Deep Neural Networks With Trainable Activations and Controlled Lipschitz Constant

Shayan Aziznejad, Harshit Gupta, Joaquim Campos, and Michael Unser, *Fellow, IEEE*

***Abstract*—We introduce a variational framework to learn the activation functions of deep neural networks. Our aim is to increase the capacity of the network while controlling an upper-bound of the actual Lipschitz constant of the input-output relation. To that end, we first establish a global bound for the Lipschitz constant of neural networks. Based on the obtained bound, we then formulate a variational problem for learning activation functions. Our variational problem is infinite-dimensional and is not computationally tractable. However, we prove that there always exists a solution that has continuous and piecewise-linear (linear-spline) activations. This reduces the original problem to a finite-dimensional minimization where an ℓ_1 penalty on the parameters of the activations favors the learning of sparse nonlinearities. We numerically compare our scheme with standard ReLU network and its variations, PReLU and LeakyReLU and we empirically demonstrate the practical aspects of our framework.**

***Index Terms*—Deep learning, deep splines, learned activations, lipschitz regularity, representer theorem.**

I. Introduction

In SUPERVISED learning, the goal is to approximate an unknown mapping from a set of noisy samples [2]. Specifically, one aims at determining the function $f : \mathbb{R}^d \to \mathbb{R}^{d'}$, given a dataset of size M that consists of pairs of the form $(\boldsymbol{x}_m, \boldsymbol{y}_m)$ such that $\boldsymbol{y}_m \approx f(\boldsymbol{x}_m)$ for $m = 1, 2, \ldots, M$, without over-fitting.

In the scalar case $d' = 1$, a classical formulation of this problem is through the minimization

$$\min_{f\in\mathcal{H}(\mathbb{R}^d)} \left(\sum_{m=1}^{M} \mathrm{E}\left(y_m, f(\boldsymbol{x}_m)\right) + \lambda \|f\|_{\mathcal{H}}^2 \right), \tag{1}$$

where $\mathcal{H}(\mathbb{R}^d)$ is a reproducing-kernel Hilbert space (RKHS), $\mathrm{E} : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ is an arbitrary convex loss function, and λ is a positive constant that controls the regularity of the model [3], [4]. Although this is an infinite-dimensional problem over a Hilbert space, the kernel representer theorem [5], [6] states that the solution of (1) is unique and admits the parametric form

$$f(\boldsymbol{x}) = \sum_{m=1}^{M} a_m \mathrm{k}(\boldsymbol{x}, \boldsymbol{x}_m), \tag{2}$$

where $\mathrm{k}(\cdot, \cdot)$ is the unique reproducing kernel of $\mathcal{H}(\mathbb{R}^d)$. Expansion (2) is the key element of kernel-based algorithms in machine learning, including the framework of support-vector machines [7]. It also reveals the intimate link between kernel methods, splines, and radial basis functions [8]–[10].

During the past decade, there has been an increasing interest in deep-learning methods as they outperform kernel-based schemes in a variety of tasks such as image classification [11], inverse problems [12], and segmentation [13]. The main idea is to replace the kernel expansion (2) by a parametric deep neural network that is a repeated composition of affine mappings intertwined with pointwise nonlinearities (a.k.a. neuronal activations) [14], [15]. The challenge then is to optimize the parameters of this model by minimizing a (typically non-convex) cost function.

The classical choice for the activation is the sigmoid function due to its biological interpretation and universal approximation property [16]. However, neural networks with sigmoidal activations suffer from vanishing gradients which essentially makes training difficult and slow. This stems from the fact that the Sigmoid function is bounded and horizontally asymptotic at large positive and negative values. Currently, the preferred activations are rectified linear unit $\mathrm{ReLU}(x) = \max(x, 0)$ [17] and its variants such as LeakyReLU, defined as $\mathrm{LReLU}(x) = \max(x, ax)$ for some $a \in (0, 1)$ [18]. ReLU-based activations have a wide range, which prevents the network from having vanishing gradients.

Neural networks with ReLU activations have been considered thoroughly in the literature [14]. Their input-output relation is a continuous piecewise-linear (CPWL) mapping [19]. Interestingly, the converse of this result also holds: Any CPWL function can be represented by a deep ReLU neural network [20]. One can also interpret a ReLU activation as a linear spline with one knot. This observation allows one to interpret deep ReLU networks as hierarchical splines [21]. In addition, Unser showed that linear-spline activations are optimal in the sense that they have a minimal second-order total variation and, hence, are maximally regularized [22]; this also provides a variational justification for ReLU-based activations.

Manuscript received January 16, 2020; revised May 27, 2020 and August 3, 2020; accepted August 3, 2020. Date of publication August 10, 2020; date of current version August 24, 2020. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Ali Tajer. This work was supported in part by the Swiss National Science Foundation, Grant 200020_184646/1 and in part by the European Research Council (H2020-ERC Project GlobalBioIm) under Grant 692726. An early version of this work has been presented at the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brighton, U.K., May 2019 [1]. *(Corresponding author: Shayan Aziznejad).*

The authors are with Biomedical Imaging Group, École polytechnique fédérale de Lausanne, 1015 Lausanne, Switzerland (e-mail: shayan.aziznejad@epfl.ch; harshit.gupta@epfl.ch; joaquim.campos@epfl.ch; michael.unser@epfl.ch).

Digital Object Identifier 10.1109/TSP.2020.3014611

1053-587X © 2020 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Although the ReLU networks are favorable, both from a theoretical and practical point of view, one may want to go even farther and learn the activation functions as well. The minimal attempt is to learn the parameter a in LeakyReLU activations, which is known as the parametric ReLU (PReLU) [23]. More generally, one can consider a parametric form for the activations and learn the parameters in the training step. There is a rich literature on the learning of activations represented by splines, a parametric form characterized by optimality and universality [24], [25]. Examples are perceptive B-splines [26], Catmull-Rom cubic splines [27], [28], and adaptive piecewise linear splines [29], to name a few.

In theoretical analyses of deep neural networks, the Lipschitz-continuity of the network and the control of its regularity is of great importance and is crucial in several schemes of deep learning, for example in Wasserstein GANs [30], in providing compressed sensing type guarantees for generative models [31], in showing the convergence of CNN-based projection algorithms to solve inverse problems [32], and in understanding the generalization property of deep neural networks [33]. Moreover, the Lipschitz regularity drives the stability of neural networks, a matter that has been tackled recently [34]–[36].

In this article, we propose a variational framework to learn the activation functions with the motivation of increasing the capacity of the network while controlling its Lipschitz regularity. To that end, we first provide a global bound for the Lipschitz constant of the input-output relation of neural networks that have second-order bounded-variation activations. Based on the minimization of this bound, we propose an optimization scheme in which we learn the linear weights and the activation functions jointly. We show that there always exists a global solution of our proposed minimization made of linear spline activations. We also demonstrate that our proposed regularization has a sparsity-promoting effect on the parameters of the spline activations. Let us remark that our regularization, which is based on an upper-bound, does not ensure that the actual Lipschitz constant of the neural network is minimized—it only prevents it from exceeding a certain range.

Our framework is inspired from [22] and brings in the following new elements:

- After a slight modification of the regularization term that was proposed in [22], we identify a global bound for the Lipschitz constant of the network (see Theorem 1).
- We prove the *existence* of a linear-spline solution in our framework (see Theorem 3 and the discussion after).
- By providing numerical examples, we show how to take advantage of our main results to improve the expressivity of neural networks. This is of practitioner's relevance, as our activation learning module can be used to replace classical activation functions like ReLU and its variants.

The article is organized as follows: In Section II, we provide mathematical preliminaries. In Section III, we discuss the properties of neural networks that have second-order bounded-variation activations and provide a global bound for their Lipschitz constant. We then introduce our variational formulation and study its solutions in Section IV. In Section V, we illustrate our framework with numerical examples.

II. PRELIMINARIES

The Schwartz space of smooth and rapidly decaying functions is denoted by $\mathcal{S}(\mathbb{R})$. Its continuous dual $\mathcal{S}'(\mathbb{R})$ is the space of tempered distributions [37]. The space of continuous functions that vanish at infinity is denoted by $\mathcal{C}_0(\mathbb{R})$. It is a Banach space equipped with the supremum norm $\|\cdot\|_\infty$ and is indeed the closure of $\mathcal{S}(\mathbb{R})$ with this norm. Its continuous dual is the space of Radon measures $\mathcal{M}(\mathbb{R})$ that is also a Banach space with the total-variation norm defined as [38]

$$\|w\|_{\mathcal{M}} \triangleq \sup_{\substack{\varphi\in\mathcal{S}(\mathbb{R})\\ \|\varphi\|_\infty=1}} \langle w,\varphi\rangle. \tag{3}$$

The Banach space $(\mathcal{M}(\mathbb{R}),\|\cdot\|_{\mathcal{M}})$ is a generalization of $(L_1(\mathbb{R}),\|\cdot\|_{L_1})$, in the sense that $L_1(\mathbb{R})\subseteq\mathcal{M}(\mathbb{R})$ and, for any $f\in L_1(\mathbb{R})$, the relation $\|f\|_{L_1}=\|f\|_{\mathcal{M}}$ holds. However, it is larger than $L_1(\mathbb{R})$. For instance, it contains the shifted Dirac impulses $\delta(\cdot-\boldsymbol{x}_0)$ with $\|\delta(\cdot-\boldsymbol{x}_0)\|_{\mathcal{M}}=1$, for all $\boldsymbol{x}_0\in\mathbb{R}$ that are not included in $L_1(\mathbb{R})$.

The space of functions with second-order bounded variations is denoted by $\mathrm{BV}^{(2)}(\mathbb{R})$ and is defined as

$$\mathrm{BV}^{(2)}(\mathbb{R})=\{f\in\mathcal{S}'(\mathbb{R}):\quad \|\mathrm{D}^2f\|_{\mathcal{M}}<+\infty\}, \tag{4}$$

where $\mathrm{D}:\mathcal{S}'(\mathbb{R})\to\mathcal{S}'(\mathbb{R})$ is the generalized derivative operator [39]. Let us mention that the second-order total variation $\mathrm{TV}^{(2)}(f)\triangleq\|\mathrm{D}^2f\|_{\mathcal{M}}$ is only a semi-norm in this space, since the null space of the linear operator D^2 is nontrivial and consists of degree-one polynomials (affine mappings in $\mathbb{R}$). However, it can become a *bona fide* Banach space with the $\mathrm{BV}^{(2)}$ norm

$$\|f\|_{\mathrm{BV}^{(2)}}\triangleq\mathrm{TV}^{(2)}(f)+|f(0)|+|f(1)|. \tag{5}$$

This space has been extensively studied in [22] and in a more general setting in [40]. We summarize some of its important properties in Appendix A.

