Technische Universität Berlin
Faculty IV
Electrical Engineering and Computer Science

Learning Multivariate Functions with Simple Structures

A Thesis Submitted in Partial Fulfillment
of the Requirements for the Degree of

Master of Science

in

Electrical Engineering

by

Justus Prass

Matriculation Number: 490692

First Examiner:	Prof. Dr.-Ing. Slawomir Stańczak
Second Examiner:	tbd
Supervisors:	Dr. rer. nat. Igor Bjelaković
	M.Sc Navneet Agrawal
Submitted On:	June 5, 2025

Declaration of Authorship

By: Justus, Prass

Matrikelnumber: 490692

I hereby declare that the thesis submitted is my own, unaided work, completed without any external help. Only the sources and resources listed were used. All passages taken from the sources and aids used, either unchanged or paraphrased, have been marked as such. Where generative AI tools were used, I have indicated the product name, manufacturer, the software version used, as well as the respective purpose (e.g. checking and improving language in the texts, systematic research). I am fully responsible for the selection, adoption, and all results of the AI-generated output I use. I have taken note of the Principles for Ensuring Good Research Practice at TU Berlin dated 8 March 2017.

I further declare that I have not submitted the thesis in the same or similar form to any other examination authority [Dep24].

Berlin, June 5, 2025

Place, Date **Signature**

Disclaimer on the Use of AI Tools

Generative AI tools were employed to improve the quality and phrasing of the text. Specifically, ChatGPT by OpenAI, based on GPT-4-turbo [Cha25] and DeepL Write by DeepL SE [Dee25] were used for language refinement. Their use was strictly limited to enhancing grammar, structure, and style; no content generation or scientific reasoning was delegated to these tools. All academic content, interpretations, and conclusions were developed independently by the author.

Abstract

Machine learning has undergone a period of accelerated growth and development in recent years, particularly through the implementation of deep neural networks, which have emerged as one of the most effective architectures for solving complex learning tasks. Nevertheless, despite their remarkable success, deep neural networks are encumbered by several notable limitations. Consequently, the exploration of alternative architectures has become a compelling area of research.

This thesis investigates a recently introduced approach within the machine learning framework known as Kolmogorov Arnold networks, which are examined as a potential alternative to conventional deep neural networks.

The exploration of this alternative unfolds in two main parts. The initial part of the thesis establishes the necessary theoretical background and extends prior theoretical work, including that related to Kolmogorov Arnold networks, to develop a more refined mathematical framework. In particular, we derive an upper bound on the approximation error, which contributes to a deeper understanding of the model's theoretical properties.

The subsequent part of the thesis builds on this refined mathematical framework to propose a novel machine learning approach, situated within the broader context of deep neural networks. The proposed method is situated within the existing machine learning landscape by providing the relevant theoretical underpinnings. Subsequently, extensive empirical evaluations are conducted to assess the potential and limitations of the proposed framework. The approach is benchmarked not only against closely related models such as Kolmogorov Arnold networks, but also against conventional deep neural networks, to evaluate its practical viability and comparative performance.

The collective contributions of this thesis aspire to further the development of theoretically grounded and practically competitive alternatives to current deep learning paradigms.

Zusammenfassung

In den zurückliegenden Jahren durchlief das maschinelle Lernen eine Phase des beschleunigten Wachstums und der Entwicklung, die insbesondere durch die Implementierung tiefer neuronaler Netze charakterisiert war. Letztere haben sich als eine der effektivsten Architekturen zur Lösung komplexer Lernaufgaben etabliert. Trotz ihres bemerkenswerten Erfolges sind tiefe neuronale Netze jedoch mit mehreren signifikanten Einschränkungen konfrontiert. Infolgedessen hat die Erforschung alternativer Architekturen zunehmend an Bedeutung gewonnen.

Die vorliegende Arbeit widmet sich einem kürzlich eingeführten Ansatz im Bereich des maschinellen Lernens, den sogenannten Kolmogorov Arnold Netzen, und untersucht deren Potenzial als Alternative zu klassischen tiefen neuronalen Netzen.

Die Untersuchung dieser Alternative erfolgt in zwei Hauptteilen. Im ersten Teil der Arbeit wird der notwendige theoretische Hintergrund bereitgestellt und frühere theoretische Arbeiten, einschließlich derer zu Kolmogorov Arnold Netzen, um einen verfeinerten mathematischen Rahmen erweitert. Von besonderer Relevanz ist die Ableitung einer oberen Schranke für den Approximationsfehler, die zu einem vertieften Verständnis der theoretischen Eigenschaften der Modelle beiträgt.

Der zweite Teil der Arbeit präsentiert, aufbauend auf dem verfeinerten mathematischen Rahmen, einen neuartigen Ansatz im Bereich des maschinellen Lernens, der im Kontext tiefer neuronaler Netze eingeordnet wird. Die vorgestellte Methode wird zunächst in die bestehende Forschungslage eingeordnet, wobei die relevanten theoretischen Grundlagen dargelegt werden. Im Anschluss werden umfangreiche empirische Evaluierungen durchgeführt, um sowohl das Potenzial als auch die Grenzen des entwickelten Ansatzes zu bewerten. In diesem Zusammenhang werden nicht nur verwandte Modelle wie die Kolmogorov Arnold Netze, sondern auch klassische tiefe neuronale Netze herangezogen, um die praktische Anwendbarkeit und Leistung des entwickelten Ansatzes im Vergleich zu bewährten Methoden zu evaluieren.

Die vorliegenden kollektiven Beiträge haben das Ziel, die Entwicklung theoretisch fundierter und zugleich anwendungsorientierter Alternativen zu den gegenwärtigen Paradigmen der tiefen neuronalen Netze voranzutreiben.

Contents

List of Figures

List of Tables

List of Abbreviations

KAN	Kolmogorov Arnold network
KANN	Kolmogorov Arnold neural network
DNN	deep neural network
MIA	membership inference attack
RMSE	root mean squared error
fs	fixed-shape
aw	arbitrary-width

1 Introduction

In recent years, the field of machine learning has undergone rapid advancement, propelled by the increasing availability of large datasets, improved computational resources, and significant algorithmic innovations. Among the most transformative developments in this domain is the emergence of deep neural networks, which have proven to be highly effective in modeling complex, nonlinear relationships across a wide range of tasks. These networks have become foundational in numerous applications due to their flexibility and expressive power. However, despite their widespread adoption, deep neural networks exhibit several significant limitations. Among the most notable are their tendency to overfit, particularly when trained on limited or noisy data [SL19], and the opacity of their internal decision-making processes, which hinders interpretability [ZTLT21]. Furthermore, these models frequently necessitate substantial amounts of labeled data to function optimally, a prerequisite that is not invariably feasible in practical real-world scenarios [Mar18]. Their training and deployment are computationally intensive, demanding substantial hardware, energy, and time resources [TGLM22]. Finally, deep neural networks frequently struggle to generalize to out-of-distribution data, limiting their adaptability in dynamic environments [HSW+21]. These limitations have prompted the proposal of an alternative architecture, called Kolmogorov Arnold networks, by Liu et al. [LWV+24]. In their study, Liu et al. hypothesized that this alternative architecture could mitigate some of the existing limitations. Inspired by the aforementioned Kolmogorov Arnold networks, this thesis offers a novel perspective by investigating the learning and representation of a specific set of functions that can be decomposed into compositions of "simpler" functions.

The primary objective of this thesis is twofold. First, it seeks to build upon the theoretical foundations established in previous research, such as [LWV+24], to gain a deeper understanding of the existing methodologies. Second, it aims to leverage these insights to propose a novel approach in the field of deep neural networks.

The discussion unfolds in several steps: In the preliminary Section 2 of this thesis, the notation employed in the subsequent sections is restated. However, the majority of the notation utilized will be introduced within the respective sections.

Section 3 introduces the theoretical underpinnings essential for comprehending the subject matter.

Section 4 focuses on concrete methods for approximating univariate functions, the results of which are revisited later.

The subsequent Section 5 is pivotal as it unveils the central tenets of the thesis. This segment encompasses the delineation of the set of functions to be approximated and the substantiation of pivotal theoretical outcomes.

The ensuing Section 6 builds upon these foundational results by introducing a novel methodology in the field of deep neural networks. This methodology is then subjected to a thorough analysis and evaluation. The proposed approach is benchmarked against existing methods, such as Kolmogorov Arnold networks and deep neural networks, through extensive empirical evaluations. These evaluations extend beyond the scope of those conducted in [LWV+24], as they are situated within the context of engineering applications, incorporating noise and irrelevant features. This expanded scope offers valuable insights, particularly concerning robustness and generalization. The section culminates in a discourse on prospective avenues for future research.

Finally, Section 7 offers a concise summary of the primary findings and emphasizes their practical ramifications.

2 Notation

In this thesis, the following notation is employed:

- $\bigcirc_{\mathrm{i\,=\,1}}^{\mathrm{N}}$ denotes the composition of N functions, where the N-th function represents the outermost function. Specifically, for functions $f_i : A_i \to B_i$, $i = 1, \dots, N$, with $A_{i+1} \subseteq B_i$ for $i = 1, 2, \dots, N-1$, the composition is given by

$$\bigcirc_{\mathrm{i\,=\,1}}^{\mathrm{N}} f_i := f_N \circ f_{N-1} \circ \cdots \circ f_1.$$

- $f : U \to V$ describes the function f defined on $U := \operatorname{dom} f$ and taking values in $V := \operatorname{codom} f$. For scalar-valued functions, we write $f : U \to V : u \mapsto f(u) \in V$, where $U \subset \mathbb{R}^n$ and $V \subset \mathbb{R}$. For vector-valued functions, we write $f : U \to V : u \mapsto f(u) := \{f^1(u), \dots, f^m(u)\} \in V$, where $U \subset \mathbb{R}^n$, $V \subset \mathbb{R}^m$, and f^k is scalar-valued, representing the k-th component of f.

- $f^{(j)} := D^j f :=$ j-th derivative of f.

- $C^d[U, V] := \{f : U \to V : \ D^i f \text{ continuous } 0 \leq i \leq d, i \in \mathbb{N}\}$. When the co-domain is understood we write $C^d[U]$

- $C_b^d[U] = \left\{f \in C^d[U] : \sup_{x \in U} |Df(x)| \leq M \text{ for some } M > 0\right\}$

- $\|x\|_{X,p}$ denotes the L_p-norm of $x \in X$, defined as:

$$\|x\|_{X,p} = \left(\sum_{i=1}^{n} |x_i|^p\right)^{\frac{1}{p}},$$

  if X is a vector space. For $p = \infty$, $\|x\|_{X,\infty}$ denotes the maximum norm:

$$\|x\|_{X,\infty} = \max_i |x_i|.$$

- If X is a function space of scalar-valued functions, the L_p-norm of $f \in X$ is given by:

$$\|f\|_{X,p} = \left(\int_{\operatorname{dom}(f)} |f(x)|^p \, dx\right)^{\frac{1}{p}},$$

  for $p \geq 1$. For $p = \infty$, $\|f\|_{X,\infty}$ denotes the supremum norm:

$$\|f\|_{X,\infty} = \sup_{x \in \operatorname{dom}(f)} |f(x)|.$$

- $\|A\|_{X,p}$ denotes the operator norm of a linear operator A on a vector space X, defined as:

$$\|A\|_p = \sup_{\|x\|_p=1} \|Ax\|_p.$$

- For normed linear spaces $(V, \|\cdot\|_V)$ and $(W, \|\cdot\|_W)$, and a linear map $f : V \to W$, the operator norm of f is given by:

$$\|f\| = \inf \{c \geq 0 \mid \|f(x)\|_W \leq c\|x\|_V \text{ for all } x \in V\}.$$

  Clearly, for all $x \in V$, we have $\|f(x)\|_W \leq \|f\|\|x\|_V$.

- $\mathcal{F}(A)$ denotes the space of all functions over A

- $S_{d,t}$ denotes the space of splines of degree at most d and grid t.

- $P_d[I^m]$ denotes the space of multivariate polynomials ($p : I^m \to \mathbb{R}$) of degree at most d.

- $\mathcal{S}^{\sigma,l}$ denotes the set of shallow neural networks of width l and activation function σ.

- $\mathcal{D}^{\sigma,t,l}$ denotes the set of deep neural networks of width l, depth t and activation function σ.

- $\omega(g, h) := \sup\{|g(x) - g(y)| : x, y \in [a, b], |x - y| \leq h\}$ denotes the modulus of continuity for $g \in C[a, b]$

- We denote by $\mathrm{Lip}_{L,\alpha}[I]$ the set of all functions f defined on the interval I such that

$$\omega(f, h) \leq Lh^\alpha \quad \text{for all } h > 0,$$

  where $L > 0$ is referred to as the Lipschitz constant and $0 < \alpha \leq 1$. When $\alpha = 1$, this condition corresponds to Lipschitz continuity, and we simply write $\mathrm{Lip}_L[I] := \mathrm{Lip}_{L,1}[I]$. For $0 < \alpha < 1$, the corresponding set is known as the set of Hölder continuous functions of order α. Moreover, we note that $\mathrm{Lip}_{L,\alpha}[I] \subseteq \mathrm{Lip}_{L',\alpha}[I]$ whenever $L \leq L'$.

- For a tuple t, the notation $|t|$ denotes its cardinality (i.e., the number of elements in the tuple).

- We say a tuple t is sorted if $t_1 \leq t_2 \leq \cdots \leq t_n$

- For any sorted tuple t:

$$\Delta^k_{min} t := \min\{t_{i+k} - t_i \mid t_{i+k} - t_i \neq 0\},$$

  and

$$\Delta^k t := \Delta^k_{max} t := \max\{t_{i+k} - t_i \mid t_{i+k} - t_i \neq 0\},$$

  when k is omitted it is understood that $k = 1$.

- We say a function f is bounded if $\sup_{x \in dom f} |f(x)| < \infty$. We say f is B-bounded if $\sup_{x \in dom f} |f(x)| \leq B, B \in \mathbb{R}$
- $\langle \cdot, \cdot \rangle : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$, $(x, y) \mapsto \sum_{i=1}^{n} x_i y_i$, denotes the inner product.
- For defining sets, we use the notation $\{a_i\}_{i \in X} := \{a_i \mid i \in X\}$. When the set X is understood, we may simply write $\{a_i\}$, which carries the same meaning.
- For sums, we use the notation $\sum_{i=1}^{n} x_i = x_1 + x_2 + \cdots + x_n$. When the bounds of summation are understood we write $\sum_i x_i$, which carries the same meaning.
- For products, we use the notation $\prod_{i=1}^{n} x_i = x_1 \cdot x_2 \cdot \cdots \cdot x_n$. When the bounds of the product are understood, we write $\prod_i x_i$, which carries the same meaning.
- $\mathcal{N}(\mu, \sigma^2)$ denotes the normal distribution with mean μ and variance σ^2.

3 Foundations of the Theory for Approximating Univariate Functions

Approximation theory constitutes a foundational domain within the realm of mathematical analysis, encompassing the representation and approximation of functions. In this and the subsequent section, our primary focus is on the approximation of univariate functions. As this thesis progresses, the objective is to extend the theory of univariate function approximation to make broader statements about a specific set of multivariate functions. In this section, we will explore the key concepts, methods, and results that allow us to approximate a given univariate function $f : \mathbb{R} \to \mathbb{R}$. The objective is to identify representations that preserve crucial properties of the original function, such as continuity or smoothness, while ensuring that the approximation error, which will be introduced subsequently, is minimal. Prior to defining the aforementioned approximation error, it is imperative to establish the necessary foundations. This necessitates a consensus on how to quantify the distance between two functions. In order to do so, the notions of metric and normed spaces will be relevant.

3.1 Metric Spaces

The distance between two elements of a set is referred to as a metric. A metric is defined as a function that is applied to a set and that satisfies the fundamental properties expected of a distance measure. The subsequent formal definition is provided for the purpose of further elucidation.

Definition 3.1 (Metric)**.** *On a set X a function $d_X : X \times X \to \mathbb{R}^+$ is a metric if and only if it satisfies*

1. $d_X(x, x) = 0, \forall_{x \in X}$ *(Reflexivity).*
2. $d_X(x, y) > 0$, $\forall_{x,y \in X, x \neq y}$ *(Positivity).*
3. $d_X(x, y) = d_X(y, x), \forall_{x,y \in X}$ *(Symmetry).*

4. $d_X(x,z) \leq d_X(x,y) + d_X(y,z), \forall_{x,y,z\in X}$ *(Triangle inequality).*

The ordered pair (X, d_X) *is designated as a metric space.*

We say that (X, d_X) has dimension n, denoted by $dimX = n < \infty$, if for every $f \in X$ we can write $f = \sum_{i=1}^{n} a_i g_i$, where $g_i \in X$ are fixed linearly independent bases. With some abuse of notation, we say that X is infinite dimensional if $n = \infty$. Let (X, d_X) be a metric space, then we say $Y \subset X$ is bounded if there exists some $M > 0$ such that for all $x, y \in Y$ we have $d_X(x,y) < M$. We say that $Y \subset X$ is closed if it contains all its boundary points. This means that if a sequence $\{x_n\} \in Y$ converges to some $x \in X$, then $x \in Y$. Equivalently, Y is closed if the complement $Y^c = X \setminus Y$ is open. That is, for every point $x \in X \setminus Y$, there exists some $\epsilon > 0$ such that the sphere $B(x, \epsilon)$ is contained in $X \setminus Y$. The term ”dense in X” is employed to denote the condition where the closure of $Y \subset X$ is equivalent to X. This means $\overline{Y} = X$, where $\overline{Y}$ denotes the closure of Y, which consists of all points in Y together with all its boundary points. Alternatively, Y is dense in X if, for every point $x \in X$ and every $\epsilon > 0$, there exists a point $y \in Y$ such that $d_X(x,y) < \epsilon$. We say that $Y \subset X$ is compact if every sequence $\{x_n\}$ in Y has a subsequence $\{x_{n_k}\}$ that converges to a point $x \in Y$, formally we get that

$$\lim_{k\to\infty} x_{n_k} = x \quad \text{for some} \quad x \in Y$$

holds.

3.2 Normed Spaces

Normed linear spaces are a specialization of the concept of metric spaces, extending the notion of distance to the ”length” or ”size” of their elements. The magnitude or extent of elements in a vector space is quantified by a norm. The subsequent formal definition is derived from this concept.

Definition 3.2 (Norm)**.** *On a set* X *a function* $\|\cdot\|_X : X \to \mathbb{R}^+$ *is called a norm if and only if it satisfies*

1. $\|x\| \geq 0 \ \forall_{x\in X}$ *and* $\|x\| = 0$ *if and only if* $x = 0$ *(Positivity).*

2. $\|\alpha x\| = |\alpha| \|x\| \forall_{x\in X}, \alpha \in \mathbb{R}$ *(Homogeneity).*

3. $\|x + y\| \leq \|x\| + \|y\| \forall_{x,y\in X}$ *(Triangle inequality).*

The ordered pair $(X, \|\cdot\|_X)$ *is referred to as a normed linear space.*

Given a norm $\|\cdot\|_X$ on X, we can define a corresponding metric $d_X : X \times X \to \mathbb{R}^+ : \{f, g\} \mapsto \|f - g\|$. We say d_X is the metric induced by $\|\cdot\|$. The demonstration that d_X indeed defines a metric is outlined in [Che01, Chapter 1, pg.8]. Consequently, it can be concluded that every normed linear space is also a metric space, thereby providing a justification for the perspective that norms represent a specialization of the concept of metrics.

3.3 Definition of Approximation Error

Before defining the concept of approximation error, it is necessary to extend the notion of distance between points (i.e., metrics as discussed in Section 3.1) to encompass the distance between a set and a point. To this end, let (X, d_X) be a metric space, and let $Y \subseteq X$ and $f \in X$ be given, then we define the distance between these objects denoted by dist as follows:

$$\operatorname{dist}(f, Y) = \inf_{\hat{f} \in Y} d_X(f, \hat{f}). \tag{3.1}$$

In the context of approximation, this can be interpreted as the extent to which a function, $f \in X$, can be represented using elements from a set, $Y \subset X$. It is noteworthy that the latter set is often characterized as a "simpler" set of functions when compared to the original set, X. The point that minimizes the distance between the point $f \in X$ and the set Y (cf. Eq. (3.1)) is called the projection of f onto Y. Formally, we have the following relation:

$$\operatorname{proj}(f, Y) = \arg\min_{\hat{f} \in Y} d_X(f, \hat{f}). \tag{3.2}$$

The proj operator is equivalent to finding a point in $\hat{f} \in Y$ that is closest to some point $f \in X$ with respect to the metric d_X. For the projection operator to be well defined, it is necessary that it be a nonempty set; that is, the minimum must exist. This condition is guaranteed by the following theorem.

