The Role of Neural Network Activation Functions

Rahul Parhi and Robert D. Nowak, *Fellow, IEEE*

***Abstract*—A wide variety of activation functions have been proposed for neural networks. The Rectified Linear Unit (ReLU) is especially popular today. There are many practical reasons that motivate the use of the ReLU. This paper provides new theoretical characterizations that support the use of the ReLU, its variants such as the leaky ReLU, as well as other activation functions in the case of univariate, single-hidden layer feedforward neural networks. Our results also explain the importance of commonly used strategies in the design and training of neural networks such as "weight decay" and "path-norm" regularization, and provide a new justification for the use of "skip connections" in network architectures. These new insights are obtained through the lens of spline theory. In particular, we show how neural network training problems are related to infinite-dimensional optimizations posed over Banach spaces of functions whose solutions are well-known to be fractional and polynomial splines, where the particular Banach space (which controls the order of the spline) depends on the choice of activation function.**

***Index Terms*—Neural networks, regularization, activation functions, inverse problems.**

I. Introduction

VARIANTS of the well-known universal approximation theorem for neural networks state that *any* continuous function can be approximated arbitrarily well by a single-hidden layer neural network, under mild conditions on the activation function [1]–[5]. While such results show that most nonlinear activation functions suffice for universal approximation in the ultra-wide limit, it is clear that the sequence of approximating functions, as well as the nature of functions learned by fitting networks to data, depends strongly on the choice of activation. Recent work on the approximation theory of neural networks has characterized how approximation rates depend on the choice of activation function [6], [7]. However, these results do not consider the *practical problem* of understanding the properties of functions *learned* by neural networks fit to data. In this paper, we consider this problem in the univariate, single-hidden layer case.

As neural networks provide a rich space of functions, learning with neural networks is inherently ill-posed. Thus, regularization plays an important role in neural network training. One of the most common regularizers is *weight decay* [8], which corresponds to the regularizer being the Euclidean norm of the network weights. Regularization is popular since neural networks trained with regularization often generalize well on new, unseen data [9]–[11].

In this paper we show how regularization in the finite-dimensional space of *neural network parameters* is actually the same as regularization in the infinite-dimensional *space of functions*. In particular, we show how training neural networks with appropriate regularization results in functions that are solutions to an infinite-dimensional *variational problem* posed over functions, where the regularizer is then a seminorm defining a Banach space that depends on the choice of activation function. We consider univariate, single-hidden layer feedforward neural networks mapping $\mathbb{R} \to \mathbb{R}$ of the form

$$x \mapsto \sum_{k=1}^{K} v_k \, \rho(w_k x - b_k) + c(x), \tag{1}$$

where $\rho : \mathbb{R} \to \mathbb{R}$ is a fixed *activation function*, K is the *width* of the network, for $k = 1, \ldots, K$, $v_k, w_k \in \mathbb{R}$, $w_k \neq 0$ are the *weights* and $b_k \in \mathbb{R}$ are the first layer *biases*, and $c(\cdot)$ is a "generalized bias"[1] term in the last layer.

Our results rely on the key observation that in the univariate case, single-hidden layer neural networks are essentially *spline functions*. Indeed, a spline function admits a representation

$$x \mapsto \sum_{k=1}^{K} v_k \, \rho(x - b_k) + c(x). \tag{2}$$

The key difference between (1) and (2) is that the atoms of the neural network are *translates* and *dilates* of the activation function, while the atoms of the spline are *only translates* of the "activation function". To this end, we use tools from the recently developed variational framework of L-splines [12], to show that single-hidden layer neural networks trained with appropriate regularization are solutions to certain variational inverse problems. The dilations by input layer weights play a key role in the design of the neural network regularizers.

A. Contributions

In this paper we introduce the notion of *admissible* activation functions. Roughly speaking, these are activation functions that allow for a rigorous connection between conventional neural network training and variational problems over an associated Banach space. Common activation functions such as the popular Rectified Linear Unit (ReLU) and modifications such as the leaky ReLU [13], are admissible and thus each is associated with its particular Banach space.