Given generic Banach spaces $(\mathcal{X},\|\cdot\|_{\mathcal{X}})$ and $(\mathcal{Y},\|\cdot\|_{\mathcal{Y}})$, a function $f:\mathcal{X}\to\mathcal{Y}$ is said to be Lipschitz-continuous if there exists a finite constant $C>0$ such that

$$\|f(x_1)-f(x_2)\|_{\mathcal{Y}}\le C\|x_1-x_2\|_{\mathcal{X}},\quad \forall x_1,x_2\in\mathcal{X}. \tag{6}$$

The minimal value of C is called the Lipschitz constant of f.

In this article, we consider fully connected feed forward neural networks. An L-layer neural network $\mathbf{f}_{\mathrm{deep}}:\mathbb{R}^{N_0}\to\mathbb{R}^{N_L}$ with the layer descriptor $(N_0,N_1,\dots,N_L)$ is the composition of the vector-valued functions $\mathbf{f}_l:\mathbb{R}^{N_{l-1}}\to\mathbb{R}^{N_l}$ for $l=1,\dots,L$ as

$$\mathbf{f}_{\mathrm{deep}}:\mathbb{R}^{N_0}\to\mathbb{R}^{N_L}:\boldsymbol{x}\mapsto\mathbf{f}_L\circ\cdots\circ\mathbf{f}_1(\boldsymbol{x}). \tag{7}$$

Each vector-valued function $\mathbf{f}_l$ is a layer of the neural network $\mathbf{f}_{\mathrm{deep}}$ and consists of two elementary operations: linear transformations and point-wise nonlinearities. In other words, for the lth layer, there exists weight vectors $\mathbf{w}_{n,l}\in\mathbb{R}^{N_{l-1}}$ and nonlinear functions (activations) $\sigma_{n,l}:\mathbb{R}\to\mathbb{R}$ for $n=1,2,\dots,N_l$ such that

$$\mathbf{f}_l(\boldsymbol{x})=\left(\sigma_{1,l}(\mathbf{w}_{1,l}^T\boldsymbol{x}),\sigma_{2,l}(\mathbf{w}_{2,l}^T\boldsymbol{x}),\dots,\sigma_{N_l,l}(\mathbf{w}_{N_l,l}^T\boldsymbol{x})\right). \tag{8}$$

One can also consider an alternative representation of the lth layer by defining the matrix $\mathbf{W}_l=[\,\mathbf{w}_{1,l}\ \mathbf{w}_{2,l}\ \cdots\ \mathbf{w}_{N_l,l}\,]^T$ and

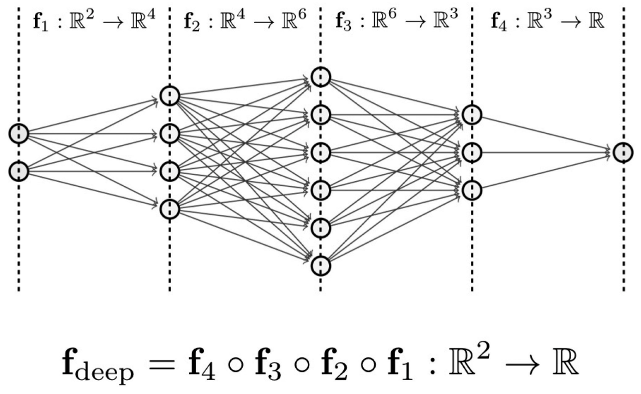

$\mathbf{f}_1 : \mathbb{R}^2 \to \mathbb{R}^4$
$\mathbf{f}_2 : \mathbb{R}^4 \to \mathbb{R}^6$
$\mathbf{f}_3 : \mathbb{R}^6 \to \mathbb{R}^3$
$\mathbf{f}_4 : \mathbb{R}^3 \to \mathbb{R}$
$\mathbf{f}_{\text{deep}} = \mathbf{f}_4 \circ \mathbf{f}_3 \circ \mathbf{f}_2 \circ \mathbf{f}_1 : \mathbb{R}^2 \to \mathbb{R}$

Fig. 1. Schematic view of a neural network with the layer descriptor (2,4,6,3,1). Each layer consists of linear weights (arrows) and point-wise nonlinearities (circles).

the vector-valued nonlinear function $\boldsymbol{\sigma}_l : \mathbb{R}^{N_l} \to \mathbb{R}^{N_l}$ as the mapping

$$(x_1, \ldots, x_{N_l}) \mapsto (\sigma_{1,l}(x_1), \sigma_{2,l}(x_2), \ldots, \sigma_{N_l,l}(x_{N_l})). \tag{9}$$

With this notation, the lth layer has simply the form $\mathbf{f}_l = \boldsymbol{\sigma}_l \circ \mathbf{W}_l$.

Lastly, for any $p \in [1, +\infty)$, we define the $(\mathrm{BV}^{(2)}, p)$-norm of the nonlinear layer $\boldsymbol{\sigma}_l$ as

$$\|\boldsymbol{\sigma}_l\|_{\mathrm{BV}^{(2)},p} = \left(\sum_{n=1}^{N_l} \|\sigma_{n,l}\|_{\mathrm{BV}^{(2)}}^p \right)^{\frac{1}{p}}. \tag{10}$$

III. SECOND-ORDER BOUNDED-VARIATION ACTIVATIONS

We consider activations from the space of second-order bounded-variation functions $\mathrm{BV}^{(2)}(\mathbb{R})$. This ensures that the corresponding neural network satisfies several desirable properties which we discuss in this section. The key feature of these activations is their Lipschitz continuity, as stated in Proposition 1. The proof is provided in Appendix B.

Proposition 1: Any function with second-order bounded variation is Lipschitz-continuous. Specifically, for any function $\sigma \in \mathrm{BV}^{(2)}(\mathbb{R})$ and any $x_1, x_2 \in \mathbb{R}$, we have that

$$|\sigma(x_1) - \sigma(x_2)| \leq \|\sigma\|_{\mathrm{BV}^{(2)}} |x_1 - x_2|. \tag{11}$$

Lipschitz functions are known to be continuous and differentiable almost everywhere [41]. Moreover, in Proposition 2, we show that any element of $\mathrm{BV}^{(2)}(\mathbb{R})$ has well-defined right and left derivatives at any point. This is an important property for activation functions, since it is the minimum requirement for performing gradient-based algorithms that take advantage of the celebrated back-propagation scheme in the training step [42].

Proposition 2: For any function $\sigma \in \mathrm{BV}^{(2)}(\mathbb{R})$ and any $x_0 \in \mathbb{R}$, the left and right derivatives of σ at the point $x = x_0$ exist and are finite.

The proof of Proposition 2 is available in Appendix C. Let us mention that Lipschitz functions in general do not have one-sided derivatives at all points; it is a property that is specific of $\mathrm{BV}^{(2)}$ functions. As an example, consider the function $f : \mathbb{R} \to \mathbb{R}$ with

$$f(x) = \begin{cases} x \sin(\log(x)), & x > 0 \\ 0, & x \leq 0. \end{cases} \tag{12}$$

One readily verifies that f is Lipschitz-continuous with the constant $C = \sqrt{2}$. However, for positive values of h, the function $\frac{f(h)}{h} = \sin(-\log(h))$ oscillates between (-1) and 1 as h goes to zero. Hence, f does not have a right derivative at the point $x_0 = 0$.

In Theorem 1, we prove that any neural network with activations from $\mathrm{BV}^{(2)}(\mathbb{R})$ specifies a Lipschitz-continuous input-output relation. Moreover, we provide an upper-bound for its Lipschitz constant. The proof can be found in Appendix D.

Theorem 1: Any feed forward fully connected deep neural network $\mathbf{f}_{\text{deep}} : \mathbb{R}^{N_0} \to \mathbb{R}^{N_L}$ with second-order bounded-variation activations $\sigma_{n,l} \in \mathrm{BV}^{(2)}(\mathbb{R})$ is Lipschitz-continuous. Moreover, if we consider the ℓ_p for $p \in [1, \infty]$ topology in the input and output spaces, the neural network satisfies the global Lipschitz bound

$$\|\mathbf{f}_{\text{deep}}(\boldsymbol{x}_1) - \mathbf{f}_{\text{deep}}(\boldsymbol{x}_2)\|_p \leq C \|\boldsymbol{x}_1 - \boldsymbol{x}_2\|_p \tag{13}$$

for all $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathbb{R}^{N_0}$, where

$$C = \left(\prod_{l=1}^{L} \|\mathbf{W}_l\|_{q,\infty} \right) \cdot \left(\prod_{l=1}^{L} \|\boldsymbol{\sigma}_l\|_{\mathrm{BV}^{(2)},p} \right), \tag{14}$$

$q \in [1, \infty]$ is such that $\frac{1}{p} + \frac{1}{q} = 1$ and $\|\mathbf{W}_l\|_{q,\infty} = \max_n \|\mathbf{w}_{n,l}\|_q$ is the mixed norm $(\ell_q - \ell_\infty)$ of the lth linear layer.

When the standard Euclidean topology is assumed for the input and output spaces, Proposition 3 provides an alternative bound for the Lipschitz constant of the neural network. The proof of Proposition 3 is available in Appendix E.

Proposition 3: Let $\mathbf{f}_{\text{deep}} : \mathbb{R}^{N_0} \to \mathbb{R}^{N_L}$ be a fully connected feed forward neural network with activations selected from $\mathrm{BV}^{(2)}(\mathbb{R})$. For all $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathbb{R}^{N_0}$ we have that

$$\|\mathbf{f}_{\text{deep}}(\boldsymbol{x}_1) - \mathbf{f}_{\text{deep}}(\boldsymbol{x}_2)\|_2 \leq C_E \|\boldsymbol{x}_1 - \boldsymbol{x}_2\|_2, \tag{15}$$

where

$$C_E = \left(\prod_{l=1}^{L} \|\mathbf{W}_l\|_F \right) \cdot \left(\prod_{l=1}^{L} \|\boldsymbol{\sigma}_l\|_{\mathrm{BV}^{(2)},1} \right). \tag{16}$$

Remark 1: In Proposition 3, it is possible to replace the ℓ_1 outer-norm of the nonlinear layers by ℓ_p for any $p \in [1, \infty]$. This is due to the equivalence of norms in finite-dimensional vector-spaces. In general, such replacements come at the cost of multiplying the Lipschitz bound by a constant. In the special case $p = 2$, no constant is required and we achieve an even tighter bound (following (43) in Appendix E). However, we favour ℓ_1 due to its globally sparsifying effect (see Section V-E for a numerical illustration).

Proposition 3 will take a particular relevance in Section IV, where (16) will allow us to design a joint-optimization problem to learn the linear weights and activations. Interestingly, the proposed minimization is compatible with the use of weight decay [43] in the training of neural networks (see (17) with $\mathrm{R}(\mathbf{W}) = \|\mathbf{W}\|_F^2$).

IV. LEARNING ACTIVATIONS

In this section, we propose a novel variational formulation to learn Lipschitz activations in a deep neural network. We select $\mathrm{BV}^{(2)}(\mathbb{R})$ as our search space to ensure the Lipschitz continuity of the input-output relation of the global network (see Theorem 1).

Similarly to the RKHS theory, the (weak*) continuity of the sampling functional is needed to guarantee the well-posedness of the learning problem. This is stated in Theorem 2 whose proof can be found in Appendix F. (We define the notion of weak*-convergence of neural networks in Appendix A.)