Theorem 3.1 (Theorem on Existence of Best Approximations in a Metric Space [Che01, Chapter 1, pg.4]). *Let (X, d_X) be a metric space, then for all compact sets $Y \subset X$ we have that for all $f \in X$ proj(f, Y) is non empty.*

The present focus is on the approximation of points with specific properties; that is, the necessity for distances between sets arises. In this context, let (X, d_X) be a metric space and consider $Y \subseteq X$ and $Z \subseteq X$. The distance between these sets is then defined as follows:

$$\operatorname{dist}(Y, Z) = \sup_{f \in Y} \operatorname{dist}(f, Z) = \sup_{f \in Y} \inf_{\hat{f} \in Z} d_X(f, \hat{f}). \tag{3.3}$$

In the context of approximation, this can be interpreted as the extent to which we can represent any function with certain properties (Y) — such as smoothness — using elements from the "simpler" set of functions Z. Eq. (3.3) pertains to the scenario where it is possible to determine a point $\hat{f} \in Z$ in the $\arg\inf_{\hat{f} \in Z} d_X(f, \hat{f})$ of the distance measure for any given point $f \in Y$. However, in most practical cases, this is not feasible, which motivates the introduction of an approximation operator. To this end, let (X, d_X) be a metric space and $Y \subset X$. A mapping $A : X \to Y$ is called an approximation operator because it maps points from the potentially complex set X to the presumably simpler set Y, providing an approximate representation. The distance between an operator A and a set $Z \subset X$ is defined as follows:

$$\operatorname{dist}(Z, A) = \sup_{f \in Z} d_X(f, Af), \tag{3.4}$$

where $\operatorname{dist}(Z, A)$ can be interpreted as the maximum approximation error when representing elements of Z using the operator A. A more thorough examination of the properties of approximation operators necessitates the introduction of the concept of sets induced by approximation operators. Let (X, d_X) be a metric space and A an arbitrary approximation operator on X. Additionally, consider the set $Y \subset X$, then the set $AY := \{Af \mid f \in Y\} \subset X$ is the set induced by A with respect to Y. In the event that the condition $dist(Y, A) = dist(Y, AY)$ is satisfied, it can be deduced that A projects elements of Y onto AY. If A projects elements of Y onto AY, then for any approximation operator $B : Y \to AY$, we have $dist(Y, A) \leq dist(Y, B)$. It is noteworthy that the projection of A onto AY is possible under the condition that AY is compact, as substantiated by Theorem 3.1. The operator A_1 is said to be superior to the operator A_2 if $dist(Y, A_1) < dist(Y, A_2)$. It is noteworthy that A_1Y and A_2Y may represent entirely distinct sets.

4 Approximation Methods

Two prominent methods for reconstructing functions from data are introduced below: polynomial methods and neural networks. On the one hand, polynomial approximation techniques are primarily concerned with representing functions as sums of monomial terms, often focusing on interpolation, i.e., exact fitting to the given data points. This category includes classical polynomial fitting techniques and methods employing piecewise polynomials, such as splines. Conversely, neural networks offer a more flexible and powerful framework, whereby the focus shifts from interpolation to learning, with the objective being to approximate a function that generalizes well to unseen data.

4.1 Polynomial Approximation Methods

The ensuing discourse briefly pertains to the approximation of univariate functions by means of polynomial methods. The present study is chiefly informed by the presentation of De Boor [dB01].

4.1.1 Polynomials

A function p is said to be a polynomial if it can be expressed in the form:

$$p(x) = a_1 + a_2 x + a_3 x^2 \cdots = \sum_{j=0}^{N} a_j x^j, \tag{4.5}$$

where $a_i \in \mathbb{R}$. A polynomial is said to have degree d if the coefficient $a_d \neq 0$, and there does not exist any index $i > d$ such that $a_i \neq 0$ (cf. Eq. (4.5)). While the set of all polynomials of degree d does not form a linear space, the set of all polynomials of degree at most d, denoted by P_d, does. This property renders the latter set particularly conducive to further analysis. The dimension of the space P_d is $d + 1$, as the set $\{1, x, x^2, \ldots, x^d\}$ forms a basis for P_d. The employment of polynomials in the approximation of functions is a recurring phenomenon, often manifesting through the process of interpolation.

Definition 4.1 (Interpolation)**.** *We say a function $f : I \to \mathbb{R}$ interpolates another func-*

tion $g : I \to \mathbb{R}$ at the points $\{\tau_i\}_{1\leq i\leq n}, \tau_i \in I$ if and only if

$$\forall_i \;\, f(\tau_i) = g(\tau_i).$$

In accordance with the foregoing definition, it is readily demonstrable that for any function $g : I \to \mathbb{R}$, there exists a polynomial of degree $n-1$ that interpolates g at the points $\{\tau_i\}_{1\leq i\leq n}$. This assertion can be substantiated through the utilization of the Lagrange polynomial, which is defined as

$$l_i(x) = \prod_{j\neq i}^{n} \frac{x-\tau_j}{\tau_i-\tau_j}, \tag{4.6}$$

which is a polynomial of degree $n-1$ with the property that

$$l_i(\tau_j) = \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{if } i \neq j \end{cases}.$$

Therefore, we may write $p_{n-1} = \sum_i g(\tau_i)l_i$ and that p_{n-1} interpolates g at $\{\tau_i\}_{1\leq i\leq n}$. For the sake of subsequent discourse, we define the operator

$$\mathcal{P}_n : C[I] \to P_n[I] : g \mapsto \sum_{i}^{n+1} g(\tau_i)l_i = p_n.$$

In fact, for any set of values $\{a_i \in \mathbb{R} \mid i = 1, \ldots, n+1\}$ and any corresponding set of interpolation points $\{\tau_i\}_{i=1}^{n+1}$. The polynomial of degree n, denoted by p_n, that satisfies $p_n(\tau_i) = \alpha_i$ for $i = 1, \ldots n+1$, has been demonstrated to be unique, as outlined in [dB01, pg. 2]. A foundational concept in the analysis of polynomial interpolation is the concept of the k-th order divided difference.

Definition 4.2 (Divided Difference [dB01, Definition 5, Chapter 1])**.** *The k-th order divided difference of a function g at the points $\{\tau_1, \ldots, \tau_k\}$ is defined as the leading coefficient of the unique polynomial of degree $k-1$ that interpolates g at these points. This divided difference is denoted by:*

$$[\tau_1, \ldots, \tau_k]g.$$

Divided Difference Properties. *The k-th divided difference possesses the following properties (in the absence of proof or reference, please refer to [dB01, Chapter 1]).*

1. ***Symmetry**: $[\tau_1, \ldots, \tau_k]g$ is a symmetric function in its arguments $\tau_1, \ldots, \tau_k$.*

2. ***Linearity**: Let f and g be functions and $\lambda, \mu \in \mathbb{R}$, then*

$$[\tau_1, \ldots, \tau_k](\lambda g + \mu f) = \lambda[\tau_1, \ldots, \tau_k]g + \mu[\tau_1, \ldots, \tau_k]f$$

holds.

3. ***Leibniz Rule**: Let f and g be functions, then*

$$[\tau_1, \dots, \tau_k](fg) = \sum_{r=1}^{k} [\tau_1, \dots, \tau_r] f [\tau_r, \dots, \tau_k] g$$

holds.

4. ***Constant Polynomials**: Let p_d be a polynomial of degree d, then*

$$[\tau_1, \dots, \tau_{d-1}] p_d$$

is constant with respect to the arguments $\tau_1, \dots, \tau_{d-1}$. Furthermore the relation

$$[\tau_1, \dots, \tau_k] p_d = 0, \quad \mathit{for} \quad k \geq d$$

holds.

5. ***Mean Value Theorem**: Let $g \in C^k[I]$, then*

$$[\tau_1, \dots, \tau_k] g = \frac{D^{k-1} g(\xi)}{(k-1)!}$$

holds for some $\xi \in I$.

6. ***Polynomial Interpolation**: Let p_{d-1} be the polynomial that interpolates g at the points $T = \{\tau_1, \dots, \tau_d\}$. Additionally, let τ_{d+1} be given, and let p_d be the polynomial that interpolates g at the points $T \cup \{\tau_{d+1}\}$. Then the following relation holds:*

$$p_d(x) = p_{d-1}(x) + \prod_{i=1}^{d} (x - \tau_i) [\tau_1, \dots, \tau_{d+1}] g.$$

To see this, note that $p_d - p_{d-1}$ vanishes at T, hence can be written in the form $C \prod_{i=1}^{d} (x - \tau_1)$, with $C = [\tau_1, \dots, \tau_{d+1}] g$. This follows immediately from the fact that p_d interpolates g at $T \cup \{\tau_{d+1}\}$, and from the definition of the divided difference, which gives the leading coefficient of p_d as $[\tau_1, \dots, \tau_{d+1}] g$.

7. ***Derivatives**: Let $g \in C^n$, then*

$$[\tau_1, \dots, \tau_n] g = D^n g(t), \quad \mathit{if} \quad \tau_1 = \tau_2 = \dots = \tau_n = t$$

holds.

8. ***Lagrange Interpolation**: Let g be a function, then for the k-th divided difference the relation*

$$[\tau_1, \dots, \tau_k] g = \sum_{j=1}^{k} \frac{g(\tau_j)}{\prod_{l=1:l \neq j}^{k} (\tau_j - \tau_l)}$$

holds. To see this, we apply equation (4.6) to express the interpolating polynomial p_k of g as $p_k = \sum_i g(\tau_i) l_i$, where l_i are the Lagrange basis polynomials. We then observe *that the leading monomial term of l_j is given by $\frac{x^{k-1}}{\prod_{l=1:l\neq j}^{k}(\tau_j-\tau_l)}$. Thus, the leading coefficient of $g(\tau_j)l_j$ is $\frac{g(\tau_j)}{\prod_{l=1:l\neq j}^{k}(\tau_j-\tau_l)}$. Finally, summing the leading monomial terms across all j yields the desired result.*

The application of the divided difference property 6 is currently underway to facilitate the observation of the following:

$$\begin{aligned} p_n(x) &= p_1(x) + (p_2(x) - p_1(x)) + \cdots + (p_n(x) - p_{n-1}(x)) \\ &= [\tau_1]g + (x-\tau_1)[\tau_1,\tau_2]g + \cdots + (x-\tau_1)\cdots(x-\tau_n)[\tau_1,\dots\tau_n]g, \end{aligned} \tag{4.7}$$

which is referred to as the Newton form (cf. [dB01, Chapter 1, pg. 4]). While this may seem promising for approximation, as we can now add the points $\{\tau_i\}_{1\leq i\leq n}$ one at a time, it is essential to exercise caution when selecting the interpolation points. To elucidate this point, consider the following relation:

$$\|\mathcal{P}_n g\|_\infty = \left\|\sum_i g(\tau_i)l_i(x)\right\|_\infty \leq \max_i |g(\tau_i)| \left\|\sum_i l_i\right\|_\infty \leq \|g\|_\infty \|\lambda_n\|_\infty,$$

where $\lambda_n := \sum_i^n |l_i|$, and the final two inequalities are derived from the triangle inequality (see Section 3.2). It can be observed that:

$$\|\lambda_n\|_\infty \sim \frac{2^n}{en\log(n)},$$

cf. [dB01, pg. 20], [Bru21, pg. 114]. This result suggests that $\mathcal{P}_n g$ may not adequately approximate the function g as n increases. This phenomenon can be understood by considering that $\|\mathcal{P}_n g\|_\infty$ could grow unbounded with n. A prominent illustration of this phenomenon is the Runge phenomenon (cf. [dB01, pg. 17]). Moreover, for the interpolation error e_n of a function $g : [a,b] \to \mathbb{R}$ by a polynomial $\mathcal{P}_n g$, we obtain:

$$e_n(x) = g(x) - \mathcal{P}_n g(x) = \mathcal{P}_n g(x) + \prod_{i=1}^{d}(x-\tau_i)[\tau_1,\dots,\tau_n,x]g - \mathcal{P}_n g(x) = \prod_{i=1}^{d}(x-\tau_i)[\tau_1,\dots,\tau_n,x]g,$$

where we have applied the divided difference property 6, which implies

$$g(x) = \mathcal{P}_n g(x) + \prod_{i=1}^{d}(x-\tau_i)[\tau_1,\dots,\tau_n,x]g.$$

Subsequently, invoking the mean-value divided difference property 5, we arrive at the following bound:

$$\operatorname{dist}(g,P_n) \leq \|g-\mathcal{P}_n g\|_\infty \leq \underbrace{\left\|\prod_{i=1}^{n}(\cdot-\tau_i)\right\|_\infty}_{(*)} \frac{\|D^n g\|_\infty}{n!}.$$

It has been well established that the so-called Chebyshev interpolation nodes minimize $(*)$ with respect to the choice of interpolation points $\{\tau_1,\dots,\tau_n\}$, achieving the value $2\left(\frac{b-a}{4}\right)^n$

(cf. [dB01, Chapter 2, pg. 23]). Consequently, we obtain the refined bound:

$$\operatorname{dist}(g, P_n) \leq \|g - \mathcal{P}_n g\|_\infty \leq 2\frac{(b-a)^n}{4^n n!}\|D^n g\|_\infty.$$

Thus, the approximation quality of polynomials of degree n for a function $g \in C[I]$ is influenced by both the smoothness of g and the length of the interval I. Prior to formulating a theorem that establishes a sharp bound on the approximation error for functions with particular regularity properties, it is necessary to introduce the concept of the modulus of continuity.

Definition 4.3 (Modulus of Continuity)**.** *Let $f : (X, d_X) \to (Y, d_Y)$ be a function. The modulus of continuity $\omega(f, \cdot) : \mathbb{R}^+ \to \mathbb{R}^+$ of f is defined as*

$$\omega(f, h) := \sup\{d_Y(f(x), f(x')) : x, x' \in X, d_X(x, x') \leq h\}.$$

In the case where $f \in C^1[I]$ (i.e., f is continuously differentiable on an compact interval I), the modulus of continuity is given by

$$\omega(f, h) = \sup\{|f(x) - f(x')| : x, x' \in I, |x - x'| < h\} \leq \|Df\|_{C[I],\infty} \cdot h, \tag{4.8}$$

where $\|Df\|_{C[I],\infty}$ denotes the supremum of the derivative Df on the interval I. If a function $f \in C[I]$ satisfies

$$\omega(f, h) \leq L \cdot h^\alpha,$$

for some constants $L > 0$ and $\alpha \in (0, 1)$, then f is said to be Hölder continuous with Lipschitz constant L and exponent α. The set of all such functions is denoted by $\mathrm{Lip}_{L,\alpha}$. For $L > 0$ and $\alpha = 1$ we say the corresponding function is Lipschitz continuous. We are now prepared to state the following theorem.

Theorem 4.1 (Jackson's Theorem [dB01, Theorem 22])**.** *Let $g \in C^r[a, b]$, then for $d \geq r + 2$, we have*

$$\operatorname{dist}(P_d[a, b], g) \leq C_r \frac{(b-a)^r}{d^r} \omega\left(D^r g, \frac{b-a}{2(d-r)}\right),$$

where $C_r \in \mathbb{R}$ is a constant dependent on r.

4.1.2 Piecewise Polynomials

As established in the previous section (cf. Theorem 4.1), the approximation error of a function $g : I \to \mathbb{R}$ by a polynomial depends significantly on the length of the interval over which it is defined and the degree of the polynomial used for approximation. Specifically, decreasing the length of the interval generally results in a reduced approximation error, and a similar effect can be achieved by increasing the degree of the approximating polynomial. However, given that the interval is generally fixed, an alternative approach involves partitioning it into smaller, compact, and connected subintervals. This approach enables for local approximation of the function using polynomials on each subinterval, which naturally gives rise to the concept of piecewise polynomial approximation. Suppose we partition the interval $I = [a, b]$ into k disjoint subintervals, denoted by $I_1, I_2, \ldots, I_k$, such that

$$\bigcup_{i=1}^{k} I_i = I,$$

we then approximate g on each subinterval I_i using a polynomial $p_{i,d} \in P_d[I_i]$. The resulting piecewise polynomial function is given by

$$pp(x) = \begin{cases} p_{i,d}(x), & \text{if } x \in I_i, \quad i = 1, \ldots, k. \end{cases}$$

It is evident that the space of all such piecewise polynomial functions has dimension $(d+1)k$. This assertion can be demonstrated through the consideration of functions of the following form:

$$f_{(d,k)}(x) = \begin{cases} x^d \text{ if } x \in I_k \\ 0 \text{ otherwise} \end{cases} .$$

It is noteworthy that both the partitioning of the interval into k uniform subintervals, and increasing the degree of the polynomial to kd, denoted as $\mathcal{P}_{kd}g$, result in a reduction of the approximation error bound from Theorem 4.1 by a factor of $(\frac{1}{k})^r$. However, piecewise polynomials offer several additional advantages in terms of both computational efficiency and numerical stability. From a computational perspective, evaluating a piecewise polynomial at a given point $x \in I$ requires storing and utilizing only $d+1$ coefficients per subinterval. In contrast, a single polynomial of degree kd requires the management of $kd+1$ coefficients. As the degree of the polynomial increases, numerical instability may arise due to excessive oscillations, particularly near the endpoints (cf. Section 4.1.1), [dB01, pg. 27]. Conversely, piecewise polynomial approximation offers considerably greater flexibility. The objective function is approximated locally within each subinterval, thereby eliminating the risk of global oscillations. This approach consequently yields more stable and accurate approximations. Additionally, the selection of subintervals is independent of the function space, thereby enabling the approximation of any continuous function by selecting sufficiently small subintervals. This approach stands in contrast to standard polynomial approximation, which is meticulous in its selection of interpolation points (cf. Section 4.1.1). The subsequent discussion will provide a formal definition of piecewise polynomials.

Definition 4.4 ([dB01, pg. 69])**.** *A piecewise polynomial of degree $d \in \mathbb{N}$ is a function $f : [\xi_1, \xi_{n+1}] \to \mathbb{R}$ of the form*

$$f(x) = p_{i,d}(x) \text{ if } \xi_i \leq x < \xi_{i+1}, i = 1, \ldots, n,$$

where $\xi := \{\xi_i\}_{1 \leq i \leq (n+1)}$ is a strictly increasing sequence of points and $\{p_{i,d}\}_{1 \leq i \leq n}$ are polynomials of degree at most d, that is $p_{i,d} \in P_d, i = 1, \ldots n$. The space of all such functions for fixed degree d and grid ξ is denoted by $P_{d,\xi}$.

In a manner analogous to standard polynomials, piecewise polynomials constitute a linear space. Nevertheless, they are not generally continuous at the partition points ξ_i. In addressing this issue, it is frequently necessary to impose continuity constraints by requiring a specified number of continuous derivatives at each partition point, denoted by $\nu_i \in \mathbb{N}$. The corresponding linear space is denoted by $P_{d,\xi,\nu} \subset P_{d,\xi}$ and is formally defined as follows:

$$P_{d,\xi,\nu} := \left\{ pp \in P_{d,\xi} \,\middle|\, D^{j-1}pp(\xi_i) \text{ exists for } i = 2, \dots, n,\ j = 1, \dots, \nu_i \right\}. \tag{4.9}$$

As stated in [dB01, pg. 84] the dimensionality of the specified space is determined by the following relation:

$$\dim P_{d,\xi,\nu} = kn - \sum_{i=2}^{n} \nu_i. \tag{4.10}$$

Here we have identified the piecewise polynomials as being smooth at the endpoints ξ_1 and ξ_{n+1} respectively. This ensures that any smoothness constraints are trivially satisfied at those points. Alternatively, one could consider the endpoints to be neglected, and define the first and last polynomials over the intervals $\{x \in \mathbb{R} \mid x \leq \xi_2\}$ and $\{x \in \mathbb{R} \mid x \geq \xi_n\}$, respectively. Functions in the space $P_{d,\xi,\nu}$, where $\nu_i \geq 1$ for all i, are commonly referred to as splines. In essence, splines can be defined as continuous piecewise polynomials. The objective at this juncture is to establish a suitable basis for the aforementioned space. Towards this aim, the following definition is proposed.

Definition 4.5 (B-Spline [dB01, pg. 87, Definition 1])**.** *Let $t := \{t_j\}$ be a non decreasing sequence. Then the j-th B-spline of degree d for the knot sequence t, denoted by $B_{j,d,t}$, is defined as follows:*

$$B_{j,d,t}(x) = (t_{j+d+1} - t_j)[t_j, \dots, t_{j+d+1}](\max(0, t - x))^d, \tag{4.11}$$

where the divided difference is to be understood with respect to the variable t.

B-Splines Properties. *B-splines possess the following properties:*

1. ***Small Support**: The function $B_{j,d,t}$ is nonzero only on the interval $I := [t_j, t_{j+d+1}]$ and vanishes elsewhere. This follows from the fact that*

$$f(t) = (\max(0, t - x))^d$$

is identically zero on I for $x > t_{j+d+1}$, and it is a polynomial of degree d on I for $x < t_j$ (cf. Figure 1), implying that its $(d+1)$-th order divided difference is zero by divided difference property 4.

2. ***Piecewise Polynomial Representation**: If the knots $\{t_j, \dots, t_{j+d+1}\}$ are distinct, then $B_{j,d,t} \in P_{d,t} \cap C^{(d-1)}$. Applying divided difference property 8 to Eq. (4.11), we obtain*

$$B_{j,d,t}(x) = \sum_{i=j}^{j+d+1} a_i \max(0, t_i - x)^d,$$

where

$$a_i = \frac{t_{i+d+1} - t_i}{\prod_{\substack{l \neq i \\ l=j}}^{i+d+1} (\tau_i - \tau_l)}.$$

Since $\max(0, t_j - x)^d$ belongs to $P_{d,t} \cap C^{(d-1)}$, the property follows, particularly at $x = t_j$. This argument follows the proof of [Flo23, Theorem 3.2].

