\,k}_{mW+Nw,\,1+(L'-1)\ell,\,N(1+m)}$".
- Page 326 line 2: "$\mathtt{NN}^{\varrho,\,d,\,k}_{W+Nw,\,1+(L-1)(\ell+1),\,N(2+m)}$", should be "$\mathtt{NN}^{\varrho,\,d,\,k}_{mW+Nw,\,1+(L-1)\ell,\,N(1+m)}$".
- Page 330 line 23: "$f_i^{(n)} \to f_i^{(0)}$", should be "$f_i^{(n)} \xrightarrow[n\to\infty]{} f_i^{(0)}$".
- Page 333 line 10: "second part", should be "first part".
- Page 333 line 30: "$\leq \delta_m$", should be "$\leq C\delta_m$".
- Page 336 line $17, 23$: "$\frac{1}{(4R)^d}$", should be "$\frac{1}{(2R)^d}$".
- Page 336 line $18-19$: "is (only) true since we are considering *generalized* neural networks", is misleading since it holds for strict as well.
- Page 337 line 15: "L_p", should be "$X_p(\Omega)$".
- Page 337 line 16: Note that "g" defined on line 16 is not the same one given in Claim (2) in Theorem 3.23 on page 290 line 6.
- Page 337 line 17: "$\|F_i - G_i\|^{p_0}_{X_p^k(\Omega)}$", should be "$\|F_i - G_i\|^{p_0}_{X_p^1(\Omega)}$".
- Page 337 line 18: "$g \in$", should be "$f \in$".
- Page 337 line 30: There is an error between the reference and the numbers. If the reference 39 is correct "Theorem 4 and Proposition 5(a)", should be "Theorem 3 and Proposition 2(a)", and if the number is correct the reference 39 on page 365 line $54-55$ "**14**(2), 235-266 (2008)", should be "**14**, 1-29 (2007)".
- Page 338 line 9: (B.1) also holds for $0 < a < 1$, so "$a \geq 1$" can be omitted.
- Page 346 line 13: "$4n(2^j - 1)$", should be "$6n(2^j - 1)$".
- The article is inconsistent in the use of $\subseteq$ and $\subset$. For instance, both $\subset$ on page 272 line 2 and $\subseteq$ on page 270 line 27 allows equality.

E | Code overview

This chapter contains an overview of which files belong to which parts of the experiment: The initial experiment is the following:

Initial Experiment:

(i) Generating B-splines:
- Generating B-splines: `Bspline` in *FunctionGeneration*.

(ii) Construction of Neural Networks:
- Generate neural networks: `NNtoBSpline` in *Interpolation*, using `CustomModels` in *Interpolation/NeuralNetworkMisc*. `CustomModels` construct the layers in the neural network model and uses `CustomActivation` and `CustomInitializer` in *Interpolation/NeuralNetworkMisc*.
- Train neural networks on B-spline: `BaseBsplineCoef` trains the neural networks, and saves the trained weights in *Interpolation/NeuralNetworkMisc/BaseCoef*.

(iii) Comparison:
- Plot both B-spline and the neural networks both untrained and trained: `InitialExperiment` using the previous files.

Next is the overview of the main experiment:

Main Experiment:

(i) Generation of Target Functions:
- Polynomial and continuous piecewise polynomials: `PSF` in *FunctionGeneration* using `RandomCoefficients`, and `RandomIntervals` in *FunctionGeneration*.

(ii) Approximation Methods:
- ***Spline***: `SplineApprox` in *Interpolation*.
- ***Equiv*** and ***Fully***: `NNtoBspline` in *Interpolation*.

(iii) Evaluation:
- Evaluations of functions and approximations: `MainExperiment` using `BSpline`, `PSF` in *FunctionGeneration*, and `NNtoBSpline`, `SplineApprox` in *Interpolation*. `MainExperiment` construct plots, mean square error, and maximum error of functions and approximations.
- Selection of results: `PlotFinder` using the result from `Experiment`.

Lastly is an overview of the code to the figures in the report constructed in Python:

Illustration in the report:
- Front page: `FrontPage` in *Plotgenerator*.
- Activation function: `ActivationFunctions` in *Plotgenerator*.
- B-splines: `BSplines` in *Plotgenerator*.