Theorem 2: For any $\boldsymbol{x}_0 \in \mathbb{R}^{N_0}$, the sampling functional $\delta_{\boldsymbol{x}_0} : \mathbf{f}_{\mathrm{deep}} \mapsto \mathbf{f}_{\mathrm{deep}}(\boldsymbol{x}_0)$ is weak*-continuous in the space of neural networks with second-order bounded-variation activations.

Given the data-set (X, Y) of size M that consists in the pairs $(\boldsymbol{x}_m, \boldsymbol{y}_m) \in \mathbb{R}^{N_0} \times \mathbb{R}^{N_L}$ for $m = 1, 2, \dots, M$, we then consider the following cost functional

$$\mathcal{J}(\mathbf{f}_{\mathrm{deep}}; X, Y) = \sum_{m=1}^{M} \mathrm{E}\left(\boldsymbol{y}_m, \mathbf{f}_{\mathrm{deep}}(\boldsymbol{x}_m)\right) + \sum_{l=1}^{L} \mu_l \mathrm{R}_l(\mathbf{W}_l) + \sum_{l=1}^{L} \lambda_l \|\boldsymbol{\sigma}_l\|_{\mathrm{BV}^{(2)},1}, \tag{17}$$

where $\mathbf{f}_{\mathrm{deep}}$ is a neural network with linear layers $\mathbf{W}_l$ and nonlinear layers $\boldsymbol{\sigma}_l = (\sigma_{1,l}, \dots, \sigma_{N_l,l})$, as specified in (7) and (8), $\mathrm{E}(\cdot,\cdot)$ is an arbitrary loss function, and $\mathrm{R}_l : \mathbb{R}^{N_l \times N_{l-1}} \to \mathbb{R}$ is a regularization functional for the linear weights of the lth layer. The standard choice for weight regularization is the Frobenius norm $\mathrm{R}(\mathbf{W}) = \|\mathbf{W}\|_F^2$, which corresponds to weight decay scheme in deep learning. Finally, the positive constants $\mu_l, \lambda_l > 0$ balance the regularization effect in the training step.

Theorem 3 states that, under some natural conditions, there always exists a solution of (17) with continuous piecewise-linear activation functions, which we refer to as a *deep-spline* neural network. The proof is given in Appendix G.

Theorem 3: Consider the training of a deep neural network via the minimization

$$\min_{\substack{\mathbf{w}_{n,l} \in \mathbb{R}^{N_{l-1}},\\ \sigma_{n,l} \in \mathrm{BV}^{(2)}(\mathbb{R})}} \mathcal{J}(\mathbf{f}_{\mathrm{deep}}; X, Y), \tag{18}$$

where $\mathcal{J}(\cdot; X, Y)$ is defined in (17). Moreover, assume that the loss function $\mathrm{E}(\cdot,\cdot)$ is proper, lower semi-continuous, and coercive. Assume that the regularization functionals R_l are continuous, and coercive. Then, there always exists a solution $\mathbf{f}^*_{\mathrm{deep}}$ of (17) with activations $\sigma_{n,l}$ of the form

$$\sigma_{n,l}(x) = \sum_{k=1}^{K_{n,l}} a_{n,l,k} \mathrm{ReLU}(x - \tau_{n,l,k}) + b_{1,n,l} x + b_{2,n,l}, \tag{19}$$

where $K_{n,l} \leq M$ and, $a_{n,l,k}, \tau_{n,l,k}, b_{\cdot,n,l} \in \mathbb{R}$ are adaptive parameters.

Theorem 3 suggests an optimal ReLU-based parametric to learn activations. This is a remarkable property as it translates the original infinite-dimensional problem (18) into a finite-dimensional parametric optimization, where one only needs to determine the ReLU weights $a_{n,l,k}$ and positions $\tau_{n,l,k}$ together with the affine terms $b_{1,n,l}, b_{2,n,l}$. Let us also mention that the baseline ReLU network and its variations (PReLU, LeakyReLU) are all included in this scheme as special cases of an activation of the form (19) with $K = 1$.

A similar result has been shown in the deep-spline representer theorem of Unser in [22]. However, there are three fundamental differences. Firstly, we relax the assumption of having normalized weights due to the practical considerations and the optimization challenges it brings. Secondly, we slightly modify the regularization functional that enables us to control the global Lipschitz constant of the neural network. Lastly, we show the existence of a minimizer in our proposed variational formulation that is, to the best of our knowledge, the first result of existence in this framework.

We remark that the choice of our regularization restrains the coefficients $b_{1,n,l}$ and $b_{2,n,l}$ from taking high values. This enables us to obtain the global bound (14) for the Lipschitz constant of the network, as opposed to the framework of [22], where only a semi-norm has been used for the regularization. The payoff is that, in [22], the activations have at most $(M-2)$ knots, which are the junctions between the consecutive linear pieces of a piecewise linear function, while our bound is $K_{n,l} \leq M$. This is the price to pay for controlling the Lipschitz regularity of the network. However, this is inconsequential in practice since there are usually much fewer knots than data points, because of the regularization penalty. The latter is justified through the computation of the $\mathrm{BV}^{(2)}$ norm of an activation of the form (19). It yields

$$\|\sigma_{n,l}\|_{\mathrm{BV}^{(2)}} = \|\boldsymbol{a}_{n,l}\|_1 + |\sigma(1)| + |\sigma(0)|, \tag{20}$$

where $\boldsymbol{a}_{n,l} = (a_{n,l,1}, \dots, a_{n,l,K_{n,l}})$ is the vector of ReLU coefficients. This shows that the $\mathrm{BV}^{(2)}$-regularization imposes an ℓ_1 penalty on the ReLU weights in the expansion (19), thus promoting sparsity [44]. In Section V, we illustrate the sparsity-promoting effect of the $\mathrm{BV}^{(2)}$-norm with numerical examples (see Figs. 3 and 4).

Another interesting property of the variational formulation (18) is the relation between the energy of consecutive linear and nonlinear layers. In Theorem 4, we exploit this relation. Its proof can be found in Appendix H.

Theorem 4: Consider Problem (18) with the weight regularization $\mathrm{R}_l(\mathbf{W}_l) = \|\mathbf{W}_l\|_F^2$ and positive parameters $\mu_l, \lambda_l > 0$ for all $l = 1, 2, \dots, L$. Then, for any of its local minima with the linear layers $\mathbf{W}_l$ and nonlinear layers $\boldsymbol{\sigma}_l$, we have that

$$\lambda_l \|\boldsymbol{\sigma}_l\|_{\mathrm{BV}^{(2)},1} = 2\mu_{l+1} \|\mathbf{W}_{l+1}\|_F^2, \quad l = 1, 2, \dots, L-1. \tag{21}$$

Theorem 4 shows that the regularization constants μ_l and λ_l provide a balance between the linear and nonlinear layers. In our experiments, we use the outcome of this theorem to determine the value of λ_l. More precisely, we select λ such that (21) holds in the initial setup. This is relevant in practice as it reduces the number of hyper-parameters that one needs to tune and results in a faster training scheme. We also show experimentally that this choice of λ is desirable.

V. NUMERICAL ILLUSTRATIONS

In this section, we discuss the practical aspects of our framework and conduct numerical experiments in which we compare the performance of our method to ReLU neural networks and its variants: LeakyReLU and PReLU activations.

According to Theorem 3, one can translate the original infinite-dimensional problem (18) to the minimization

$$\min_{\substack{\mathbf{w}_{n,l}\in\mathbb{R}^{N_{l-1}},\\ \boldsymbol{a}_{n,l}\in\mathbb{R}^{K_{n,l}}\\ b_{i,n,l}\in\mathbb{R}}} \sum_{m=1}^{M} \mathrm{E}\left(\boldsymbol{y}_m, \mathbf{f}_{\mathrm{deep}}(\boldsymbol{x}_m)\right) + \sum_{l=1}^{L}\mu_l \sum_{n=1}^{N_l} \|\mathbf{w}_{n,l}\|_2^2$$
$$+ \sum_{l=1}^{L}\lambda_l \sum_{n=1}^{N_l} \left(\|\boldsymbol{a}_{n,l}\|_1 + |\sigma_{n,l}(1)| + |\sigma_{n,l}(0)|\right), \tag{22}$$

where $\mathbf{f}_{\mathrm{deep}}$ is the global input-output mapping and $\sigma_{n,l}$ follows the parametric form given in (19). The optimization hence is over a set of finitely many variables, namely, the linear weights $\mathbf{w}_{n,l}$ and the unknown parameters of $\sigma_{n,l}$ in (19) for each neuron (n, l). The main challenge is that the number $K_{n,l}$ of ReLUs in the representation (19) is unknown *a priori*. To overcome this issue, we fix $K_{n,l}$ to a large value (we took $K = 21$ in our experiments) and rely on the sparsifying effect of $\mathrm{BV}^{(2)}$ regularization to promote a sparse expansion in the ReLU basis and remove the nonessential ReLUs. Thus, one may use the standard optimization schemes such as SGD or ADAM to learn the activations, jointly with the other parameters of the network. At the end of training, we perform a sparsifying step in which we annihilate some ReLU coefficients that are selected in such a way that the training error changes less than 1 percent.

Let us mention that the parametric form of the deep spline activation function has linear dependencies to its parameters $\boldsymbol{a}_{n,l} \in \mathbb{R}^{K_{n,l}}$ and $b_{i,n,l} \in \mathbb{R}, i = 1, 2$. However, this does not reduce the global optimization problem (22) to the learning of a linear classifier (or regressor). Indeed, for a fixed data point $\boldsymbol{x}_m$, the quantity $\mathbf{f}_{\mathrm{deep}}(\boldsymbol{x}_m)$ in general has nonlinear (and even nonconvex) dependencies to the parameters $\boldsymbol{a}_{n,l} \in \mathbb{R}^{K_{n,l}}$ and $b_{i,n,l} \in \mathbb{R}, i = 1, 2$ of the activation function of the neuron (n, l).

A. Setup

We designed a simple experiment in which the goal is to classify points that are inside a circle of area 2 centred at the origin. This is a classical two-dimensional supervised-learning problem, where the target function is

$$\mathbb{1}_{\mathrm{Circle}}(x_1, x_2) = \begin{cases} 1, & x_1^2 + x_2^2 \le \frac{2}{\pi} \\ 0, & \text{otherwise.} \end{cases} \tag{23}$$

The training dataset is obtained by generating $M = 1000$ random points from a uniform distribution on $[-1, 1]^2$. The points that lie inside and outside of the circle are then labeled as 1 and 0, respectively.

To illustrate the effect of our proposed scheme, we consider a family of fully connected architectures with layer descriptors of the form $(2, 2\,W, 1)$, where the width parameter $W \in \mathbb{N}$ governs the complexity of the architecture. We follow the classical choice of using a sigmoid activation in the last layer, together with the binary cross-entropy loss

$$\mathrm{E}(y, \widehat{y}) = -y\log(\widehat{y}) - (1-y)\log(1-\widehat{y}). \tag{24}$$

We take $\mu_1 = \mu_2 = \mu$ and, in each scheme, we tune the single hyper-parameter $\mu > 0$.