The property extends to arbitrary knot sequences as follows, let t have multiplicities $\{m_i\}$, that is

$$t = \{\underbrace{t_1, \dots, t_1}_{m_1}, \dots, \underbrace{t_i, \dots, t_i}_{m_i}, \dots, \underbrace{t_n, \dots, t_n}_{m_n}\},$$

then $B_{j,d,t} \in P_{d,t}$ and is $(d - m_i)$-times continuously differentiable at t_i. This follows from divided difference property 7 and the fact that $\max(0, t_j - x)^{d-r}$ is $(d - r - 1)$-times continuously differentiable. If a knot appears more than d times, the corresponding B-spline is no longer continuous.

3. ***Partition of Unity**: The B-splines form a partition of unity, that is*

$$\sum_j B_{j,d,t}(x) = 1, \forall_{x \in [t_1, t_n]}$$

holds, see [Flo23, Section 4.2].

4. ***Linear Independence**: The B-splines form a linearly independent set, that is*

$$\sum_j a_j B_{j,d,t} = 0 \implies a_j = 0 \quad \forall j$$

holds, see [Flo23, Section 4.2].

5. ***Recurrence Relation**: B-splines satisfy the recurrence relation, that is*

$$B_{j,d,t}(x) = \omega_{j,d}(x) B_{j,d-1,t}(x) + (1 - \omega_{j+1,d}(x)) B_{j+1,d-1,t}(x),$$

where

$$\omega_{j,d,t}(x) = \frac{x - t_j}{t_{j+d} - t_j}.$$

For a proof, see [dB01, pg. 90].

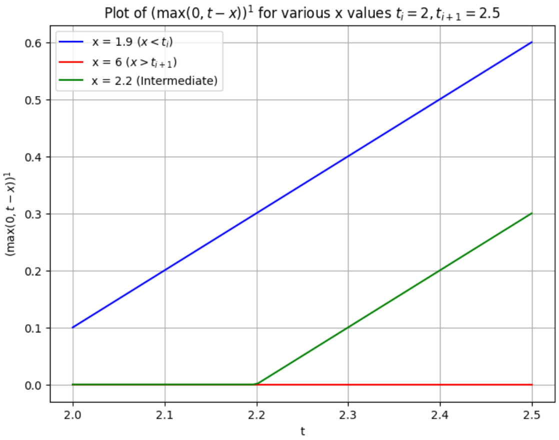


Figure 1: The function $f(t) = \max(0, t - x)$ over some interval $I := [t_i, t_{i+1}] = [2., 2.5]$ for three different fixed values of x. The blue and red functions are polynomials of degree at most 1 over I, the green function is not a polynomial, over I.

The ensuing discourse will pertain to the linear space:

$$S_{d,t} = \{\sum_j a_j B_{j,d,t} | a_j \in \mathbb{R}\}. \tag{4.12}$$

It is evident that $S_{d,t} \subset P_{d,\xi}$. However, a more thorough examination reveals that $S_{d,t} = P_{d,\xi,\nu}$. This equivalence is derived from the following theorem, which in turn provides a suitable basis for $P_{d,\xi,\nu}$ as desired.

Theorem 4.2 (Curry-Schonenberger [dB01, Chapter 9 Theorem 44])**.** *The set $\{B_{j,d,t}\}_j$ forms a basis for $P_{d,\xi,\nu}$, see Eq. (4.9), where the associated non decreasing sequence $\{t_i\}_{1 \le i \le \dim P_{d,\xi,\nu}}$ admits the form*

$$(t_{d+2}, \dots, t_n) = \{ \underbrace{\xi_2, \dots, \xi_2}_{d-\nu_2}, \underbrace{\xi_3, \dots, \xi_3}_{d-\nu_3}, \dots, \underbrace{\xi_{n-1}, \dots, \xi_{n-1}}_{d-\nu_{n-1}} \},$$

additionally

$$t_{d+1} \le \xi_1, \quad \textit{and} \quad \xi_n \le t_{n+1}.$$

For further calculations, we introduce another useful property, known as the dual functional property (cf. [dB01, pg. 102]). Specifically, for functions $f \in S_{d,t}$, it holds that

$$f = \sum_j \lambda_{j,d} f B_{j,d,t},$$

where for each j $\lambda_{j,d} : C^d[I] \to \mathbb{R}$ is the linear functional given by

$$\begin{aligned} \lambda_{j,d} f &= \sum_{r \le d} \omega_{j,r} f^{(r)}(\tau_i), \\ \text{with } \omega_{j,r} &= \frac{(-1)^{d-r} \psi_{j,d}^{(d-r)}(\tau_j)}{d!}, \\ \text{where } \psi_{j,d}(\tau_j) &= (t_{j+1} - \tau_j) \dots (t_{j+d} - \tau_j), \end{aligned} \tag{4.13}$$

and $t_j \le \tau_j \le t_{j+d+1}$. It is then evident that the following relation is established:

$$\lambda_{i,d}(\sum_j a_j B_{j,d,t}) = a_i,$$

as demonstrated in [dB01, pg. 102]. For the sake of comprehensiveness, it should be noted that the restriction of B-splines to at most $(d-1)$-times continuously differentiable at the knots in t (cf. B-Spline Property 2) is addressed by identifying $\tau_j \in t$ with the one-sided limits $\lim_{t \to \tau_j^+}$ or $\lim_{t \to \tau_j^-}$. This mitigates the theoretical issues that arise only at those points. The following definition of an approximation operator for smooth functions is proposed:

$$A_{t,d} : C^d[I] \to S_{d,t} : g \mapsto \sum_{j=1}^n (\lambda_{j,d} g) B_{j,d,t}, \tag{4.14}$$

where $\lambda_{j,d}$ is as in Eq. (4.13). The operator was initially investigated by De Boor and Fix [dBF73], which lead to the following theorem.

Theorem 4.3 ([dBF73, Section 3, Thm 2.1])**.** *Let $f \in C^d[I]$ and let $A_{t,d}$ be as in Eq. (4.14), then for the approximation error in the r-th derivative the following relation holds:*

$$\|D^r f - D^r A_{t,d} f\|_\infty \leq C_r \omega(f^d, \Delta t)\Delta t^{d-r}, r \leq d, \tag{4.15}$$

where C_r is some constant dependent on r and $\Delta t = \max_i t_i - t_{i-1}$.

This approximation error provides a striking estimate, as the distance, with respect to the metric induced by $\|\cdot\|_\infty$, between splines and smooth functions decreases at most as fast as Δt^{d+1} (when $D^d f$ is Lipschitz continuous). Furthermore, it can be demonstrated that for smooth functions, this rate of convergence is optimal and cannot be improved, except in trivial cases (cf. [dB01, Chapter 12, Theorem 10]). In the case of non-smooth functions, however, further analysis is necessary to ascertain the precise nature of this convergence. It has been determined [dB01, pg. 147] that the following relation holds:

$$\operatorname{dist}(C[I], S_{d,t}) \leq 2\omega\left(g; \frac{b-a}{\sqrt{2d}}\Delta t\sqrt{\frac{d+1}{12}}\right). \tag{4.16}$$

This provides an approximation error that decays at a rate of at most Δt^α for $\operatorname{Lip}_{L,\alpha}[I] \subset C[I]$. This result is derived from Schonberg's variation diminishing approximation operator [dB01, pg. 141], which is defined as follows:

$$Vg := \sum_j g(\tau^*_{jd}) B_{j,d,t}, \tag{4.17}$$

where $\tau^*_{jd} = \frac{t_{j+1}+\cdots+t_{j+d}}{d}$. In a subsequent section of this text, the moduli of continuity of splines (cf. Definition 4.3) will emerge as a particularly salient property. The subsequent theorem plays a pivotal role in its definition.

Theorem 4.4 ([dB01, pg. 116])**.** *For $d \geq 1$, the derivative operator $D : S_{d,t} \to S_{d-1,t}$ is given by*

$$Ds = D\sum_{i=1}^{n} a_j B_{j,d,t} = \sum_j d\frac{a_j - a_{j-1}}{t_{j+d} - t_j} B_{j,d-1,t}.$$

Accordingly, an upper bound for the first derivative of a spline $s \in S_{d,t} \mid_{(t_\mu, t_{\mu+1})}$ can be derived as follows:

$$\|Ds\|_\infty = \left\| D\sum_{i=1}^{n} a_j B_{j,d,t} \right\|_\infty = \left\| \sum_{j=\mu-d+1}^{\mu+1} d\frac{a_j - a_{j-1}}{t_{j+d} - t_j} B_{j,d-1,t} \right\|_\infty \tag{4.18}$$

$$\leq d \sum_{j=\mu-d+1}^{\mu+1} \frac{\Delta|a|}{d\Delta_{\min} t} = (d-1)\frac{\Delta|a|}{\Delta_{\min} t}, \tag{4.19}$$

where the equality in Eq. (4.18) is deduced directly from Theorem 4.4, along with a subsequent application of the B-spline property 1. In Eq. (4.19), we define $\Delta_{\min} t = \min_i t_i - t_{i-1}$ and $\Delta|a| = \max_i |a_i - a_{i-1}|$. The relation then follows immediately from B-spline property 3. This argument leads to the following lemma.

Lemma 4.1. *The functions $f \in S_{d,t}$ are Lipschitz continuous with constant $L_{d,t}$. The Lipschitz constant $L_{d,t}$ is upper bounded by*

$$L_{d,t} \leq (d-1)\frac{\Delta|a|}{\Delta_{min}t},$$

where Δ_{min} and $\Delta|\cdot|$ were defined directly above.

Proof. Applying the results obtained in Eq. (4.19) to Eq. (4.8) yields the assertion. □

It is evident that the operator V in Eq. (4.17) induces the functional $\lambda_{i,d} : f \mapsto f(\tau_{id}^*)$ (as in Eq. (4.13)). Therefore, when approximating the function f using the operator V, it can be observed that:

$$\Delta|a| = \max_i |\lambda_{i,d}f - \lambda_{i-1,d}f| = \max_i |f(\tau_{id}^*) - f(\tau_{i-1d}^*)| \leq \max_i \omega\left(f, \frac{t_{i+d} - t_i}{d}\right) \leq \omega\left(f, \Delta t\right).$$

Thus, it can be concluded that the Lipschitz constant of Vf is upper-bounded by:

$$L_{Vf} \leq (d-1)\frac{\omega\left(f, \Delta t\right)}{\Delta_{min}t}. \tag{4.20}$$

4.2 Neural Networks

In the subsequent discussion, the concept of density is to be understood in the context of the metric induced by the uniform norm, denoted by $\|\cdot\|_\infty$.

The subsequent section will focus on the approximation operator $\mathbf{S} : C[\mathbb{R}^n, \mathbb{R}^m] \to \mathcal{S}^{\sigma,l}[\mathbb{R}^n, \mathbb{R}^m]$, where $\mathcal{S}^{\sigma,l}[\mathbb{R}^n, \mathbb{R}^m]$ denotes the set of all shallow neural networks with width l. In order to define this operator, it is first necessary to introduce the functions $f \in \mathcal{S}^{\sigma,l}$. Following this introduction, several results will be established, the establishment of which is essential for the discussions that ensue in the following sections. This section builds upon the arguments presented in [Weg23]. The fundamental component of a neural network, the neuron, is defined first.

Definition 4.6 (Neuron [Weg23, Defintion 16.1])**.** *A function*

$$f : \mathbb{R}^n \to \mathbb{R} : x \mapsto \sigma(\langle w, x\rangle + b)$$

is called artifical neuron with weight vector $w = (w_1, w_2, \ldots, w_n) \in \mathbb{R}^n$, bias $b \in \mathbb{R}$ and activation function $\sigma : \mathbb{R} \to \mathbb{R}$.

A shallow neural network can thus be conceptualized as an ensemble of such neurons arranged in a layered configuration, thereby giving rise to the subsequent definition.

Definition 4.7 (Shallow Neural Network [Weg23, Definition 16.5])**.** *A shallow, fully-connected neural network of width l is a function of the form*

$$f : \mathbb{R}^n \to \mathbb{R}^m : x \mapsto A\begin{bmatrix} n_1(x) \\ n_2(x) \\ \vdots \\ n_l(x) \end{bmatrix}, \tag{4.21}$$

where $n_1, n_2, \dots, n_l$ are neurons and $A \in \mathbb{R}^{m\times l}$ is a matrix. Assuming that all neurons possess identical activation functions σ and that the application is comprehended elementwise, the function f in Eq. (4.21) can be expressed as follows:

$$f : x \mapsto A\sigma(Wx + b) = \mathit{Aff}_2 \circ \sigma \circ \mathit{Aff}_1(x),$$

where $W \in \mathbb{R}^{n\times l}$ represents the matrix obtained by stacking the weight vectors of the respective neurons, and the functions $\mathit{Aff}_k : \mathbb{R}^{a_k} \to \mathbb{R}^{a_{k+1}} : x \mapsto W_k x + b_k$ denote an affine transformation with $a = (n, l, m)$. When referring to a neural network with activation function σ, we mean a network in which each neuron applies the σ activation function.

We denote by $\mathcal{S}^{\sigma,l}[\mathbb{R}^n, \mathbb{R}^m]$ the set of all shallow fully-connected neural networks of width l, as per Definition 4.7. We use the convention $\mathcal{S}^{\sigma,l}[\mathbb{R}^n] := \mathcal{S}^{\sigma,l}[\mathbb{R}^n, \mathbb{R}]$. Furthermore, we denote by

$$\mathcal{S}^{\sigma}[\mathbb{R}^n, \mathbb{R}^m] := \bigcup_{l=1}^{\infty} \mathcal{S}^{\sigma,l}[\mathbb{R}^n, \mathbb{R}^m],$$

the set of all shallow neural networks, or equivalently the shallow neural networks with arbitrary width. We now focus on the set of functions that can be represented by a shallow neural network. A function f is said to be representable by a set of functions X if $\operatorname{dist}(f, X) \leq \epsilon$, for every $\epsilon > 0$ (cf. Eq. (3.1)). According to the aforementioned definition a set of functions X is said to be representable by another set Y if every function $f \in X$ is representable in Y, i.e $\operatorname{dist}(X, Y) \leq \epsilon$, (cf. Eq. (3.3)). This is tantamount to asserting that the set Y is dense in the set X. In the context of approximating functions using a set of approximation functions Y, the universal approximation property represents a strong form of representational power. This property ensures that the set of approximation functions Y can represent the entire set of continuous functions. In the specific case of shallow neural networks, we want $\mathcal{S}^{\sigma}[\mathbb{R}^n, \mathbb{R}^m]$ to be dense in $C[\mathbb{R}^n, \mathbb{R}^m]$, thereby providing the universal approximation property. It has been demonstrated that the analysis can be constrained to the case $S^{\sigma}[\mathbb{R}]$ in view of the following theorem.

Theorem 4.5 ([Weg23, Thm 16.12])**.** *Let $\sigma : \mathbb{R} \to \mathbb{R}$ be continuous and such that the shallow neural networks $\mathcal{S}^{\sigma}[\mathbb{R}]$ (cf. Definition 4.7) are dense in the continuous functions $C[\mathbb{R}]$ over compacta. Then $\mathcal{S}^{\sigma}[\mathbb{R}^n]$ is dense in $C[\mathbb{R}^n]$ over compacta.*

The statements in Theorem 4.5, combined with the intuitive fact that if $S^{\sigma}[\mathbb{R}^n]$ is dense in $C[\mathbb{R}^n]$, then $S^{\sigma}[\mathbb{R}^n, \mathbb{R}^m]$ is dense in $C[\mathbb{R}^n, \mathbb{R}^m]$ (see [Weg23, Proposition 16.8]), suggest that we can focus on the univariate case. It is demonstrated that a sufficient condition on the activation function, σ can be established to ensure the density of $\mathcal{S}^{\sigma}[\mathbb{R}]$ in $C[\mathbb{R}]$ over compacta. To establish this result, it is first necessary to introduce the concept of a discriminatory function.

Definition 4.8 ([Weg23, Definition 16.13])**.** *We say a borel-measurable function $\sigma : \mathbb{R} \to \mathbb{R}$ is discriminatory if for compact $\Omega \subset \mathbb{R}$ and for every complex borel measure $\mu \in M(\Omega)$ the following holds*

$$\forall_{w,b\in\mathbb{R}} : \int_{\Omega} \sigma(wx + b)d\mu = 0 \implies \mu = 0.$$

With this notion we can then formulate the sufficient condition for $S^\sigma[\mathbb{R}]$ to be dense in $C[\mathbb{R}]$, over compacta.

Lemma 4.2 ([Weg23, Lemma 16.17])**.** *$S^\sigma[\mathbb{R}] \subset C[\mathbb{R}]$ is dense on compacta, if $\sigma : \mathbb{R} \to \mathbb{R}$, is continuous and discriminatory.*

Lemma 4.2 emphasizes how crucial discriminatory activation functions are in the context of neural networks. Consequently, a number of examples of such functions are provided herein.

- ReLU: $\mathbb{R} \to \mathbb{R} : x \mapsto \max(0, x)$, for a proof that this function is in fact discriminatory, please see [Weg23, Corollary 16.21].
- Every bounded and measurable function of the form

  $$\sigma(t) = \begin{cases} 1 \text{ as } t \to \infty \\ 0 \text{ as } t \to -\infty \end{cases},$$

  is discriminatory, see [Cyb89, Lemma 1].

This indicates that a considerable proportion of functions indeed satisfy the desired density result, as stated in the aforementioned lemma. However, a practical challenge arises because it is not possible to use an arbitrarily large number of neurons in real-world applications. Notably, reordering the neurons can yield substantial performance enhancement, even when the number of neurons remains constant. This process is referred to as deep learning, denoted by the approximation operator $\mathbf{D} : C[\mathbb{R}^n, \mathbb{R}^m] \to \mathcal{D}^{\sigma,t}[\mathbb{R}^n, \mathbb{R}^m]$. The following definition formalizes this process.

Definition 4.9 (Deep Neural Network [Weg23, Definition 16.23])**.** *A deep, fully-connected neural network of depth t and width l is a function $f : \mathbb{R}^n \to \mathbb{R}^m$, which takes the form*

$$f = A \circ n^{t-1} \circ \cdots \circ n^1, \text{ where } n^k = \begin{bmatrix} n_1^k \\ n_2^k \\ \vdots \\ n_l^k \end{bmatrix}, \tag{4.22}$$

and n_i^k are neurons, as per Definition 4.6. If all neurons have the same activation σ, and we understand the application elementwise, then the function f in Eq. (4.22) has the form

$$\begin{aligned} f : x \mapsto & W_t\sigma\left(W_{t-1} \cdots \sigma\left(W_2\sigma\left(W_1 x + b_1\right) + b_2\right) \cdots + b_{t-1}\right) \\ &= \mathit{Aff}_t \circ \sigma_{t-1} \circ \cdots \circ \sigma_1 \circ \mathit{Aff}_1(x), \end{aligned} \tag{4.23}$$

where the function $\mathit{Aff}_k : \mathbb{R}^{a_k} \to \mathbb{R}^{a_{k+1}} : x \mapsto W_k x + b_k$, with $a = (n, l, \ldots, l, m)$, represents an affine transformation.

We denote by $\mathcal{D}^{\sigma,l,t}[\mathbb{R}^n, \mathbb{R}^m]$ the set of all deep neural networks (DNNs) of depth t and width l. The theoretical underpinnings of DNNs are predicated on the foundational principles of shallow neural networks, as articulated in the ensuing lemma.

Lemma 4.3 ([Weg23, Theorem 16.24])**.** *Let $f_{shallow} \in \mathcal{S}^{ReLU,l}[\mathbb{R}^d]$ be a shallow neural network with width l and $\Omega \subset \mathbb{R}^d_+$ compact. Then there exist a DNN $f_{deep} \in \mathcal{D}^{ReLU,d+3,l}[\mathbb{R}^d]$, such that*

$$f_{shallow}(x) = f_{deep}(x)\ \forall_{x\in\Omega}$$

As demonstrated in Lemma 4.3, it is possible to represent shallow neural networks as DNNs. Consequently, the conclusions about density that were previously established continue to apply (c.f Lemma 4.2). However, it is important to note that in constructing such a representation, many neurons may merely replicate the inputs or act as the zero function [Weg23, pg. 247]. Consequently, this representation can be regarded as suboptimal, indicating that in practical applications, a reduced depth and width might be sufficient to attain the desired outcomes. A subsequent section will elucidate that a notable property of a DNN is its modulus of continuity (cf. Defintion 4.3).

