DEEP SPLINE NETWORKS WITH CONTROL OF LIPSCHITZ REGULARITY

Shayan Aziznejad and Michael Unser

Biomedical Imaging Group, École polytechnique fédérale de Lausanne (EPFL), Switzerland

ABSTRACT

The motivation for this work is to improve the performance of deep neural networks through the optimization of the individual activation functions. Since the latter results in an infinite-dimensional optimization problem, we resolve the ambiguity by searching for the sparsest and most regular solution in the sense of Lipschitz. To that end, we first introduce a bound that relates the properties of the pointwise nonlinearities to the global Lipschitz constant of the network. By using the proposed bound as regularizer, we then derive a representer theorem that shows that the optimum configuration is achievable by a deep spline network. It is a variant of a conventional deep ReLU network where each activation function is a piecewise-linear spline with adaptive knots. The practical interest is that the underlying spline activations can be expressed as linear combinations of ReLU units and optimized using ℓ_1-minimization techniques.

Index Terms— Deep learning, Lipschitz regularity, learned activations, deep spline, representer theorem.

1. INTRODUCTION

Supervised learning is often formulated as a data-fitting problem (a.k.a. regression). There, the goal is to estimate a map $f : \mathbb{R}^d \to \mathbb{R}$ from a set of (possibly inaccurate) samples $y_m \approx f(\boldsymbol{x}_m), m = 1, \dots, M$ [1]. Researchers have exploited the connection with splines and regularization theory to justify the use of kernel estimators; in particular, radial basis functions [2, 3, 4]. In the reproducing kernel Hilbert space (RKHS) framework, the learning problem is formulated via the minimization

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

where $E(\cdot, \cdot)$ is an arbitrary error function and $\mathcal{H}$ is an RKHS with the corresponding reproducing kernel $\mathrm{k} : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$. Remarkably, it can be shown (see [5]) that the solution of (1) can always be expressed as the kernel expansion

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

which then yields a discretization scheme for determining the optimal coefficients $a_1, a_2, \dots, a_M$.

The classical RKHS formulation is elegant but suffers from one major drawback: It requires as many basis functions as there are data samples. This is the reason why researchers have developed schemes to reduce the number of active kernels, for example by using a sparsity-enforcing loss function such as the ϵ-insensitive norm in the SVM regression [6, 7] or by replacing the quadratic regularization supported by the theory of RKHS by a sparsity-enforcing penalty such as the ℓ_1-norm, which results in the generalized LASSO [8].

While kernel methods used to be a major player in machine learning since the mid '90s, they have been recently outperformed by deep neural networks in many real-world applications such as image classification [9] and segmentation [10]. The leading idea of deep learning is to build powerful architectures via the repeated composition of elementary blocks that consist of a linear transformation (with learnable weights) followed by a layer of (fixed) pointwise nonlinearities (neuron activations) [11, 12]. While practitioners have considered a variety of activation functions, such as the sigmoid, a preferred choice that has emerged over the years is the rectified linear unit $\mathrm{ReLU}(x) = x_+ \stackrel{\Delta}{=} \max(x, 0)$ [13].

Deep networks with ReLUs perform remarkably well and are typically easy to train [11]. Moreover, they implement a global input-output relation that is continuous and piecewise-linear [14]. This property is due to the ReLU itself being a linear spline, which has prompted Poggio *et al.* to interpret deep neural networks as hierarchical splines [15]. Recently, Unser was able to establish theoretically the optimality of linear spline activations in the sense that they satisfy the minimization of a second-order total-variation criterion [16].

Lipschitz continuity is a desirable property of a deep neural network. It has been assumed in various analyses of deep learning, for example in the analysis of Wasserstein GAN's [17]; the convergence of CNN-based projection algorithms in inverse problems [18] and in the analysis of the generalization property of deep neural networks [19].