In our Lipschitz-based design, we use Xavier's rule [45] to initialize the linear weights. For the activations, we consider the simple piecewise-linear functions *absolute value* and *soft-thresholding*, defined as

$$f_{\mathrm{abs}}(x) = \begin{cases} x, & x \ge 0 \\ -x, & x < 0, \end{cases} \tag{25}$$

$$f_{\mathrm{soft}}(x) = \begin{cases} x - \frac{1}{2}, & x \ge \frac{1}{2} \\ 0, & x \in (-\frac{1}{2}, \frac{1}{2}) \\ x + \frac{1}{2}, & x \le -\frac{1}{2}. \end{cases} \tag{26}$$

We then initialize half of the activations with f_{abs} and the other half with f_{soft}. Intuitively, such initializations may allow the network to be flexible to both even and odd functions.

Moreover, we deploy Theorem 4 to tune the parameter λ. A direct calculation reveals that

$$\|f_{\mathrm{abs}}\|_{\mathrm{BV}^{(2)}} = 3, \qquad \|f_{\mathrm{soft}}\|_{\mathrm{BV}^{(2)}} = \frac{5}{2}. \tag{27}$$

This allows us to tune λ so that the optimality condition (21) holds in the initial setup. Due to the Xavier initialization, the linear weights of the second layer have variance $\sigma^2 = 2/(2W + 1)$. Therefore, we obtain that

$$\lambda = \frac{16}{11(2W+1)}\mu. \tag{28}$$

For an informed comparison, we also count the total number of parameters that is used in each scheme to represent the learned function. More specifically, with the layer descriptor $(2, 2\,W, 1)$, there are $6\,W$ linear weights and one bias for the last (sigmoidal) activation. In addition, there are parameters that depend on the specific activation we are using: There is a bias parameter in ReLU and LeakyReLU activations. In addition to bias, PReLU activation has an extra parameter (the slope in the negative part of the real line) as well and finally, the number of parameters in our scheme is adaptive and is equal to the number of active ReLUs plus the null-space coefficients in the representation (19).

B. Comparison with ReLU-Based Activations

We display in Fig. 2 the learned function $f : \mathbb{R}^2 \to \mathbb{R}$ in each case. We also disclose in Table I the performance and the number of active parameters of each scheme. One verifies that our scheme, already in the simplest configuration with layer descriptor (2,2,1), outperforms all other methods, even when

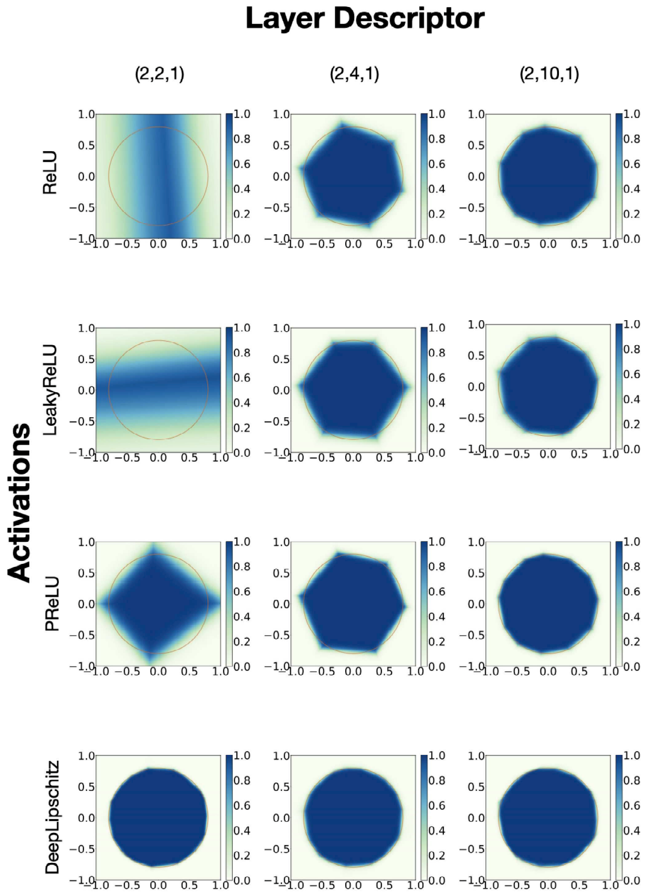


Fig. 2. Area classification with different activations, namely ReLU, LeakyReLU, PReLU, and our proposed scheme, which we refer to as *Deep Lipschitz*. In each case, we consider $W = 1, 2, 5$ hidden neurons.

TABLE I
NUMBER OF PARAMETERS AND PERFORMANCE IN THE AREA CLASSIFICATION EXPERIMENT

	Architecture	N_{param}	Performance
ReLU	(2, 10, 1)	41	98.15
LeakyReLU	(2, 10, 1)	41	98.12
PReLU	(2, 10, 1)	51	98.19
Deep Lipschitz	(2, 2, 1)	**23**	**98.54**

they are deployed over the richer architecture (2,10,1). Moreover, there are fewer parameters in the final representation of the target function in our scheme, as compared to the other methods. This experiment shows that the learning of activations in simple architectures is beneficial as it compensates the low capacity of the network and contributes to the generalization power of the global learning scheme. In the minimal case $W = 1$, we expect the network to learn parabola-type activations. This is due to the fact that the target function can be represented as

$$\mathbb{1}_{\text{Circle}}(x_1, x_2) = \mathbb{1}_{[0,\frac{2}{\pi}]}(x_1^2 + x_2^2), \tag{29}$$

which is the composition of the sum of two parabolas and a threshold function. To verify this intuition, we have also plotted the learned activations for the case $W = 1$ in Fig. 3.

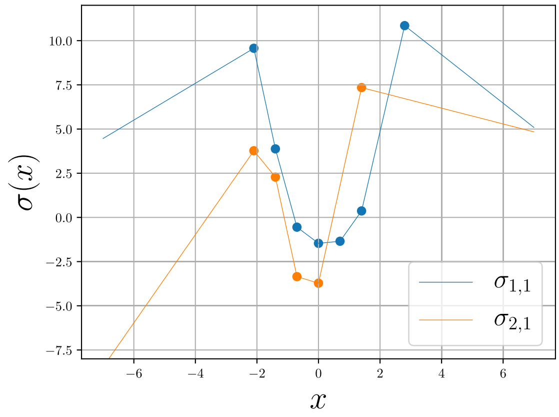


Fig. 3. Learned activations in the area classification experiment for a simple network with layer descriptor (2,2,1).

C. *Sparsity-Promoting Effect of* $\mathrm{BV}^{(2)}$*-Regularization*

Despite allowing a large number of ReLUs in the expansion (19) ($K = 21$), the learned activations (see Fig. 3) have sparse expansion in the ReLU basis. This is due to the sparsity-promoting effect of the $\mathrm{BV}^{(2)}$-norm on the ReLU coefficients and also the thresholding step that we added at the end of training.

D. *Effect of the Parameter* λ

To investigate the effect of the parameter λ in our experiments, we have set the weight decay parameter to $\mu = 10^{-4}$ and plotted in Fig. 4 the error rate, our proposed Lipschitz bound, and the total number of active ReLUs versus λ. As expected, the sparsity and Lipschitz regulariy of the network increases with λ. Consequently, one can control the overall regularity/complexity of the network by tuning this parameter.

As for the error rate, a definite transition occurs as λ varies. This suggests a range of “proper” values of λ (in this case, $\lambda < 10^{-3}$) in which the error would not change much. The critical value $\lambda = 10^{-3}$ is certainly the best choice, since it has a small error and, in addition, the overall network is maximally regularized in the sense of Lipschitz. However, one is required to compute these curves for each value of μ to find the optimal λ, which can be time consuming. A heuristic (but faster) approach is to honor (28). In this case, it yields $\lambda \approx 0.5 \times 10^{-5}$, which lies within the favourable range of each plot.

E. ℓ_1 *versus* ℓ_2 *Outer-Norms*

As mentioned in Remark 1, we can replace the ℓ_1 outer norm in our Lipschitz bound by ℓ_2, which results in a tighter bound. We compare the two cases in the area classification experiment, where we consider a network with the layer descriptor (2,10,1) and train it with two outer-norms. The results are reported in Table II. As can be seen, the two cases have similar performances. However, the ℓ_1 outer-norm results in a much sparser network with fewer parameters, due to its global sparsifying effect.

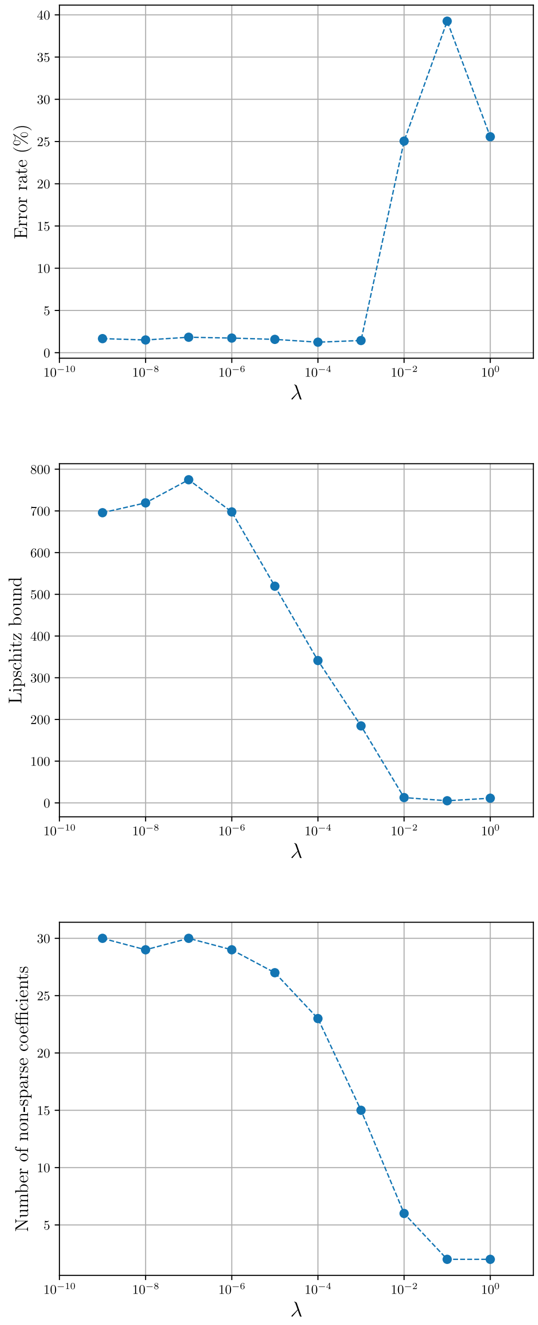


Fig. 4. From top to bottom: error rate, Lipschitz bound, and total number of nonzero ReLU coefficients versus λ in the simple architecture with layer descriptor (2,2,1).