We instantiate our main result and show that training single-hidden layer neural networks with particular *power activation*

Manuscript received August 2, 2020; revised September 18, 2020; accepted September 24, 2020. Date of publication September 29, 2020; date of current version October 14, 2020. This work was supported in part by the AFOSR/AFRL under Grant FA9550-18-1-0166 and in part by the NSF Research Traineeship (NRT) under Grant 1545481. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Victor Sanchez. *(Corresponding author: Rahul Parhi.)*

The authors are with the Department of Electrical and Computer Engineering, University of Wisconsin–Madison, Madison 53706 USA (e-mail: rahul@ece.wisc.edu; rdnowak@wisc.edu).

Digital Object Identifier 10.1109/LSP.2020.3027517

[1]We will later see that $c(\cdot)$ corresponds to a "simple" function, e.g., a low degree polynomial, which depends on the activation function.

functions, introduced in Example 11, which include the ReLU and the leaky ReLU, and appropriate weight regularization produce *optimal fractional and polynomial splines* fits to the data. In other words, neural network training solves infinite-dimensional optimizations over the Banach spaces of functions of higher order bounded variation. Crucially, the regularizers are variants of the well-known path-norm [14] and weight decay [8] regularizers that are "matched" to the activation function. We also show that admissible activation functions are necessarily these power activation functions.

Furthermore, for activation functions such as the ReLU and leaky ReLU, the generalized bias term exactly corresponds to the well-known notion of *skip connections* [15] and thus our result also provides theoretical insight into the use of skip connections in neural network architectures. Finally, another interesting result of this paper is that it suffices to simply train a (sufficiently wide) neural network to solve certain variational inverse problems as opposed to more standard multiresolution or grid-based approaches [16], [17].

B. Related work

The choice of activation function plays an important role in the efficacy of neural networks. While the traditional sigmoid activation function was used for many years, the ReLU activation has become the preferred choice. Its initial motivation was to promote sparsity (in the sense of decreasing the number of active neurons) [18]. It has also been empirically observed that the training of neural networks is much faster with ReLU activations [19]. Furthermore, variants of the ReLU, such as the leaky ReLU [13], have been proposed to avoid the problem of *vanishing gradients* in neural network training.

More recently, several recent works have made connections between splines and neural networks. In particular, the authors of [20] show that the "connect-the-dots" linear spline is a solution to the problem of training a single-hidden layer ReLU network with weight decay subject to data fitting constraints. Another related, but different work, is concerned with the "optimal shaping" of activation functions in *deep neural networks* [21], [22] in which the authors consider *learnable activation functions* and show that linear spline activation functions satisfy a minimal second-order total variation criterion. In our own work in [23], we relate neural network training to a variational problem over a Banach space in the multivariate case. We remark that in the univariate case explored in this paper, a much broader class of activation functions are *admissible*. This is discussed further in Remark 17.

II. Preliminaries

Let $\mathscr{S}(\mathbb{R})$ be the Schwartz space of smooth and rapidly decaying test functions on $\mathbb{R}$ with continuous dual $\mathscr{S}'(\mathbb{R})$, the space of tempered distributions on $\mathbb{R}$. We will be interested in the space $\mathcal{M}(\mathbb{R})$ of finite Radon measures on $\mathbb{R}$. The space $\mathcal{M}(\mathbb{R})$ can be viewed as a subspace of $\mathscr{S}'(\mathbb{R})$ with the norm

$$\|u\|_{\mathcal{M}(\mathbb{R})} = \sup_{\varphi\in\mathscr{S}(\mathbb{R}),\|\varphi\|_{L^\infty(\mathbb{R})}=1} \langle u, \varphi\rangle,$$

which is exactly the *total variation norm in the sense of measures*. We are interested in $\mathcal{M}(\mathbb{R})$ since it is a "generalization" of $L^1(\mathbb{R})$. Indeed, we have $L^1(\mathbb{R}) \subset \mathcal{M}(\mathbb{R})$ and for any $f \in L^1(\mathbb{R})$ we have $\|f\|_{L^1(\mathbb{R})} = \|f\|_{\mathcal{M}(\mathbb{R})}$, but the translated Dirac impulses $\delta(\cdot - x_0)$, $x_0 \in \mathbb{R}$, are not in $L^1(\mathbb{R})$ but are in $\mathcal{M}(\mathbb{R})$ with $\|\delta(\cdot - x_0)\|_{\mathcal{M}(\mathbb{R})} = 1$.

We will now state the relevant background from the framework of L-splines [12].