In this paper, we propose a new variational framework for deep neural networks with the motivation of controlling the Lipschitz regularity of the whole network. To do so, we propose a new regularization that gives a bound on the Lipschitz constant. We then derive a representer theorem for this problem, showing that the optimal solution takes the form of a

This work was funded by the Swiss National Science Foundation under Grant 200020-162343 / 1

deep spline network where each activation is a linear combination of ReLU units plus a linear term.

2. MATHEMATICAL DESCRIPTION

2.1. Notations and Definitions

First, we recall some relevant mathematical definitions. $\mathcal{C}_0(\mathbb{R})$ is the Banach space of continuous functions that vanish at infinity, equipped with the supremum norm $\|f\|_\infty \triangleq \sup_{x\in\mathbb{R}} f(x)$. The topological dual of $\mathcal{C}_0(\mathbb{R})$, denoted by $\mathcal{M}(\mathbb{R})$, is the space of Radon measures over $\mathbb{R}$ equipped with the $\mathcal{M}$-norm (the total-variation norm in the sense of measure theory)

$$\|w\|_{\mathcal{M}} \triangleq \sup_{\varphi\in\mathcal{C}_0(\mathbb{R})} \frac{\langle w, \varphi\rangle}{\|\varphi\|_\infty}. \tag{3}$$

The space $\mathcal{M}(\mathbb{R})$ is an extension of $L_1(\mathbb{R})$, since $L_1(\mathbb{R}) \subseteq \mathcal{M}(\mathbb{R})$ and, for any $f \in L_1(\mathbb{R})$, $\|f\|_{L_1} = \|f\|_{\mathcal{M}}$. However, $\mathcal{M}(\mathbb{R})$ contains the shifted Dirac distributions $\delta(\cdot - x)$ with $\|\delta(\cdot - x)\|_{\mathcal{M}} = 1$, which shows that it is larger than $L_1(\mathbb{R})$.

Generally, a function $f : \mathcal{X} \to \mathcal{Y}$ ($\mathcal{X}$ and $\mathcal{Y}$ are normed spaces with their corresponding norms denoted by $\|\cdot\|_{\mathcal{X}}$, $\|\cdot\|_{\mathcal{Y}}$, respectively) is Lipschitz if, for all $x_1, x_2 \in \mathcal{X}$, there exists a constant C such that

$$\|f(x_1) - f(x_2)\|_{\mathcal{Y}} \leq C\|x_1 - x_2\|_{\mathcal{X}}. \tag{4}$$

Classical supervised-learning algorithms are tied to specific classes of parametric vector-valued functions $\mathbf{f} : \mathbb{R}^N \to \mathbb{R}^{N'}$, which may also serve as elementary modules of more advanced architectures. A deep feedforward network results from the sequential composition of L such units under the implicit assumption that the domain and range of consecutive operators are compatible. The primary transformations are (i) linear maps with adjustable weights, and (ii) component-wise nonlinearities. Accordingly, we represent a standard deep neural network (DNN) as

$$\mathbf{f}_{\text{deep}} : \mathbb{R}^{N_0} \to \mathbb{R}^{N_L} : \boldsymbol{x} \mapsto \mathbf{f}_L \circ \cdots \circ \mathbf{f}_1(\boldsymbol{x}) \tag{5}$$

with associated dimensionality/layer descriptor $(N_0, \ldots, N_L)$. The ℓth layer $\mathbf{f}_\ell : \mathbb{R}^{N_{\ell-1}} \to \mathbb{R}^{N_\ell}$ of the network is then described by

$$\mathbf{f}_\ell(\boldsymbol{x}) = \left(\sigma_{1,\ell}(\mathbf{w}_{1,\ell}^T\boldsymbol{x}), \sigma_{2,\ell}(\mathbf{w}_{2,\ell}^T\boldsymbol{x}), \ldots, \sigma_{N_\ell,\ell}(\mathbf{w}_{N_\ell,\ell}^T\boldsymbol{x})\right), \tag{6}$$

where $\mathbf{w}_{n,\ell} \in \mathbb{R}^{N_{\ell-1}}$ encodes the linear weights and $\sigma_{n,\ell} : \mathbb{R} \to \mathbb{R}$ denotes the neural activation function associated to any particular neuron indexed by (n, ℓ). When the network is fully connected, its training results from the optimization of a cost function with respect to the linear weights $\mathbf{w}_{n,\ell}$. The regularization is achieved via a functional that constrains the magnitude of the weights, for instance, $R_{\text{weights}}(\mathbf{f}_{\text{deep}}) = \sum_{\ell=1}^{L}\sum_{n=1}^{N_\ell} \|\mathbf{w}_{n,\ell}\|_2^2$, which is the squared ℓ_2-norm of all linear weights.