TABLE II
EFFECT OF THE ℓ_1 VS. ℓ_2 OUTER NORMS IN THE AREA CLASSIFCATION EXPERIMENT

Outer-norm	N_{param}	Performance
ℓ_1	66	98.61
ℓ_2	89	98.39

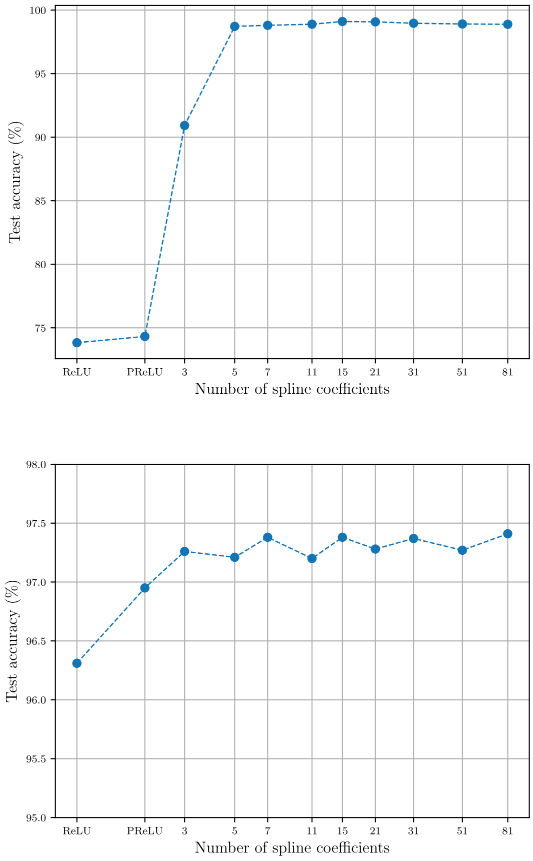


Fig. 5. The performance versus the number K of spline knots of each activation functions in the area classification (above) and in the MNIST experiment (below).

F. Effect of the Parameter K

Until now, we have performed all experiments with $K = 21$ spline knots. In this section, we let K vary and examine how this effects. We consider the area classification problem described in Section V-A and train a simple neural network with layer descriptor (2,2,1).

We also perform this experiment on the MNIST dataset [46] that consists of 28×28 grayscale images of digits from 0 to 9. In this case, we used a neural network that consists of three blocks. The first two are each composed of three layers: 1) a convolutional layer with a filter of size 5×5 and two output channels; 2) a nonlinear layer that has shared activations across each output channels (two activations in each layer are being learned); and 3) a max-pooling layer with kernel and stride of size 2. The third block is composed of a fully connected layer with output of size 10 followed by soft-max. The output of the network represents the probability of each digit.

The results are depicted in Fig. 5. In both cases, we also indicated the performance of ReLU and PReLU for comparison. We observe that the performance monotonically increases with K until it reaches saturation. We conclude that, although finding the best value for K is challenging, suboptimal value still leads to substantial improvements in the performance of the network and typically to better performances than ReLU networks and its variants.

VI. CONCLUSION

In this paper, we have introduced a variational framework to learn the activations of a deep neural network while controlling its global Lipschitz regularity. We have considered neural networks with second-order bounded-variation activations and we provided a global bound for their Lipschitz constants. We have showed that the solution of our proposed variational problem exists and is in the form of a deep-spline network with continuous piecewise linear activation functions. Our future work in this direction is to explore how the simplification of architectures can be compensated by the deployment of more complex activations.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their constructive feedbacks that improved the quality of the paper. They also would like to thank Dr. Julien Fageot, Dr. Pol del Aguila Pla, Dr. Jaejoon Yoo and, Pakshal Bohra for fruitful discussions.

APPENDIX

A. Topological Structure of $\mathrm{BV}^{(2)}(\mathbb{R})$

For a dual pair $(\mathcal{X}, \mathcal{X}')$ of Banach spaces, the sequence $\{w_n\}_{n\in\mathbb{N}} \in \mathcal{X}'$ converges in the weak*-topology to $w_{\lim} \in \mathcal{X}'$ if, for any element $\varphi \in \mathcal{X}$, we have that

$$\langle w_n, \varphi\rangle \to \langle w_{\lim}, \varphi\rangle, \quad n \to +\infty. \tag{30}$$

Consequently, a functional $\nu : \mathcal{X}' \to \mathbb{R}$ is weak*-continuous if $\nu(w_n) \to \nu(w_{\lim})$ for any sequence $\{w_n\}_{n\in\mathbb{N}} \in \mathcal{X}'$ that converges in the weak*-topology to $w_{\lim}$. This can be shown to be equivalent to the inclusion $\nu \in \mathcal{X}$. In other words, the predual space $\mathcal{X}$ is isometrically isomorphic to the space of weak*-continuous functionals over $\mathcal{X}'$.

We say that a sequence of neural networks converges in the weak*-topology if

- the networks all have the same layer descriptor (architecture);
- for any neuron in the architecture, the corresponding linear weights converge in the Euclidean topology and the corresponding activations converge in the weak*-topology of $\mathrm{BV}^{(2)}(\mathbb{R})$.

We conclude Appendix A by mentioning two important properties of the space of second-order bounded-variation functions.

1) The sampling functionals defined as $\delta_{x_0} : f \mapsto f(x_0)$ for $x_0 \in \mathbb{R}$ are weak*-continuous in the topology of $\mathrm{BV}^{(2)}(\mathbb{R})$ [22, Theorem 1].
2) Any function $f \in \mathrm{BV}^{(2)}(\mathbb{R})$ can be uniquely represented as

$$f(x) = \int_{\mathbb{R}} h(x, y)u(y)\mathrm{d}y + b_1 + b_2 x, \tag{31}$$

where $h(x, y) = (x - y)_+ - (1 - x)(-y)_+ - x(1 - y)_+$, $u = \mathrm{D}^2 f$, $b_1 = f(0)$, and $b_2 = f(1) - f(0)$. This result is a special case of Theorem 5 in [40].

B. Proof of Proposition 1

Proof: First, one needs to verify that the kernel function $h(\cdot, y_0)$ in (31) is Lipschitz-continuous with the constant 1 for all $y_0 \in \mathbb{R}$. Indeed, for any $x_1, x_2, y_0 \in \mathbb{R}$, we have that

$$|h(x_1, y_0) - h(x_2, y_0)| \le |x_1 - x_2|. \tag{32}$$

Now, using the representation (31) together with the triangle inequality, we have that

$$\begin{aligned}
&|f(x_1) - f(x_2)| = \\
&\left|\int_{\mathbb{R}} (h(x_1, y) - h(x_2, y))\, w(y)\mathrm{d}y + b_2(x_1 - x_2)\right| \\
&\le \left|\int_{\mathbb{R}} (h(x_1, y) - h(x_2, y))\, w(y)\mathrm{d}y\right| + |b_2||x_1 - x_2| \\
&\le \int_{\mathbb{R}} |h(x_1, y) - h(x_2, y)|\, |w(y)|\mathrm{d}y + |b_2||x_1 - x_2| \\
&\le |x_1 - x_2| \int_{\mathbb{R}} |w(y)|\mathrm{d}y + |b_2||x_1 - x_2| \\
&= |x_1 - x_2| \|w\|_{\mathcal{M}} + |b_2||x_1 - x_2| \\
&\le \|f\|_{\mathrm{BV}^{(2)}} |x_1 - x_2|.
\end{aligned}$$

■

C. Proof of Proposition 2

Proof: We prove the existence of the right derivative at $x = 0$ and deduce the existence of the left derivative by symmetry. Moreover, since $\mathrm{BV}^{(2)}(\mathbb{R})$ is a shift-invariant function space, the existence of left and right derivatives will be ensured at any point $x_0 \in \mathbb{R}$.

Let us denote

$$\Delta\sigma(a, b) = \frac{\sigma(a) - \sigma(b)}{a - b}, \quad a, b \in \mathbb{R}, a \ne b. \tag{33}$$

From Proposition 1, we have that

$$-\|\sigma\|_{\mathrm{BV}^{(2)}} \le \Delta\sigma(a, b) \le \|\sigma\|_{\mathrm{BV}^{(2)}}, \quad a, b \in \mathbb{R}, a \ne b. \tag{34}$$

Define quantities $M_{\sup}$ and $M_{\inf}$ as

$$\begin{aligned}
M_{\sup} &= \limsup_{h\to 0^+} \Delta\sigma(h, 0), \\
M_{\inf} &= \liminf_{h\to 0^+} \Delta\sigma(h, 0).
\end{aligned}$$

The finiteness of $|M_{\sup}|$ and $|M_{\inf}|$ is guaranteed by (34). Now, it remains to show that $M_{\sup} = M_{\inf}$ to prove the existence of the right derivative. Assume, by contradiction, that $M_{\sup} > M_{\inf}$. Consider a small value of $0 < \epsilon < \frac{M_{\sup} - M_{\inf}}{3}$ and define the constants $C_1 = (M_{\sup} - \epsilon)$ and $C_2 = (M_{\inf} + \epsilon)$. Clearly, we have $C_1 - C_2 \ge \epsilon > 0$. Moreover, due to the definition of lim sup and lim inf, there exist sequences $\{a_n\}_{n=0}^{\infty}$ and $\{b_n\}_{n=0}^{\infty}$ that are monotonically decreasing to 0 and are such that

$$\begin{aligned}
&\Delta\sigma(a_n, 0) > C_1, \quad \Delta\sigma(b_n, 0) < C_2, \\
&a_n > b_n > a_{n+1}, \quad \forall n \in \mathbb{N}.
\end{aligned}$$

One then has that

$$\sigma(a_n) > C_1 a_n + \sigma(0), \quad \sigma(b_n) < C_2 b_n + \sigma(0)$$

and, consequently,

$$\Delta\sigma(a_n, b_n) \geq \frac{C_1 a_n - C_2 b_n}{a_n - b_n} > C_1, \tag{35}$$

$$\Delta\sigma(b_n, a_{n+1}) \leq \frac{C_2 b_n - C_1 a_{n+1}}{b_n - a_{n+1}} < C_2. \tag{36}$$

From the definition of second-order total variation and using (35) and (36), we obtain that

$$\begin{aligned}\|\mathrm{D}^2\sigma\|_{\mathcal{M}} &\geq \sum_{n=0}^{\infty} |\Delta\sigma(a_n, b_n) - \Delta\sigma(b_n, a_{n+1})| \\ &\geq \sum_{n=0}^{\infty} \left(\frac{C_1 a_n - C_2 b_n}{a_n - b_n} - \frac{C_2 b_n - C_1 a_{n+1}}{b_n - a_{n+1}} \right) \\ &\geq \sum_{n=1}^{\infty} (C_1 - C_2) = \sum_{n=0}^{\infty} \epsilon = +\infty,\end{aligned}$$

which contradicts the original assumption $\sigma \in \mathrm{BV}^{(2)}(\mathbb{R})$. Hence, $M_{\mathrm{sup}} = M_{\mathrm{inf}}$ and the right derivative exists. ■