Definition 1 (Definition 1 of [12]): A linear operator $\mathrm{L} : \mathscr{S}'(\mathbb{R}) \to \mathscr{S}'(\mathbb{R})$ is called *spline-admissible* if

1) it is translation-invariant, i.e., $\mathrm{L}\,\mathfrak{T}_{x_0} = \mathfrak{T}_{x_0}\,\mathrm{L}$, where $\mathfrak{T}_{x_0}\{f\}(x) = f(x - x_0)$ is the *translation operator*;
2) there exists a function $rho_{\mathrm{L}} : \mathbb{R} \to \mathbb{R}$ such that $\mathrm{L}\rho_{\mathrm{L}} = \delta$, i.e., ρ_{L} is a Green's function of L;
3) the null space $\mathcal{N}_{\mathrm{L}} = \{q : \mathrm{L}q = 0\}$ has finite-dimension $N_0 \geq 0$.

Definition 2 (Definition 2 of [12]): A function $s : \mathbb{R} \to \mathbb{R}$ is said to be a *nonuniform* L-*spline* if

$$\mathrm{L}\{s\} = \sum_{k=1}^{K} v_k\, \delta(\cdot - b_k),$$

where $\{v_k\}_{k=1}^K$ is a sequence of weights and the locations of Dirac impulses are at the spline knots $\{b_k\}_{k=1}^K$.

Remark 3: Notice that the spline representation in (2) with ρ being a Green's function of L is clearly a nonuniform L-spline, so long as $c(\cdot) \in \mathcal{N}_{\mathrm{L}}$. The finite-dimensionality is required in Definition 2, so that $c(\cdot)$ can be represented by a finite number of coefficients. We refer to the representation in (2) as the *canonical spline representation*.

The fundamental result of [12] is the following *representer theorem* regarding the structure of the solutions to variational problems with generalized total variation regularization.

Proposition 4 (Based on Theorems 1 and 2 of [12]): Let L be a spline-admissible operator in the sense of Definition 1. Then, the extreme points of the solutions of

$$\min_{f\in\mathcal{M}_{\mathrm{L}}(\mathbb{R})} \|\mathrm{L}f\|_{\mathcal{M}(\mathbb{R})} \quad \text{s.t.} \quad \langle \nu_n, f\rangle = y_n,\ n = 1, \ldots, N \tag{3}$$

are necessarily non-uniform L-splines of the form in (2) with the $K \leq N - N_0$ knots, where ρ is a Green's function of L and $c(\cdot) \in \mathcal{N}_{\mathrm{L}}$, $\boldsymbol{\nu} : f \mapsto (\langle \nu_1, f\rangle, \ldots, \langle \nu_N, f\rangle) \in \mathbb{R}^N$ is a weak*-continuous *measurement operator*, and $\mathcal{M}_{\mathrm{L}}(\mathbb{R})$ is the *native space* of L defined by $\mathcal{M}_{\mathrm{L}}(\mathbb{R}) := \{f \in \mathscr{S}'(\mathbb{R}) : \mathrm{L}f \in \mathcal{M}(\mathbb{R})\}$.

Remark 5: For appropriate choices of loss function,[2] the result of Proposition 4 also holds for *regularized problems*:

$$\min_{f\in\mathcal{M}_{\mathrm{L}}(\mathbb{R})} \sum_{n=1}^{N} \ell(y_n, \langle \nu_n, f\rangle) + \lambda\|\mathrm{L}f\|_{\mathcal{M}(\mathbb{R})} \tag{4}$$

where $\ell(\cdot,\cdot)$ is the loss function and $\lambda > 0$ is an adjustable regularization parameter.

Remark 6: In machine learning, the measurement model is taken to be *ideal sampling*, i.e., $\nu_n = \delta(\cdot - x_n)$ for some $x_n \in \mathbb{R}$. In other words, the machine learning problem considers fitting the data $\{(x_n, y_n)\}_{n=1}^N \subset \mathbb{R} \times \mathbb{R}$. In the rest of this paper, we will only be interested in this setting. A sufficient condition for weak*-continuity of $\delta(\cdot - x_n)$ is continuity of the Green's function of L. For a detailed proof in the case that $\mathrm{L} = \mathrm{D}^2$, the second derivative operator, see [21, Theorem 1].

[2] A strictly convex, coercive, lower semi-continuous loss function suffices.

III. Neural Network Training and Regularization

In this section we will state our main results.