2.2. Learning the Activations

The standard paradigm in deep learning is to fix the shape of the neuronal responses to $\sigma_{n,\ell}(x) = \sigma(x - b_{n,\ell})$, where σ is the activation function (*e.g.*, sigmoid or ReLU), common to all neurons, and $b_{n,\ell}$ an adjustable bias.

Our proposal is to allow $\sigma_{n,\ell}$ to vary on a neuron-by-neuron basis and to optimize the shape of the activations during the training process, under the constraint that the linear weights are normalized. In the original work of [16], the training of the activation functions has been formulated by introducing an additional regularization term $R_{\text{neurons}}(\mathbf{f}_{\text{deep}}) = \sum_{\ell=1}^{L}\sum_{n=1}^{N_\ell} \mathrm{TV}^{(2)}(\sigma_{n,\ell})$, where $\mathrm{TV}^{(2)}(\cdot)$ is the second-order total-variation

$$\mathrm{TV}^{(2)}(\sigma) = \|\mathrm{D}^2\sigma\|_{\mathcal{M}} \tag{7}$$

and D is the derivative operator. Unser's representer theorem then states that the optimal solution is achieved with a deep spline network; that is, a network whose individual activation functions are adaptive and piecewise-linear.

While the above optimality result is a good starting point, it is not entirely suitable for our purpose because the second-order total-variation is only a semi-norm. The reason for this is that the null space of the second-order derivative D^2 is non-trivial, since it is composed of all polynomials of the form $b_1 + b_2x$. Our proposal here is to add an additional term to obtain a *bona fide* norm. This leads us to define the $\mathrm{BV}^{(2)}$-norm as

$$\|f\|_{\mathrm{BV}^{(2)}} \triangleq \mathrm{TV}^{(2)}(f) + |f(0)| + |f(1) - f(0)|. \tag{8}$$

The corresponding Banach space is

$$\mathrm{BV}^{(2)}(\mathbb{R}) = \{f : \mathbb{R} \to \mathbb{R} : \|f\|_{\mathrm{BV}^{(2)}} < \infty\}. \tag{9}$$

A fundamental property of $\mathrm{BV}^{(2)}(\mathbb{R})$ is stated in Proposition 1. It is a special case of [20, Theorem 5] with $\mathrm{L} = \mathrm{D}^2$.

Proposition 1. *For any function $f \in \mathrm{BV}^{(2)}(\mathbb{R})$, there exist a unique $w = \mathrm{D}^2 f \in \mathcal{M}(\mathbb{R})$, $b_1 = f(0), b_2 = (f(1) - f(0)) \in \mathbb{R}$ such that*

$$f(x) = \int_{\mathbb{R}} h(x, y)w(y)\mathrm{d}y + b_1 + b_2x, \tag{10}$$

where $h(\cdot, \cdot)$ is given as

$$h(x, y) = (x - y)_+ - (1 - x)(-y)_+ - x(1 - y)_+. \tag{11}$$

The $\mathrm{BV}^{(2)}$-norm of f then simplifies to

$$\|f\|_{\mathrm{BV}^{(2)}} = \|w\|_{\mathcal{M}} + |b_1| + |b_2|. \tag{12}$$

In this paper, we propose to learn the activation functions by constraining their $\mathrm{BV}^{(2)}$-norm. To avoid the propagation of scaling from one layer to the next, we also impose a scaling constraint on the linear weights. Specifically, we assume that the sup norm of the weight vectors $\mathbf{w}_{n,\ell}$ is normalized; i.e., $\|\mathbf{w}_{n,\ell}\|_\infty = 1$ for all neurons. Note that all the analysis in this paper remains valid for other norms as well since all norms are equivalent for a finite-dimensional space.

