SIAM J. MATH. DATA SCI.
Vol. 5, No. 2, pp. 306–322

Approximation of Lipschitz Functions Using Deep Spline Neural Networks*

Sebastian Neumayer†, Alexis Goujon†, Pakshal Bohra†, and Michael Unser†

Abstract. Although Lipschitz-constrained neural networks have many applications in machine learning, the design and training of expressive Lipschitz-constrained networks is very challenging. Since the popular rectified linear-unit networks have provable disadvantages in this setting, we propose using learnable spline activation functions with at least three linear regions instead. We prove that our choice is universal among all componentwise 1-Lipschitz activation functions in the sense that no other weight-constrained architecture can approximate a larger class of functions. Additionally, our choice is at least as expressive as the recently introduced non-componentwise Groupsort activation function for spectral-norm-constrained weights. The theoretical findings of this paper are consistent with previously published numerical results.

Key words. deep learning, learnable activations, universality, robustness, Lipschitz continuity, linear splines

MSC codes. 26A16, 26B40, 41A15, 41A29, 65D07, 68T01, 94A15

DOI. 10.1137/22M1504573

1. Introduction.

Lipschitz-constrained neural networks (NNs) have proven to be useful in several areas of machine learning, for instance in the context of provably convergent Plug-and-Play algorithms [18, 24, 28, 31, 34, 37], to obtain robustness guarantees [16, 26, 35], or in Wasserstein generative adverserial networks (GANs) [2, 15]. However, the design and training of Lipschitz-constrained NNs is difficult, as the computation of the Lipschitz constant of multilayer models is known to be NP-hard. A simple upper bound is given by the product of the Lipschitz constant of each layer, but it is usually very coarse. There exist more precise estimators based on semidefinite programming [13, 21], adversarial training [8, 27], or the derivation of sharper estimates for the composition of layers [38]. Unfortunately, these methods are either computationally expensive or do not provide a proper upper bound.

A possible strategy to address these shortcomings is to design the model architecture so that the fast-to-evaluate bounds become sharper. A general overview of NN architectures and, in particular, Lipschitz-constrained ones can be found in [9]. The most common approach toward Lipschitz-constrained architectures is to control the norm of each linear layer, typically with the spectral or other p-norms [14, 25, 29], or by enforcing orthogonality of the weight matrices [17, 18, 19]. In combination with 1-Lipschitz activations, this results in architectures

* Received by the editors June 27, 2022; accepted for publication (in revised form) January 19, 2023; published electronically May 15, 2023. Sebastian Neumayer and Alexis Goujon are contributed equally to this work.
https://doi.org/10.1137/22M1504573
Funding: The research leading to these results was supported by the European Research Council (ERC) under European Union's Horizon 2020 (H2020), grant agreement - Project 101020573 FunLearn and by the Swiss National Science Foundation, grant 200020 184646/1.

† Biomedical Imaging Group, École polytechnique fédérale de Lausanne (EPFL), CH-1015 Lausanne, Switzerland (sebastian.neumayer@epfl.ch, alexis.goujon@epfl.ch, pakshal.bohra@epfl.ch, michael.unser@epfl.ch).

with a Lipschitz constant bounded by the product of the norms of the weights. However, this estimate is, in general, quite pessimistic, especially for deep models. Consequently, this additional structural constraint often leads to vanishing gradients [22] and a seriously reduced expressivity of the model. Remarkably, the commonly used rectified linear-unit (ReLU) activation aggravates the situation. For instance, it is shown in [20] that ReLU NNs with ∞-norm weight constraints have a second-order total variation that is bounded independently of the depth. Further, it is proven in [1] that, under spectral norm constraints, any scalar-valued ReLU NN Φ with $\|\nabla\Phi\|_2 = 1$ a.e. is necessarily linear. To circumvent the described issues, several new activation functions have been proposed recently, such as Groupsort [1] or the related Householder [30] activation functions. Note that, contrary to ReLU, all of these activation functions are multivariate. Analyzing the expressivity of the resulting NNs and determining their applicability in practice is an active area of research.

It is by no means trivial to specify which class of functions can be approximated by a generic NN with 1-Lipschitz layers. Ideally, given a compact set $D \subset \mathbb{R}^d$ equipped with the p-norm, it is desirable to approximate all scalar-valued 1-Lipschitz functions, which are denoted by $\mathrm{Lip}_{1,p}(D)$. The first result in this direction was provided in [1], where the authors show that the use of the Groupsort activation function and ∞-norm-constrained weights indeed allows for the universal approximation of $\mathrm{Lip}_{1,p}(D)$. The behavior of such NNs was then further investigated in [11, 32]. Unfortunately, the proof strategies published so far cannot be generalized to other norms and not even partial results are known for this very challenging problem. Therefore, being able to compare the approximation capabilities of different architectures is an important first step. For example, the approximation of the absolute value function, for which an exact representation with ReLU is impossible, provides a classic benchmark to compare architectures. From a practical perspective, Groupsort NNs have yielded promising results and compare favorably against ReLU NNs with similar architectures [1].

Currently, the most substantial results in this area rely on multivariate activation functions. Although the ReLU activation function is indeed too limiting, we claim that the class of componentwise activation functions ought not to be dismissed off-hand. Following this idea, we analyze deep spline NNs, whose activation functions are learnable linear splines [3, 5, 36]. Since bounds on the Lipschitz constant of compositions are usually too pessimistic, our rationale is to increase the expressivity of the activation function while still being able to efficiently control its Lipschitz constant. As reported first in [6], Lipschitz-constrained deep spline NNs perform well in practice and a more systematic comparison against other frameworks can be found in [12]. In this work, we shed light on the theoretical benefits of these NNs over ReLU-like NNs. In particular, we prove that the choice of learnable linear spline activation functions with three regions is universal among all componentwise 1-Lipschitz activation functions. In other words, no other weight-constrained NN with componentwise activation functions can approximate a larger class of functions. Moreover, for the spectral-norm constraint, which is commonly used in practice, we show that deep spline NNs are at least as expressive as Groupsort NNs.

Outline and contributions. In section 2, we revisit 1-Lipschitz continuous piecewise-linear (CPWL) functions and 1-Lipschitz NNs In particular, we show that they can approximate any function in $\mathrm{Lip}_{1,p}(D)$. Since the construction of 1-Lipschitz NNs is nontrivial, we briefly discuss two architectures for this task, namely deep spline and Groupsort NNs. Then, in

section 3, we extend some known results on the limitations of weight-constrained NNs with ReLU activation functions. More precisely, we show that ReLU-like NNs cannot represent certain simple functions for any p-norm weight constraint. Based on a second-order total variation argument, we further show that they cannot be universal approximators for ∞-norm weight constraints. Next, in section 4, we study the approximation properties of deep spline NNs. Here, we prove our main result, according to which deep spline NNs with three linear regions achieve the maximum expressivity among NNs with componentwise activation functions. Further, we discuss the relation between deep spline and Groupsort NNs. Finally, we draw conclusions in section 5.

2. Lipschitz-constrained NNs.

In this paper, we investigate feedforward NN architectures that consist of $K \in \mathbb{N}$ layers with widths $n_1, \dots, n_K$ that are given by mappings $\Phi \colon \mathbb{R}^d \to \mathbb{R}^{n_K}$ of the form

$$\Phi(x) := A_K \circ \sigma_{K-1,\alpha_{K-1}} \circ A_{K-1} \circ \sigma_{K-2,\alpha_{K-2}} \circ \cdots \circ \sigma_{1,\alpha_1} \circ A_1(x). \tag{2.1}$$

Here, the affine functions $A_k \colon \mathbb{R}^{n_{k-1}} \to \mathbb{R}^{n_k}$ are given by

$$A_k(x) := W_k x + b_k, \qquad k = 1, \dots, K, \tag{2.2}$$

with weight matrices $W_k \in \mathbb{R}^{n_k, n_{k-1}}$, $n_0 = d$ and bias vectors $b_k \in \mathbb{R}^{n_k}$. For multilayer perceptrons, W_k is learned as a full matrix, while for convolutional NNs, W_k is parametrized via a convolution operator whose kernel is learned. The model includes parameterized nonlinear activation functions $\sigma_{k,\alpha_k} \colon \mathbb{R}^{n_k} \to \mathbb{R}^{n_k}$ with corresponding parameters α_k, $k = 1, \dots, K-1$. For the case of componentwise activation functions, we have that $\sigma_{k,\alpha_k}(x) = (\sigma_{k,\alpha_k,j}(x_j))_{j=1}^{n_k}$. We sometimes drop the index k in the activation function σ_{k,α_k} to simplify the notation. The complete parameter set of the NN is denoted by $u := (W_k, b_k, \alpha_k)_{k=1}^K$ and the NN by $\Phi(\cdot, u)$ whenever the dependence on the parameters is explicitly needed. For an illustration, see Figure 2.1. Architecture (2.1) results in a CPWL function whenever the activation functions themselves are CPWL functions such as the ReLU. Next, we investigate the approximation properties of this architecture under Lipschitz constraints on $\Phi(\cdot, u)$.