Lemma 4.4. *The DNNs $f_{deep} \in \mathcal{D}^{\sigma,l,t}[\mathbb{R}^n, \mathbb{R}^m]$, as defined in Defintion 4.9, are Lipschitz continuous with constant $L_{f_{deep}}$ if the activation functions $\{\sigma_i\}$ are Lipschitz continuous. The Lipschitz constant $L_{f_{deep}}$ is upper bounded by*

$$L_{f_{deep}} \leq L_{Aff_t} \prod_{i=1}^{t-1} L_{Aff_i} L_{\sigma_i},$$

where L_{Aff_i} and L_{σ_i} represent the Lipschitz constants of the affine transformation functions Aff_i and the activation functions σ_i, respectively (cf. Eq. (4.23)).

Proof. Before giving the proof we state a technical lemma.

Lemma 4.5. *Let $g : (X, d_X) \to (Y, d_Y)$ and $f : (Y, d_Y) \to (Z, d_Z)$ be Lipschitz with constant L_g and L_f respectively. Then $f \circ g \in Lip_{L_{f\circ g}}$ with*

$$L_{f\circ g} \leq L_f L_g$$

Proof. By definition, we observe that

$$d_Z(f \circ g(x), f \circ g(x')) \leq L_f\, d_Y(g(x), g(x')) \leq L_f L_g\, d_X(x, x'),$$

which completes the assertion. □

Using Lemma 4.5 on Eq. (4.23) yields the assertion. □

An immediate consequence of Lemma 4.5 is that L_{Aff_i} and L_{σ_i} can be computed with respect to any metric induced by an L_p-norm on the respective spaces. In which case we have that $L_{\text{Aff}_i} = \|W_i\|_p$, where $\text{Aff}_i : x \mapsto W_i x + b$ and $\|\cdot\|_p$ denotes the operator norm on the respective space. However, it is important to note that the bound in the aforementioned lemma is generally not tight.

5 Multivariate Functions Represented as Compositions of Superpositions of Univariate Functions

The subsequent section is devoted to the presentation of the primary outcomes of this thesis. As delineated in the introduction of Section 3, the present study proposes an extension to the theory of univariate function approximation, thereby enabling the formulation of more general statements concerning a particular set of multivariate functions.

5.1 Definition of the Set of Functions $\mathcal{F}_{X,I,n}$

This section delineates the set of functions, called multivariate functions represented as compositions of superpositions of univariate functions, which we aim to approximate. In order to define a multivariate function $f : I_1^{n_1} \to I_N^{n_N}$ represented as a composition of superpositions of univariate functions, it is first necessary to define the univariate functions that serve as the building blocks. Subsequent to this, superpositions are constructed, aggregating their contributions to yield the desired composition. For the univariate functions consider:

$$\begin{gathered} \Psi_{l,k,j} : I_l \to \mathbb{R}, \\ I_l \subseteq I, \quad l \in \{1, \dots, N-1\}, \quad k \in \{1, \dots, n_{l+1}\}, \quad j \in \{1, \dots, n_l\}, \\ N = |n|, \end{gathered} \tag{5.24}$$

where $n = (n_1, \dots, n_N)$ is a tuple with $N \geq 2$ and $n_i \in \mathbb{N}, i = 1, \dots, N$. I is a compact set. As previously mentioned, the objective is to apply compositions to these functions, which necessitates a constraint on the co-domain of the functions $\Psi_{l,k,j}$. To elaborate, the constraint stipulates the existence of sets $I_1, I_2, \dots I_{N-1} \subseteq I$ such that

$$\Psi_{l,k,1}(I_l) + \Psi_{l,k,2}(I_l) + \dots + \Psi_{l,k,n_l}(I_l) \subseteq I_{l+1} \tag{5.25}$$

$$l = 1 \dots N-1, \ j = 1, \dots n_l$$

holds. The application of the superposition principle yields the following expression for the function Φ_l^k:

$$\Phi_l^k : I_l^{n_l} \to I_{l+1} : x \mapsto \sum_{j=1}^{n_l} \Psi_{l,k,j}(x_j), \tag{5.26}$$

which, from the previous context, makes the constraint in Eq. (5.25) evident. Subsequently, the functions Φ_l^k are arranged in a layered configuration to establish the mapping Φ_l. The formal expression is as follows:

$$\Phi_l : I_l^{n_l} \to I_{l+1}^{n_{l+1}} : x \mapsto (\Phi_l^1(x), \dots, \Phi_l^{n_{l+1}}(x)). \tag{5.27}$$

Sometimes we refer to the functions Φ_i as ”inner” functions. It is finally posited that a function $f : I^{n_1} \to I^{n_N}$ is a multivariate function represented as a composition of superpositions of univariate functions, if it can be written as

$$f = \bigcirc_{l=1}^{N-1} \Phi_l, \tag{5.28}$$

for some tuple n and some compact interval I. It is important to acknowledge that I_N is not necessarily a subset of I (see Eq. (5.25)). Rather, I_N is an arbitrary set because we can refrain from enforcing any conditions on I_N, as Φ_{N-1}^k is not applied to a subsequent function. It is important to note that the compactness of I_N is contingent upon the continuity of each function $\Psi_{l,k,j}$, as evidenced by the compactness of I and the observation that the image of a continuous function, when considered over a compact set, is itself compact. As well as the fact that the sum of compact sets in a finite-dimensional space results in another compact set. The set of all such functions is denoted by

$$\mathcal{F}_{I,(n_1,\ldots,n_N)} := \{f : I_1^{n_1} \to I_N^{n_N} | f(x) = (\bigcirc_{l=1}^{N-1} \Phi_l)(x), \text{ as per Eq. } ((5.24) \text{ - } (5.28))\}. \tag{5.29}$$

If the tuple n is understood, we write $\mathcal{F}_{I,n} := \mathcal{F}_{I,(n_1,\ldots,n_N)}$. In certain circumstances, it can be advantageous to delineate the characteristics of functions in terms of the number of "inner" functions, denoted by $N-1$ (cf. Eq. (5.28)). We write $f_{N-1} \in \mathcal{F}_{I,n}$ if $|n| = N$. More specifically for a function $f \in \mathcal{F}_{I,n}$, f_j denotes the composition of the first j inner functions. That is to say:

$$f = \Phi_{N-1} \circ \Phi_{N-2} \circ \cdots \circ \underbrace{\Phi_j \circ \cdots \circ \Phi_1}_{f_j}. \tag{5.30}$$

Figure 2 provides a graphical representation of a function $f_3 \in \mathcal{F}_{I,(2,3,3,1)}$.

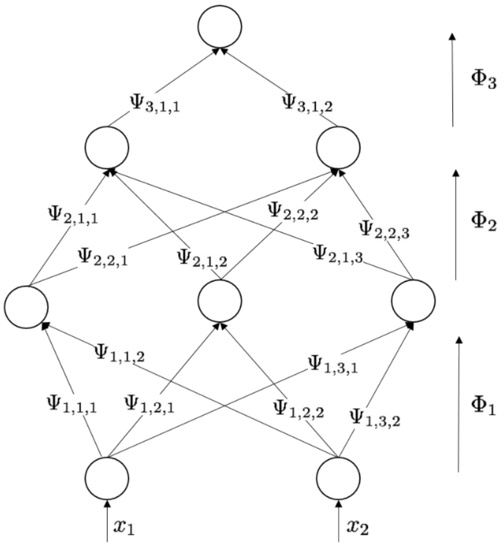


Figure 2: Graphical representation of a function $f \in \mathcal{F}_{I,(2,3,2,1)}$. Each arrow represents a function $\Psi_{l,k,j}$ applied to an input, which is depicted by a circle. The graph is read from bottom to top, where multiple lines joining at a circle in the upward direction indicate the application of a summation operation.

The following are some examples of functions $f \in \mathcal{F}_{I,n}$.

1. Obviously, all univariate functions over a compact domain I are in $\mathcal{F}_{I,(1,1)}$
2. By [Buc79, Theorem 6] we have that any function defined over a compact n-cell I^n is in $\mathcal{F}_{I,(n,1,1)}$, that is
$$\mathcal{F}[I^n] = \mathcal{F}_{I,(n,1,1)}[I^n]. \tag{5.31}$$
The set $\mathcal{F}_{I,(n,1,1)}$ is referred to as the set of nomographic functions [Buc79] . The set $\mathcal{F}[I^n]$ denotes the set of all functions over I^n.

It is imperative to recognize that the objective of this study is to approximate the function $f \in \mathcal{F}_{I,n}$ by approximating each univariate function $\Psi_{l,k,j}$. The quality of this approximation is significantly influenced by the following factors:

1. The properties of the individual functions $\Psi_{l,k,j}$, as for example the functions $\Psi_{l,k,j}$ in Eq. (5.31) may even be discontinuous and therefore unfeasible for approximation. To streamline the ensuing discussion, we introduce the following notation that highlights the key properties of the individual univariate functions $\Psi_{l,k,j}$:
$$\mathcal{F}_{X,I,n} = \{f : I_1^{n_1} \to I_N^{n_N} | f(x) = (\bigcirc_{l=1}^{N-1} \Phi_l)(x), \text{ as per Eq. } (5.24 \text{ - } 5.28), \Psi_{l,k,j} \in X\}, \tag{5.32}$$
where X denotes a predefined set of functions, such as continuous functions, Lipschitz continuous functions, or smooth functions. It is posited that a set of functions, denoted by K, admits a representation of type X if, for some tuple n and a compact interval I, we can write $K \subseteq \mathcal{F}_{X,I,n}$.
2. The method employed for approximating univariate functions, as some approximation techniques can yield substantially different results depending on the properties of the function $\Psi_{l,k,j}$ being approximated (cf. Section 4). To streamline the ensuing discourse, let $A : X \to Y$ be an approximation operator for univariate functions, as defined in Section 3.3. We subsequently proceed to define an approximation operator for functions $f \in \mathcal{F}_{X,I,n}$, as
$$\mathbf{A} : \mathcal{F}_{X,I,n} \to \mathcal{F}_{Y,I,n}, \tag{5.33}$$
where $\mathbf{A}$ is the composition- and component-wise application of the operator A. For functions $f \in \mathcal{F}_{X,I,n}$, this is tantamount to applying the operator A to each $\Psi_{l,k,j}$ individually. We say the operator A induces $\mathbf{A}$.

It is finally possible to quantify the quality of approximation of a function $f \in \mathcal{F}_{C[I],I,n}$ by $\|f - \mathbf{A}f\|_{\mathcal{F}_{C[I],I,n},\infty}$, where we have defined the normed linear space:
$$\begin{gathered}(C[I_1^{n_1}, I_N^{n_N}], \|\cdot\|_{\mathcal{F}_{C[I],I,n},\infty}),\\ \text{where } \|f\|_{\mathcal{F}_{C[I],I,n},\infty} = \sup_{x \in I_1^{n_1}} \|f(x)\|_{I_N^{n_N},\infty}.\end{gathered} \tag{5.34}$$

The fact that $\|f\|_{\mathcal{F}_{C[I],I,n},\infty}$ indeed defines a norm, as per the axioms outlined in Section 3.2, is discussed in Appendix A.5.1. The subsequent discussion will focus on several examples of functions in the set $\mathcal{F}_{X,I,n}$.

1. Every function $f \in C[I^n]$ belongs to the set $\mathcal{F}_{C[I],I,(n,2n+1,1)}$, which follows directly from the Kolmogorov-Arnold theorem [Kol57]. This theorem posits that any continuous function can be expressed as a superposition of univariate continuous functions. Specifically, for any continuous function $f : [0,1]^n \to \mathbb{R}$, there exist univariate continuous functions $\phi_{q,p} : [0,1] \to \mathbb{R}$ and $\Phi_q : \mathbb{R} \to \mathbb{R}$ such that the relation

$$f(\mathbf{x}) = f(x_1, \ldots, x_n) = \sum_{q=0}^{2n} \Phi_q \left(\sum_{p=1}^{n} \phi_{q,p}(x_p) \right) \tag{5.35}$$

   holds. In fact each of inner functions $(\phi_{q,p})$ are independent of f. Since $\phi_{q,p}$ is a continuous function defined on a compact domain for each q and for all p, its image remains compact. Consequently, Φ_q, which is applied to the sum of such functions, also has a compact domain. This is because, in finite-dimensional spaces, the sum of compact sets is compact. However, the Kolmogorov–Arnold theorem does not guarantee any desirable properties—such as smoothness—for the resulting univariate functions. In fact, these univariate functions often exhibit highly irregular behavior and may even have fractal-like properties.

2. Every function $f \in \mathcal{S}^{ReLU,l}[\mathbb{R}^n]$ (cf. Section 4.2) is in $\mathcal{F}_{S_{1,t},I,(n,l,1)}$, with $|t| = 3$. Recall that $S_{1,t}$ denotes the space of continuous piecewise polynomials of degree 1 (cf. Section 4.1.2). This can be easily verified by directly defining the mappings Φ_1 and Φ_2 (cf. Eq. (5.27)). We define Φ_1, as

$$\Phi_1 : x \mapsto \{\sum_{j=1}^{n} w_{1,j}x_j + \frac{b_1}{n}, \sum_{j=1}^{n} w_{2,j}x_j + \frac{b_2}{n}, \cdots, \sum_{j=1}^{n} w_{l,j}x_j + \frac{b_l}{n}\},$$

   where $w_{i,j} \in \mathbb{R}$ and $b_i \in \mathbb{R}$. We then define Φ_2, as

$$\Phi_2 : (z_1, \ldots z_l) \mapsto \sum_{j=1}^{l} a_j ReLU(z_j),$$

   where $a_j \in \mathbb{R}$. Combining this with the fact that $a_j\,\mathrm{ReLU} \in S_{1,t}$ for all $a_j \in \mathbb{R}$ and $t = (t_1, 0, t_3)$, we obtain the desired result.

In the following examples, we provide both an analytical as well as a polynomial representation of multivariate polynomials, utilizing the developed framework.

3. Any multivariate polynomial $p \in P_d[a,b]^m$ where $a > 0$ is an element of $\mathcal{F}_{C^\infty[I],I,(m,\binom{d+m}{m}-1,1)}$.

This can be easily verified by directly defining the mapping Φ_1 (cf. Eq. (5.27)), as

$$\Phi_1 : x \mapsto \{\sum_{j=1}^{m_1} a_{1,j}\log(x_j), \sum_{j=1}^{m_2} a_{2,j}\log(x_j), \dots, \sum_{j=1}^{m_n} a_{\binom{d+m}{m},j}\log(x_j)\}$$

$$\text{such that for all } i = 1, \dots, \binom{d+m}{m} - 1 \sum_{j=1}^{m_i} a_{i,j} \le d, a_{i,j} \in \mathbb{N},$$

$$\text{and the mapping } \Phi_2 \text{, as}$$

$$\Phi_2 : x \mapsto \sum_j \gamma_j \exp x_j,$$

where each sum in the mapping Φ_1 represents a monomial term. This follows directly from the identity

$$\prod_{i=1}^{n} x_i^{\alpha_i} = \exp\left(\sum_{i=1}^{n} \alpha_i \log(x_i)\right).$$

The mapping Φ_2, is then employed to apply the corresponding coefficients γ_j. We note that the constant monomial has been neglected; however, by adding its coefficient to any of the univariate functions in Φ_2, it can be easily incorporated. To enumerate the monomials in a polynomial of degree d, we examine the following equation:

$$a_1 + a_2 + \dots + a_n = k,$$

which has exactly $\binom{k+n-1}{n-1}$ integer solutions for the a_i's. By summing over all possible values of k, we obtain the following identity:

$$\sum_{k=0}^{d} \binom{k+n-1}{n-1} = \binom{d+n}{n},$$

which is a well-known identity in combinatorics. Thus providing the total number of monomials in a polynomial of degree d in n variables.

4. Before presenting the subsequent example, it is necessary to state a technical lemma.

Lemma 5.1. *There exist exactly $\binom{d-1}{w}$ tuples $\{x_1, x_2, \dots, x_w\} \subset \mathbb{N}$ of cardinality w, such that $\sum_{i=1}^{w} x_i < d$.*

Proof. We refer the reader to Appendix A.5.4. □

Any multivariate polynomial $p \in P_d[I^n]$ belongs to the space $\mathcal{F}_{P_d[I],I,(n,\sum_{w=1}^{d} 2^w \binom{n}{w}\binom{d}{w},1)}$. This can be verified as follows. Initially, the subsequent crucial identity is presented:

$$C\prod_{i=1}^{n} x_i = \sum_{S \subseteq \{x_1,\dots,x_n\}} (-1)^{n-|S|} \left(\sum_{x_i \in S} x_i\right)^n, \tag{5.36}$$

for a proof, the reader is referred to Appendix A.5.2. We proceed to define the set S as the collection of all univariate monomials with coefficient 1, formed from the variables $\{x_1, \dots, x_n\}$, up to degree d, excluding the constant term; that is,

$$S = \{x_1^1, x_1^2, \dots, x_1^d, \dots, x_n^1, x_n^2, \dots, x_n^d\}.$$

In the subsequent step, we identify suitable subsets of S according to Eq. (5.36). These subsets are selected such that the total sum of their respective elements' exponents is less than or equal to d. This set is denoted by $\mathcal{T}_d$. Formally, we obtain

$$\mathcal{T}_d(S) = \left\{ \mathcal{P}(S') \mid S' \subseteq S, \sum_{y_i \in S'} \text{exponent}(y_i) \leq d \right\},$$

with *exponent* $: \mathbb{R} \to \mathbb{N} : x_i^k \mapsto k$. For the sake of clarity, it should be noted that each y_i in the aforementioned equation represents an element $x_i^k \in S$. We note that $|\mathcal{T}_d(S)| \leq \sum_{w=1}^d \binom{n}{w}\binom{d}{w}$, where w can be conceptualized as the quantity of elements contained within the appropriate subsets $S' \subset S$. $\binom{d}{w}$ represents the number of possible exponent combinations for the w-variables that satisfy the degree condition, which directly follows from Lemma 5.1. By $\mathcal{T}_d^i(S)$ we denote the i-th element in $\mathcal{T}_d(S)$ and $\mathcal{P}$ denotes the power set. Now by defining the tuple $T_{d,flat}(S) := \bigcup_i \mathcal{T}_d^i(S)$ we can define, using Eq. (5.36), the mapping Φ_1 as in Eq. (5.27), as

$$\Phi_1 : x \mapsto \left\{\Phi_1^1(x), \dots \Phi_1^N(x)\right\},$$
$$\text{with } \Phi_1^i : \{x_1, \dots, x_n\} \mapsto \left\{ \sum_{y \in T_{d,flat}^i(S)} y \right\},$$

where $N = |\mathcal{T}_{d,flat}(S)| \leq \sum_{w=1}^d 2^w \binom{n}{w}\binom{d}{w}$. Finally, we define the second mapping Φ_2 as

$$\Phi_2 : \{x_1, \dots, x_n\} \mapsto \sum_k C'_k x_k^{d'}, \quad d' \leq d.$$

The reconstructed polynomial has coefficients $\alpha_i = C'_k C$. Note that C is the same for all terms reconstructing a monomial, as indicated in Eq. (5.36). Here, d' denotes the number of terms in the corresponding multivariate monomial (n in Eq. (5.36)).

Remark 5.1. *The latter example is particularly noteworthy because it generalizes the former by representing polynomials over arbitrary compact intervals. Moreover, it utilizes univariate polynomials, which occupy a distinctive position in approximation theory. These polynomials are known for their ease of approximation and represent one of the simplest classes of functions used for approximating more complex functions (see Section 4.1.1). Finally, the famous Stone-Weierstrass theorem (see for example [Wer18, Theorem VIII.4.7, pg. 455]) ensures the validity of the following statement:* $\bigcup_{d=1}^\infty P_d[I^n]$ *is dense in* $C[I^n]$*, and consequently,*

$$\bigcup_{d=1}^{\infty} \mathcal{F}_{P_d,I,(n,\sum_{w=1}^d 2^w \binom{n}{w}\binom{d}{w},1)},$$

is dense in $C[I^n]$*, offering a universal approximation property, analogous to that of neural networks (cf. Section 4.2). An extension to deeper representation, similar to the one* *in Lemma 4.3, is straightforward. This extension is achieved by copying the inputs to subsequent layers via the identity function. The identity function $I : x \mapsto x$ is a polynomial. This representation is particularly advantageous compared to the one given in Example 1 in that it expresses functions through "simple" univariate functions rather than the highly intricate form presented in Example 1.*

The dedicated reader might argue that the universal approximation property is already implied by Example 2. However, Examples 3 and 4 underscore the utility of more expressive univariate functions, emphasizing their role in capturing nuances beyond the mere composition of affine transformations and nonlinear activations.

5.2 Approximation of Functions in $\mathcal{F}_{Lip_L,I,n}$

Pursuant to the definitions established in the preceding Section 5.1, we are now prepared to proceed with the articulation of the ensuing theorem.

Theorem 5.1 (Approximation Theorem)**.** *For any $f \in \mathcal{F}_{Lip_L,I,n}$ and any approximation operator A on $Lip_L[I]$, we have the inequality,*

$$\|f - \mathbf{A}f\|_{\mathcal{F}_{C[I],I,n},\infty} \leq C_{L,n}\,\mathrm{dist}(Lip_L, A),$$

where $C_{L,n}$ is a constant depending on the Lipschitz-constant L of function $\Psi_{l,k,j} \in Lip_L[I] \subset C[I]$ and on the tuple n. The distance measure dist *is defined in Eq. (3.4) and is calculated with respect to the metric induced by $\|\cdot\|_\infty$. The operator $\mathbf{A}$ is induced by A, cf. Eq. (5.33).*

Proof. We prove the statement using induction on $N-1$, where N denotes the cardinality of the tuple n.