D. Proof of Theorem 1

Proof: From Proposition 1, for any $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathbb{R}^{N_{l-1}}$, we have that

$$\left|\sigma_{n,l}(\mathbf{w}_{n,l}^T \boldsymbol{x}_1) - \sigma_{n,l}(\mathbf{w}_{n,l}^T \boldsymbol{x}_2)\right| \leq \|\sigma_{n,l}\|_{\mathrm{BV}^{(2)}} \left|\mathbf{w}_{n,l}^T(\boldsymbol{x}_1 - \boldsymbol{x}_2)\right|. \tag{37}$$

Now, by using Hölder's inequality, we bound the Lipschitz constant of the linear layers as

$$\left|\mathbf{w}_{n,l}^T(\boldsymbol{x}_1 - \boldsymbol{x}_2)\right| \leq \|\mathbf{w}_{n,l}\|_q \|\boldsymbol{x}_1 - \boldsymbol{x}_2\|_p. \tag{38}$$

By combining (37) and (38) and using the fact that $\|\mathbf{w}_{n,l}\|_q \leq \|\mathbf{W}_l\|_{q,\infty}$, we obtain that

$$\begin{aligned}&\left|\sigma_{n,l}(\mathbf{w}_{n,l}^T \boldsymbol{x}_1) - \sigma_{n,l}(\mathbf{w}_{n,l}^T \boldsymbol{x}_2)\right|^p \\ &\quad \leq \|\mathbf{W}_l\|_{q,\infty}^p \|\sigma_{n,l}\|_{\mathrm{BV}^{(2)}}^p \|\boldsymbol{x}_1 - \boldsymbol{x}_2\|_p^p,\end{aligned} \tag{39}$$

which is a Lipschitz bound for the (n, l)th neuron of the neural network. By summing up over the neurons of layer l, we control the output of this layer as

$$\|\mathbf{f}_l(\boldsymbol{x}_1) - \mathbf{f}_l(\boldsymbol{x}_2)\|_p \leq \|\mathbf{W}_l\|_{q,\infty} \|\boldsymbol{\sigma}_l\|_{\mathrm{BV}^{(2)},p} \|\boldsymbol{x}_1 - \boldsymbol{x}_2\|_p. \tag{40}$$

Now, the composition of the layer inequalities results in the inequality (13) with the constant introduced in (14). ■

E. Proof of Proposition 3

Following Proposition 1 and using Cauchy–Schwarz' inequality, we obtain that

$$\begin{aligned}&\left|\sigma_{n,l}(\mathbf{w}_{n,l}^T \boldsymbol{x}_1) - \sigma_{n,l}(\mathbf{w}_{n,l}^T \boldsymbol{x}_2)\right| \\ &\quad \leq \|\sigma_{n,l}\|_{\mathrm{BV}^{(2)}} \|\mathbf{w}_{n,l}\|_2 \|\boldsymbol{x}_1 - \boldsymbol{x}_2\|_2.\end{aligned} \tag{41}$$

Combining it with the known hierarchy between the discrete ℓ_p norms and, in particular, the inequality $\|\boldsymbol{x}\|_2 \leq \|\boldsymbol{x}\|_1$ for any $\boldsymbol{x} \in \mathbb{R}^{N_l}$, we obtain that

$$\begin{aligned}\|\mathbf{f}_l(\boldsymbol{x}_1) - \mathbf{f}_l(\boldsymbol{x}_2)\|_2 &\leq \|\mathbf{f}_l(\boldsymbol{x}_1) - \mathbf{f}_l(\boldsymbol{x}_2)\|_1 \\ &= \sum_{n=1}^{N_l} \left|\sigma_{n,l}(\mathbf{w}_{n,l}^T \boldsymbol{x}_1) - \sigma_{n,l}(\mathbf{w}_{n,l}^T \boldsymbol{x}_2)\right| \\ &\leq \sum_{n=1}^{N_l} \|\sigma_{n,l}\|_{\mathrm{BV}^{(2)}} \|\mathbf{w}_{n,l}\|_2 \|\boldsymbol{x}_1 - \boldsymbol{x}_2\|_2 \\ &\leq \|\boldsymbol{\sigma}_l\|_{\mathrm{BV}^{(2)},2} \|\mathbf{W}_l\|_F \|\boldsymbol{x}_1 - \boldsymbol{x}_2\|_2^2.\end{aligned} \tag{42}$$

Note that in the last inequality of (43), we have again used Cauchy-Schwarz' inequality. Combining with $\|\boldsymbol{\sigma}_l\|_{\mathrm{BV}^{(2)},2} \leq \|\boldsymbol{\sigma}_l\|_{\mathrm{BV}^{(2)},1}$, we have that

$$\|\mathbf{f}_l(\boldsymbol{x}_1) - \mathbf{f}_l(\boldsymbol{x}_2)\|_2 \leq \|\boldsymbol{\sigma}_l\|_{\mathrm{BV}^{(2)},1} \|\mathbf{W}_l\|_F \|\boldsymbol{x}_1 - \boldsymbol{x}_2\|_2. \tag{43}$$

Finally by composing (43) through the layers, we obtain the announced bound.

F. Proof of Theorem 2

We first prove Lemma 1. We recall that a sequence of functions $f_t : \mathcal{X} \to \mathcal{Y}, t \in \mathbb{N}$, converges pointwise to $f_{\lim} : \mathcal{X} \to \mathcal{Y}$ if, for all $x \in \mathcal{X}$,

$$f_{\lim}(x) = \lim_{t \to +\infty} f_t(x). \tag{44}$$

Lemma 1: Given the Banach spaces $\mathcal{X}, \mathcal{Y}$, and $\mathcal{Z}$, consider the two sequences of functions $f_t : \mathcal{X} \to \mathcal{Y}$ and $g_t : \mathcal{Y} \to \mathcal{Z}$ such that they converge pointwise to the functions $f_{\lim} : \mathcal{X} \to \mathcal{Y}$ and $g_{\lim} : \mathcal{Y} \to \mathcal{Z}$, respectively. Moreover, assume that the functions g_t are all Lipschitz-continuous with a shared constant $C > 0$, so that, for any $y_1, y_2 \in \mathcal{Y}$, one has that

$$\|g_t(y_1) - g_t(y_2)\|_{\mathcal{Z}} \leq C\|y_1 - y_2\|_{\mathcal{Y}}, \quad \forall t \in \mathbb{N}. \tag{45}$$

Then, the composed sequence $h_t : \mathcal{X} \to \mathcal{Z}$ with $h_t = g_t \circ f_t$ converges pointwise to $h_{\lim} = g_{\lim} \circ f_{\lim}$.

Proof: We use the triangle inequality to obtain that

$$\begin{aligned}\|h_t(x) - h_{\lim}(x)\|_{\mathcal{Z}} &\leq \|h_t(x) - g_t(f_{\lim}(x))\|_{\mathcal{Z}} \\ &\quad + \|g_t(f_{\lim}(x)) - h_{\lim}(x)\|_{\mathcal{Z}}\end{aligned} \tag{46}$$

for all $x \in \mathcal{X}$. The uniform Lipschitz-continuity of g_t then yields that

$$\begin{aligned}\|h_t(x) - g_t(f_{\lim}(x))\|_{\mathcal{Z}} &= \|g_t(f_t(x)) - g_t(f_{\lim}(x))\|_{\mathcal{Z}} \\ &\leq C\|f_t(x) - f_{\lim}(x)\|_{\mathcal{Y}} \to 0\end{aligned} \tag{47}$$

as $t \to +\infty$. This is due to the pointwise convergence of $\{f_t\} \to f_{\lim}$. Similarly, the pointwise convergence $\{g_t\} \to g_{\lim}$ implies that

$$\|g_t(f_{\lim}(x)) - g_{\lim}(f_{\lim}(x))\|_{\mathcal{Z}} \to 0, \quad t \to +\infty$$

which, together with (47) and (46), proves the pointwise convergence of $h_t \to h_{\lim}$ as $t \to +\infty$. ■

Proof of Theorem 2: Assume that the sequence $\{\mathbf{f}_{\text{deep}}^{(t)}\}$ of neural networks with layers $\mathbf{f}_l^{(t)} = \boldsymbol{\sigma}_l^{(t)} \circ \mathbf{W}_l^{(t)} : \mathbb{R}^{N_{l-1}} \to \mathbb{R}^{N_l}$ for $l = 1, \dots, L$ (described in (7)) converges in the weak*-topology to

$$f_{\text{deep}}^{\lim} = \mathbf{f}_L^{\lim} \circ \dots \circ \mathbf{f}_1^{\lim}, \quad \forall l : \mathbf{f}_l^{\lim} = \boldsymbol{\sigma}_l^{\lim} \circ \mathbf{W}_l^{\lim}.$$

By definition, every element of $\boldsymbol{\sigma}_l^{(t)}$ converges in the weak*-topology to the corresponding element in $\boldsymbol{\sigma}_l^{\lim}$. The convergence is also pointwise due to the weak*-continuity of the sampling functional in the space of activation functions $\mathrm{BV}^{(2)}(\mathbb{R})$. The conclusion is thaty $\mathbf{f}_l^{(t)}$ also converges pointwise to $\mathbf{f}_l^{\lim}$.

In addition, knowing that any norm is weak*-continuous in its corresponding Banach space, one can find the uniform constant $T_1 > 0$ such that, for all $t > T_1$ and for all $l = 1, \dots, L$, we have that

$$\|\boldsymbol{\sigma}_l^{(t)}\|_{\mathrm{BV}^{(2)},2} \leq C_1 = 2 \max_l \|\boldsymbol{\sigma}_l^{\lim}\|_{\mathrm{BV}^{(2)},2}. \tag{48}$$

Similarly, from the convergence $\mathbf{W}_l^{(t)} \to \mathbf{W}_l^{\lim}$, one deduces that there exists a constant $T_2 > 0$ such that, for all $t > T_2$ and for all $l = 1, \dots, L$, we have that

$$\|\mathbf{W}_l^{(t)}\|_{2,\infty} \leq C_2 = 2 \max_l \|\mathbf{W}_l^{\lim}\|_{2,\infty}. \tag{49}$$

Now, from (49) with $p = q = 2$ and using (40), (48), we deduce that, for $t > \max(T_1, T_2)$, each layer of $\mathbf{f}_{\text{deep}}^{(t)}$ is Lipschitz-continuous with the shared constant $C = C_1 C_2$. Combining it with the pointwise convergence $\mathbf{f}_l^{(t)} \to \mathbf{f}_l^{\lim}$, one completes the proof by sequentially using the outcome of Lemma 1. ■

G. *Proof of Theorem 3*

Proof: We divide the proof in two parts. First, we show the existence of the solution of (17) and, then, we show the existence of a solution with activations of the form (19).