2.1. Universality of 1-Lipschitz ReLU networks.

First, we briefly revisit the approximation of Lipschitz function by CPWL functions, for which we give a precise definition.

Definition 2.1. *A continuous function $f \colon \mathbb{R}^d \to \mathbb{R}^n$ is called continuous and piecewise linear if there exist a finite set $\{f^m \colon m = 1, \dots, M\}$ of affine functions, also called affine pieces, and closed sets $(\Omega_m)_{m=1}^M \subset \mathbb{R}^d$ with nonempty and pairwise-disjoint interiors, also called projection regions [33], such that $\cup_{m=1}^M \Omega_m = \mathbb{R}^d$ and $f_{|\Omega_m} = f^m_{|\Omega_m}$.*

Assume that we are given a collection of tuples $(x_i, y_i) \in \mathbb{R}^d \times \mathbb{R}$, $i = 1, \dots, N$, which can be interpreted as samples from a function $f \colon \mathbb{R}^d \to \mathbb{R}$. Let

$$L^p_{x,y} := \max_{i \neq j} \frac{|y_i - y_j|}{\|x_i - x_j\|_p} \tag{2.3}$$

denote the Lipschitz constant associated with these points. Then, a first natural question is whether it is always possible to find an interpolating CPWL function g with p-norm Lipschitz constant $\mathrm{Lip}_p(g) = L^p_{x,y}$.

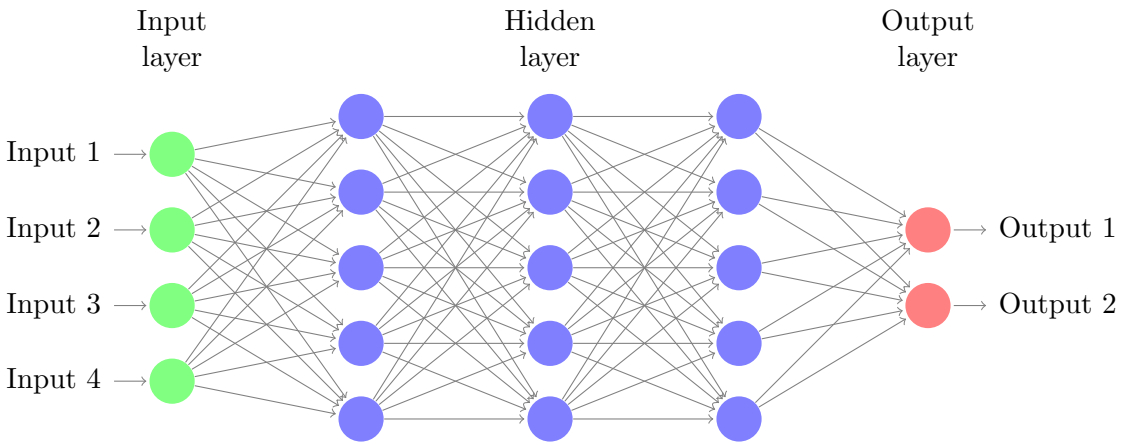

Figure 2.1. *Model of a feedforward NN with three hidden layers, where* $d=4$, $K=4$, $n_1=n_2=n_3=5, n_4=2$.

Proposition 2.2. *For the tuples* $(x_i, y_i) \in \mathbb{R}^d \times \mathbb{R}$, $i=1,\dots,N$ *and* $p \in [1,+\infty]$, *there exists a CPWL function* f *with* $\mathrm{Lip}_p(f) = L^p_{x,y}$ *such that* $g(x_i) = y_i$ *for all* $i=1,\dots,N$.

Since we are unaware of a proof for general p, we provide one below.

Proof. Let q be such that $1/p + 1/q = 1$. For $p < +\infty$, define $u_{i,j} \in \mathbb{R}^d$ as the vector given by

$$(u_{i,j})_k = \mathrm{sgn}((x_i - x_j)_k)|(x_i - x_j)_k|^{p/q}. \tag{2.4}$$

If $p=+\infty$, we choose k_0 with $\|x_i - x_j\|_\infty = |(x_i - x_j)_{k_0}|$, and define $(u_{i,j})_{k_0} = \mathrm{sgn}(x_i - x_j)_{k_0}$ with all other components of $u_{i,j}$ set to 0. This saturates Hölder's inequality with

$$\langle u_{i,j}, x_j - x_i\rangle = \sum_{k=1}^{d} |(u_{i,j})_k (x_j - x_i)_k| = \|u_{i,j}\|_q \|x_j - x_i\|_p, \tag{2.5}$$

where we used that $u_{i,j}$ and $(x_j - x_i)$ have components with the same sign. For $i \neq j$, we define the linear function

$$f_{i,j}(x) = y_i + \frac{y_j - y_i}{\|x_j - x_i\|_p \|u_{i,j}\|_q} \langle u_{i,j}, x - x_i\rangle, \tag{2.6}$$

which is such that $f_{i,j}(x_i) = y_i$ and $\mathrm{Lip}_p(f_{i,j}) = |y_j - y_i|/\|x_j - x_i\|_p$, as $\sup_{\|x\|_p \leq 1} \langle u_{i,j}, x\rangle = \|u_{i,j}\|_q$. Next, set $f_i(x) = \max_{j, j\neq i} f_{i,j}(x)$ for which it holds that $f_i(x_i) = y_i$ and $\mathrm{Lip}_p(f_i) = \max_j |y_j - y_i|/\|x_j - x_i\|_p$. Then, we define $f(x) = \min_i f_i(x)$ and directly obtain that $f(x_j) \leq y_j$ for any $j = 1,\dots,N$. However, we also have that

$$f_i(x_j) \geq f_{i,j}(x_j) = y_i + y_j - y_i = y_j, \tag{2.7}$$

which then implies that $f(x_j) = y_j$ for any $j = 1,\dots,N$. Further, we directly get that $\mathrm{Lip}_p(f) = L^p_{x,y}$. Finally, by recalling that the maximum and the minimum of any number of CPWL functions is CPWL as well [33], we conclude that f is CPWL and the claim follows. ∎