Base Case:

We shall now proceed to derive the distance between f_1[1] and $\mathbf{A}f_1$, measured in the norm $\|\cdot\|_{\mathcal{F}_{C[I],I,(n_1,n_2),\infty}}$. Consequently, the subsequent relation is derived:

$$\|f_1 - \mathbf{A}f_1\|_{\mathcal{F}_{C[I],I,(n_1,n_2)}\infty} = \|\Phi_1 - \mathbf{A}\Phi_1\|_{\mathcal{F}_{C[I],I,(n_1,n_2),\infty}} \tag{5.37}$$

$$= \sup_{k\in\{1,\dots,n_2\}} \|\Phi_1^k - \mathbf{A}\Phi_1^k\|_{\mathcal{F}_{C[I],I,(n_1,1),\infty}} \tag{5.38}$$

$$= \sup_{k\in\{1,\dots,n_2\}} \sup_{x\in I_1^{n_1}} |\sum_{j=1}^{n_1} \Psi_{1,k,j}(x_j) - A\Psi_{1,k,j}(x_j)| \tag{5.39}$$

$$\leq n_1 \sup_{\substack{k\in\{1,\dots,n_2\}\\ j\in\{1,\dots,n_1\}}} \|\Psi_{1,k,j} - A\Psi_{1,k,j}\|_{C[I],\infty} \tag{5.40}$$

$$\leq n_1\,\mathrm{dist}(Lip_L, A), \tag{5.41}$$

where the equality in Equation (5.37) follows directly from the definition of the set $\mathcal{F}_{\mathrm{Lip}_L,I,n}$ (cf. Section 5.1). The equivalence between Eq.s (5.37) and (5.38) follows immediately from the definition of the norm $\|\cdot\|_{\mathcal{F}_{C[I],I,(n_1,n_2),\infty}}$ (see Eq. (5.34)). By said definition, this is also equal to the expression in Eq. (5.39). Applying the triangle inequality to Eq. (5.39) yields

[1] This notation is defined in Eq. (5.30))

Eq. (5.40), and finally, the inequality to Eq. (5.41) follows from the definition of the distance measure dist (cf. Eq. (3.4)).

Induction Hypothesis:

We define the residue R_m as

$$R_m = f_m - \mathbf{A} f_m.$$

The induction hypothesis, is then given by the following inequality:

$$\|R_m\|_{\mathcal{F}_{C[I],I,(n_1,\dots,n_{m+1}),\infty}} \leq C'_{L,n} \operatorname{dist}(Lip_L, A),$$

where $C'_{L,n}$ is a constant, as in the statement of the theorem.

Induction Step:

We shall now proceed to derive an upper bound on $\|R_{m+1}\|_{\mathcal{F}_{C[I],I,n,\infty}}$. Consequently, the subsequent relation is derived:

$$\begin{aligned} \|R_{m+1}\|_{\mathcal{F}_{C[I],I,n,\infty}} &= \|f_{m+1} - \mathbf{A} f_{m+1}\|_{\mathcal{F}_{C[I],I,n},\infty} \\ = \|\Phi_{m+1} \circ f_m - \mathbf{A}(\Phi_{m+1} \circ f_m)&\|_{\mathcal{F}_{C[I],I,(n_1,\dots,n_{m+1},n_{m+2}),\infty}} \end{aligned} \tag{5.42}$$

$$= \sup_{k\in\{1,\dots,n_{n+2}\}} \sup_{x\in I_1^{n_1}} |\sum_{j=1}^{n_{m+1}} \Psi_{m+1,k,j} \circ f_m^j(x) - \mathbf{A}\left(\Psi_{m+1,k,j} \circ f_m^j(x)\right)| \tag{5.43}$$

$$\leq n_{m+1} \sup_{\substack{k\in\{1,\dots,n_{m+2}\} \\ j\in\{1,\dots,n_{m+1}\}}} \|\Psi_{m+1,k,j} \circ f_m^j - \mathbf{A}(\Psi_{m+1,k,j} \circ f_m^j)\|_{\mathcal{F}_{C[I],I,(n_1,\dots n_{m+1},1),\infty}}, \tag{5.44}$$

where the relation in Eq. (5.42) is derived form the utilization of the fact that any $f_{m+1} \in \mathcal{F}_{C[I],I,(n_1,\dots n_{m+2})}$ can be expressed as $f_{m+1} = \Phi_{m+1} \circ f_m$ (cf. Eq. (5.30)). Subsequently, the equivalence to Eq. (5.43) follows in a manner similar to the base case. The notation f_m^j refers to the j-th component of f_m. Finally, the inequality between Eq. (5.43) and Eq. (5.44) follows by applying the triangle inequality, analogous to the base case. We shall now focus on the term $\|\cdot\|_{\mathcal{F}_{C[I],I,(n_1,\dots n_{m+1},1)}}$ in Eq. (5.44). By adding 0 and applying the triangle inequality, we obtain the following inequality:

$$\begin{aligned} &\left\|\Psi_{m+1,k,j} \circ f_m^j - \mathbf{A}(\Psi_{m+1,k,j} \circ f_m^j) + \Psi_{m+1,k,j} \circ \mathbf{A} f_m^j - \Psi_{m+1,k,j} \circ \mathbf{A} f_m^j\right\|_{\mathcal{F}_{C[I],I,(n_1,\dots,n_{m+1},1),\infty}} \\ \leq &\underbrace{\left\|\Psi_{m+1,k,j} \circ f_m^j - \Psi_{m+1,k,j} \circ \mathbf{A} f_m^j\right\|_{\mathcal{F}_{C[I],I,(n_1,\dots,n_{m+1},1),\infty}}}_{\Box} \\ &+ \underbrace{\left\|\Psi_{m+1,k,j} \circ \mathbf{A} f_m^j - \mathbf{A}(\Psi_{m+1,k,j} \circ f_m^j)\right\|_{\mathcal{F}_{C[I],I,(n_1,\dots,n_{m+1},1),\infty}}}_{\triangle}. \end{aligned}$$

We now proceed to derive an upper bound by analyzing each summand separately.

1. For □ we obtain the upper bound

$$\begin{aligned} &\|\Psi_{m+1,k,j} \circ f_m^j - \Psi_{m+1,k,j} \circ \mathbf{A} f_m^j\|_{\mathcal{F}_{C[I],I,(n_1,\dots n_{m+1},1)}} \\ &\leq \omega\left(\Psi_{m+1,k,j}, \|f_m^j - \mathbf{A} f_m^j\|_{\mathcal{F}_{C[I],I,(n_1,\dots n_m,1),\infty}}\right) \\ \leq L\|f_m^j - \mathbf{A} f_m^j\|_{\mathcal{F}_{C[I],I,(n_1,\dots n_m,1)}} &\leq C''_{L,n_{m+1}} \|R_m\|_{\mathcal{F}_{C[I],I,(n_1,\dots n_m,n_{m+1}),\infty}}, \end{aligned} \tag{*}$$

where ω was already defined before (cf. Definition 4.3) and the last inequality follows from the fact that

$$\|f_m^j - \mathbf{A} f_m^j\|_{\mathcal{F}_{C[I],I,(n_1,\dots n_m,1),\infty}} \leq \|f_m - \mathbf{A} f_m\|_{\mathcal{F}_{C[I],I,(n_1,\dots n_{m+1}),\infty}}$$

2. For $\triangle$ we obtain, analogous to the Base Case, the upper bound

$$\|\Psi_{m+1,k,j} \circ \mathbf{A} f_m^j - \mathbf{A}(\Psi_{m+1,k,j} \circ f_m^j)\|_{\mathcal{F}_{C[I],I,(n_1,\dots n_{m+1},1),\infty}} \leq \operatorname{dist}(Lip_L, A). \qquad (**)$$

The Inequalities (*) and (**), together with the induction hypothesis, yield the assertion. □

Theorem 5.1 essentially states that if each of the functions $\Psi_{l,k,j}$ is Lipschitz continuous, then the approximation error, resulting from approximating each $\Psi_{l,k,j}$ individually, increases linearly with $\operatorname{dist}(Lip_L[I], A)$. A slightly converse statement is made in the following remark.

Remark 5.2. *For any approximation operator $\mathbf{A}$ (cf. Eq. (5.33)) induced by a operator of the form $A_L : C[I] \to \mathrm{Lip}_L[I]$, the statements of Theorem 5.1 remain valid for functions $f \in \mathcal{F}_{C[I],I,n}$, except that the constant $C_{L,n}$ now depends on the Lipschitz constant L of the approximation operator A_L. It is noteworthy that all approximation operators presented in this thesis are Lipschitz-continuous (cf. Lemma 4.1 and Lemma 4.4). For a proof of the remark we refer the reader to Appendix A.5.3*

The statement of Remark 5.2 may appear promising; however, it should be noted that, in general, the Lipschitz constant of an approximating function increases with the complexity of the target function. To illustrate, the approximation of Hölder-continuous functions with splines and an approximation error of at most $\gamma > 0$ necessitates a finer grid t (cf. Eq. (4.16)) compared to smooth functions (cf. Theorem 4.3). As demonstrated in Lemma 4.1 and Eq. (4.20), this results in a larger Lipschitz constant. A similar phenomenon occurs in the case of neural networks approximating a more complex function (e.g., Hölder-continuous functions) to some degree of accuracy necessitates a greater number of neurons, which in turn leads to a larger Lipschitz constant (cf. Lemma 4.4). We now proceed to provide an example that aims to place the work of Liu et al. [LWV⁺24] within the methodology and definitions developed in this thesis.

Example 5.1. *In this example, the focus will be on the approximation of a function.*

$$f \in \mathcal{F}_{C_b^{d+1}[I],I,n} \text{[2]} \qquad (5.45)$$

by the operator

$$\mathbf{A}_{\mathbf{t},\mathbf{d}} : \mathcal{F}_{C_b^d[I],I,n} \to \mathcal{F}_{S_{d,t},I,n}, \qquad (5.46)$$

which is induced by the operator

$$A_{t,d} : C^d[I] \to S_{d,t}, \qquad (5.47)$$

[2] $\mathcal{C}_b^d[U] = \{f \in \mathcal{C}^d[U] \mid \sup_{x\in U} |D^i f(x)| \leq M$ for some $M \geq 0$ and $i = 1, 2, \dots d\} \subset Lip_L[I]$

as defined in Eq. (4.14)*. The latter operator utilizes splines to approximate smooth functions. The distance measure* $\operatorname{dist}(C^d, A_{t,d})$ *(cf. Eq.* (3.4)*), with respect to the uniform norm, is upper bounded in Theorem 4.3 for* $r = 0$*. In accordance with the established result in Theorem 5.1, the subsequent inequality is derived:*

$$\|f - \mathbf{A_{t,d}} f\|_{\mathcal{F}_{C[I],I,n}} \leq K_{L,n,L_d,C_0} (\Delta_{max} t)^{d+1}.$$

Here, K_{L,n,L_d,C_0} *is a constant that depends on* L*, defined as* $L = \sup_{l,j,k} \|D\Psi_{l,j,k}\|_{C[I],\infty}$*, on the tuple* n*, on the constant* C_0 *given in Theorem 4.3, and on* L_d*, defined as* $L_d = \sup_{l,j,k} \|D^{d+1}\Psi_{l,j,k}\|_{C[I],\infty}$*, which clarifies why we consider functions in* $C_b^{d+1} \subset C_b^d$ *in Eq.* (5.45)*, rather than functions in* C^d *as in Eq.* (4.14)*. The constant* K_{L,n,L_d,C_0} *can be bounded, following the steps of the proof of Theorem 5.1, by the inequality*

$$K_{L,n,L_d,C_0} \leq \sum_{l=1}^{|n|-1} \prod_{k=l}^{|n|-1} n_k L L_d C_0 \leq \left(\prod_{i=1}^{|n|-1} n_i \right) (L L_d C_0)^{|n|-1} (|n| - 1).$$

The statements in Example 5.1 delineate the process of approximating a function, $f \in \mathcal{F}_{C_b^d[I],I,n}$ with a Kolmogorov Arnold network (KAN), as defined in [LWV$^+$24]. It should be noted that the minor exception exists of considering functions in $\mathcal{F}_{C_b^d[I],I,n}$, whereas the original paper focused on functions $f \in \mathcal{F}_{C^d[I],I,n}$, as stated in [LWV$^+$24, Thm 2.1].

6 Proposed Approach and Evaluation

In this section, the derived results will be applied to develop a new machine learning approach, which will be referred to as Kolmogorov Arnold neural networks (KANNs). We will subsequently assess its performance on common regression problems to evaluate its effectiveness and compare it with existing methods. To streamline the ensuing discussion of model evaluation, we first provide a concise overview of the training process. Training refers to the process of learning a representation from a finite dataset. Formally, a dataset D is defined as a finite collection of input-output pairs:

$$D = \{(x_i, y_i) \mid x_i \in X \subseteq \mathbb{R}^n, y_i \in Y \subseteq \mathbb{R}^m, i = 1, \ldots, N\}.$$

Here, each x_i represents an input vector in $\mathbb{R}^n$, and the corresponding y_i is the associated output in $\mathbb{R}^m$. The objective of training is to ascertain a set of parameters w that minimizes a specified loss function, denoted by $L : Y \times Y \to \mathbb{R}^+$, which quantifies the discrepancy between the model's predictions and the actual outputs. The following is a formal definition of the aforementioned process:

$$\min_w \sum_i L(f_w(x_i), y_i),$$

where f_w represents the model parameterized by the parameters w. In this work, we utilize the root mean squared error (RMSE) as the performance criterion, analogous to the loss function, which is defined as:

$$L_{\mathrm{RMSE}}(f_w, D) = \sqrt{\frac{1}{N} \sum_{(x_i, y_i) \in D} (y_i - f_w(x_i))^2}.$$

We note that L_{RMSE} can be interpreted as a discrete approximation of the metric induced by the L_2-norm. Within the framework of machine learning, a distinction is often drawn between training loss and test loss. The training loss quantifies the model's error on the dataset used for optimization, denoted by D. Conversely, the test loss serves to evaluate the model's performance on previously unseen data points $(x, y) \notin D$, thereby providing an estimate of its generalization ability.

6.1 Kolmogorov Arnold Neural Networks

This section is initiated with a concise review of the previously introduced notation. The operator A is employed to denote an approximation operator that maps functions of a designated type to a typically less complex set. In this section, the operator $A : X \to Y$, is employed for continuous univariate functions, that is $X, Y \subset C[I]$. Subsequently, the operator $\mathbf{A}$ represents the composition- and component-wise application of the operator A to functions in $\mathcal{F}_{X,I,n}$, that is

$$\mathbf{A} : \mathcal{F}_{X,I,n} \to \mathcal{F}_{Y,I,n}.$$

We say that A induces $\mathbf{A}$. A comprehensive investigation of this matter was conducted within Section 5.1.

In the ensuing discourse, an examination will be conducted of the operator $A : C[I] \to \mathcal{D}^{\sigma,t,l}[I]$ (cf. Definition 4.9) within the framework of the methodology that has been developed in Section 5. The operator A induces, as per Eq. (5.33), the operator

$$\mathbf{A} : \mathcal{F}_{C[I],I,n} \to \mathcal{F}_{\mathcal{D}^{\sigma,t,l},I,n}. \tag{6.48}$$

The subsequent lemma furnishes an especially compelling property of the set $\mathcal{F}_{\mathcal{D}^{\sigma,t,l},I,n}$.

Lemma 6.1. *The relation:*

$$\mathcal{F}_{\mathcal{D}^{\sigma,t,l},I,n} \subset \mathcal{D}^{\sigma}[\mathbb{R}^{n_1}, \mathbb{R}^{n_N}],$$

where $\mathcal{D}^{\sigma} := \bigcup_{t,l}^{\infty} \mathcal{D}^{\sigma,t,l}$ *(cf. Section 4.2), is valid.*

Proof. For a proof we refer the reader to Appendix A.6.1 □

The set $\mathcal{F}_{\mathcal{D}^{\sigma},I,n}$ is referred to as the set of KANNs. Figure 3 illustrates the relationship between a KANN and a DNN. Of particular note is the emphasis on the $\subset$ relation between the respective approaches. It is imperative to recall that a DNN is a function of the following form:

$$f : x \mapsto A\sigma\left(W_{t-1} \cdots \sigma\left(W_2 \sigma\left(W_1 x + b_1\right) + b_2\right) \cdots + b_{t-1}\right),$$

c.f Section 4.2. For the specific case illustrated in the figure ($t = 2$), the sparse matrices W_1 and A assume the following form:

$$W_1 = \begin{bmatrix} w_{11} & 0 \\ w_{21} & 0 \\ 0 & w_{32} \\ 0 & w_{42} \\ w_{51} & 0 \\ w_{61} & 0 \\ 0 & w_{72} \\ 0 & w_{82} \end{bmatrix}, w_{ij} \in \mathbb{R}.$$

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & a_{25} & a_{26} & a_{27} & a_{28} \end{bmatrix}, w_{ij} \in \mathbb{R}.$$

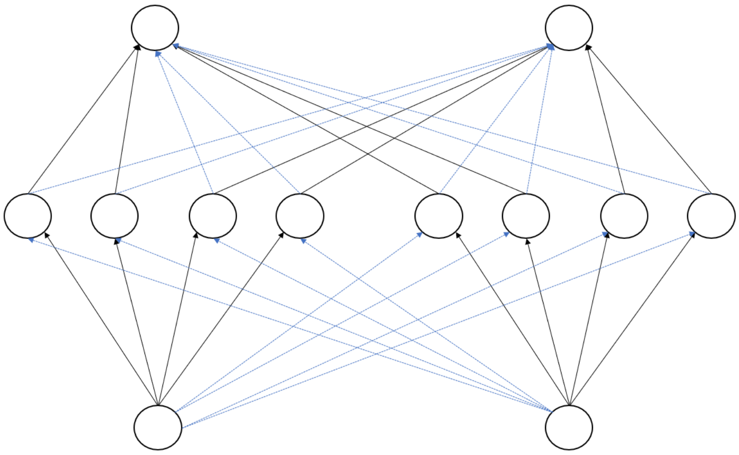


Figure 3: Visualization illustrating the inclusion $\mathcal{F}_{\mathcal{D}^{\sigma,1,2},I,\{2,2\}} \subset \mathcal{D}^{\sigma,8,1}[\mathbb{R}^2,\mathbb{R}^2]$. In this diagram, the circles represent the neurons in the DNN. The black lines represent the weights of the KANN $\mathcal{F}_{\mathcal{D}^{\sigma,1,2},I,\{2,2\}}$, while the combination of black and blue lines illustrates the fully connected DNN $\mathcal{D}^{\sigma,8,1}[\mathbb{R}^2,\mathbb{R}^2]$.

6.2 Evaluation

The code utilized in this section is available at [Pra25].

In this section, the objective is to assess the efficacy of operators $\mathbf{A}$ as delineated in Eq. (5.33). A particular focus of this study is the analysis of the effects of employing the structure[3] of functions $f \in \mathcal{F}_{Lip_L,I,n}$, as defined in Section 5. In order to facilitate our analysis, it is necessary to reduce the effects of the univariate function operator A on the approximation error. The effects in question are encapsulated by the expression $dist(Lip_L, A)$, as elucidated in Theorem 5.1. Alternatively, these effects can be characterized in terms of the Lipschitz constant of functions belonging to the co-domain of A. This notion was further elaborated upon in Remark 5.2. To explore this, we compare the operator $\mathbf{A}$ induced by $A : C[I] \to \mathcal{D}^{\sigma,l,t}$ with DNN based operators $\mathbf{D} : C[\mathbb{R}^{n_1},\mathbb{R}^{n_N}] \to \mathcal{D}^{\sigma,t}[\mathbb{R}^{n_1},\mathbb{R}^{n_N}]$ (discussed

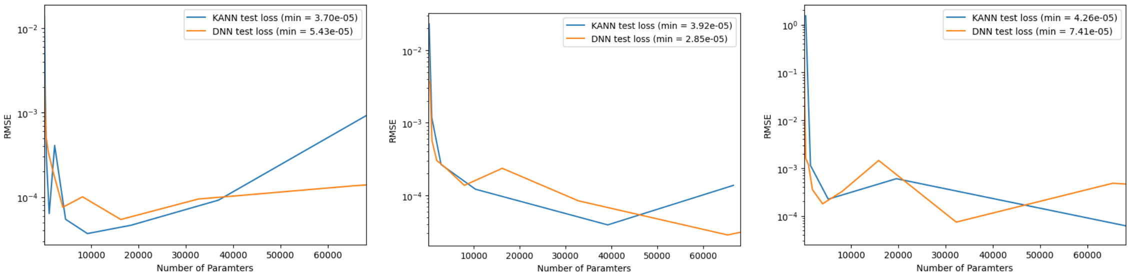
[3] the structure is described by the tuple n

in Section 4.2). This comparison is meaningful in terms of the structural properties due to the subset relationship between these operators (see Lemma 6.1). Moreover, the operator **A** furnishes a lucid framework for appraising the strengths and limitations of our approach in relation to standard DNNs, as it sequesters performance from the influence of *different* optimization techniques or training strategies, a distinction that likewise emanates from the subset relationship.