The global Lipschitz continuity of the neural network can only be ensured if each of its each individual neurons is Lipschitz-continuous over $\mathbb{R}$. Lemma 1 reveals the tight connection between the $\mathrm{BV}^{(2)}$-norm and the Lipschitz property at the elementary level of a scalar nonlinearity.

Lemma 1. *Any function $\sigma \in \mathrm{BV}^{(2)}(\mathbb{R})$ is Lipschitz-continuous with constant $C = \|\sigma\|_{\mathrm{BV}^{(2)}}$. Indeed, for any $x, y \in \mathbb{R}$, we have*

$$|\sigma(x) - \sigma(y)| \leq \|\sigma\|_{\mathrm{BV}^{(2)}} |x - y|. \tag{13}$$

Proof. The first step is to show that, for any $x_1, x_2 \in \mathbb{R}$,

$$\sup_{y \in \mathbb{R}} |h(x_1, y) - h(x_2, y)| \leq |x_1 - x_2|. \tag{14}$$

Hence,

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

□

This Lipschitz-continuity property implies that the members of $\mathrm{BV}^{(2)}(\mathbb{R})$ are continuous on $\mathbb{R}$ and differentiable almost everywhere. This is a minimal requirement for the activation functions of a neural network in order to be able to deploy the back-propagation algorithm in the training step.

3. LIPCHITZ BOUND FOR DEEP NEURAL NETWORKS

We now provide a global Lipschitz bound for the whole network that involves the $\mathrm{BV}^{(2)}$-norm of each neuron.

Theorem 1 (Lipschitz regularity of deep neural networks)**.** *Any feed-forward fully-connected deep neural network with the nonlinearity selected from the space $\mathrm{BV}^{(2)}(\mathbb{R})$ and normalized linear weights (with respect to the ℓ_∞-norm) specifies an input-output relation that is Lipschitz with respect to the ℓ_1-norm with constant*

$$C = \prod_{\ell=1}^{L} \left(\sum_{n=1}^{N_\ell} \|\sigma_{n,\ell}\|_{\mathrm{BV}^{(2)}} \right). \tag{15}$$

In other words, for all $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^{N_0}$, we have

$$\|\mathbf{f}_{\mathrm{deep}}(\boldsymbol{y}) - \mathbf{f}_{\mathrm{deep}}(\boldsymbol{x})\|_1 \leq C \|\boldsymbol{y} - \boldsymbol{x}\|_1. \tag{16}$$

Proof. Due to Lemma 1, we have

$$\begin{aligned} \left|\sigma_{n,\ell}(\mathbf{w}_{n,\ell}^T \boldsymbol{y}) - \sigma_{n,\ell}(\mathbf{w}_{n,\ell}^T \boldsymbol{x})\right| &\leq \|\sigma_{n,\ell}\|_{\mathrm{BV}^{(2)}} \left|\mathbf{w}_{n,\ell}^T (\boldsymbol{y} - \boldsymbol{x})\right| \\ &\leq \|\sigma_{n,\ell}\|_{\mathrm{BV}^{(2)}} \|\mathbf{w}_{n,\ell}\|_\infty \|\boldsymbol{y} - \boldsymbol{x}\|_1 \\ &= \|\sigma_{n,\ell}\|_{\mathrm{BV}^{(2)}} \|\boldsymbol{y} - \boldsymbol{x}\|_1, \end{aligned}$$

where the last step follows from the Hölder inequality. This yields a Lipschitz bound for the ℓth layer of the network, as

$$\|\mathbf{f}_\ell(\boldsymbol{y}) - \mathbf{f}_\ell(\boldsymbol{x})\|_1 \leq \left(\sum_{n=1}^{N_\ell} \|\sigma_{n,\ell}\|_{\mathrm{BV}^{(2)}} \right) \|\boldsymbol{y} - \boldsymbol{x}\|_1.$$

Now, by composing the layer inequalities, we obtain the announced Lipschitz bound.