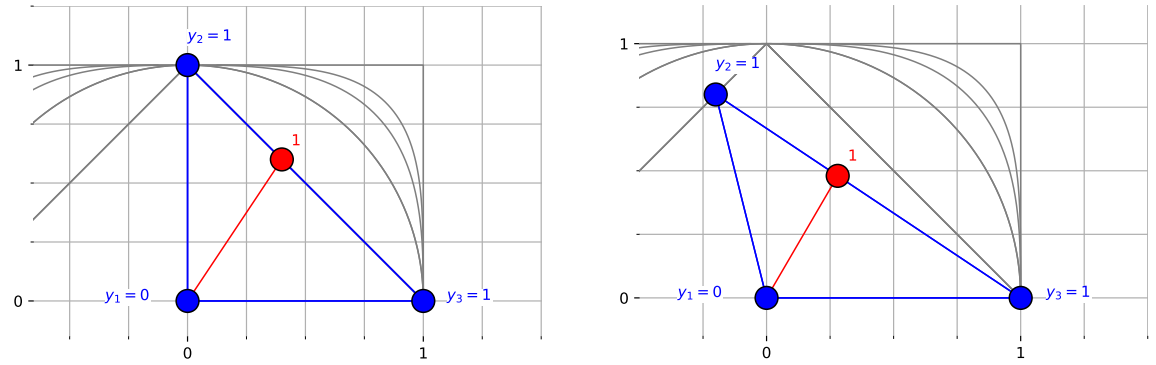

Figure 2.2. *Interpolation based on a triangulation: Let $x_1, x_2, x_3 \in \mathbb{R}^2$ be input data points (blue dots) with corresponding target values $y_1 = 0$, $y_2 = 1$, and $y_3 = 1$. The gray curves depict the ℓ_p unit balls for $p \in \{1, 2, 3, 4, +\infty\}$. For the left plot, we set $p > 1$ and get $L^p_{x,y} = 1$. In the right, we set $p = 1$ and also get $L^p_{x,y} = 1$. The unique affine function $g\colon \mathbb{R}^2 \to \mathbb{R}$ interpolating the data is the simplest CPWL function that fits the data. On any point x lying between x_2 and x_3 (red dot), it holds that $g(x) = 1$, hence $|g(x_1) - g(x)| = 1$. However, in both settings x is in the unit ball for the according p which implies that $\|x_1 - x\|_p < 1$. Hence, $\mathrm{Lip}_p(g) > L^p_{x,y}$ and g does not interpolate the data with the minimal Lipschitz constant.*

Remark 2.3. The d-dimensional construction is more involved than the one-dimensional (1D) case, for which a simple interpolation is sufficient. A natural way to fit the data in any dimension is to form a triangulation with vertices $(x_i)_{i=1}^N$. Then, with the use of the CPWL hat basis functions of the triangulation, one can directly form an interpolating CPWL function. Unfortunately, the Lipschitz constant of this function can exceed $L^p_{x,y}$. An example of this issue is provided in Figure 2.2.

Since the maximum and minimum of finitely many affine functions can be represented by ReLU NNs, the same holds true for the CPWL function constructed in Proposition 2.2. This directly leads us to a well-known corollary.

Corollary 2.4. *Let $D \subset \mathbb{R}^d$ be compact, and let $p \in [1, +\infty]$. Then, the ReLU NNs $\Phi\colon D \to \mathbb{R}$ with $\mathrm{Lip}_p(\Phi) \leq 1$ are dense in $\mathrm{Lip}_{1,p}(D)$.*

Since computing the Lipschitz constant of a generic NN is NP-hard, Corollary 2.4 has limited practical relevance. To circumvent this issue, either algorithms that provide tight estimates, or special architectures with simple yet sharp bounds, are necessary. In this paper, we pursue the second direction. To this end, we introduce tools to build Lipschitz-constrained architectures in the remainder of this section and investigate the universality of these architectures in section 4.

2.2. 1-Lipschitz network architectures. A first step toward Lipschitz-constrained NNs is to constrain the norm of the weights. As we are aiming for 1-Lipschitz NNs, we always constrain them by one, but remark that other values are possible as well. If we further impose that all activation functions $\sigma_{k,\alpha}$ are 1-Lipschitz, then the resulting NN is also 1-Lipschitz.

Operator-norm constraints. The $p \to q$ operator norm is given for $W \in \mathbb{R}^{n,m}$ and $p, q \in [1, +\infty]$ by

$$\|W\|_{p,q} := \max_{x \in \mathbb{R}^m, \|x\|_p = 1} \|Wx\|_q \tag{2.8}$$

and $\|\cdot\|_p := \|\cdot\|_{p,p}$. Note that $\|\cdot\|_1$ and $\|\cdot\|_\infty$ correspond to the maximum ℓ_1 norm of the columns and rows of W, respectively. The norm $\|\cdot\|_2$, also known as spectral norm, corresponds to the largest singular value of W. To obtain a nonexpansive NN of the form (2.1) in the p-norm sense, the weight matrices can be constrained as

$$\|W_k\|_p \leq 1, \quad k = 1, \dots, K, \tag{2.9}$$

which we shall henceforth refer to as p-norm-constrained weights. For matrices $W \in \mathbb{R}^{1,n}$ it holds that $\|W\|_p = \|W^T\|_q$ with $1/p + 1/q = 1$. In other words, if we interpret these matrices as vectors, then we have to constrain the q-norm instead. In the case of scalar-valued NNs, we can also constrain the weights as $\|W_k\|_q \leq 1$, $k = 2, \dots, K$, and $\|W_1\|_{p,q} \leq 1$, since all standard norms are identical in $\mathbb{R}$. There exist several methods to enforce such constraints in the training stage [14, 25, 29]; see Remark 2.5 for more details.

Orthonormality constraints. Instead of imposing $\|W\|_2 \leq 1$, we can also require that either $W^T W = \mathrm{Id}$ or $W W^T = \mathrm{Id}$, depending on the shape of W. This constraint corresponds to imposing that either W or W^T lie in the so-called Stiefel manifold. Compared to the spectral-norm constraint, the orthonormality constraint enforces all singular values of W to be unity. From a computational perspective, this approach is more challenging than the previous one but helps to mitigate the problem of vanishing gradients in deep NNs. For more details, including possible implementations, we refer to [17, 18, 19].

Remark 2.5. Many of the implementations for the schemes of section 2.2 enforce the p-norm constraint or orthonormality only approximately. For theoretical guarantees, it is, however, necessary to ensure that the constraint is satisfied exactly. In practice, this means that sufficient numerical accuracy or additional postprocessing after training might be necessary.

2.3. Special activation functions. While the quest for optimal activation functions in the last decade leaves us with many choices, the 1-Lipschitz constraint is the game-changer and the relevance of each activation function must be reassessed. In section 3, we provide results that explain why the ReLU activation function is actually not suited in a Lipschitz-constrained setting. Hence, we need to resort to other activation functions that lead to increased expressivity of the resulting NN. There is a fundamental conceptual difference between componentwise and general multivariate activation functions. In particular, finding a good trade-off in terms of representational power and computational complexity is necessary. In the following, we briefly discuss two corresponding families of activation functions, which have been shown experimentally to be well suited in the constrained setting. Then, we further explore their usability in the norm-constrained case and investigate the relations between the two approaches.

Deep spline NNs. A deep spline NN [4, 5, 36] uses learnable componentwise linear-spline activation functions; see Figure 2.3. It is known that deep spline NNs are solutions of a functional optimization problem; namely, the training of a neural network with free-form activation functions whose second-order total-variation is regularized [36]. A linear-spline activation function is fully characterized by its linear regions and the corresponding values at the boundaries. In the unconstrained setting, any linear spline can be implemented by means of a scalar one-hidden-layer ReLU NN as

$$x \mapsto \sum_{m=1}^{M} u_m \mathrm{ReLU}(v_m x + b_m), \tag{2.10}$$

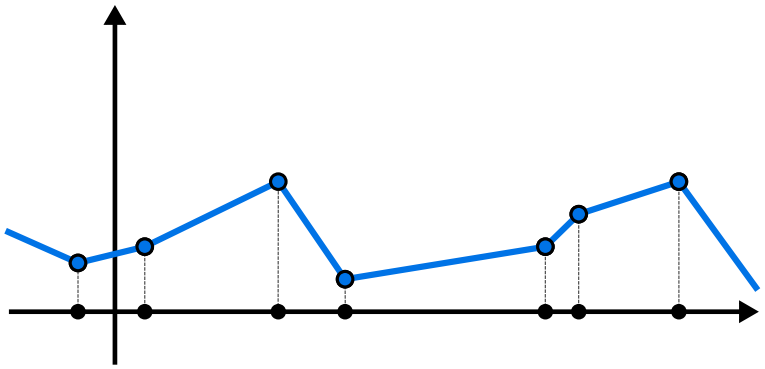

Figure 2.3. *Linear spline with seven knots (also known as breakpoints) and eight linear regions.*

where $u_m, v_m, b_m \in \mathbb{R}$ and $M \in \mathbb{N}$. This parameterization, however, lacks expressivity under p-norm constraints on the weights, as it is not able to produce linear splines with second-order total variation greater than 1, as discussed in Lemma 3.2 and section 3.2. Instead, it is more convenient to rely on local B-spline atoms [5]. In practice, the linear-spline activation functions have a fixed number of uniformly spaced breakpoints—typically between 10 and 50—and are expressed as a weighted sum of cardinal B-splines. This amounts to adding a learnable parameter for each breakpoint and two additional ones to set the slope at both ends for a linear extrapolation. This local parameterization yields an evaluation complexity that remains independent of the number of breakpoints, in contrast with (2.10). The B-spline framework can easily be adapted to learn 1-Lipschitz activation functions via the use of a suitable projector on the B-splines coefficients [12].