In the subsequent stage of our investigation, we will undertake an empirical evaluation of performance across a range of scenarios. As a preliminary measure, a comparison is made of the parameter efficiency of the two approaches. Initially, we examine the task of learning functions $f \in P_2[I^2]$ (cf. Section 4.1.1), a set of functions that is particularly well-suited for this analysis, as it admits the following form:

$$P_2[I^2] \subset \mathcal{F}_{P_2,I,(2,m,1)}, \tag{6.49}$$

as illustrated in Example 4. The coefficients $\{a_i\}$ of the monomials of f, where randomly chosen according to $a_i \sim \mathcal{N}(0,1)$, yet are maintained constant across all trials. Figure 4 shows the performance of a KANN $f_{KANN} \in \mathcal{F}_{\mathcal{D}^{\sigma,t,l},I,(2,12,1)}$ in comparison with a DNN $f_{DNN} \in \mathcal{D}^{\sigma,t,l}$ with respect to their parameter count. Each plot presents the respective test losses for a fixed t while steadily increasing the parameter l. For clarity, after increasing the parameter count, each model was trained starting with randomly distributed parameters. The results demonstrate that KANNs perform optimally when the univariate functions are approximated using shallow neural networks. Specifically, the test loss reaches its minimum at $\mathcal{F}_{\mathcal{D}^{\sigma,2,32},I,n}$, which supports the notion of approximating complex multivariate functions through simpler univariate ones. However, in this scenario, the DNNs outperform the KANNs, achieving the lowest overall test loss.



(a) $\mathcal{F}_{\mathcal{D}^{\sigma,2,l},I,(2,12,1)}$ vs. $\mathcal{D}^{\sigma,3,l}$ (b) $\mathcal{F}_{\mathcal{D}^{\sigma,3,l},I,(2,12,1)}$ vs. $\mathcal{D}^{\sigma,5,l}$ (c) $\mathcal{F}_{\mathcal{D}^{\sigma,4,l},I,(2,12,1)}$ vs. $\mathcal{D}^{\sigma,7,l}$

Figure 4: At each parameter count (biases neglected) the models were trained over 1000 epochs, each epoch consists of 10 optimizing steps, where at each step gradients were computed with respect to 512 randomly drawn samples. The KANNs use tuple $n = (2,12,1)$ (cf. Example 4). The neural networks used had shape $[2, \underbrace{r, \dots, r}_{t-times}, 1]$, where r was increased, in order to increase the total number of parameters. The loss was computed on 10000 unseen samples. As optimizer AdamW was employed (cf. [PyT25])

Nonetheless, it can be posited that the functions $f \in P_2[I^2]$, may not possess the requisite complexity to ensure the reliability and robustness of the results, a consequence of their relatively uncomplicated structural nature. This simplicity is reflected in the small value of m and $n_1 = 2$ in Eq. (6.49). Consequently, in the subsequent stage, we will examine learning functions $f \in P_2[I^{12}]$, which exhibit a similar complexity of inner functions to those in Eq. (6.49), but with a considerably larger value of m. This increase in m introduces a substantial degree of complexity, rendering it impractical to identify all univariate polynomials in a straightforward manner. Figure 5 illustrates the test loss of the respective approaches with respect to the parameter count. Due to hardware limitations, a detailed analysis was only feasible for $f \in \mathcal{F}_{\mathcal{S}^\sigma,I,n}$[4]. This limitation will be discussed in more detail in Section 6.3.1. As demonstrated, KANNs not only exhibit a test loss that is roughly 3 times lower than that of standard DNNs, but they also attain this enhanced performance with a substantially reduced parameter count. Notably, the optimum was attained with $f \in \mathcal{F}_{\mathcal{S}^{\sigma,14},I,n}$.

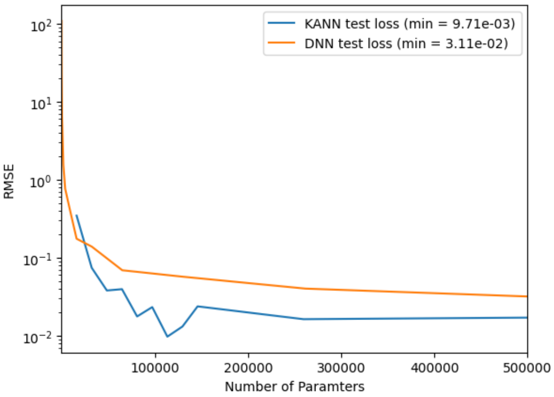


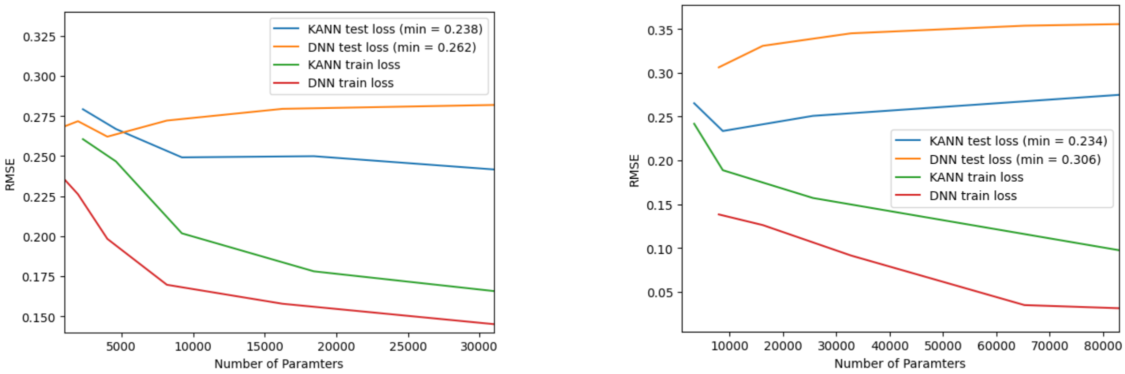
Figure 5: At each parameter count (biases neglected) the models were trained over 1000 epochs, each epoch consists of 10 optimizing steps, where at each step gradients were computed with respect to 512 randomly drawn samples. The KANNs use tuple $n = (12, 312, 1)$ (cf. Example 4). The neural networks used had shape $[12, \underbrace{r, \dots, r}_{t-times}, 1]$, where r was increased, in order to increase the toal number of parameters. The loss was computed on 10000 unseen samples. As optimizer AdamW was employed (cf. [PyT25])

The observed increase in test loss around $150,000$ parameters can be attributed to the introduction of redundant and unnecessary complexity. At this juncture, the KANN has already captured the essential patterns, and the addition of further parameters no longer contributes meaningfully to the learning process. Consequently, this results in inefficient weight updates. This assertion aligns with the concept of approximating complex multivariate functions through simpler univariate ones. It is imperative to acknowledge that prior experiments primarily concentrated on the general expressiveness, with respect to the number of parameters, of the respective approaches. This is because the training data was sampled directly from the objective function, thereby effectively inducing a dataset D (cf. Section 6) with an infinite number of samples. Within the scope of the conducted experiments KANNs attain a level of expressiveness that is comparable to, if not superior to, that of DNNs.

Subsequently, the practical applicability of the proposed method is evaluated through a comparative analysis of the various approaches on the California housing regression

[4] Recall, that $\mathcal{S}^\sigma = \mathcal{D}^{\sigma,2}$, cf. Section 4.2.

dataset [lD24]. The primary distinction is that the optimal tuple n for the function $f \in \mathcal{F}_{\mathcal{D}^{\sigma,t,l},I,n}$ is unknown, prompting an investigation into the effects of using specific structures. Figure 6 compares $\mathcal{F}_{\mathcal{S}^{\sigma,32},I,(8,\underbrace{m,\dots,m}_{\nu\text{-times}},1)}$ with $\mathcal{D}^{\sigma,\nu+2,l}$, where the parameter count was increased by increasing the parameters m and l respectively. It is evident from the conducted experiments that KANNs exhibit superior performance in estimating functions with unknown representations, compared to the baseline DNN approaches.



(a) $f \in \mathcal{F}_{\mathcal{S}^{\sigma,32},I,(8,m,1)}$ vs $\mathcal{D}^{\sigma,3,l}$

(b) $f \in \mathcal{F}_{\mathcal{S}^{\sigma,32},I,(8,m,m,1)}$ vs $\mathcal{D}^{\sigma,5,l}$

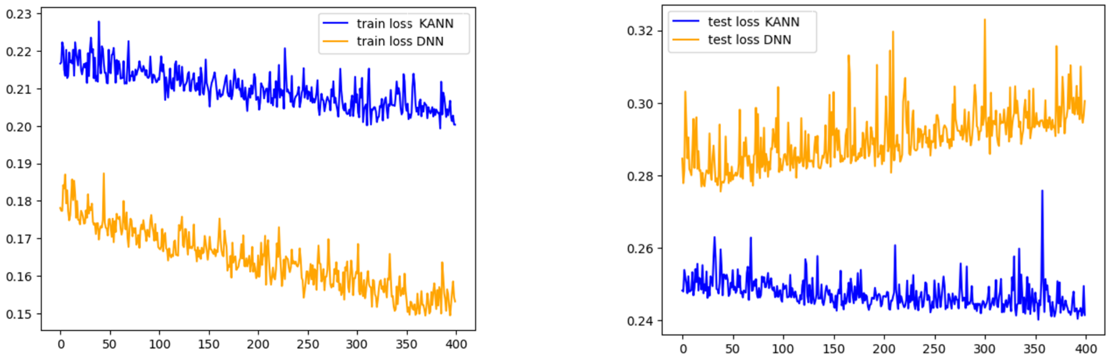
Figure 6: Comparison of parameter efficiency between KANNs and DNNs. The number of parameters was increased by varying m and l, respectively. The networks were trained on the California Housing dataset [lD24] for 1000 epochs using the AdamW optimizer [PyT25].

It can be posited that, in the case of this particular experiment, KANNs mitigate the repercussions of overfitting. Overfitting manifests when a model acquires a memory of the training data instead of discerning general patterns, resulting in the capture of noise rather than the inherent structure of the data. This phenomenon is most commonly observed when a model contains a large number of parameters, as the model may become overly complex and fit even the noise in the data, thereby reducing its ability to generalize. As illustrated in Figure 6, both approaches exhibit distinct patterns in their respective train and test losses. The DNNs consistently outperform the KANNs in terms of train loss, yet the reverse is observed in test loss. Furthermore, as the number of parameters increases, the test loss of the DNNs increases, which does not occur for the test loss of the KANNs.

The findings from both observations lend support to the hypothesis that, in the case of this particular experiment, KANNs are capable of mitigating the effects of overfitting. This conjecture is also observable during the training process. Figure 7 illustrates the final 400 training epochs of the models, with a parameter count of around $1 \cdot 10^4$ (cf. Figure 6a). It is evident that, the train loss of the DNN demonstrates a consistent decline at a rate that is notably more rapid compared to the train loss of the KANNs (cf. Figure 7a). However, the test loss for the DNNs does increase. This phenomenon is indicative of overfitting, wherein the model exhibits difficulty in generalizing to novel, unseen data. Rather than discerning underlying patterns, the model tends to memorize the training data, including its noise and particular details. Consequently, the model's performance on the test set experiences a decline, leading to an increase in the test loss. In contrast, the test loss for the KANNs remains consistent or exhibits a decline, as evidenced in Figure

7b. The following remark proffers an explanation as to why this may be the case. It is important to note, however, that the general validity of these statements is not claimed.

Remark 6.1. *The phenomenon of mitigating the effects of overfitting using KANNs aligns with the observations detailed in Remark 5.1, wherein it was demonstrated that any function can be represented by $\mathcal{F}_{P[I],I,n}$ a composition of ”simple”[5] univariate functions. The KANNs efficiently capture these localized univariate functions with a limited number of parameters, thereby reducing the likelihood of overfitting. This is due to the fact that estimating a function with fewer parameters reduces the estimation's complexity, thereby preventing it from fitting noise. Conversely, DNNs learn more complex global structures with a large number of parameters.*



(a) Train Loss

(b) Test Loss

Figure 7: Comparison of train and test loss between KANNs $f_{\text{KAN NN}} \in \mathcal{F}_{\mathcal{S}^{\sigma,32},I,(8,32,1)}$ (9,216 parameters) and DNNs $f_{DNN} \in \mathcal{D}^{\sigma,3,95}$ (9,880 parameters) over the final 400 epochs of a total 1,000 training epochs. The models were trained on the California housing dataset.

Finally, we will assess the robustness of each approach to noise. Specifically, the approaches will be evaluated using the so-called Friedmann regression dataset [Fri91], which considers, among others, functions of the form:

$$f_{friedmann} : \mathbb{R}^{n_{in}} \to \mathbb{R} : x \mapsto g(x_1, \ldots, x_m) + \mathcal{N}(0, \sigma^2), \tag{6.50}$$

where $g : \mathbb{R}^m \to \mathbb{R}$ and in general $n_{in} \gg m$. These datasets are particularly useful for testing robustness because they introduce noise in high-dimensional settings, simulating real-world scenarios where irrelevant features can obscure the true signal, making it challenging for models to generalize effectively. We employ $f_{friedmann} \in \mathcal{F}_{Lip_L,I,n}$, where $n = (n_{in}, 6, 2, 1)$. The following approaches are considered for the regression task:

- fixed-shape (fs) KANNs, where the Friedmann function is estimated as $\hat{f}_{\text{Friedmann}} \in \mathcal{F}_{\mathcal{S}^{\sigma},I,n}$.
- arbitrary-width (aw) KANNs, where the Friedmann function is estimated as $\hat{f}_{\text{Friedmann}} \in \mathcal{F}_{\mathcal{S}^{\sigma},I,(n_{\text{in}},m,1)}$ for some $m \in \mathbb{N}$. This is in line with the statements in Remark 5.1.

[5] Here, ”simple” refers to functions that can be effectively approximated with a small number of parameters, such as univariate polynomials.

- Standard DNNs.

In addition, to guarantee the comparability with contemporary methodologies, the following methods will be given due consideration:

- fs KANs, where the Friedmann function is estimated as $\hat{f}_{\text{Friedmann}} \in \mathcal{F}_{S_{d,t},I,n}$, as in Example 5.1.

- aw KANs, where the Friedmann function is estimated as $\hat{f}_{\text{Friedmann}} \in \mathcal{F}_{S_{d,t},I,(n_{\text{in}},m,1)}$ for some $m \in \mathbb{N}$.

Table 1 illustrates the test losses for the various approaches. It is evident that when training is conducted on the regression task devoid of noise and a limited number of irrelevant features, the KAN-based approaches demonstrate superior performance in comparison to alternative methods. However, as the presence of noise or irrelevant features is augmented, the performance undergoes a steady decline, a phenomenon that can be anticipated when employing an interpolation technique such as splines (cf. Section 4.1). Additionally, in scenarios with minimal noise, the DNNs consistently exhibit superior performance in comparison to the KANNs. This superiority can be attributed to the utilization of a larger number of parameters by the DNNs. Conversely, as the input dimension and additive noise increase (cf. Eq. (6.50)), the performance of the DNNs declines progressively. In contrast, KANN-based methods exhibit stable performance, as evidenced by their significantly lower variance in test loss. In light of these observations, it can be concluded that, within the scope of the conducted experiments, KANN-based methods exhibit remarkable robustness against noise and irrelevant features and exhibit a notable performance advantage over other approaches in this regard. Moreover, the findings suggest that aw KANNs surpasses other approaches in terms of performance on the given regression problem, particularly demonstrating superior performance compared to fs KANNs. The hypothesis is proposed that increasing the number of parameters in fs KANNs could reduce the mean loss, potentially enabling them to outperform the aw versions. However, a more detailed analysis was not feasible due to practical limitations, which will be discussed later (cf. Section 6.3.1). Conversely, the impact of augmenting the number of parameters of the DNNs on its performance was not found to be substantial.

		fs KANN		DNN	aw KANN	fs KAN	aw KAN
σ^2	n_{in}	$\mathcal{F}_{\mathcal{S}^{\sigma,16},I,n}$	$\mathcal{F}_{\mathcal{S}^{\sigma,32},I,n}$	$\mathcal{D}^{\sigma,3,128}$	$\mathcal{F}_{\mathcal{S}^{\sigma,32},I(n_{in},16,1)}$	$\mathcal{F}_{S_{3,t},I,n}$	$\mathcal{F}_{S_{3,t},I,(n_{in},16,1)}$
0.0	5	$2.3 \cdot 10^{-2}$	$2.3 \cdot 10^{-2}$	$6 \cdot 10^{-3}$	$1 \cdot 10^{-3}$	$6 \cdot 10^{-3}$	$\mathbf{1.8 \cdot 10^{-5}}$
0.0	10	$3.7 \cdot 10^{-2}$	$7.5 \cdot 10^{-3}$	$2.3 \cdot 10^{-3}$	$1 \cdot 10^{-2}$	$1.2 \cdot 10^{-4}$	$\mathbf{3.1 \cdot 10^{-5}}$
0.0	15	$2.5 \cdot 10^{-2}$	$1.7 \cdot 10^{-2}$	$4.5 \cdot 10^{-3}$	$2.5 \cdot 10^{-2}$	$5.3 \cdot 10^{-2}$	$\mathbf{2.5 \cdot 10^{-4}}$
0.0	100	$5.7 \cdot 10^{-2}$	$1.1 \cdot 10^{-2}$	$2.4 \cdot 10^{-2}$	$\mathbf{2.3 \cdot 10^{-3}}$	1.8	$7.1 \cdot 10^{-1}$
0.2	5	$2.3 \cdot 10^{-2}$	$\mathbf{5 \cdot 10^{-3}}$	$8 \cdot 10^{-3}$	$7 \cdot 10^{-3}$	$7.8 \cdot 10^{-3}$	$3.8 \cdot 10^{-2}$
0.2	10	$2.2 \cdot 10^{-2}$	$2 \cdot 10^{-2}$	$\mathbf{1 \cdot 10^{-2}}$	$\mathbf{1 \cdot 10^{-2}}$	$1.9 \cdot 10^{-2}$	$7.7 \cdot 10^{-2}$
0.2	15	$2.6 \cdot 10^{-2}$	$2.5 \cdot 10^{-2}$	$2.2 \cdot 10^{-2}$	$\mathbf{2.1 \cdot 10^{-2}}$	$3.1 \cdot 10^{-1}$	$9.4 \cdot 10^{-2}$
0.2	100	$2.3 \cdot 10^{-2}$	$2.1 \cdot 10^{-2}$	$4 \cdot 10^{-2}$	$\mathbf{1.5 \cdot 10^{-2}}$	2.8	$2.6 \cdot 10^{-1}$
0.5	5	$3 \cdot 10^{-2}$	$\mathbf{1 \cdot 10^{-2}}$	$2.3 \cdot 10^{-2}$	$\mathbf{1 \cdot 10^{-2}}$	$7.3 \cdot 10^{-2}$	$2.6 \cdot 10^{-1}$
0.5	10	$3.6 \cdot 10^{-2}$	$4.5 \cdot 10^{-2}$	$9.3 \cdot 10^{-2}$	$\mathbf{1.5 \cdot 10^{-2}}$	$3.1 \cdot 10^{-2}$	$4.5 \cdot 10^{-1}$
0.5	15	$4.2 \cdot 10^{-2}$	$3.8 \cdot 10^{-2}$	$7.2 \cdot 10^{-2}$	$\mathbf{2.9 \cdot 10^{-2}}$	$1.8 \cdot 10^{-1}$	$5.3 \cdot 10^{-1}$
0.5	100	$4.1 \cdot 10^{-2}$	$3.6 \cdot 10^{-2}$	$1.6 \cdot 10^{-1}$	$\mathbf{2.3 \cdot 10^{-2}}$	$8 \cdot 10^{-1}$	2.2
1.0	5	$3.9 \cdot 10^{-2}$	$3 \cdot 10^{-2}$	$9 \cdot 10^{-2}$	$\mathbf{2.5 \cdot 10^{-2}}$	$1 \cdot 10^{-1}$	1.03
1.0	10	$6 \cdot 10^{-2}$	$6 \cdot 10^{-2}$	$2.3 \cdot 10^{-1}$	$\mathbf{3.1 \cdot 10^{-2}}$	$2 \cdot 10^{-1}$	1.8
1.0	15	$5.7 \cdot 10^{-2}$	$5.5 \cdot 10^{-2}$	$2.5 \cdot 10^{-1}$	$\mathbf{3.7 \cdot 10^{-2}}$	$8 \cdot 10^{-1}$	2.1
1.0	100	$\mathbf{7.6 \cdot 10^{-2}}$	$1.1 \cdot 10^{-1}$	$6 \cdot 10^{-1}$	$9 \cdot 10^{-2}$	3.7	2.7
Mean		$3.8 \cdot 10^{-2}$	$3.2 \cdot 10^{-2}$	$1 \cdot 10^{-1}$	$2.1 \cdot 10^{-2}$	$6.7 \cdot 10^{-1}$	$7.6 \cdot 10^{-1}$
Variance		$2.5 \cdot 10^{-4}$	$6.5 \cdot 10^{-4}$	$2.2 \cdot 10^{-2}$	$4.1 \cdot 10^{-4}$	1.1	$7.8 \cdot 10^{-1}$

Table 1: Performance evaluation of models on the Friedmann dataset [ld25a] under varying noise levels and input dimensions. The reported values correspond to the RMSE on noise-free test data. The fs models shape is given by $n = (n_{in}, 6, 2, 1)$. The KANNs and DNNs were trained for 1000 epochs using the AdamW optimizer [PyT25]. The KANs were trained on ordered tuples t with cardinalities of 3, 10, and 20, with the best-performing result across all cardinalities presented in the table. Additionally, the KANs were optimized using the L-BFGS algorithm [Dev24] for 100 epochs.