□

Remark 1. *We can replace the ℓ_1-norm in* (16) *by any norm due to the norm-equivalence property of finite-dimensional vector spaces.*

Due to (15), the per-layer optimization of $\sum_{n=1}^{N_\ell} \mathrm{BV}(\sigma_{n,\ell})$ contributes to a decrease of the overall Lipchitz constant of the network. This is our main motivation for including such terms in the regularization functional.

4. DEEP-SPLINE REPRESENTER THEOREM

Let us now get back to our initial problem: the determination of the optimal set of parameters in (6)— the linear weight vectors $\mathbf{w}_{n,\ell}$ and the best shape of activation functions $\sigma_{n,\ell} : \mathbb{R} \to \mathbb{R}$—during the training of the neural network. Our strategy is to augment the usual cost functional by adding a new regularization term that controls the overall Lipchitz regularity of the neural network. To that end, we define our training problem as

$$\begin{aligned} \min_{\substack{\|\mathbf{w}_{n,\ell}\|_\infty = 1 \\ \sigma_{n,\ell} \in \mathrm{BV}^{(2)}(\mathbb{R})}} \sum_{m=1}^{M} E\big(\mathbf{y}_m, \mathbf{f}(\boldsymbol{x}_m)\big) + \mu \sum_{\ell=1}^{L} \sum_{n=1}^{N_\ell} R_\ell(\mathbf{w}_{n,\ell}) \\ + \lambda \sum_{\ell=1,}^{L} \left(\sum_{n=1}^{N_\ell} \|\sigma_{n,\ell}\|_{\mathrm{BV}^{(2)}} \right), \end{aligned} \tag{17}$$

where $E : \mathbb{R}^{N_L} \times \mathbb{R}^{N_L} \to \mathbb{R}_{\geq 0}$ is an arbitrary error function such that $E(\mathbf{y}, \mathbf{y}) = 0$ for any $\mathbf{y} \in \mathbb{R}^{N_L}$, $R_\ell : \mathbb{R}^{N_{\ell-1}} \to \mathbb{R}_{\geq 0}$ is some arbitrary cost that favors certain types of linear transformations, and $\lambda, \mu \in \mathbb{R}_{>0}$ are two adjustable regularization parameters.

Theorem 2 ($\mathrm{BV}^{(2)}$ optimality of deep splines)**.** *If the solution of* (17) *exists, then it is achieved by a deep spline network with individual activations of the form*

$$\sigma_{n,\ell}(x) = b_{1,n,\ell} + b_{2,n,\ell}x + \sum_{k=1}^{K_{n,\ell}} a_{k,n,\ell}\mathrm{ReLU}(x - \tau_{k,n,\ell}), \tag{18}$$

with adaptive parameters $K_{n,\ell} \leq M$*,* $\tau_{1,n,\ell}, \dots, \tau_{K_{n,\ell},n,\ell} \in \mathbb{R}$*, and* $b_{1,n,\ell}, b_{2,n,\ell}, a_{1,n,\ell}, \dots, a_{K_{n,\ell},n,\ell} \in \mathbb{R}$*.*

Proof. Consider an arbitrary solution of (17) with linear weights denoted by $\mathbf{w}^0_{n,\ell}$ and the non-linearities $\sigma^0_{n,\ell}$ for $\ell = 1, 2, \dots, L$ and $n = 1, 2, \dots, N_\ell$. We denote $s_{m,n,\ell}$ and $z_{m,n,\ell}$ as the input and output of the neuron (n, ℓ) respectively for the input vector $\boldsymbol{x}_m = (x_{m,1}, x_{m,2}, \dots, x_{m,N_0})$. More precisely, we set $z_{m,n,0} = x_{m,n}$ and, then, for $\ell = 1, 2, \dots, L$, we inductively define $\boldsymbol{z}_{m,\ell-1}$, $s_{m,n,\ell}$, and $z_{m,n,\ell}$ as

$$\begin{aligned}
\boldsymbol{z}_{m,\ell-1} &= (z_{m,1,\ell-1}, z_{m,2,\ell-1}, \dots, z_{m,N_{\ell-1},\ell-1}),\\
s_{m,n,\ell} &= \mathbf{w}^{0}_{n,\ell}{}^{T} \boldsymbol{z}_{m,\ell-1},\\
z_{m,n,\ell} &= \sigma^0_{n,\ell}(s_{m,n,\ell}).
\end{aligned}$$