Among weight-constrained NNs with componentwise activation functions, deep spline NNs achieve the optimal representational power.

Lemma 2.6. *Let $(x_n, y_n) \in (\mathbb{R}^d, \mathbb{R}^p)$, $n = 1, \dots, N$, and Φ a NN with K layers, parameter set u, p-norm weight constraints and 1-Lipschitz activation functions. Then, there exists a deep spline NN denoted by* DS *with the same architecture, where the activation functions are replaced by 1-Lipschitz linear splines with no more than $(N-1)$ linear regions such that*

$$\Phi(x_n, u) = \mathrm{DS}(x_n, u) \quad \textit{for } n = 1, \dots, N. \tag{2.11}$$

Proof. On the data points $(x_n, y_n)_{n=1}^N$, the activation functions of Φ are evaluated for at most N different values. Hence, the result directly follows by interpolating these values using a linear spline, which yields 1-Lipschitz linear-spline activation functions. ∎

This result is somehow still unsatisfying as the number of linear regions grows with the number of training points. Later, we show that linear-spline activation functions with three linear regions are actually sufficient. This amounts to six tunable parameters per activation function.

Groupsort. The sort operation takes a vector of dimension n and simply outputs its components sorted in ascending order. This operation has complexity $\mathcal{O}(n \log(n))$, which is slightly worse than the linear complexity of componentwise activation functions. The Groupsort activation function [1] is a generalization of this operation: it splits the preactivation into groups

of prescribed length and performs the sort operation within each group. This results in near-linear complexity when the group length are small enough. If the group length is two, then the activation function is known as the MaxMin or norm-preserving orthogonal-permutation linear unit [10]. Let us remark that any arbitrary Groupsort activation function can be written as composition of MaxMin activation functions, i.e., larger group lengths do not increase the theoretical expressivity. Although not obvious at first glance, the Groupsort activation function is actually a CPWL operation. The rationale for this activation function is to perform a nonlinear and norm-preserving operation, which mitigates the issue of vanishing gradients in deep constrained architectures. More precisely, we have that the Jacobian of the Groupsort activation function is a.e. given by a permutation matrix, which is indeed an orthogonal matrix. Motivated by this observation, this approach was recently generalized [30] to yield the Householder activation functions $\sigma_v \colon \mathbb{R}^d \to \mathbb{R}^d$ with $v \in \mathbb{R}^d$, $\|v\|_2 = 1$, given by

$$\sigma_v(z) = \begin{cases} z & \text{if } v^T z > 0, \\ (\mathrm{Id} - 2vv^T)z & \text{otherwise.} \end{cases} \tag{2.12}$$

On the hyperplane that separates the two cases (i.e., $v^T z = 0$) we have that $(I - 2vv^T)z = z - 2(v^T z)v = z$. Thus, σ_v is continuous and, moreover, the Jacobian is either I or $(I - 2vv^T)$, which are both square orthogonal matrices. For practical purposes, the authors of [30] recommend using groups of length 2. This construction can be iterated to obtain higher-order Householder activation functions with more linear regions.

3. Limitations of certain architectures.

In this section, we provide results that explain why the use of activation functions that are more complex than the ReLU is indeed necessary for weight-constrained NNs.

3.1. Diminishing Jacobians.

Componentwise and monotone activation functions are detrimental to the expressivity of NNs with spectral-norm-constrained weights [1, Thm. 1]. Here, we generalize this result to NNs with p-norm-constrained weights and certain CPWL activation functions, along with a more precise characterization. In particular, we also cover the case where $\|J\Phi\|_p$ is not 1 a.e.

Proposition 3.1. *Let $p \in (1, +\infty]$, let $I \subset \mathbb{R}$ be a closed interval, and let $\sigma \colon \mathbb{R} \to \mathbb{R}$ be a CPWL activation function satisfying*

- *$\sigma(x) = x + b$, $b \in \mathbb{R}$, for $x \in I$,*
- *$|\sigma'(x)| < 1$ for $x \notin I$.*

Then, any NN $\Phi \colon \mathbb{R}^d \to \mathbb{R}$ of the form (2.1) with p-norm-constrained weights and activation function σ has at most one affine region Ω_i with $\|J\Phi|_{\Omega_i}\|_p = 1$.

Proof. We proceed via induction over the number K of layers of Φ. For $K = 1$, the mapping is affine and the statement holds trivially. Now, assume that the result holds for some $K > 1$. Let

$$\Phi_{K+1} = A_{K+1} \circ \sigma \circ A_K \circ \cdots \circ \sigma \circ A_1, \tag{3.1}$$

which we decompose as $\Phi_{K+1} = \Phi_K \circ h$ with $\Phi_K = A_{K+1} \circ \sigma \circ A_K \circ \cdots \circ \sigma \circ A_2$ and $h = \sigma \circ A_1$. The induction assumption implies that $\|J\Phi_K\|_p < 1$ on all affine regions except possibly one.

The corresponding affine function $f_K^1 \colon \mathbb{R}^{n_1} \to \mathbb{R}$ with projection region $\Omega_K \subset \mathbb{R}^{n_1}$ takes the form $x \mapsto v^T x + c$, where $v \in \mathbb{R}^{n_1}$ is such that $\|v\|_q \leq 1$, $1/p + 1/q = 1$, and $c \in \mathbb{R}$. Now, we define the set

$$\Omega_{K+1} = \{x \in \mathbb{R}^d : (A_1(x))_l \in I \text{ for any } l \text{ s.t. } v_l \neq 0\} \cap h^{-1}(\Omega_K). \tag{3.2}$$

By construction, Φ_{K+1} is affine on Ω_{K+1} and coincides with $\Phi_K \circ (A_1 + b)$ on this set. Any other affine piece of Φ_{K+1} can be written in the form of $f_K^i \circ h^j$, where f_K^i and h^j are affine pieces of Φ_K and h, respectively. For this composition, either of the following holds:

(i) It holds that $f_K^i \neq f_K^1$, which results in $\|J(f_K^i \circ h^j))\|_p < 1$ due to $\|Jf_K^i\|_p < 1$.
(ii) It holds that $f_K^i = f_K^1$. Further, note that $Jh^j = \operatorname{diag}(d)W_1$ for some $d \in \mathbb{R}^{n_1}$ with entries $|d_l| \leq 1$. Due to the definition of Ω_{K+1}, there exists l^* such that $v_{l^*} \neq 0$ and $|d_{l^*}| < 1$. Hence, the Jacobian of the affine piece is given by $\tilde{v}^T W_1$ with $\tilde{v} = \operatorname{diag}(d)v$. Since $p \neq 1$, we get that $q < +\infty$ and $\|\tilde{v}\|_q < \|v\|_q \leq 1$. Consequently, $\|J(f_K^i \circ h^j)\|_p = \|\tilde{v}^T W_1\|_p \leq \|\tilde{v}\|_q \|W_1\|_p < 1$.

This concludes the induction argument. ∎

For $p > 1$, Proposition 3.1 implies that ReLU NNs with p-norm constraints on the weights can reproduce neither the absolute value nor a whole family of simple functions, including the triangular hat function (also known as the B-spline of degree 1) and the soft-thresholding function. Further, this result suggests that activation functions with more than one region with maximal slope are better suited within the scope of this approximation framework. Typically, learnable spline activation functions are capable of having this property.