It is noteworthy that the KANs were optimized using the L-BFGS algorithm, a choice that may shed light on their observed performance decline in the presence of irrelevant features and additive noise. However, it was observed that employing the Adam optimizer resulted in a substantially diminished performance. The approaches are evaluated for the other Friedmann-regression problems [Fri91] in Appendix A.6.2. These approaches address issues concerning non-Lipschitz-continuous representations and the effective convergence during the training process.

6.3 Future Work

Despite the promising results discussed in Section 6.2, further improvements are necessary before KANNs can be reliably used in practical applications. This section highlights some of the key challenges that remain.

6.3.1 Algorithmic Inefficiency

Our evaluation in Section 6.2 does not yet determine whether KANNs outperform standard DNNs on more complex tasks, which typically require a much larger number of parameters. This is primarily due to the current implementation's poor scalability in both time and space complexity as the number of parameters in $\mathcal{F}_{\mathcal{D}^{\sigma,l,t},I,n}$ increases. Specifically, the augmentation of the number of parameters, by widening l for $\mathcal{D}^{\sigma,l,t}$ or expanding the elements of n, results in very large weight matrices W (cf. Lemma 6.1). As a reminder, the weight matrices of a single layer can contain up to $\max_i n_i^3 l$ parameters, with the majority of these being zero. However, the present implementation does not leverage the sparsity of these matrices, which could enhance the efficiency of matrix multiplication and parameter updates. To address this limitation, we are developing an algorithm that leverages matrix sparsity. In future work, we intend to evaluate this approach on more complex learning tasks.

6.3.2 Enhancing Kolmogorov Arnold Neural Networks with Increased Depth

A critical factor contributing to the efficacy of DNNs is their capacity to efficiently train models with increasing depth, enabling them to capture intricate representations and attain remarkable performance across a diverse array of tasks. However, standard techniques that facilitate deep representation learning, such as batch normalization, regularization methods, or residual connections, have yet to be implemented or tested. Additionally, for deep representation, that is $f \in \mathcal{F}_{I,n}$, with $|n| \gg 1$, we recall that a reduction in the Lipschitz constant of the univariate approximation functions contributes positively to the overall approximation error (cf. Remark 5.2). This finding renders training strategies that implicitly reduce the Lipschitz constant (i.e., lower absolute values of the weights, cf. Lemma 4.4), such as weight decay or pruning strategies, particularly appealing. A comprehensive evaluation of these strategies will be reserved for future research endeavors. In addition, the question of how to introduce depth in KANNs must be addressed. One approach would be to increase the cardinality of the tuple n. Alternatively, the depth of univariate DNNs could be extended. On the one hand, the deployment of deeper neural networks has the potential to enhance expressive power while maintaining a lower Lipschitz constant. This is of particular relevance due to Remark 5.2. The aforementioned conjecture is based on the fact that two consecutive affine transformations, each with a weight matrix that is bounded in operator norm ($\|W\|_p \leq 1$), result in an overall Lipschitz constant that is less than or equal to 1. We assume a activation function with Lipschitz constant less than 1 (cf. Lemma 4.4). This fact is illustrated by means of a univariate deep neural network of the following form:

$$f : x \mapsto \mathrm{Aff}_t \circ \sigma_{t-1} \circ \cdots \circ \sigma_1 \circ \mathrm{Aff}_1(x),$$

where $\mathrm{Aff}_k : x \mapsto W_k x + b_k$. Let $\|W_k\|_p = 1$ for each $k = 1, \ldots, t$ and assume that σ_k is a function with a Lipschitz constant equal to 1. Under these conditions, the Lipschitz constant of the entire network is bounded by 1 for any value of t. Thus, it is possible to incorporate more parameters while maintaining strict control over the Lipschitz constant. However, the parameters of the individual weight matrices W_k cannot be augmented arbitrarily without consequence, as the operator norm of a matrix is directly proportional to the number of parameters it contains, which in turn is related to the input and output dimensions of the corresponding layer. Consequently, deeper networks could preserve the same Lipschitz bound while incorporating a greater number of parameters. Conversely, increasing the cardinality of the tuple n may facilitate a more effective representation of hierarchical relationships, as suggested in Appendix A.6.2, albeit in a somewhat weak manner.

6.3.3 Exploration of Multivariate Extensions

Another potential direction for research involves KANNs that build on multivariate functions, rather than univariate ones, leading to smaller values $n_i \in n$ for $i = 2, \dots N-1$, where n denotes the structure tuple for function $\mathcal{F}_{I,n}$ (cf. Section 5.1). Such a representation could facilitate the learning of more complex multivariate features while still benefiting from the structural advantages of the KANN architecture. A brief extension of the theoretical results from the univariate case can be found in Appendix A.5.5.

6.3.4 Non-Uniform Distribution of Parameters

As previously stated, augmenting the number of parameters generally increases the Lipschitz constant of the univariate approximation functions (cf. Lemma 4.1 and 4.4). In accordance with Remark 5.2, this is an effect we aim to mitigate. However, it is noteworthy that for the initial layer, the Lipschitz constant does not induce adverse effects, as substantiated by the findings outlined in Theorem 5.1. Consequently, we hypothesize that augmenting the number of parameters, particularly in this layer, may yield enhanced outcomes. It is noteworthy that a similar effect could be achieved by introducing layer-wise differentiated regularization. A thorough examination of this approach is reserved for subsequent studies.

6.3.5 Extension to other Architectures

The developed approach can be seamlessly extended to other neural network-based machine learning architectures, including recurrent neural networks, gated recurrent units, long short-term memory networks, or attention layers. The extension and evaluation of these architectures will be addressed in future research endeavors.

7 Conclusion and Applications

This section offers a synopsis of the primary contributions of the present thesis, in addition to an examination of the potential applications of KANNs.

7.1 Summary and Contributions

The field of alternatives to DNNs has emerged as a significant area of research, with growing interest from the machine learning community. Specifically, the recent introduction of KANs [LWV+24] has generated considerable interest within the machine learning community. While the underlying concept was not entirely novel, these networks were hypothesized to offer solutions to some of the fundamental challenges associated with DNNs.

This thesis expounded on the theoretical underpinnings of KANs, with a particular emphasis on approximation theory. The focus was directed towards methodologies for univariate function approximation. A seminal contribution of this study was the definition of multivariate functions represented as compositions of superpositions of univariate functions. Moreover, multivariate polynomials were shown to be representable as compositions of superpositions of univariate polynomials, thereby providing a robust theoretical foundation for the structure employed in KANs.

Moreover, key results were established and proven, including the derivation of an upper bound on the approximation error that is contingent upon the function to be approximated. These theoretical findings were thoroughly discussed from multiple perspectives, providing a comprehensive understanding of the approach's potential.

In addition to the theoretical analysis, this thesis introduced a novel approach based on these foundations, which demonstrated superior performance in a range of experiments. In summary, this work proffered a compelling new perspective on machine learning and introduced a promising enhancement to traditional DNNs.

7.2 Potential Applications of the Proposed Method

The following section explores practical applications that are particularly well-suited for KANNs. In particular, the remarkable robustness of KANNs against irrelevant features renders it especially well-suited for problems involving high-dimensional datasets with numerous redundant variables. A notable illustration of this is found in the field of genomics, where "biologically speaking, there is only a limited number of genes that are associated with a disease and, as such, only expression levels of certain genes can differentiate between cases and controls" [AU20]. Conventional methods frequently employ statistical techniques, such as p-value computation, to filter out statistically insignificant features. However, these methods face challenges in discerning between statistical insignificance and actual feature irrelevance [AU20]. In contrast, our approach is designed to inherently suppress the influence of irrelevant features, which has the potential to eliminate the need for this error-prone preprocessing step and to result in enhanced performance.

Additionally, the robustness of the KANNs could render them a compelling solution in the domain of adversarial machine learning, particularly in scenarios where defending against membership inference attacks (MIAs) is imperative. A MIA is a privacy attack in which an adversary attempts to ascertain whether a particular data point was incorporated into a machine learning model's training dataset by examining the model's behavior, such as its prediction confidence. In the context of defending against MIAs, the robustness of KANNs is particularly effective in high-dimensional datasets, where the risk of overfitting and the exposure of sensitive data is heightened. In domains such as healthcare and finance, where voluminous and intricate datasets are prevalent, KANNs exhibit a natural aptitude for mitigating overfitting, thereby impeding the capacity of adversaries to discern between training and non-training data. In contrast to traditional models, which are susceptible to overfitting and can make high-confidence predictions on training data that attackers can exploit, KANNs offer a more robust and secure alternative.

Another promising application of KANNs lies in large language models, where parameter efficiency is of utmost importance. As these models continue to scale, the need for more efficient use of parameters grows, especially given the computational and storage constraints associated with training and deploying such models. In conjunction with the strategies outlined in Section 6.3.2, we believe there may be opportunities to improve current practices, particularly in terms of reducing the parameter count without sacrificing model performance. Large language models are particularly vulnerable to overfitting, especially when trained on vast, unstructured data. The proposed method, by virtue of its resilience to irrelevant features and noise, could provide a solution to this problem. By addressing overfitting more effectively, the method could enhance the generalization capabilities of these models, ensuring they perform better across a wider range of tasks. This could result in more reliable, efficient, and scalable language models, which would make them more applicable in real-world settings where data may be noisy or incomplete.

Together, these applications underscore the versatility and robustness of KANNs, highlighting their potential to address key challenges across diverse high-dimensional, privacy-sensitive, and computationally demanding domains.

A Appendix

A.5 Section 5

A.5.1 Validity of the Norm Defined in Eq. (5.34)

We now proceed to show that the normed linear space given by:

$$\begin{gathered}(C[I_1^{n_1}, I_N^{n_N}], \|\cdot\|_{\mathcal{F}_{C[I],I,n},\infty}),\\ \text{where } \|f\|_{\mathcal{F}_{C[I],I,n},\infty} = \sup_{x\in I_1^{n_1}} \|f(x)\|_{I_N^{n_N},\infty},\end{gathered} \tag{A.5.37}$$

in fact satisfies the defining axioms of such. In particular we show that the norm $\|f\|_{\mathcal{F}_{C[I],I,n},\infty} = \sup_{x\in I_1^{n_1}} \|f(x)\|_{I_N^{n_N},\infty}$, indeed statisfies the defining axioms of a norm.

1. (Triangle Inequality) For $f, g \in \mathcal{F}_{C[I],I,n}$

$$\begin{aligned}\|f+g\|_{\mathcal{F}_{C[I],I,n},\infty} \le \sup_{x\in I_1^{n_1}} \|f(x)+g(x)\|_{I_N^{n_N},\infty} &\le \sup_{x\in I_1^{n_1}} \|f(x)\|_{I_N^{n_N},\infty} + \sup_{x\in I_1^{n_1}} \|g(x)\|_{I_N^{n_N},\infty}\\ &= \|f\|_{\mathcal{F}_{C[I],I,n},\infty} + \|g\|_{\mathcal{F}_{C[I],I,n},\infty}.\end{aligned}$$

2. (Homogeneity) For $f \in \mathcal{F}_{C[I],I,n}, \alpha \in \mathbb{R}$

$$\|\alpha f\|_{\mathcal{F}_{C[I],I,n},\infty} = |\alpha| \|f\|_{\mathcal{F}_{C[I],I,n},\infty},$$

   we note here that f is bounded by definition, as continuous functions over compact domains are necessarily bounded.

3. (Positivity) For $f \in \mathcal{F}_{C[I],I,n}$,

$$\|f\|_{\mathcal{F}_{C[I],I,n},\infty} \ge 0, \|f\|_{\mathcal{F}_{C[I],I,n},\infty} = 0 \iff \forall_i f^i = 0,$$

   which is a direct consequence of the definition of $\|\cdot\|_{I_N^{n_N},\infty}$.

A.5.2 Proof of Eq. (5.36)

We now proceed to proof that the identity:

$$C \prod_{i=1}^{n} x_i = \sum_{S\subseteq\{x_1,\dots,x_n\}} (-1)^{n-|S|} \left(\sum_{x_i\in S} x_i\right)^n \tag{A.5.39}$$

holds. Before stating the proof, we introduce the following notation for convenience. Let $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_n)$ be a vector, we then define the following norms:

$$\|\alpha\|_0 = |\{i \mid \alpha_i \ne 0,\, i = 1, 2, \dots, n\}|,$$

$$\|\alpha\|_1 = \sum_{i=1}^{n} |\alpha_i|.$$

Additionally we state that the multinomial expansion of $(x_1 + x_2 + \cdots + x_k)^n$ is given by:

$$\begin{aligned}(x_1 + x_2 + \cdots + x_k)^n &= \sum_{a_1+a_2+\cdots+a_k=n} \binom{n}{a_1, a_2, \ldots, a_k} x_1^{a_1} x_2^{a_2} \ldots x_k^{a_k} \\ &= \sum_{\|\alpha\|_1=n} \binom{n}{\alpha_1, \alpha_2, \ldots, \alpha_k} x_1^{\alpha_1} x_2^{\alpha_2} \ldots x_k^{\alpha_k},\end{aligned} \tag{A.5.40}$$

where

$$\binom{n}{a_1, a_2, \ldots, a_k} = \frac{n!}{a_1! a_2! \ldots a_k!}.$$

Proof. We begin by slight rewriting the right hand side of Eq. (A.5.39) to obtain

$$\sum_{S \subseteq \{x_1,\ldots,x_n\}} (-1)^{n-|S|} \left(\sum_{x_i \in S} x_i \right)^n = \sum_{k=1}^{n} (-1)^{n-k} \sum_{\substack{S \subseteq \{x_1,\ldots,x_n\} \\ |S|=k}} (\sum_{x_i \in S} x_i)^n.$$

Now, expanding the inner two sums, we get using Eq. (A.5.40)

$$\sum_{\substack{S \subseteq \{x_1,\ldots,x_n\} \\ |S|=k}} (\sum_{x_i \in S} x_i)^n = \sum_{\substack{\|\alpha\|_1=n \\ \|\alpha\|_0 \le k}} \binom{n}{\alpha_1, \ldots, \alpha_n} \prod_{i=1}^{n} x_i^{\alpha_i} \binom{n - \|\alpha\|_0}{k - \|\alpha\|_0}, \tag{A.5.41}$$

where the binomial coefficient $\binom{n-\|\alpha\|_0}{k-\|\alpha\|_0}$ represents the number of ways a sequence $s \subseteq \{x_1, x_2, \ldots, x_n\} := S$ of cardinality $\|\alpha\|_0 = |s|$ can be part of a subset of size k drawn from the set S of cardinality n. In other words, if we consider all subsets of size k from S, the sequence s appears in exactly $\binom{n-\|\alpha\|_0}{k-\|\alpha\|_0}$ of them. This binomial coefficient precisely quantifies the frequency with which subsets of size k contribute the term $\binom{n}{\alpha_1,\ldots,\alpha_n} \prod_{i=1}^{n} x_i^{\alpha_i}$ in Eq. (A.5.41), which is readily verified by Eq. (A.5.40). As a result, the final polynomial expression in Eq. (A.5.39) contains each multivariate monomial term $\prod_{i=1}^{n} x_i^{\alpha_i}$ with corresponding coefficient:

$$\binom{n}{\alpha_1, \ldots, \alpha_n} \sum_{k=1}^{n} (-1)^{n-k} \binom{n - \|\alpha\|_0}{k - \|\alpha\|_0}.$$

Finally, we notice that

$$\sum_{k=1}^{n} (-1)^{n-k} \binom{n - \|\alpha\|_0}{k - \|\alpha\|_0} = \begin{cases} 1 & \text{if } \|\alpha\|_0 = n \\ 0 & \text{else} \end{cases},$$

where we used the convention that $\binom{n}{m} = 0$ if $n < 0 \vee m < 0$. □

A.5.3 Proof of Remark 5.2

Proof. Analogous to the proof of Theorem 5.1 we obtain

$$\|\Psi_{n+1,k,j} \circ f_n^j - \mathbf{A}(\Psi_{n+1,k,j} \circ f_n^j)\|_{\mathcal{F}_{C[I],I,(n_1,\ldots n_{n+1},1),\infty}}$$

(cf. Eq (5.44)). By adding 0 we then obtain

$$\|\Psi_{n+1,k,j} \circ f_n^j - \mathbf{A}(\Psi_{n+1,k,j} \circ f_n^j) + \mathbf{A}(\Psi_{n+1,k,j}) \circ f_n^j - \mathbf{A}(\Psi_{n+1,k,j}) \circ f_n^j\|_{\mathcal{F}_{C[I],I,(n_1,\dots n_{n+1},1),\infty}},$$

from this, it follows that Remark 5.2 holds if $A\Psi_{l,k,j}$ is Lipschitz continuous. To see this, consider the following inequality:

$$\|\mathbf{A}(\Psi_{n+1,k,j}) \circ f_n^j - \mathbf{A}(\Psi_{n+1,k,j} \circ f_n^j)\| \leq \omega(A\Psi_{n+1,k,j}, \|f_n^j - \mathbf{A}f_n^j\|)$$

and the reasoning used in the proof of Theorem 5.1. The same argument yields the assertion. □

A.5.4 Proof of Lemma 5.1

Lemma A.5.1. *There exist exactly* $\binom{d-1}{w}$ *tuples* $\{x_1, x_2, \dots, x_w\} \subset \mathbb{N}$ *of cardinality* w*, such that* $\sum_{i=1}^{w} x_i < d$.

Proof. Let $y_i = x_i - 1$, then

$$\sum_{i=1}^{w}(y_i + 1) = \sum_{i=1}^{w} x_i < d$$

Rearranging gives:

$$\sum_{i=1}^{w} y_i \leq d - w - 1.$$

Counting the number of non-negative integer solutions to $\sum_{i=1}^{w} y_i = k$, we obtain $\binom{k+w-1}{w-1}$. Hence the total number of solutions is:

$$\sum_{k=0}^{d-w-1} \binom{k+w-1}{w-1}.$$

Using the identity for the sum of binomial coefficients:

$$\sum_{k=0}^{m} \binom{k+r}{r} = \binom{m+r+1}{r+1},$$

where $m = d - w - 1$ and $r = w - 1$, we find:

$$\sum_{k=0}^{d-w-1} \binom{k+w-1}{w-1} = \binom{(d-w-1)+(w-1)+1}{w} = \binom{d-1}{w}.$$

□

A.5.5 Extension to Multivariate Functions Represented as a Composition of Superpositions of Functions of Fewer Variables

We shall now extend Theorem 5.1 to multivariate functions ($f : I_1^{n_1} \to I_N^{n_N}$), represented as a composition of superpositions of functions of fewer variables. To achieve this objective, a minor adjustment is made to the definition presented in the opening of Section 5.1.

Specifically, the following modification is made to Eq. (5.24):

$$\Psi_{l,k,j} : I_l{}^m \to \mathbb{R}, m \leq n_l, \tag{A.5.27}$$

which results in functions $\Phi_l^k : I_l^{n_l} \to I_{l+1} : x \mapsto \sum_{j=1}^{2^{n_l}-2} \Psi_{l,k,j}(X_j), X_j \subset \{x_1, x_2, \ldots x_{n_l}\}$. This concept is visualized in Figure 8. Similar to the definition of functions represented by compositions of superpositions of univariate functions (cf. Eq. (5.32)), we can define the set of functions of multivariate functions represented as the composition of superposition of functions of fewer variables, in the following way:

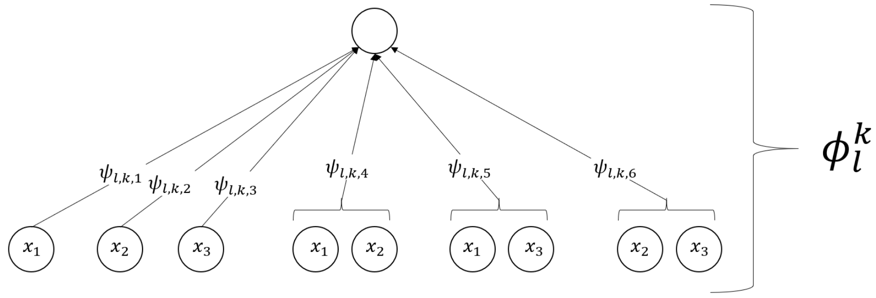


Figure 8: Graphical representation of a function $\Phi_l^k : I_l^3 \to I_{l+1}$. Each arrow represents a function $\Psi_{l,k,j}$ applied to an input, which is depicted by a circle. The graph is read from bottom to top, where multiple lines joining at a circle in the upward direction indicate the application of a summation operation.