E.1 Results from Initial Experiment

Degree	Mean Square Error Untrained	Mean Square Error Trained	Maximum Error Untrained	Maximum Error Trained
0	0.000e+00	0.000e+00	0.000e+00	0.000e+00
1	3.382e-16	3.382e-16	5.913e-08	5.913e-08
2	5.160e-14	5.160e-14	1.431e-06	1.431e-06
3	4.002e-13	4.003e-13	5.722e-06	5.722e-06
4	4.229e-12	4.230e-12	1.526e-05	1.526e-05
5	2.869e-11	2.912e-11	4.578e-05	4.578e-05
6	3.719e-10	3.957e-10	1.831e-04	1.831e-04
7	3.024e-09	1.480e-08	7.324e-04	9.766e-04
8	3.802e-08	3.001e-07	2.441e-03	3.174e-03
9	1.233e-06	1.565e-04	1.270e-02	4.492e-02
10	3.307e-06	6.820e-01	2.393e-02	3.295e+00
11	9.691e-05	1.162e+02	1.211e-01	4.546e+01
12	4.845e-04	3.111e+06	3.438e-01	9.341e+03
13	5.125e-03	2.399e+09	1.250e+00	2.674e+05
14	5.822e-02	7.727e+01	5.000e+00	7.650e+01
15	4.407e-01	9.068e+10	9.000e+00	2.083e+06

Figure E.1: *Overview of mean square error and maximum error between B-spline and the neural network, both untrained and trained, from the initial experiment.*

Bibliography

[1] J. M. Almira and U. Luther. Generalized approximation spaces and applications. Math. Nachr. 263(264), 3–35, 2004.

[2] S. Arpin. Analysis Prelim August 2011. University of Colorado Boulder, Mathematics Department, 2011.

[3] C. M. Bishop. Pattern Recognition and Machine Learning. Springer, 2006.

[4] H. Q. Bui and R. Laugesen. Affine systems that span Lebesgue spaces. The Journal of Fourier Analysis and Applications, 11(5), 533–556, 2005.

[5] P. G. Casazza. Every Frame is a Sum of Three (But Not Two) Orthonormal Bases–and Other Frame Representations. The Journal of Fourier Analysis and Applications, 1998.

[6] I. Daubechies, R. DeVore, S. Foucart, B. Hanin, and G. Petrova. Nonlinear Approximation and (Deep) ReLU Networks. Constructive Approximation, Springer, 2021.

[7] C. de Boor. A Practical Guide to Splines. Springer, 2001.

[8] R. DeVore, B. Hanin, and G. Petrova. Neural network approximation. Cambridge University Press, 2021.

[9] R. A. DeVore. Nonlinear approximation. Acta Numerica (1998), pp. 51-150, 2021.

[10] R. A. DeVore and G. G. Lorentz. Contructive Approximation. Springer-Verlag, 1993.

[11] G. Folland. Real Analysis: Modern Techniques and Their Applications. Pure and Applied Mathematics, second edition, 1999.

[12] R. Gribonval, G. Kutyniok, M. Nielsen, and F. Voigtlaender. Approximation Spaces of Deep Neural Networks. Constructive Approximation, Springer, 2021.

[13] R. Laugesen. Affine synthesis onto L-p when $0 < p \leq 1$. The Journal of Fourier Analysis and Applications, 14, 235–266, 2008.

[14] C. Moler. Numerical Computing with MATLAB. SIAM, 2008.

[15] M. Nielsen. General approximation with deep neural networks. Department of Mathematical Sciònces, Aalborg University, 2023.

[16] J. Obando-Ceron, A. Courville, and P. S. Castro. In deep reinforcement learning, a pruned network is a good network. Google DeepMind, 2024.
https://arxiv.org/pdf/2402.12479.pdf [Visited 23/04-2024].

[17] E. T. Poulsen. Funktioner af en og flere variable. Institut for Matematik, Aarhus Universitet, 1st edition, 2016.

[18] scipy.interpolate.splrep. SciPy, 2024.
https://docs.scipy.org/doc/scipy/reference/generated/scipy.interpolate.splrep.html [Visited 0/05-2024].

[19] G. Sokar, R. Agarwal, P. S. Castro, and U. Evci. The Dormant Neuron Phenomenon in Deep Reinforcement Learning. Proceedings of the 40 th International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.
https://proceedings.mlr.press/v202/sokar23a/sokar23a.pdf [Visited 23/04-2024].

[20] R. Yeh, Y. S. G. Nashed, T. Peterka, and X. Tricoche. Fast Automatic Knot Placement Method for Accurate B-spline Curve Fitting, 2020.
https://www.sciencedirect.com/science/article/pii/S0010448520300981.