3.2. Limited expressivity. A meaningful metric for the expressivity of a model is its ability to produce functions with high variations. In this section, we investigate the impact of the Lipschitz constraint on the maximal second-order total variation of such an NN. Note that we partially rely on results from [20] for our proofs. The second-order total variation of a function $f \colon \mathbb{R} \to \mathbb{R}$ is defined as $\mathrm{TV}^{(2)}(f) := \|\mathrm{D}^2 f\|_{\mathcal{M}}$, where $\|\cdot\|_{\mathcal{M}}$ is the total-variation norm related to the space $\mathcal{M}$ of bounded Radon measures, and D is the distributional derivative operator. The space of functions with bounded second-order total variation is denoted by

$$\mathrm{BV}^{(2)}(\mathbb{R}) = \{f \colon \mathbb{R} \to \mathbb{R} \text{ s.t. } \mathrm{TV}^{(2)}(f) < +\infty\}. \tag{3.3}$$

For more details, we refer the reader to [7, 36]. Further, we recall that $\mathrm{TV}^{(2)}$ is a seminorm that, for a CPWL function on the real line, is given by the finite sum of its absolute slope changes. Based on Lemma 3.2, we infer for the p-norm-constrained setting that, in general, a linear-spline activation function cannot be replaced with a one-layer ReLU NN without losing expressivity.

Lemma 3.2. *Let $f \colon \mathbb{R} \to \mathbb{R}$ be parameterized by a one-hidden-layer NN with componentwise activation function σ and p-norm-constrained weights, $p \in [1, +\infty]$. If $\sigma \in \mathrm{BV}^{(2)}(\mathbb{R})$, then*

$$\mathrm{TV}^{(2)}(f) \leq \mathrm{TV}^{(2)}(\sigma). \tag{3.4}$$

Proof. Let f be given by $x \mapsto u^T \sigma(wx + b) = \sum_{n=1}^N u_n \sigma(w_n x + b_n)$ with $u := (u_1, \ldots, u_N) \in \mathbb{R}^N$, $w := (w_1, \ldots, w_N) \in \mathbb{R}^N$, and $b := (b_1, \ldots, b_N) \in \mathbb{R}^N$. The p-norm weight constraints imply that $\|w\|_p \leq 1$ and $\|u\|_q \leq 1$ with $1/p + 1/q = 1$. Since $\mathrm{TV}^{(2)}$ is a seminorm, we get

$$\mathrm{TV}^{(2)}(f) \leq \sum_{n=1}^{N} |u_n| \mathrm{TV}^{(2)}(\sigma(w_n \cdot + b_n)) \leq \sum_{n=1}^{N} |u_n w_n| \mathrm{TV}^{(2)}(\sigma) \leq \mathrm{TV}^{(2)}(\sigma), \tag{3.5}$$

where the last step follows by Hölder's inequality. ∎

In principle, the composition operation suffices to increase the second-order total variation of a mapping exponentially. For instance, the n-fold composition f_n of $f\colon \mathbb{R} \to \mathbb{R}$ with $x \mapsto 2|x - 1/2|$ yields the sawtooth function with 2^n linear regions and

$$\mathrm{TV}^{(2)}(f_n) = 2(2^n - 1). \tag{3.6}$$

This highly desirable property is, however, not achievable by ReLU NNs with ∞-norm-constrained weights [20, Thm. 1]. As shown in Proposition 3.3, this has a drastic impact on the size of the class of functions that can be approximated by ReLU NNs.

Proposition 3.3. *Let $D \subset \mathbb{R}^d$ be compact with nonempty interior. Then, there exists $f \in \mathrm{Lip}_{1,\infty}(D)$ that cannot be approximated by ReLU NNs $\Phi\colon \mathbb{R}^d \to \mathbb{R}$ with architecture (2.1), and ∞-norm-constrained weights.*

Proof. By [20, Thm. 1], we know that for any $u \in \mathbb{R}^d$ with $\|u\|_\infty = 1$ and any ReLU NN Φ with ∞-norm weight constraint, it holds that

$$\mathrm{TV}^{(2)}(\Phi \circ \varphi_u) \leq 2, \tag{3.7}$$

where $\varphi_u\colon \mathbb{R} \to \mathbb{R}^d$ with $t \mapsto tu$. Let $(\Phi_n)_{n\in\mathbb{N}}$ be a sequence of ReLU NNs with ∞-norm-constrained weights that converges uniformly to Φ on D. Since D has nonempty interior, we can pick $u \in \mathbb{R}^d$ with $\|u\|_\infty = 1$ such that $\varphi_u^{-1}(D)$ contains an open interval $I \subset \mathbb{R}$. Then, $(\Phi_n \circ \varphi_u)_{n\in\mathbb{N}}$ converges uniformly to $\Phi \circ \varphi_u$ on I. Since $\mathrm{TV}^{(2)}$ is lower semicontinuous with respect to uniform convergence [7, Prop. 3.14], we infer that the restriction to I satisfies

$$\mathrm{TV}^{(2)}(\Phi \circ \varphi_u) \leq 2. \tag{3.8}$$

In other words, any $f \in \mathrm{Lip}_{1,\infty}(D)$ with $\mathrm{TV}^{(2)}(f \circ \varphi_u) > 2$ cannot be approximated by ∞-norm-constrained ReLU NNs. However, there exist sawtooth-like functions on I that have this property, with an explicit example constructed in Proposition 3.4. ∎

Unlike ReLU networks, deep spline networks can produce arbitrarily complex mappings thanks to the composition operation, even in the norm-constrained setting.

Proposition 3.4. *Let $C > 0$, $p \in [1, +\infty]$, $I \subset \mathbb{R}$ open, and $u \in \mathbb{R}^d$. Then, there exists an NN $\Phi\colon \mathbb{R}^d \to \mathbb{R}$ with architecture (2.1), p-norm-constrained weights, and 1-Lipschitz linear-spline activation functions with one knot such that, for $\varphi_u\colon I \to \mathbb{R}^d$ with $\varphi(t) = tu$, it holds that*

$$\mathrm{TV}^{(2)}(\Phi \circ \varphi_u) > C. \tag{3.9}$$

Proof. Pick $b \in \mathbb{R}$, $c > 0$ such that $[b-c, b+c] \subset I$. Let σ_1 with $x \mapsto (|x-b| - c/2)$, σ_k with $x \mapsto (|x| - c/2^k)$, $k = 2, \dots, m$, and $F_m = \sigma_m \circ \cdots \circ \sigma_1$. The function F_m is a sawtooth-like CPWL function with 2^m linear regions all contained in $[b-c, b+c]$. Further, it holds for all $t \in \mathbb{R}$ that $|F_m'(t)| = 1$, and the sign of the slope is different for neighboring regions. From this, we directly infer that

$$\mathrm{TV}^{(2)}(F_m) = 2(2^m - 1). \tag{3.10}$$

Now, we build a deep spline NN $\Phi_K : \mathbb{R}^d \to \mathbb{R}$ with K hidden layers of widths $n_1, \dots, n_K = d$ and $n_{K+1} = 1$. The activation function used in the kth hidden layer is σ_k for the first neuron and zero otherwise, the weight matrices are chosen as the identity matrix except for the last layer, where it is chosen such that

$$\Phi_K(x) = F_K(x_1). \tag{3.11}$$

This construction results for $\varphi_{e_1} : I \to \mathbb{R}^d$ in

$$\mathrm{TV}^{(2)}(\Phi_K \circ \varphi_{e_1}) = 2(2^K - 1), \tag{3.12}$$

and the claim follows for $u = e_1$ by taking K sufficiently large. The general case $u \neq e_1$ follows by using an appropriate weight matrix in the first layer. ∎

4. Approximation of 1-Lipschitz functions.

In this section, we investigate the approximation of 1-Lipschitz functions using the NN architecture (2.1) together with different activation functions and weight constraints. Compared to the setting in section 2.1, the situation is much more involved.

4.1. Networks with componentwise activation functions.

Here, we investigate NNs with architecture as in (2.1), p-norm-constrained weights, and with 1-Lipschitz componentwise activation functions. As first step toward a better understanding, we restrict our attention to functions on the real line. In particular, we show that any CPWL activation function $\sigma : \mathbb{R} \to \mathbb{R}$ can be written as a composition of simple linear splines.