$\mathcal{F}^{mult}_{X,I,n} = \{f : I_1^{n_1} \to I_N^{n_N} | f(x) = (\bigcirc_{l=1}^{N-1} \Phi_l)(x)$, as per Eq. ((5.25) - (5.28) and (A.5.27)), $\Psi_{l,k,j} \in X\}$. We define the set $\mathcal{L}_{n,L} := \bigcup_{k=1}^{n-1} \mathrm{Lip}_L[I^k]$. Analogously to the proof of Theorem 5.1, it follows that for any $f \in \mathcal{F}^{\mathrm{mult}}_{\mathcal{L}_L,I,n}$, the following holds:

$$\|f - \mathbf{A}f\|_{\mathcal{F}^{mult}_{C[I],I,n},\infty} \leq C_{L,n} dist(\mathcal{L}_{\max n,L}, A),$$

where $A : C[I^m] \to C[I^m]$ is defined for all m and $\mathbf{A}$ is induced by A as per Eq. (5.33). For dist we use the measure of distance:

$$d(f,g) = \begin{cases} \sup_{x \in dom(f)} |f(x) - g(x)|, & \text{if } dom(f) = dom(g), \\ \infty, & \text{else.} \end{cases}$$

Note that this is not a metric, as it takes values outside of $\mathbb{R}^+$. However, by the definition of the operator A, it still provides reliable results for our purposes.

A.6 Section 6

A.6.1 Proof of Lemma 6.1

We shall now proceed to proof the subsequent lemma.

Lemma A.6.1. *The relation:*

$$\mathcal{F}_{\mathcal{D}^{\sigma,t,l},I,n} \subset \mathcal{D}^{\sigma}[\mathbb{R}^{n_1}, \mathbb{R}^{n_N}],$$

where $\mathcal{D}^\sigma := \bigcup_{t,l}^\infty \mathcal{D}^{\sigma,t,l}$ *(cf. Section 4.2), is valid.*

Before giving the proof we state an auxiliary lemma.

Lemma A.6.2. *Let* $f : \mathbb{R}^l \to \mathbb{R}^m$ *and* $f' : \mathbb{R}^n \to \mathbb{R}^l$ *be functions such that* $f, f' \in \mathcal{D}^\sigma$*, then*

$$f \circ f' \in \mathcal{D}^\sigma$$

Proof. As per Eq. (4.23) the functions in $\mathcal{D}^\sigma$ take the form:

$$f : x \mapsto A\sigma\left(W_{t-1} \cdots \sigma\left(W_2\sigma\left(W_1 x + b_1\right) + b_2\right) \cdots + b_{t-1}\right).$$

The composition of two such functions is then given by:

$$f \circ f' : x \mapsto A\sigma\left(W_{t-1} \cdots \sigma\left(W_1 A'\sigma\left(W'_{t-1} \cdots \sigma\left(W'_1 x + b'_1\right) \cdots + b'_{t-1}\right) + b_1\right) \cdots + b_{t-1}\right),$$

which is of the form given in Eq. (4.23) and hence yields the assertion. □

We are now prepared to continue with the proof of Lemma A.6.1.

Proof. We prove the statement using induction on $N-1$, where N denotes the cardinality of the tuple n.

Base Case:

The univariate functions of $f \in \mathcal{F}_{\mathcal{D}^{\sigma,t,l},I,(n_1,n_2)}$ are, as per Eq. (4.23), given in the form:

$$\Psi_{1,k,j} : x \mapsto A_{k,j}\sigma\left(W_{k,j,t-1} \cdots \sigma\left(W_{k,j,2}\sigma\left(W_{k,j,1}x + b_{k,j,1}\right) + b_{k,j,2}\right) \cdots + b_{k,j,t-1}\right),$$

where $k = 1, \ldots, n_2$ and $j = 1, \ldots, n_1$. We can now generalize the previous representation by incorporating matrices W_m and biases b_m, for $m = 1, \ldots, t-1$, which denotes the depth of the univariate DNNs, in the following form:

$$W_m = \begin{bmatrix} W_{1,1,m} & 0 & \cdots & 0 \\ 0 & W_{1,2,m} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & W_{1,n_1,m} \\ W_{2,1,m} & 0 & \cdots & 0 \\ 0 & W_{2,2,m} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & W_{2,n_1,m} \\ \vdots & \vdots & \vdots & \vdots \\ W_{n_2,1,m} & 0 & \cdots & 0 \\ 0 & W_{n_2,2,m} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & W_{n_2,n_1,m} \end{bmatrix}$$

and

$$b_m = \begin{bmatrix} b_{1,1,m} \\ b_{1,2,m} \\ \vdots \\ b_{1,n_1,m} \\ b_{2,1,m} \\ \vdots \\ b_{n_2,n_1,m} \end{bmatrix}.$$

Hence we may write for $f \in \mathcal{F}_{\mathcal{D}^{\sigma,t,l},I,(n_1,n_2)}$:

$$f : x \mapsto A\sigma\left(W_{t-1} \cdots \sigma\left(W_2\sigma\left(W_1 x + b_1\right) + b_2\right) \cdots + b_{t-1}\right),$$

where A introduces the summation operator and is defined as follows:

$$A = \begin{bmatrix} A_{1,1} & A_{1,2} & \cdots & A_{1,n_1} & 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & A_{2,1} & \cdots & A_{2,n_1} & 0 & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots & \ddots & & & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & \cdots & \cdots & \cdots & A_{n_2,1} & \cdots & A_{n_2,n_1} \end{bmatrix},$$

which yields the base case. An example of the construction of these matrices can be found in Section 6.1.

Induction Hypothesis:

We shall now assume that $\mathcal{F}_{\mathcal{D}^{\sigma,t,l},I,(n_1,n_2,\ldots,n_m)} \subset \mathcal{D}^\sigma$

Induction Step:

In accordance with the results from Section 5.1 we now consider the form $f_{m+1} = \Phi_{m+1} \circ f_m$ (cf. Eq. (5.30)), where $\Phi_{m+1} \in \mathcal{F}_{\mathcal{D}^{\sigma,t,l},I,(n_m,n_{m+1})}$, $f_m \in \mathcal{F}_{\mathcal{D}^{\sigma,t,l},I,(n_1,n_2,\ldots,n_m)}$ and $f_{m+1} \in \mathcal{F}_{\mathcal{D}^{\sigma,t,l},I,(n_1,n_2,\ldots,n_{m+1})}$. As per the results shown in the base case $\Phi_{m+1} \in \mathcal{D}^\sigma$ and in accordance with the induction hypothesis $f_m \in \mathcal{D}^\sigma$. These facts in conjunction with Lemma A.6.2 yield the assertion. □

A.6.2 Friedmann 2 and Friedmann 3

Remark A.1. *The results of Theorem 5.1 can be extended to the set $\mathcal{F}_{\mathrm{Lip}_{L,\alpha},I,n}$ for $\alpha < 1$ as follows:*

$$\|f - \mathbf{A}f\|_{\mathcal{F}_C[I],I,n} \leq C_{L,n} \, \mathrm{dist}(\mathrm{Lip}_{\mathrm{L},\alpha}[\mathrm{I}], \mathrm{A})^{\alpha^{\mathrm{N}\text{-}2}},$$

where $N = |n|$. The proof of this extension follows similarly to the one presented for Theorem 5.1.

Tables 2 and 3 present the performance of the respective approaches introduced in Section 6.2 for Friedmann regression problems. The regression problems are given by

$$\begin{aligned} f_2(x_1,x_2,x_3,x_4) &= \left(x_1^2 + (x_4x_2 - (x_4x_3)^{-1})^2\right)^{0.5} + \mathcal{N}(0,\sigma^2) \qquad \text{Table 2} \\ f_3(x_1,x_2,x_3,x_4) &= \arctan\left(\frac{x_2x_3}{x_1} - \frac{1}{x_1x_2x_4}\right) + \mathcal{N}(0,\sigma^2) \qquad \text{Table 3}, \end{aligned} \tag{A.6.28}$$

they denote the magnitude of the impedance and the phase of a series RLC circuit respectively (cf. [Fri91, Eq. 63]). It is observed that neither of these functions belongs to $\mathcal{F}_{Lip_L,I,n}$. Specifically, the function f_2 contains Hölder-continuous functions, i.e., $f(x) = \sqrt{x}$. We emphasize that for f_2 in Eq. (A.6.28), the $\sqrt{\cdot}$ function is applied to values in the interval $[2.6 \times 10^{-9}, 3.1 \times 10^{6}]$. Although it is Lipschitz continuous on any closed interval bounded away from zero, the extremely small lower bound results in a large Lipschitz constant. As a result, and to better reflect the behavior near the lower end of the interval, we treat f_2 as Hölder continuous in the following analysis. Conversely, the function f_3 does not even admit a uniformly continuous representation as $x_1 \in [0, 100]$ (cf. [ld25c])[6]. Moreover, both functions in Eq. (A.6.28) possess a broadly analogous overall structure, with the exception of the outermost function. The function f_3 smoothed the outputs by applying a Lipschitz-continuous, sigmoidal-like function (arctan), which bound the range and mitigated extreme variations. Conversely, the function f_2 introduces a Hölder-continuous function $(\cdot^{0.5})$ to its outputs, resulting in increased variability and irregularity. This, in turn, leads to heightened sensitivity to input fluctuations. In these cases, we do not include irrelevant features, resulting in smaller input dimensions. This allows us to add more parameters to each approach for improved flexibility in model tuning.

	fs KANN		DNN	aw KANN		aw KAN
σ	$\mathcal{F}_{\mathcal{S}^{\sigma,32},I,n}$	$\mathcal{F}_{\mathcal{S}^{\sigma,128},I,n}$	$\mathcal{D}^{\sigma,3,512}$	$\mathcal{F}_{\mathcal{S}^{\sigma,64},I(4,16,1)}$	$\mathcal{F}_{\mathcal{S}^{\sigma,16},I(4,128,1)}$	$\mathcal{F}_{S_{3,t},I,(4,64,1)}$
0.0	$2.9 \cdot 10^{-1}$	$3.9 \cdot 10^{-1}$	$5.7 \cdot 10^{-1}$	$4.11 \cdot 10^{-1}$	$1.7 \cdot 10^{-1}$	$\mathbf{4.1 \cdot 10^{-2}}$
0.5	1.02	6.2	2	$4.17 \cdot 10^{-1}$	$2.8 \cdot 10^{-1}$	$\mathbf{7.1 \cdot 10^{-2}}$
1.0	3.8	$3.7 \cdot 10^{-1}$	5	$6 \cdot 10^{-1}$	$4.6 \cdot 10^{-1}$	$\mathbf{2.1 \cdot 10^{-1}}$
2.0	1.5	1.1	1.6	$8 \cdot 10^{-1}$	$\mathbf{4.8 \cdot 10^{-1}}$	1
5.0	3.7	10.8	2.8	1.46	**1.3**	9

Table 2: Performance evaluation of models on the Friedmann dataset [ld25b] across varying noise levels. The reported values represent the RMSE on test data without noise. The shape of the fixed-size approaches is given as $n = (4, 6, 3, 2, 1, 1)$. All models were trained for 1000 epochs using the AdamW optimizer [PyT25]. The input data was standardized cf. [ld21]. For the KANs we choose $|t| = 10$.

The results presented in Table 2 are obtained by training on the function f_2 in Eq. (A.6.28). Higher levels of noise are employed, as the range of this function is significantly larger, that is, $f_2(x) \in (5.1 \cdot 10^{-5}, 1762.13)$ compared to other Friedmann functions. The substantial variations in outcomes, particularly for the fs KANNs $\mathcal{F}_{\mathcal{S}^{\sigma,128},I,n}$ can be attributed to a highly unstable training process. It was observed that convergence was achieved even in the absence of noise, provided that adaptive learning rate strategies were employed, and these strategies were applied once a predefined loss threshold was attained. This approach was further extended to the DNNs [7]. However, this was not a prerequisite for the aw KANNs. One plausible explanation for the more robust training process of aw KANNs is that the increased sparsity of the respective weight matrices leads to

[6] We identify for $f : x \mapsto \frac{1}{x}$ $f(0) := \lim_{x \to 0^+} f(x)$

[7] For the fs KANs, no substantial convergence was observed during the training process.

smaller operator norms of the gradients, which in turn reduces the effective learning rate. A notable observation is that reducing the learning rate from the outset causes both the fs KANNs and DNNs to converge too slowly. Furthermore, fluctuations in the training process are observed even when adjusting the learning rate during training (cf. Table 2). Furthermore, aw KANNs may initially converge slower but still at a significant pace, remaining sufficiently fast. One possible interpretation of these phenomena is that this instability lies in the challenging trade-off between the Lipschitz constant of the univariate approximation function and the accurate estimation of univariate Hölder-continuous functions in the case of employing fs approaches (cf. Remark 5.2). The latter generally necessitates substantial Lipschitz constants for approximation functions, which can result in unstable minima, particularly in the presence of noise. Furthermore, the aw KANNs demonstrated a convergence pattern consistent with the estimation of smooth continuous functions, as discussed in Remark 5.1. It is noteworthy that we have observed satisfactory convergence even when employing aw and arbitrary-depth KANNs. However, we do not assess these in their entirety due to the extended training time required (cf. Section 6.3.1). However, for the specific case of $\mathcal{F}_{\mathcal{S}^{\sigma,16},I,(4,64,64,1)}$ and $\sigma^2 = 1$, we observed a test loss of $1.9 \cdot 10^{-1}$. The mitigation of these counterproductive effects could be achieved through the implementation of specific training strategies, as elaborated in Section 6.3.2. The results presented in Table 3 are obtained by training on the function f_3 in Eq. (A.6.28). In this context, the significant variations (leading to non-uniform continuity) observed near 0 of the functions $\frac{1}{x}$ are effectively mitigated by the sigmoidal nature of the arctan function. The behavior of the function arctan is more smooth for large values, which reduces the sensitivity to large fluctuations. This effect culminates in a more stable approximation of the function. The fs KANNs demonstrates superior performance in this instance, capitalizing on its smoothing property. Interestingly, as opposed to the other experiments, the KAN-based approaches do not exhibit superior performance in the noise free case. This can be explained by the large range of the function to which the arctan is applied, leading to large distances between the grid points of the splines. Assuming the grid points are uniformly distributed, this results in suboptimal outcomes, as the non-uniform complexity of sigmoid functions is not well captured. The results presented in Tables 1, 2, and 3 illustrate that the optimal approach varies depending on the specific task, emphasizing the importance of selecting an appropriate model for each scenario.

	fs KANN		DNN	aw KANN		fs KAN	aw KAN
σ^2	$\mathcal{F}_{\mathcal{S}^{\sigma,32},I,n}$	$\mathcal{F}_{\mathcal{S}^{\sigma,128},I,n}$	$\mathcal{D}^{\sigma,3,512}$	$\mathcal{F}_{\mathcal{S}^{\sigma,64},I(4,16,1)}$	$\mathcal{F}_{\mathcal{S}^{\sigma,16},I(4,128,1)}$	$\mathcal{F}_{S_{3,t},I,n}$	$\mathcal{F}_{S_{3,t},I,(4,64,1)}$
0.0	$\mathbf{5.1 \cdot 10^{-5}}$	$1.1 \cdot 10^{-4}$	$1.1 \cdot 10^{-4}$	$1.4 \cdot 10^{-4}$	$2.4 \cdot 10^{-4}$	$6.8 \cdot 10^{-3}$	$5.6 \cdot 10^{-2}$
0.2	$\mathbf{1.4 \cdot 10^{-3}}$	$2 \cdot 10^{-3}$	$3.1 \cdot 10^{-2}$	$2.2 \cdot 10^{-3}$	$2.8 \cdot 10^{-3}$	$1 \cdot 10^{-2}$	$3.5 \cdot 10^{-2}$
0.5	$\mathbf{6.2 \cdot 10^{-3}}$	$1.6 \cdot 10^{-2}$	$2.2 \cdot 10^{-1}$	$1.1 \cdot 10^{-2}$	$1.9 \cdot 10^{-2}$	$7.8 \cdot 10^{-2}$	$2.1 \cdot 10^{-1}$
1.0	$\mathbf{2.9 \cdot 10^{-2}}$	$6.2 \cdot 10^{-2}$	1.02	$4.7 \cdot 10^{-2}$	$7.2 \cdot 10^{-2}$	$2.8 \cdot 10^{-1}$	$8.3 \cdot 10^{-1}$

Table 3: Performance evaluation of models on the Friedmann dataset [ld25c] across varying noise levels. The reported values represent the RMSE on test data without noise. The shape of the fs KANNs is given as $n = (4, 13, 1, 1)$. All models were trained for 1000 epochs using the AdamW optimizer [PyT25]. The input data was standardized cf. [ld21]. For the KANs we choose $|t| = 10$.

References

[AU20] Majid Afshar and Hamid Usefi. High-dimensional feature selection for genomic datasets. *Knowledge-Based Systems*, 195:105724, 2020.

[Bru21] L. Brutman. Lebesgue functions for polynomial interpolation: A survey. *Department of Mathematics and Computer Science, University of Haifa*, 2021. Dedicated to Ted Rivlin, on the occasion of his 70th birthday.

[Buc79] R. Buck. Approximate complexity and functional representation. *J. Math. Anal. Appl.*, 70(1):280–298, 1979.

[Cha25] ChatGPT, based on GPT-4-turbo. Language refinement assistance. OpenAI, San Francisco, CA, 2025. Used to support phrasing and language improvements.

[Che01] E. W. Cheney. *Introduction to Approximation Theory.* AMS Chelsea Publishing, Providence, RI, 2001.

[Cyb89] G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems*, 2(4):303–314, 1989.

[dB01] Carl de Boor. *A Practical Guide to Splines (Revised Edition).* Springer, 2001.

[dBF73] C. de Boor and G. J. Fix. Spline approximation by quasiinterpolants. *Journal of Approximation Theory*, 8:19–45, 1973. Communicated by Oued Shisha.

[Dee25] DeepL SE. Deepl write. https://www.deepl.com/write, 2025. Used to improve grammar, clarity, and style.

[Dep24] Department I – Student Services, Examinations. Guidelines for Completing Bachelor's and Master's Theses, April 2024. Version from April 2024.

[Dev24] PyTorch Developers. torch.optim.lbfgs, 2024. Accessed: March 22, 2025.

[Flo23] Michael S. Floater. *An Introduction to Spline Theory.* 2023. Draft date May 22, 2023.

[Fri91] J. Friedman. Multivariate adaptive regression splines. *The Annals of Statistics*, 19(1):1–67, 1991.

[HSW$^+$21] Rui Hu, Jitao Sang, Jinqiang Wang, Rui Hu, and Chaoquan Jiang. Understanding and testing generalization of deep networks on out-of-distribution data. *arXiv preprint arXiv:2111.09190*, 2021.

[Kol57] A. N. Kolmogorov. On the representation of continuous functions of several variables by superposition of continuous functions of one variable and addition. *Doklady Akademii Nauk SSSR*, 114:953–956, 1957.

[ld21] Scikit learn developers. Standardscaler. Scikit-learn documentation, 2021. Accessed: 2025-03-19.

[lD24] Scikit learn Developers. California housing dataset, 2024.

[ld25a] Scikit learn developers. makefriedman1, 2025. Accessed: 2025-03-13.

[ld25b] Scikit learn developers. makefriedman2, 2025. Accessed: 2025-03-13.

[ld25c] Scikit learn developers. makefriedman3. Scikit-learn Documentation, 2025. Accessed: 2025-03-19.

[LWV$^+$24] Ziming Liu, Yixuan Wang, Sachin Vaidya, Fabian Ruehle, James Halverson, Marin Soljačić, Thomas Y. Hou, and Max Tegmark. KAN: Kolmogorov–Arnold Networks. *arXiv preprint arXiv:2404.19756v1*, 2024. License: CC BY 4.0.

[Mar18] Gary Marcus. Deep learning: A critical appraisal, 2018.

[Pra25] Justus Prass. https://github.com/Justus-ui/KAN_NN, 2025. Accessed: 2025-04-08.

[PyT25] PyTorch. torch.optim.adamw, 2025. Accessed: 2025-03-04.

[SL19] Shaeke Salman and Xiuwen Liu. Overfitting mechanism and avoidance in deep neural networks, 2019.

[TGLM22] Neil C. Thompson, Kristjan Greenewald, Keeheon Lee, and Gabriel F. Manso. The computational limits of deep learning, 2022.

[Weg23] Sven Ake Wegner. *Mathematische Einführung in Data Science*. Springer, 2023.

[Wer18] D. Werner. *Funktionalanalysis*. Springer Spektrum, Berlin, 8. überarbeitete auflage edition, 2018.

[ZTLT21] Yu Zhang, Peter Tino, Ales Leonardis, and Ke Tang. A survey on neural network interpretability. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 5(5):726–742, October 2021.