Definition 7: A linear operator $\mathrm{L}: \mathscr{S}'(\mathbb{R}) \to \mathscr{S}'(\mathbb{R})$ is called *neural network-admissible* if

1) it is spline-admissible in the sense of Definition 1 with a continuous[3] Green's function;
2) there exists $g: \mathbb{R} \to \mathbb{R}$ such that $\mathrm{L}\,\mathfrak{D}_w = g(w)\,\mathfrak{D}_w\,\mathrm{L}$, where $\mathfrak{D}_w\{f\}(x) := f(wx)$ is the *dilation operator*.

Definition 8: An activation function $\rho: \mathbb{R} \to \mathbb{R}$ is called *admissible* if it is the continuous Green's function of some neural network-admissible operator.

We see that single-hidden layer neural networks with admissible activation functions are in fact splines. Indeed, let ρ be an admissible activation function for the neural network-admissible operator L. Then, consider the neural network

$$f_{\boldsymbol{\theta}}(x) = \sum_{k=1}^{K} v_k\, \rho(w_k x - b_k) + c(x), \tag{5}$$

where $\boldsymbol{\theta} = (v_1, \ldots, v_K, w_1, \ldots, w_K, b_1, \ldots, b_K, c)$ contains the neural network parameters and $c(\cdot) \in \mathcal{N}_{\mathrm{L}}$. Also, let Θ be the space of all neural network parameters $\boldsymbol{\theta}$. We see that

$$\begin{aligned}
\mathrm{L}\{f_{\boldsymbol{\theta}}\} &= \sum_{k=1}^{K} v_k (\mathrm{L}\,\mathfrak{D}_{w_k})\{\rho(\cdot - b_k/w_k)\} \\
&= \sum_{k=1}^{K} v_k\, g(w_k) (\mathfrak{D}_{w_k}\,\mathrm{L})\{\rho(\cdot - b_k/w_k)\} \\
&= \sum_{k=1}^{K} v_k\, g(w_k)\, \delta(w_k(\cdot) - b_k) \\
&= \sum_{k=1}^{K} v_k \frac{g(w_k)}{|w_k|}\, \delta(\cdot - b_k/w_k)
\end{aligned} \tag{6}$$

where in the last line we used the fact that the Dirac impulse is homogeneous of degree -1. From Definition 2, we see from (6) that $f_{\boldsymbol{\theta}}$ is an L-spline with spline knots at $\{b_k/w_k\}_{k=1}^{K}$. Thus, we see that although the neural network representation is not the canonical spline representation, neural networks, with admissible activation functions, are in fact splines. By Proposition 4, this says that they are solutions to variational problems of the form in (3). We can now state our main result.

Theorem 9: Let L be a neural network-admissible in the sense of Definition 7, and let ρ be a continuous Green's function of L. Then, the solutions to

$$\min_{\boldsymbol{\theta} \in \Theta} \sum_{k=1}^{K} |v_k| \frac{|g(w_k)|}{|w_k|} \quad \text{s.t.} \quad f_{\boldsymbol{\theta}}(x_n) = y_n,\ n = 1, \ldots, N$$

with $K \geq N - N_0$ are solutions to the variational problem in (3) under the ideal sampling setting.

Proof: Consider a neural network as in (5) and assume it is in *reduced form*, i.e., the weight bias pairs (w_k, b_k) are unique. The theorem follows by taking the $\|\cdot\|_{\mathcal{M}(\mathbb{R})}$ of (6). ∎

Remark 10: Just as in Remark 5, Theorem 9 also holds for regularized problems similar to (4).

[3] See Remark 6.

Example 11: Consider the activation function defined by

$$\rho_{\alpha,\beta,\gamma}(x) := \begin{cases} \alpha\, x^{\gamma-1}, & x < 0, \\ \beta\, x^{\gamma-1}, & x \geq 0, \end{cases} \tag{7}$$

where $\alpha, \beta \in \mathbb{R}$ with $\alpha \neq \beta$ and $\gamma \geq 1$. We refer to this as an (α, β, γ)-*power activation function*, and refer to γ as the *order* of the activation function. This family of activation functions are admissible with corresponding operator being D^{γ}, the γth-order derivative operator, since, up to a constant factor, $\rho_{\alpha,\beta,\gamma}$ is a Green's function of D^{γ}. When γ is not an integer, D^{γ} is understood as the Fourier multiplier $\omega \mapsto (\mathrm{i}\omega)^{\gamma}$. In this case, $g(w) = w^{\gamma}$. Hence, the corresponding regularizer is