Proposition 4.1. *Let $g : \mathbb{R} \to \mathbb{R}$ be a 1-Lipschitz CPWL function. Then, there exist $n \in \mathbb{N}$ and 1-Lipschitz CPWL functions $g_i : \mathbb{R} \to \mathbb{R}$, $i = 1, \dots, n$, with at most three linear regions such that $g = g_n \circ \cdots \circ g_1$.*

Proof. We proceed via induction over the number m of linear regions of g. For g with up to three linear regions the claim is clearly true. Now, assume that it holds for some $m \in \mathbb{N}$, and let g be linear on the $m+1 > 3$ intervals $[a_i, a_{i+1}]$, $i = 0, \dots, m$, with $a_0 = -\infty$ and $a_{m+1} = +\infty$, and let $s_\pm = \lim_{x \to \pm\infty} g'(x)$. First, we can write $g = g_1 \circ g_2$ with

$$g_1(x) = \begin{cases} g(a_1) + \mathrm{sign}(s_-)(x - a_1) & \text{for } x < a_1, \\ g(x) & \text{for } a_1 \le x \le a_m, \\ g(a_m) + \mathrm{sign}(s_+)(x - a_m) & \text{otherwise} \end{cases} \tag{4.1}$$

and

$$g_2(x) = \begin{cases} a_1 + |s_-|(x - a_1) & \text{for } x < a_1, \\ x & \text{for } a_1 \le x \le a_m, \\ a_m + |s_+|(x - a_m) & \text{otherwise.} \end{cases} \tag{4.2}$$

Since g_2 has three linear regions and g_1 has the same number of linear regions as g, we can limit our discussion to functions g with $\lim_{x\to\pm\infty}|g'(x)|=1$.

Case 1**.** There exists some a_j, $j\in\{2,\dots,m-1\}$, such that the function g has an extremum in a_j when restricted to $(-\infty,a_j]$ or $[a_j,+\infty)$. As all possible cases are similar, we only provide the construction for $g(a_j)$ being a maximum of g on $(-\infty,a_j)$. To this end, we define the functions $\tilde{g}_1$, $\tilde{g}_2$ as

$$\tilde{g}_1(x)=\begin{cases} g(x) & \text{for } x\le a_j,\\ g(a_j)+(x-a_j) & \text{otherwise}\end{cases} \tag{4.3}$$

and

$$\tilde{g}_2(x)=\begin{cases} x & \text{for } x\le g(a_j),\\ g(x+a_j-g(a_j)) & \text{otherwise,}\end{cases} \tag{4.4}$$

which are both 1-Lipschitz piecewise-linear functions with at most m linear regions and satisfying $\lim_{x\to\pm\infty}|\tilde{g}_i'(x)|=1$. Further, it holds that $g=\tilde{g}_2\circ\tilde{g}_1$, so that we can apply the induction assumption to conclude the argument.

Case 2**.** Case 1 does not apply and $\lim_{x\to+\infty}g'(x)/g'(-x)=1$. In the following, we reduce this to Case 1. We only provide the construction for $\lim_{x\to-\infty}g'(x)=1$, the other case being similar. Here, it holds that $g(a_1)\ge g(a_i)\ge g(a_m)$ for all $i=1,\dots,m$ and we now define the functions $\tilde{g}_1$, $\tilde{g}_2$ as

$$\tilde{g}_1(x)=\begin{cases} g(x) & \text{for } x<a_1,\\ 2g(a_1)-g(x) & \text{for } a_1\le x\le a_m,\\ g(x)+2(g(a_1)-g(a_m)) & \text{otherwise}\end{cases} \tag{4.5}$$

and

$$\tilde{g}_2(x)=\begin{cases} x & \text{for } x<g(a_1),\\ 2g(a_1)-x & \text{for } g(a_1)\le x\le 2g(a_1)-g(a_m),\\ 2(g(a_m)-g(a_1))+x & \text{otherwise.}\end{cases} \tag{4.6}$$

Clearly, both of the functions satisfy $\lim_{x\to\pm\infty}|\tilde{g}_i'(x)|=1$ and are 1-Lipschitz. Here, the first function has $m+1$ linear regions and the second one has three. Further, the first function now fits Case 1 and it remains to show that $g=\tilde{g}_2\circ\tilde{g}_1$. However, this follows immediately from $g(a_1)\ge\tilde{g}_1(x)\ge(g(a_1)-g(a_m))$ for $x\in[a_1,a_m]$.

Case 3**.** Case 1 does not apply and $\lim_{x\to+\infty}g'(x)/g'(-x)=-1$. This case can be reduced to either Case 1 or Case 2. We assume that $\lim_{x\to-\infty}g'(x)=1$ and note that the other case is again similar. Then, it holds that $\min\{g(a_1),g(a_m)\}\ge g(a_i)$ for all $i=1,\dots,m$ and we choose $a^*\in\arg\max_{x\in\mathbb{R}}g(x)\in\{a_1,a_m\}$. Next, we define the functions $\tilde{g}_1$, $\tilde{g}_2$ as

$$\tilde{g}_1(x)=\begin{cases} g(x) & \text{for } x<a^*,\\ 2g(a^*)-g(x) & \text{otherwise}\end{cases} \tag{4.7}$$

and

$$\tilde{g}_2(x) = \begin{cases} x & \text{for } x < g(a^*), \\ 2g(a^*) - x & \text{otherwise.} \end{cases} \tag{4.8}$$

Note that both functions satisfy $\lim_{x\to\pm\infty} |\tilde{g}_i'(x)| = 1$ and are 1-Lipschitz. Here, the first function has $m+1$ linear regions and the second one has 2. Further, the first function now fits either Case 1 or Case 2 and, hence, it remains to show that $g = \tilde{g}_2 \circ \tilde{g}_1$. However, this follows immediately from the definition of a^*. ∎

Remark 4.2. The proof actually also shows that if g satisfies $|g'(x)| = 1$ a.e., then the same also holds true for the g_i. Further, the result can be interpreted as an approximation with an NN that has only one neuron per hidden layer. Note that a similar approximation result for ResNets without Lipschitz constraints was given in [23].

Proposition 4.1 is a strong motivation for the use of deep spline NNs. In particular, it implies that deep spline NNs with very simple activation functions already suffice to achieve the maximum representational power in (2.1).

Theorem 4.3. *Let $D \subset \mathbb{R}^d$ be compact. Then, NNs $\Psi\colon D \to \mathbb{R}^n$ with architecture (2.1), p-norm-constrained weights, and 1-Lipschitz spline activation functions with three linear regions can approximate the same functions as the corresponding NNs $\Phi\colon D \to \mathbb{R}^n$ with arbitrary 1-Lipschitz componentwise activation functions.*

Proof. We proceed by induction over the number K of layers of Φ. For $K = 1$, the NN Φ produces an affine mapping and there is nothing to show. Assume that the statement holds for K layers. Let $\Phi_{K+1}\colon \mathbb{R}^d \to \mathbb{R}^{n_{K+1}}$ be an NN of the form (2.1) with p-norm-constrained weights and $K+1$ layers. Then, $\Phi_{K+1} = A_{K+1} \circ \sigma_{\alpha_K} \circ \Phi_K$ with a K-layer NN $\Phi_K\colon \mathbb{R}^d \to \mathbb{R}^{n_K}$ of the same form. By application of the induction assumption, for any $\epsilon \in \mathbb{R}_{>0}$ there exists a deep spline NN $\Psi_1\colon \mathbb{R}^d \to \mathbb{R}^{n_K}$ with p-norm-constrained weights such that $\max_{x\in D} \|\Phi_K(x) - \Psi_1(x)\|_p \leq \epsilon/2$. Due to the finite diameter of D, the range of 1-Lipschitz functions is compact. Hence, Proposition 4.1 implies that there exists a deep spline NN $\Psi_2\colon \mathbb{R}^{n_k} \to \mathbb{R}^{n_k}$ with all affine transformations being identities such that $\max_{x\in\Phi_K(D)} \|\sigma_{\alpha_K}(x) - \Psi_2(x)\|_p \leq \epsilon/2$. For the deep spline NN $A_{K+1} \circ \Psi_2 \circ \Psi_1$ with spectral-norm-constrained weights, we can bound the error as

$$\begin{aligned} &\max_{x\in D} \|\Phi(x) - A_{K+1} \circ \Psi_2 \circ \Psi_1(x)\|_p \leq \max_{x\in D} \|\sigma_{\alpha_K} \circ \Phi_K(x) - \Psi_2 \circ \Psi_1(x)\|_p \\ &\leq \max_{x\in D} (\|\sigma_{\alpha_K} \circ \Phi_K(x) - \Psi_2 \circ \Phi_K(x)\|_p + \|\Psi_2 \circ \Phi_K(x) - \Psi_2 \circ \Psi_1(x)\|_p) \\ &\leq \epsilon/2 + \max_{x\in D} \|\Phi_K(x) - \Psi_1(x)\|_p \leq \epsilon. \end{aligned} \tag{4.9}$$