Now, we define the auxiliary interpolation problem

$$\min_{\sigma \in \mathrm{BV}^{(2)}(\mathbb{R})} \mathrm{TV}^{(2)}(\sigma) \quad s.t. \tag{19}$$

$$\begin{cases} \sigma(s_{m,n,\ell}) = z_{m,n,\ell}, & m = 1, 2, \dots, M,\\ \sigma(x) = \sigma^0_{n,\ell}(x), & x \in \{0, 1\}. \end{cases} \tag{20}$$

for the neuron (n, ℓ) of the network.

First, we show that $\sigma^0_{n,\ell}$ is a solution of (19). Assume by contradiction that there exists an activation function $\tilde{\sigma}_{n,\ell} \in \mathrm{BV}^{(2)}(\mathbb{R})$ with $\mathrm{TV}^{(2)}(\tilde{\sigma}_{n,\ell}) < \mathrm{TV}^{(2)}(\sigma_{n,\ell})$ that satisfies the feasibility conditions (20). It then follows from the feasibility conditions $\tilde{\sigma}(x) = \sigma^0_{n,\ell}(x)$ for $x = 0, 1$ and (8) that $\|\tilde{\sigma}\|_{\mathrm{BV}^{(2)}(\mathbb{R})} < \|\sigma^0_{n,\ell}\|_{\mathrm{BV}^{(2)}(\mathbb{R})}$. Now, by replacing $\sigma^0_{n,\ell}$ by $\tilde{\sigma}_{n,\ell}$ in the neuron (n, ℓ), we obtain a new network that has a lesser cost (17) since, except for the $\mathrm{BV}^{(2)}$-norm of the neuron (n, ℓ), which has been decreased, all the other terms of the cost function remains unchanged. Having a new network with a lesser cost contradicts the optimality of the original network.

Now, since (19) has a solution, it fulfills the conditions of Lemma 1 of [16] which states that there always exists a solution of (19) that is a linear spline of the form

$$\sigma_{\mathrm{spline}}(x) = b_1 + b_2 x + \sum_{k=1}^{K} a_k \mathrm{ReLU}(x - \tau_k) \tag{21}$$

with $K \leq M$ knots, as it has $M + 2$ constraints. This yields the solution form (18) for the neuron (n, ℓ). By repeating this procedure for all neurons, we obtain the optimal form of all the activations as desired. □

The main outcome of our new theorem is that the optimal architecture is a deep spline network where the action of each individual neuron is encoded by a linear spline. The nontrivial part is that these splines are *adaptive*, meaning that the number of knots $K_{n,\ell}$ associated to each neuron (n, ℓ), as well as their location $\tau_{k,n,\ell}$, is unknown a priori.

The elegant and encouraging aspect of the solution form (18) is that the standard ReLU architecture is included as a particular (and minimalistic) case with $K_{n,\ell} = 1$. However, Theorem 2 also calls for a novel optimization challenge: the optimal allocation and determination of the spline knots.

Due to $\mathrm{D}^2\{\mathrm{ReLU}(\cdot - \tau_k)\} = \delta(\cdot - \tau_k)$, we have

$$\|\sigma_{\mathrm{spline}}\|_{\mathrm{BV}^{(2)}} = \sum_{k=1}^{K} |a_k| + |b_1| + |b_2| = \|\mathbf{a}\|_1 + \|\mathbf{b}\|_1,$$

which connects our framework with ℓ_1-minimization techniques. In addition, we have $\sigma_{\mathrm{spline}} \in \mathrm{BV}^{(2)}(\mathbb{R}) \Leftrightarrow \|\mathbf{a}\|_1 + \|\mathbf{b}\|_1 < \infty$.