$$\sum_{k=1}^{K} |v_k| \frac{|g(w_k)|}{|w_k|} = \sum_{k=1}^{K} |v_k| |w_k|^{\gamma-1}, \tag{8}$$

which can be viewed as a generalized ℓ^1-path-norm regularizer [14] that is "matched" to the activation function. This path-norm is also an upper bound on the Rademacher complexity of neural neural networks [23]; thus networks with small path-norms have better generalization bounds.

Theorem 12: An admissible activation function necessarily takes the form in (7).

Proof: From Item 2 in Definition 7, we see that an admissible activation function $\rho: \mathbb{R} \to \mathbb{R}$ must satisfy

$$\rho(wx) = g(w)\rho(\mathrm{sgn}(w)x) \tag{9}$$

for some $g: \mathbb{R} \to \mathbb{R}$. Put $P(x) := \ln \rho(e^x)$. For any $h \in \mathbb{R}$,

$$\begin{aligned}
P(x+h) &= \ln \rho(e^{x+h}) = \ln \rho(e^h e^x) \\
&= \ln\{g(e^h)\rho(e^x)\} = \ln g(e^h) + P(x),
\end{aligned}$$

where in the second line we used the fact that $e^h > 0$ for all $h \in \mathbb{R}$. Next, fix $h \in \mathbb{R} \setminus \{0\}$ and consider the finite difference

$$\Delta_h\{P\}(x) := \frac{P(x+h) - P(x)}{h} = \frac{\ln g(e^h)}{h}.$$

Since the finite difference is independent of x, we see that P is piecewise linear. Consider an interval $I \subset \mathbb{R}$ in which $P(x) = ax + b$ for all $x \in I$ for some $a, b \in \mathbb{R}$. Then, for all $x \in I$ we have

$$\rho(x) = e^{P(\ln x)} = e^{a \ln x + b} = e^b x^a.$$

Finally, by Definition 7, ρ must be spline-admissible and must satisfy (9). It follows that ρ must take the form in (7). ∎

Remark 13: When γ is not an integer, the functions learned by networks with $\rho_{\alpha,\beta,\gamma}$ activation functions trained on data and regularized according to (8) are optimal γ*th-order fractional splines* [24] fit to the data. When γ is an integer, the learned functions are optimal γ*th-order polynomial splines*.

Example 14: When $(\alpha, \beta, \gamma) = (0, 1, 2)$, we have $\rho_{0,1,2} = \max\{0, \cdot\}$ which is exactly the ReLU. The generalized bias term takes the form of a skip connection, i.e., $c(x) = ux + s$, where $u, s \in \mathbb{R}$ are trainable parameters. Additionally, the regularizer in (8) is exactly the ℓ^1-path-norm regularizer proposed in [14]. This same result holds for modifications of the ReLU such as the leaky ReLU [13], which is a $(\alpha, 1, 2)$-power activation function. When trained on data, these networks learn functions that are optimal with respect to the Banach space of functions of second-order bounded variation which are *optimal linear splines* fit to the data.

Remark 15: The leaky ReLU was proposed in order to avoid the *dying ReLU problem* in the training neural networks, where weights get stuck at 0 due to the fact that the ReLU is 0 for all inputs less than 0. Since our result says that the underlying function spaces for the ReLU and leaky ReLU are the same, perhaps the leaky ReLU should be used over the ReLU.

Example 16: The truncated power functions given by $\rho_{0,1,\gamma} \propto \max\{0, \cdot\}^{\gamma-1}/(\gamma-1)!$, where γ is a positive integer, are admissible. The generalized bias term takes the form of a polynomial of degree less than γ, with trainable coefficients, which can be viewed as a generalized skip connection.

Remark 17: In our related work in [23] we consider a similar problem to this paper, but in the multivariate case and relate training multivariate single-hidden layer networks to a variational problem over a Banach space. Our result there is more restrictive in that the only admissible activation functions are power activation functions where γ is a postive even integer, and also does not make any connections to splines.

Remarkably, as noticed in [23], is that the regularizer as in (8) is related to the well-known weight-decay regularizer [8].