Existence of Solution: Consider an arbitrary neural network $\mathbf{f}_{\text{deep},0}$ with the cost $A = \mathcal{J}(\mathbf{f}_{\text{deep},0}; X, Y)$. The coercivity of R_l guarantees the existence of the constants B_l for $l = 1, 2, \dots, L$ such that

$$\|\mathbf{w}_{n,l}\|_2 \geq B_l \Rightarrow \mathrm{R}_l(\mathbf{w}_{n,l}) \geq \frac{A}{\mu_l}. \tag{50}$$

This allows us to transform the unconstrained problem (17) into the equivalent constrained minimization

$$\min_{\substack{\mathbf{w}_{n,l} \in \mathbb{R}^{N_{l-1}}, \\ \sigma_{n,l} \in \mathrm{BV}^{(2)}(\mathbb{R})}} \mathcal{J}(\mathbf{f}_{\text{deep}}), \quad \text{s.t.} \begin{cases} \|\mathbf{w}_{n,l}\|_2 \leq B_l, \\ \mathrm{TV}^{(2)}(\sigma_{n,l}) \leq A/\lambda_l, \\ |\sigma_{n,l}(0)| \leq A/\lambda_l, \\ |\sigma_{n,l}(1)| \leq A/\lambda_l. \end{cases} \tag{51}$$

The equivalence is due to the fact that any neural network that does not satisfy the constraints of (52) has a strictly bigger cost than $\mathbf{f}_{\text{deep},0}$ and, hence, is not in the solution set. Due to the decomposition (31), we can rewrite (51) as

$$\min_{\substack{\mathbf{w}_{n,l} \in \mathbb{R}^{N_{l-1}}, \\ u_{n,l} \in \mathcal{M}(\mathbb{R}) \\ b_{\cdot,n,l} \in \mathbb{R}}} \mathcal{J}(\mathbf{f}_{\text{deep}}), \quad \text{s.t.}, \begin{cases} \|\mathbf{w}_{n,l}\|_2 \leq B_l, \\ \|u_{n,l}\|_{\mathcal{M}} \leq A/\lambda_l, \\ |b_{1,n,l}|, |b_{2,n,l}| \leq 2\, A/\lambda_l. \end{cases} \tag{52}$$

Due to the Banach-Anaoglu theorem [37], the feasible set in (52) is weak*-compact. Moreover, the cost functional defined in (17) is a composition and sum of lower-semicontinuous functions and weak*-continuous functionals (see Theorem 2). Hence, it is itself weak*-lower semicontinuous. This guarantees the existence of a minimizer of (52) (and, consequently, of (18)), due to the generalized Weierstrass theorem [47].

Optimal Activations: Let $\tilde{\mathbf{f}}_{\text{deep}}$ be a solution of (18) with

$$\tilde{\mathbf{f}}_{\text{deep}} = \tilde{\mathbf{f}}_L \circ \dots \circ \tilde{\mathbf{f}}_1, \quad \forall l : \tilde{\mathbf{f}}_l = \tilde{\boldsymbol{\sigma}}_l \circ \tilde{\mathbf{W}}_l. \tag{53}$$

For any input vector $\boldsymbol{x}_m$ in the dataset $\boldsymbol{X}$, we then define the vectors $\boldsymbol{z}_{l,m} = (z_{1,l,m}, \dots, z_{N_l,l,m}), \boldsymbol{s}_{l,m} = (s_{1,l,m}, \dots, s_{N_l,l,m}) \in \mathbb{R}^{N_l}$ as

$$\boldsymbol{z}_{l,m} = \tilde{\mathbf{f}}_l \circ \dots \circ \tilde{\mathbf{f}}_1(\boldsymbol{x}_m),$$

$$\boldsymbol{s}_{l,m} = \tilde{\mathbf{W}}_l \circ \tilde{\mathbf{f}}_{l-1} \circ \dots \circ \tilde{\mathbf{f}}_1(\boldsymbol{x}_m).$$

Now, we show that the activation $\tilde{\sigma}_{n,l}$ of the neuron indexed by (n, l) is indeed a solution of the minimization

$$\min_{\sigma \in \mathrm{BV}^{(2)}(\mathbb{R})} \mathrm{TV}^{(2)}(\sigma) = \|\mathrm{D}^2 \sigma\|_{\mathcal{M}} \quad s.t.$$

$$\begin{cases} \sigma(s_{n,l,m}) = z_{n,l,m}, & m = 1, 2, \dots, M, \\ \sigma(x) = \tilde{\sigma}_{n,l}(x), & x \in \{0, 1\}. \end{cases} \tag{54}$$

Assume by contradiction that there exists a function $\sigma \in \mathrm{BV}^{(2)}(\mathbb{R})$ that satisfies the feasiblity conditions (54) and is such that $\mathrm{TV}^{(2)}(\sigma) < \mathrm{TV}^{(2)}(\tilde{\sigma}_{n,l})$. Then, we have that

$$\begin{aligned} \|\sigma\|_{\mathrm{BV}^{(2)}} &= \mathrm{TV}^{(2)}(\sigma) + |\sigma(0)| + |\sigma(1)| \\ &= \mathrm{TV}^{(2)}(\sigma) + |\tilde{\sigma}_{n,l}(0)| + |\tilde{\sigma}_{n,l}(1)| \\ &< \mathrm{TV}^{(2)}(\tilde{\sigma}_{n,l}) + |\tilde{\sigma}_{n,l}(0)| + |\tilde{\sigma}_{n,l}(1)| \\ &= \|\tilde{\sigma}_{n,l}\|_{\mathrm{BV}^{(2)}}. \end{aligned} \tag{55}$$

In addition, due to the feasiblity assumptions $\sigma(s_{n,l,m}) = z_{n,l,m}$ for $m = 1, \dots, M$, one readily verifies that, by replacing $\tilde{\sigma}_{n,l}$ by σ in the optimal neural network $\tilde{\mathbf{f}}_{\text{deep}}$, the data fidelity term $\sum_{m=1}^{M} \mathrm{E}(\boldsymbol{y}_m, \tilde{\mathbf{f}}_{\text{deep}}(\boldsymbol{x}_m))$ in (17) remains untouched. The same holds for the weight regularization term $\sum_{l=1}^{L} \mathrm{R}_l(\mathbf{W}_l)$. However, from (55), one gets a strictly smaller overall $\mathrm{BV}^{(2)}$ penalty with σ that contradicts the optimality of $\tilde{\mathbf{f}}_{\text{deep}}$. With a similar argument, one sees that, for any solution $\sigma \in \mathrm{BV}^{(2)}$ of (54), the substitution of $\tilde{\sigma}_{n,l}$ by σ yields another solution of (18). Due to Lemma 1 of [22], Problem (54) has a solution that is a linear spline of the form (19) with $K_{n,l} \leq (\tilde{M} - 2)$, where $\tilde{M} = M + 2$ is the number of constraints in (54). By using this result for every neuron (n, l), we verify the existence of a deep-spline solution of (18). ■

H. Proof of Theorem 4

Proof: For any local minima $\mathbf{f}_{\text{deep}}$ of (18) with linear weights $\mathbf{W}_l$ and nonlinear layers $\boldsymbol{\sigma}_l$ and for any layer $l^* \neq L$, consider the perturbed network $\mathbf{f}_{\text{deep},\epsilon}$ with the linear layers

$$\mathbf{W}_{l,\epsilon} = \begin{cases} \mathbf{W}_l, & l \neq l^* + 1 \\ (1+\epsilon)\mathbf{W}_{l^*}, & l = l^* + 1 \end{cases} \tag{56}$$

and the nonlinear layers

$$\boldsymbol{\sigma}_{l,\epsilon} = \begin{cases} \boldsymbol{\sigma}_l, & l \neq l^* \\ (1+\epsilon)^{-1}\boldsymbol{\sigma}_{l^*}, & l = l^* \end{cases} \tag{57}$$

for any $\epsilon \in (-1, 1)$. One readily verifies that, for any $\boldsymbol{x} \in \mathbb{R}^{N_0}$ and any $\epsilon \in \mathbb{R}$, we have that $\mathbf{f}_{\text{deep},\epsilon}(\boldsymbol{x}) = \mathbf{f}_{\text{deep}}(\boldsymbol{x})$ and, hence, both networks have the same data-fidelity penalty in the global cost (17). In fact, the only difference between their overall cost is associated to the regularization terms of the (l^*+1)th linear layer and the lth nonilnear layer. For those, the scaling property of norms yields that

$$\|\mathbf{W}_{l^*+1,\epsilon}\|_F^2 = (1+\epsilon)^2 \|\mathbf{W}_{l^*+1}\|_F^2, \tag{58}$$

$$\|\boldsymbol{\sigma}_{l^*,\epsilon}\|_{\text{BV}^{(2)},1} = (1+\epsilon)^{-1} \|\boldsymbol{\sigma}_{l^*}\|_{\text{BV}^{(2)},1}. \tag{59}$$

Due to the (local) optimality of $\mathbf{f}_{\text{deep}}$, there exists a constant $\epsilon_{\max}$ such that, for all $\epsilon \in (-\epsilon_{\max}, \epsilon_{\max})$, we have that

$$\mathcal{J}(\mathbf{f}_{\text{deep}}) \leq \mathcal{J}(\mathbf{f}_{\text{deep},\epsilon}). \tag{60}$$

Now, from (58) and (59), we have that

$$\begin{aligned} &\mu_{l^*+1}\|\mathbf{W}_{l^*+1}\|_F^2 + \lambda_l \|\boldsymbol{\sigma}_{l^*}\|_{\text{BV}^{(2)},1} \\ &\quad \leq \mu_{l^*+1}(1+\epsilon)^2\|\mathbf{W}_{l^*+1}\|_F^2 + \lambda_l(1+\epsilon)^{-1}\|\boldsymbol{\sigma}_{l^*}\|_{\text{BV}^{(2)},1}, \end{aligned}$$

for any $\epsilon \in (-\epsilon_{\max}, \epsilon_{\max})$. By simplifying the latter inequality, we get that

$$0 \leq \epsilon g(\epsilon), \quad \forall \epsilon \in (-\epsilon_{\max}, \epsilon_{\max}), \tag{61}$$

where $g(\epsilon) = \mu_{l^*+1}\|\mathbf{W}_{l^*+1}\|_F^2(\epsilon+2) - \lambda_l\|\boldsymbol{\sigma}\|_{\text{BV}^{(2)}}(1+\epsilon)^{-1}$ is a continuous function of ϵ in the interval $(-1, 1)$. This yields that $g(\epsilon)$ is nonnegative for positive values of ϵ and is nonpositive for negative values of ϵ. Hence, we get that $g(0) = 0$ and, consequently, that

$$\lambda_{l^*}\|\boldsymbol{\sigma}_{l^*}\|_{\text{BV}^{(2)}} = 2\mu_{l^*+1}\|\mathbf{W}_{l^*+1}\|_F^2. \tag{62}$$

■

REFERENCES

[1] S. Aziznejad and M. Unser, "Deep spline networks with control of Lipschitz regularity," in *Proc. 44th IEEE Int. Conf. Acoust., Speech, Signal Process.*, Brighton, U.K., May 12–17, 2019, pp. 3242–3246.