This concludes the proof. ∎

Theorem 4.3 tells us that, among all NNs of the form (2.1) with componentwise 1-Lipschitz activation functions, splines with three linear regions achieve the optimal representational power. This is corroborated by numerical experiments on function fitting, Wasserstein distance estimation, and Plug-and-Play image reconstruction for which it was found that 1-Lipschitz deep spline NNs match or outperform other 1-Lipschitz NNs including the Groupsort architecture [12]. Meanwhile, the question of whether deep spline networks with p-norm-constrained

weights are universal approximators for $\mathrm{Lip}_{1,p}(D)$ is part of ongoing research, and it appears to be a very challenging problem.

4.2. Groupsort versus linear-spline activation functions. In this section, we discuss how Groupsort NNs and deep spline NNs can be expressed in terms of each other. Here, the situation differs depending on the applied weight constraint. First, we revisit a framework specifically tailored to Groupsort NNs, where the weights in architecture (2.1) satisfy $\|W_k\|_\infty \leq 1$, $k = 2, \ldots, K$, and $\|W_1\|_{p,\infty} \leq 1$. Then, the expression of an arbitrary deep spline NN using a Groupsort NN is made possible due to the following universality result proved in [1, Thm. 3].

Proposition 4.4. *Let $D \subset \mathbb{R}^d$ be compact, and let $p \in [1, +\infty]$. The Groupsort NNs with architecture (2.1), group size at least 2, and weight constraints $\|W_k\|_\infty \leq 1$, $k = 2, \ldots, K$, and $\|W_1\|_{p,\infty} \leq 1$ are dense in $\mathrm{Lip}_{1,p}(D)$.*

Proposition 4.4, according to which density holds for all $p \in [1, +\infty]$, can be misleading as p only has little to do with the involved norm constraints. All weights but the first one have to fulfill an ∞-norm constraint, which is rarely used in practice. This somehow limits the practical relevance of the result. Nevertheless, it would be interesting if a similar result would also hold for deep spline NNs. Let us remark that the proof of Proposition 4.4 relies heavily on the maximum operation and the chosen norms, which makes it difficult to generalize it to other norm constraints or activation functions.

Now, we discuss the case of spectral-norm constraints, which are the usual choice in practice. For this setting, let us recall that it holds that

$$\max(x_1, x_2) = \frac{x_1 + x_2 + |x_1 - x_2|}{2}. \tag{4.10}$$

Hence, in the case of the spectral-constrained weights, the MaxMin activation function can be written as the deep spline NN $\mathrm{MaxMin}(x) = W_2 \sigma_1 (W_1 x)$, where

$$W_1 = W_2 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \quad \text{and} \quad \sigma_1(x) = \begin{pmatrix} x_1 \\ |x_2| \end{pmatrix}. \tag{4.11}$$

This can be extended to any Groupsort operation since the MaxMin operation has the same expressivity as Groupsort under any p-norm constraint [1]. We are not aware of any results for the reverse direction, i.e., to express a deep spline NN using a Groupsort NN with spectral-norm-constrained weights.

5. Conclusions and open problems. In this paper, we have shown that neural networks (NNs) with linear-spline activation functions with at least three linear regions can approximate the maximal class of functions among all NNs with p-norm weight constraints and componentwise activation functions. However, it remains an open question whether these NNs are universal approximators of $\mathrm{Lip}_{1,p}(D)$, $D \subset \mathbb{R}^d$, compact. While this problem appears to be very challenging, our result could be a first step toward its solution. The comparison of linear spline to non-componentwise activation functions involves subtle considerations. It is so far unclear which choice leads to more expressive NNs. For the spectral norm, deep spline NNs are at least as expressive as Groupsort NNs, but for ∞-norm-constrained weights the opposite

is true. The further investigation of the problem of universality under different constraints appears to be a promising research topic that may lead to better trainable Lipschitz-constrained NN architectures.

Regarding the question of universality, we mainly focused on the approximation of scalar-valued functions $f: \mathbb{R}^d \to \mathbb{R}$. This also reflects the current state of research, where most results are only formulated for scalar-valued NNs. The extension of these results to vector-valued functions appears highly nontrivial and is a topic for future research. Finally, we want to remark that little is known about the optimal structure for deep spline and Groupsort NNs, namely, if it is more preferable to design either deep or wide architectures.

REFERENCES

[1] C. Anil, J. Lucas, and R. Grosse, *Sorting out Lipschitz function approximation*, in Proceedings of the 36th International Conference on Machine Learning, Proceedings of Machine Learning Research 97, PMLR, 2019, pp. 291–301, https://openreview.net/pdf?id=ryxY73AcK7.

[2] M. Arjovsky, S. Chintala, and L. Bottou, *Wasserstein generative adversarial networks*, in Proceedings of the 34th International Conference on Machine Learning, Proceedings of Machine Learning Research 70, PMLR, 2017, pp. 214–223, https://proceedings.mlr.press/v70/arjovsky17a.html.

[3] S. Aziznejad, H. Gupta, J. Campos, and M. Unser, *Deep neural networks with trainable activations and controlled Lipschitz constant*, IEEE Trans. Signal Process., 68 (2020), pp. 4688–4699, https://doi.org/10.1109/TSP.2020.3014611.

[4] S. Aziznejad and M. Unser, *Deep spline networks with control of Lipschitz regularity*, in Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, IEEE, 2019, pp. 3242–3246, https://doi.org/10.1109/ICASSP.2019.8682547.

[5] P. Bohra, J. Campos, H. Gupta, S. Aziznejad, and M. Unser, *Learning activation functions in deep (spline) neural networks*, IEEE Open J. Signal Process., 1 (2020), pp. 295–309, https://doi.org/10.1109/OJSP.2020.3039379.

[6] P. Bohra, D. Perdios, A. Goujon, S. Emery, and M. Unser *Learning Lipschitz-controlled activation functions in neural networks for Plug-and-Play image reconstruction methods*, in NeurIPS, 2021 Workshop on Deep Learning and Inverse Problems, 2021, https://openreview.net/forum?id=efCsbTzQTbH.

[7] K. Bredies and M. Holler, *Higher-order total variation approaches and generalisations*, Inverse Problems, 36 (2020), 123001, https://doi.org/10.1088/1361-6420/ab8f80.

[8] L. Bungert, R. Raab, T. Roith, L. Schwinn, and D. Tenbrinck, *CLIP: Cheap Lipschitz training of neural networks*, in Scale Space and Variational Methods in Computer Vision, Springer, Cham, 2021, pp. 307–319, https://doi.org/10.1007/978-3-030-75549-2_25.

[9] O. Calin, *Deep Learning Architectures: A Mathematical Approach*, Springer, Cham, 2020, https://doi.org/10.1007/978-3-030-36721-3

[10] A. Chernodub and D. Nowicki, *Norm-Preserving Orthogonal Permutation Linear Unit Activation Functions (OPLU)*, preprint, https://arxiv.org/abs/1604.02313, 2016.

[11] J. E. Cohen, T. P. Huster, and R. Cohen, *Universal Lipschitz Approximation in Bounded Depth Neural Networks*, preprint, https://arxiv.org/abs/1904.04861, 2019.