Proposition 18 (Special case of Proposition 2.13 of [23]): Consider training neural networks as in (5) with an admissible activation function of order γ. Then, the following optimization problems are equivalent:

$$\min_{\boldsymbol{\theta}\in\Theta} \sum_{k=1}^{K} |v_k||w_k|^{\gamma-1} \quad \text{s.t.} \quad f_{\boldsymbol{\theta}}(x_n) = y_n,\ n = 1, \ldots, N$$

$$\min_{\boldsymbol{\theta}\in\Theta} \frac{1}{2}\sum_{k=1}^{K} |v_k|^2 + |w_k|^{2\gamma-2} \quad \text{s.t.} \quad f_{\boldsymbol{\theta}}(x_n) = y_n,\ n = 1, \ldots, N$$

Remark 19: These optimizations are also equivalent in the case of regularized problems similar to (4).

Remark 20: When $\gamma = 2$, the second optimization in Proposition 18 is exactly the well-known weight decay regularizer. Thus, ReLU networks and leaky ReLU networks are *intrinsically tied* to the well-known weight decay regularizer.

IV. Empirical Validation

In this section we verify empirically that the claims made in Section III hold. We use Proposition 18 and consider regularized neural network training problems of the form

$$\min_{\boldsymbol{\theta}\in\Theta} \sum_{n=1}^{N} |y_n - f_{\boldsymbol{\theta}}(x_n)|^2 + \frac{\lambda}{2}\sum_{k=1}^{K} |v_k|^2 + |w_k|^{2\gamma-2}. \quad (10)$$

To promote interpolation of the data we take $\lambda = 10^{-5}$. We specifically consider the ReLU activation which is a power activation function with $(\alpha, \beta, \gamma) = (0, 1, 2)$ and the cubic truncated power activation which is a power activation function with $(\alpha, \beta, \gamma) = (0, 1, 4)$. PyTorch was used to implement the networks and AdaGrad [25] to train the networks.

In Fig. 1, we trained a width $K = 200$ ReLU network according to (10) ($\gamma = 2$) and a width $K = 200$ cubic truncated power function network according to (10) ($\gamma = 4$). The choice of $K = 200$ was chosen so that the networks are sufficiently wide according to Theorem 9. We compare the learned functions to the standard linear and cubic splines.[4] We also illustrate the importance of regularization by also training the networks without regularization and show that they do not learn the optimal spline interpolations of the data. Indeed, we see in Fig. 1(c) that there are extra "bumps" between the first and second data point and between the second and third data point, and we see in Fig. 1(f) that there is an extra "bump" between the first and second data point. While the function learned in Fig. 1(b) is not the connect-the-dots linear spline, we see that it has the same second-order total variation and is hence a minimizer to the variational problem.

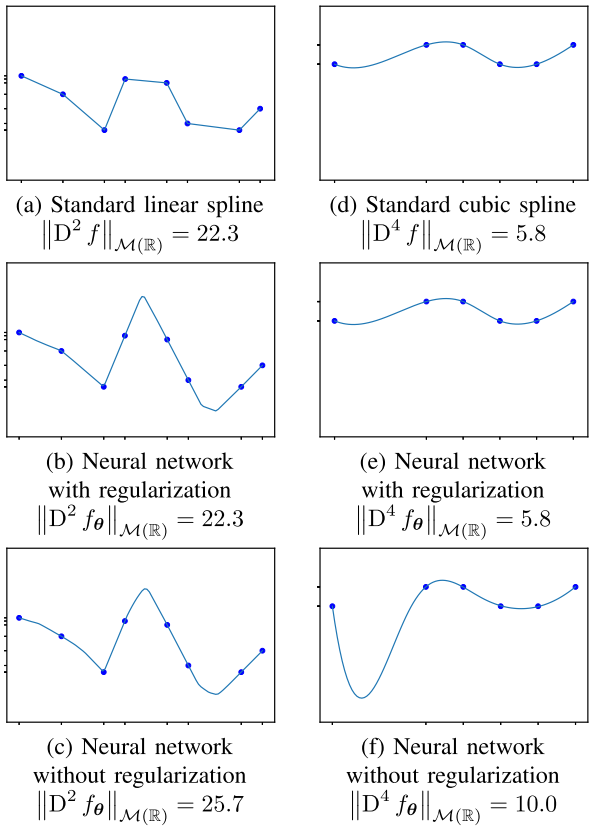

Fig. 1. In (a) (resp. (c)) we have the standard linear (resp. cubic) spline of the data. In (b) (resp. (e)) we have a ReLU (resp. cubic truncated power function) network with $K = 200$ neurons trained with regularization according to (10). In (c) (resp. (f)) we illustrate the importance of regularization. All figures plot the function (spline or neural network) vs. the input. The dots are the data.