[2] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer, 2006.

[3] N. Aronszajn, "Theory of reproducing kernels," *Trans. Amer. Math. Soc.*, vol. 68, no. 3, pp. 337–404, May 1950.

[4] G. Wahba, *Spline Models for Observational Data*. Philadelphia, PA: Society for Industrial and Applied Mathematics, 1990.

[5] G. Kimeldorf and G. Wahba, "Some results on Tchebycheffian spline functions," *J. Math. Anal. Appl.*, vol. 33, no. 1, pp. 82–95, Jan. 1971.

[6] B. Schölkopf, R. Herbrich, and A. J. Smola, "A generalized representer theorem," in *Int. Conf. Comput. Learn. Theory*. Berlin, Heidelberg: Springer, 2001, pp. 416–426.

[7] V. Vapnik, *Statistical Learning Theory*. New York, NY, USA: Wiley, 1998.

[8] T. Poggio and F. Girosi, "Networks for approximation and learning," *Proc. IEEE*, vol. 78, no. 9, pp. 1481–1497, Sep. 1990.

[9] B. Schölkopf *et al.*, "Comparing support vector machines with Gaussian kernels to radial basis function classifiers," *IEEE Trans. Signal Process.*, vol. 45, no. 11, pp. 2758–2765, Nov. 1997.

[10] F. Girosi, M. Jones, and T. Poggio, "Priors stabilizers and basis functions: From regularization to radial, tensor and additive splines," *Tech. Rep. A.I. Memo No. 1430*, 1993.

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inform. Process. Syst.*, 2012, pp. 1097–1105.

[12] K. Jin, M. McCann, E. Froustey, and M. Unser, "Deep convolutional neural network for inverse problems in imaging," *IEEE Trans. Image Process.*, vol. 26, no. 9, pp. 4509–4522, 2017.

[13] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assisted Intervention*, Munich, Germany, 2015, pp. 234–241.

[14] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nat.*, vol. 521, pp. 436–444, May 2015.

[15] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA: MIT Press Cambridge, 2016, vol. 1.

[16] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics Control, Signals Syst.*, vol. 2, no. 4, pp. 303–314, 1989.

[17] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proc. 14th Int. Conf. Artif. Intell. Statist.*, 2011, pp. 315–323.

[18] A. Maas, A. Hannun, and A. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. 30th Int. Conf. Mach. Learn.*, vol. 30, no. 1, Atlanta, Georgia, USA, 2013, p. 3.

[19] G. F. Montufar, R. Pascanu, K. Cho, and Y. Bengio, "On the number of linear regions of deep neural networks," in *Proc. Adv. Neural Inform. Process. Syst.*, 2014, pp. 2924–2932.

[20] R. Arora, A. Basu, P. Mianjy, and A. Mukherjee, "Understanding deep neural networks with rectified linear units," 2016, *arXiv:1611.01491*.

[21] T. Poggio, L. Rosasco, A. Shashua, N. Cohen, and F. Anselmi, "Notes on hierarchical splines, DCLNs and i-theory," Center for Brains, Minds and Machines (CBMM), Tech. Rep., 2015.

[22] M. Unser, "A representer theorem for deep neural networks," *J. Mach. Learn. Res.*, vol. 20, no. 110, pp. 1–30, January-Present 2019.

[23] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on Imagenet classification," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1026–1034.

[24] C. De Boor, *A Practical Guide to Splines*. New York, NY, USA: Springer-Verlag, 1978, vol. 27.

[25] M. Unser, "SPLINES: A perfect fit for signal and image processing," *IEEE Signal Process. Mag.*, vol. 16, no. 6, pp. 22–38, Nov. 1999.

[26] S. Lane, M. Flax, D. Handelman, and J. Gelfand, "Multi-layer perceptrons with B-spline receptive field functions," in *Adv. Neural Inform. Process. Syst.*, 1991, pp. 684–692.

[27] L. Vecci, F. Piazza, and A. Uncini, "Learning and approximation capabilities of adaptive spline activation function neural networks," *Neural Netw.*, vol. 11, no. 2, pp. 259–270, 1998.

[28] S. Guarnieri, F. Piazza, and A. Uncini, "Multilayer feedforward networks with adaptive spline activation function," *IEEE Trans. Neural Netw.*, vol. 10, no. 3, pp. 672–683, May 1999.

[29] F. Agostinelli, M. Hoffman, P. Sadowski, and P. Baldi, "Learning activation functions to improve deep neural networks," 2014, *arXiv:1412.6830*.

[30] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein GAN," 2017, *arXiv:1701.07875*.

[31] A. Bora, A. Jalal, E. Price, and A. G. Dimakis, "Compressed sensing using generative models," in *Proc. 34th Int. Conf. Mach. Learn.*, vol. 70, 2017, pp. 537–546.

[32] H. Gupta, K. H. Jin, H. Q. Nguyen, M. T. McCann, and M. Unser, "CNN-based projected gradient descent for consistent CT image reconstruction," *IEEE Trans. Med. Imag.*, vol. 37, no. 6, pp. 1440–1453, Jun. 2018.

[33] B. Neyshabur, S. Bhojanapalli, D. McAllester, and N. Srebro,"Exploring generalization in deep learning," in *Proc. Adv. Neural Inform. Process. Syst.*, 2017, pp. 5947–5956.

[34] S. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "DeepFool: A simple and accurate method to fool deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2574–2582.

[35] A. Fawzi, S.-M. Moosavi-Dezfooli, and P. Frossard, "The robustness of deep networks: A geometrical perspective," *IEEE Signal Process. Mag.*, vol. 34, no. 6, pp. 50–62, 2017.

[36] V. Antun, F. Renna, C. Poon, B. Adcock, and A. C. Hansen, "On instabilities of deep learning in image reconstruction and the potential costs of AI?" in *Proc. Nat. Acad. Sci.*, May 2020.

[37] W. Rudin, *Functional Analysis International Series in Pure and Applied Mathematics*. New York, NY, USA: McGraw-Hill, Inc., 1991.

[38] W. Rudin, *Real and Complex Analysis*, 3rd ed. New York, NY, USA: McGraw-Hill, Inc., 1987.

[39] M. Unser and P. D. Tafti, *An Introduction to Sparse Stochastic Processes*. Cambridge, U.K.: Cambridge Univ. Press, 2014.

[40] M. Unser, J. Fageot, and J. P. Ward, "Splines are universal solutions of linear inverse problems with generalized TV regularization," *SIAM Rev.*, vol. 59, no. 4, pp. 769–793, Nov. 2017.

[41] J. Heinonen, *Lectures on Lipschitz Analysis*. University of Jyväskylä, 2005, no. 100.

[42] D. Rumelhart, G. Hinton, and R. Williams, "Learning representations by back-propagating errors," *Cogn. Model.*, vol. 5, no. 3, p. 1, 1988.

[43] A. Krogh and J. Hertz, "A simple weight decay can improve generalization," in *Proc. Adv. Neural Inform. Process. Syst.*, 1992, pp. 950–957.

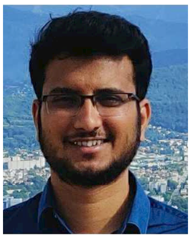
[44] D. Donoho, "For most large underdetermined systems of linear equations the minimal l1-norm solution is also the sparsest solution," *Commun. Pure Appl. Mathematics: A J. Issued by Courant Inst. Math. Sci.*, vol. 59, no. 6, pp. 797–829, 2006.

[45] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. 13th Int. Conf. Artif. Intell. Statist.*, 2010, pp. 249–256.

[46] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[47] A. Kurdila and M. Zabarankin, *Convex Functional Analysis*. Berlin, Germany: Springer, 2006.

Shayan Aziznejad received the B.Sc. degrees in electrical engineering and mathematics in 2017, from Sharif University of Technology (SUT), Tehran, Iran. He is currently pursuing the Ph.D. degree under the direction of M. Unser with the Biomedical Imaging Group at Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland. His research focuses on variational formulation of inverse problems and machine learning, spline theory, and stochastic processes. He is the recipient of the Best Student Paper Award at ICASSP 2019.

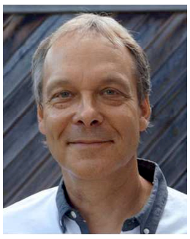


Harshit Gupta received B.Tech (Electronics and Communications Engineering) from Indian Institute of Technology (IIT), Guwahati in 2015. He is currently pursuing Ph.D. degree with the Biomedical Imaging Group, Ecole Polytechnique Fdrale de Lausanne (EPFL), Lausanne, Switzerland, under the direction of Prof. Michael Unser. He obtained B.Tech. (Electronics and Communications Engineering) from Indian Institute of Technology (IIT), Guwahati in 2015.

His research is aimed at designing deep-learning-based algorithms, backed with sound mathematics, in order to solve inverse problems in imaging. He is interested in the modalities of Cryo Electron Microscopy (Cryo-EM), Computational Tomography (CT), and Magnetic Resonance Imaging (MRI). He has also worked on spline theory, representer theorems, and variational methods.

Joaquim Campos received the B.Sc. degree in electrical engineering from Instituto Superior Técnico, Lisbon, Portugal, in 2016, and the M.Sc. degree in communication systems from École Polytechnique Fédérale de Lausanne (EPFL), Switzerland, in 2020.

He is currently working towards a Ph.D. degree with the Biomedical Imaging Group, École Polytechnique Fédérale de Lausanne, under the direction of M. Unser. His research focuses on machine learning and inverse problems.



Michael Unser (Fellow, IEEE) received the M.S. (summa cum laude) and Ph.D. degrees in electrical engineering in 1981 and 1984, respectively, from Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland. Curently, he is Professor and Director of EPFL's Biomedical Imaging Group, Lausanne, Switzerland. His primary area of investigation is biomedical image processing. He is internationally recognized for his research contributions to sampling theory, wavelets, the use of splines for image processing, stochastic processes, and computational bioimaging. He has published over 350 journal papers on those topics. He is the author with P. Tafti of the book *An introduction to sparse stochastic processes*, Cambridge University Press 2014.

From 1985 to 1997, he was with the Biomedical Engineering and Instrumentation Program, National Institutes of Health, Bethesda USA, conducting research on bioimaging.

Dr. Unser has served on the Editorial Board of most of the primary journals in his field including the IEEE TRANSACTIONS ON MEDICAL IMAGING (associate Editor-in-Chief 2003–2005), IEEE TRANS. IMAGE PROCESSING, Proceedings of IEEE, and *SIAM Journal of Imaging Sciences*. He is the Founding Chair of the technical committee on Bio Imaging and Signal Processing (BISP) of the IEEE Signal Processing Society.

Prof. Unser is a fellow of the IEEE (1999), an EURASIP fellow (2009), and a member of the Swiss Academy of Engineering Sciences. He is the recipient of several international prizes including five IEEE-SPS Best Paper Awards, two Technical Achievement Awards from the IEEE (2008 SPS and EMBS 2010), the Technical Achievement Award from EURASIP (2018), and a recent Career Achievement Award (IEEE EMBS 2020).