[12] S. Ducotterd, A. Goujon, P. Bohra, D. Perdios, S. Neumayer, and M. Unser, *Improving Lipschitz-Constrained Neural Networks by Learning Activation Functions*, preprint, https://arxiv.org/abs/2210.16222, 2022.

[13] M. Fazlyab, A. Robey, H. Hassani, M. Morari, and G. Pappas, *Efficient and accurate estimation of Lipschitz constants for deep neural networks*, in Advances in Neural Information Processing Systems, Vol. 32, Curran Associates, Red Hook, NY, 2019, pp. 11427–11438, https://openreview.net/forum?id=rkxGbHBe8S.

[14] H. Gouk, E. Frank, B. Pfahringer, and M. Cree, *Regularisation of neural networks by enforcing Lipschitz continuity*, Mach. Learn., 110 (2021), pp. 393–416. https://doi.org/10.1007/s10994-020-05929-w.

[15] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville *Improved training of Wasserstein GANs*, in Advances in Neural Information Processing Systems 30, Curran Associates, Red Hook, NY, 2017, pp. 2644–2655, https://proceedings.neurips.cc/paper˙files/paper/2017/file/892c3b1c6dccd52936e27cbd0ff683d6-Paper.pdf.

[16] P. Hagemann and S. Neumayer, *Stabilizing invertible neural networks using mixture models*, Inverse Problems, 37 (2021), 085002, https://doi.org/10.1088/1361-6420/abe928.

[17] M. Hasannasab, J. Hertrich, S. Neumayer, G. Plonka, S. Setzer, and G. Steidl, *Parseval proximal neural networks*, J. Fourier Anal., 26 (2020), 59, https://doi.org/10.1007/s00041-020-09761-7.

[18] J. Hertrich, S. Neumayer, and G. Steidl, *Convolutional proximal neural networks and plug-and-play algorithms*, Linear Algebra Appl., 631 (2021), pp. 203–234, https://doi.org/10.1016/j.laa.2021.09.004.

[19] L. Huang, X. Liu, B. Lang, A. W. Yu, Y. Wang, and B. Li, *Orthogonal weight normalization: Solution to optimization over multiple dependent Stiefel manifolds in deep neural networks*, in Proceedings of the 32nd AAAI Conference on Artificial Intelligence, AAAI Press, 2018, pp. 3271–3278, https://doi.org/10.1609/aaai.v32i1.11768.

[20] T. Huster, C.-Y. J. Chiang, and R. Chadha, *Limitations of the Lipschitz constant as a defense against adversarial examples*, in Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, Cham, 2018, pp. 16–29,https://doi.org/10.1007/978-3-030-13453-2_2.

[21] F. Latorre, P. Rolland, and V. Cevher, *Lipschitz constant estimation of neural networks via sparse polynomial optimization*, in International Conference on Learning Representations, 2020, pp. 1–14, https://openreview.net/forum?id=rJe4_xSFDB.

[22] Q. Li, S. Haque, C. Anil, J. Lucas, R. Grosse, and J.-H. Jacobsen, *Preventing gradient attenuation in Lipschitz constrained convolutional networks*, in Advances in Neural Information Processing Systems 32, Curran Associates, Red Hook, NY, 2019, pp. 15364–15376, https://openreview.net/forum?id=Syx36SBeUS.

[23] H. Lin and S. Jegelka, *ResNet with one-neuron hidden layers is a universal approximator*, in Advances in Neural Information Processing Systems 31, Curran Associates, Red Hook, NY, 2018, pp. 6172–6181, https://proceedings.neurips.cc/paper/2018/file/03bfc1d4783966c69cc6aef8247e0103-Paper.pdf.

[24] T. Meinhardt, M. Moeller, C. Hazirbas, and D. Cremers, *Learning proximal operators: Using denoising networks for regularizing inverse imaging problems*, in Proceedings of the IEEE International Conference on Computer Vision, IEEE, 2017, pp. 1799–1808, https://doi.ieeecomputersociety.org/10.1109/ICCV.2017.198.

[25] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, *Spectral normalization for generative adversarial networks*, in Sixth International Conference on Learning Representations, 2018, pp. 1–26, https://openreview.net/forum?id=B1QRgziT-.

[26] P. Pauli, A. Koch, J. Berberich, P. Kohler, and F. Allgöwer, *Training robust neural networks using Lipschitz bounds*, IEEE Control Syst. Lett., 6 (2022), pp. 121–126, https://doi.org/10.1109/LCSYS.2021.3050444.

[27] K. Roth, Y. Kilcher, and T. Hofmann, *Adversarial training is a form of data-dependent operator norm regularization*, in Advances in Neural Information Processing Systems, Vol. 33, Curran Associates, 2020, pp. 14973–14985, https://proceedings.neurips.cc/paper/2020/file/ab7314887865c4265e896c6e209d1cd6-Paper.pdf.

[28] E. Ryu, J. Liu, S. Wang, X. Chen, Z. Wang, and W. Yin, *Plug-and-play methods provably converge with properly trained denoisers*, in International Conference on Machine Learning, 2019, PMLR, pp. 5546–5557, https://proceedings.mlr.press/v97/ryu19a.html.

[29] H. Sedghi, V. Gupta, and P. M. Long, *The singular values of convolutional layers*, in International Conference on Learning Representations, 2019, pp. 1–12, https://openreview.net/forum?id=rJevYoA9Fm.

[30] S. Singla, S. Singla, and S. Feizi, *Improved Deterministic l2 Robustness on CIFAR-10 and CIFAR-100*, preprint, https://arxiv.org/abs/2108.04062, 2021.

[31] S. Sreehariand, S. V. Venkatakrishnan, and B. Wohlberg, *Plug-and-play priors for bright field electron tomography and sparse interpolation*, IEEE Trans. Comput. Imaging, 2 (2016), pp. 408–423, https://doi.org/10.1109/TCI.2016.2599778.

[32] U. Tanielian, M. Sangnier, and G. Biau, *Approximating Lipschitz continuous functions with GroupSort neural networks*, in Proceedings of the 24th International Conference on Artificial Intelligence and Statistics, PMLR, 2021, pp. 442–450, http://proceedings.mlr.press/v130/tanielian21a.html.

[33] J. M. Tarela, E. Alonso, and M. V. Martínez, *A representation method for PWL functions oriented to parallel processing*, Math. Comput. Model., 13 (1990), pp. 75–83, https://doi.org/10.1016/0895-7177(90)90090-A.

[34] M. Terris, A. Repetti, J. Pesquet, and Y. Wiaux, *Building firmly nonexpansive convolutional neural networks*, in Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, IEEE, 2020, pp. 8658–8662, https://doi.org/10.1109/ICASSP40776.2020.9054731.

[35] Y. Tsuzuku, I. Sato, and M. Sugiyama, *Lipschitz-margin training: Scalable certification of perturbation invariance for deep neural networks*, in Advances in Neural Information Processing Systems 31, Curran Associates, Red Hook, NY, 2018, pp. 6542–6551, https://proceedings.neurips.cc/paper˙files/paper/2018/file/485843481a7edacbfce101ecb1e4d2a8-Paper.pdf.

[36] M. Unser, *A representer theorem for deep neural networks*, J. Mach. Learn. Res., 20 (2019), 110, http://jmlr.org/papers/v20/18-418.html.

[37] S. V. Venkatakrishnan, C. A. Bouman, and B. Wohlberg, *Plug-and-play priors for model based reconstruction*, in Proceedings of the IEEE Global Conference on Signal and Information Processing, IEEE, 2013, pp. 945–948, https://doi.org/10.1109/GlobalSIP.2013.6737048.

[38] A. Virmaux and K. Scaman, *Lipschitz regularity of deep neural networks: Analysis and efficient estimation*, in Advances in Neural Information Processing Systems 31, Curran Associates, Red Hook, NY, 2018, pp. 3839–3848, https://proceedings.neurips.cc/paper_files/paper/2018/file/d54e99a6c03704e95e6965532dec148b-Paper.pdf.