We also note that the present solution is very similar to the one reported in [16] for $\mathrm{TV}^{(2)}$ regularization. The fundamental difference is the inclusion of the coefficients $b_{1,n,\ell}$ and $b_{2,n,\ell}$ in the regularization, which is essential for controlling the Lipchitz regularity of the network. Otherwise, there is the risk of having the Lipchitz constant grow without bounds because of a lack of penalty on the linear part of the solution; *i.e.*, the term $b_{1,n,\ell} + b_{2,n,\ell}x$, which is present in both scenarios. Mathematically, the price to pay for switching from $\mathrm{TV}^{(2)}$ to $\mathrm{BV}^{(2)}$ is a slight loss in the sharpness of the sparsity bound: $K_{n,\ell} \leq M$, as opposed to $K_{n,\ell} \leq (M - 2)$ in [16].

5. CONCLUSION

In this paper, we have proposed a variational framework for optimizing the activation functions of deep neural networks. The main motivation is to learn activations that are smooth and “sparse”, while controlling their Lipschitz regularity. We have proposed the $\mathrm{BV}^{(2)}$-norm as a suitable candidate. It gives an upper bound to the Lipschitz constant of the global network. We have proved that the solution of this variational problem is a deep spline network that has piecewise linear activation functions. They can be expressed as a linear combination of ReLU functions with a linear additive term. The $\mathrm{BV}^{(2)}$-norm also enforces ℓ_1 regularization on the expansion coefficients, which supports the idea of imposing sparsity in the network. The next step of our research is to design practical algorithms that optimally allocate and determine the spline knots and to investigate the potential performance gain of such architectures.

6. REFERENCES

[1] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

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

[3] T. Poggio and F. Girosi, "Networks for approximation and learning," *Proceedings of the IEEE*, vol. 78, pp. 1481–1497, September 1990.

[4] B. Schölkopf, K.-K. Sung, C. J. C. Burges, F. Girosi, P. Niyogi, T. Poggio, and V. Vapnik, "Comparing support vector machines with Gaussian kernels to radial basis function classifiers," *IEEE Transactions on Signal Processing*, vol. 45, pp. 2758–2765, November 1997.

[5] B. Schölkopf, R. Herbrich, and A. J. Smola, "A generalized representer theorem," in *International Conference on Computational Learning Theory*, pp. 416–426, Springer, 2001.

[6] V. Vapnik, *Statistical Learning Theory*. Wiley, New York, 1998.

[7] T. Evgeniou, M. Pontil, and T. Poggio, "Regularization networks and support vector machines," *Advances in Computational Mathematics*, vol. 13, pp. 1–50, April 2000.

[8] V. Roth, "The generalized LASSO," *IEEE Transactions on Neural Networks*, vol. 15, pp. 16–28, January 2004.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, pp. 1097–1105, 2012.

[10] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 234–241, Springer, 2015.

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

[12] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, vol. 1. MIT press Cambridge, 2016.

[13] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pp. 315–323, 2011.

[14] G. F. Montufar, R. Pascanu, K. Cho, and Y. Bengio, "On the number of linear regions of deep neural networks," in *Advances in Neural Information Processing Systems*, pp. 2924–2932, 2014.

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

[16] M. Unser, "A representer theorem for deep neural networks," *arXiv preprint arXiv:1802.09210*.

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

[18] 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 Transactions on Medical Imaging*, vol. 37, pp. 1440–1453, June 2018.

[19] B. Neyshabur, S. Bhojanapalli, D. McAllester, and N. Srebro, "Exploring generalization in deep learning," in *Advances in Neural Information Processing Systems*, pp. 5947–5956, 2017.

[20] M. Unser, J. Fageot, and J. P. Ward, "Splines are universal solutions of linear inverse problems with generalized TV regularization," *SIAM Review*, vol. 59, pp. 769–793, November 2017.