V. Conclusion & Future Work

Using tools from the variational framework of L-splines, we have shown that the choice of activation implicitly defines a neural network regularizer that corresponds to a seminorm that defines a Banach space. We showed that the resulting neural network regularizers are related to the well-known path-norm and weight decay regularizers. Finally, we verified our results with empirical validation by showing that trained neural networks are optimal splines fit to data. Understanding the functional characteristics of deep neural networks trained on data is an open question.

Acknowledgment

The authors would like to thank Jordan Ellenberg for suggesting the simple argument that appears in Theorem 12.

[4]The standard splines were computed using SciPy.

REFERENCES

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

[2] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Netw.*, vol. 2, no. 5, pp. 359–366, 1989.

[3] K.-I. Funahashi, "On the approximate realization of continuous mappings by neural networks," *Neural Netw.*, vol. 2, no. 3, pp. 183–192, 1989.

[4] A. R. Barron, "Universal approximation bounds for superpositions of a sigmoidal function," *IEEE Trans. Inf. Theory*, vol. 39, no. 3, pp. 930–945, May 1993.

[5] M. Leshno, V. Y. Lin, A. Pinkus, and S. Schocken, "Multilayer feedforward networks with a nonpolynomial activation function can approximate any function," *Neural Netw.*, vol. 6, no. 6, pp. 861–867, 1993.

[6] J. M. Klusowski and A. R. Barron, "Approximation by combinations of ReLU and squared ReLU ridge functions with ℓ^1 and ℓ^0 controls," *IEEE Trans. Inf. Theory*, vol. 64, no. 12, pp. 7649–7656, Dec. 2018.

[7] H. N. Mhaskar, "Dimension independent bounds for general shallow networks," *Neural Netw.*, vol. 123, pp. 142–152, 2020.

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

[9] J. E. Moody, "The effective number of parameters: An analysis of generalization and regularization in nonlinear learning systems," in *Proc. Adv. Neural Inf. Process. Syst.*, 1992, pp. 847–854.

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

[11] C. Wei, J. D. Lee, Q. Liu, and T. Ma, "Regularization matters: Generalization and optimization of neural nets vs their induced kernel," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 9709–9721.

[12] 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, 2017.

[13] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. Int. Conf. Mach. Learn.*, 2013, p. 3.

[14] B. Neyshabur, R. R. Salakhutdinov, and N. Srebro,"Path-SGD: Path-normalized optimization in deep neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 2422–2430.

[15] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.

[16] H. Gupta, J. Fageot, and M. Unser, "Continuous-domain solutions of linear inverse problems with Tikhonov versus generalized TV regularization," *IEEE Trans. Signal Process.*, vol. 66, no. 17, pp. 4670–4684, Sep. 2018.

[17] T. Debarre, J. Fageot, H. Gupta, and M. Unser, "B-spline-based exact discretization of continuous-domain inverse problems with generalized TV regularization," *IEEE Trans. Inf. Theory*, vol. 65, no. 7, pp. 4457–4470, Jul. 2019.

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

[19] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[20] P. H. P. Savarese, I. Evron, D. Soudry, and N. Srebro, "How do infinite width bounded norm networks look in function space?," in *Proc. Conf. Learn. Theory*, 2019, pp. 2667–2690.

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

[22] S. Aziznejad and M. Unser, "Deep spline networks with control of Lipschitz regularity," in *Proc. ICASSP IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2019, pp. 3242–3246.

[23] R. Parhi and R. D. Nowak, "Neural networks, ridge splines, and TV regularization in the Radon domain," 2020, *arXiv:2006.05626v1*.

[24] M. Unser and T. Blu, "Fractional splines and wavelets," *SIAM Rev.*, vol. 42, no. 1, pp. 43–67, 2000.

[25] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *J. Mach. Learn. Res.*, vol. 12, pp. 2121–2159, 2011.