# RIDGELESS INTERPOLATION WITH SHALLOW RELU NETWORKS IN $1D$ IS NEAREST NEIGHBOR CURVATURE EXTRAPOLATION AND PROVABLY GENERALIZES ON LIPSCHITZ FUNCTIONS

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

We prove a precise geometric description of all one layer ReLU networks $z(x;\theta)$ with a single linear unit and input/output dimensions equal to one that interpolate a given dataset $\boldsymbol{D} = \{(x_i, f(x_i))\}$ and, among all such interpolants, minimize the $\ell_2$-norm of the neuron weights. Such networks can intuitively be thought of as those that minimize the mean-squared error over $\boldsymbol{D}$ plus an infinitesimal weight decay penalty. We therefore refer to them as ridgeless ReLU interpolants. Our description proves that, to extrapolate values $z(x;\theta)$ for inputs $x \in (x_i, x_{i+1})$ lying between two consecutive datapoints, a ridgeless ReLU interpolant simply compares the signs of the discrete estimates for the curvature of $f$ at $x_i$ and $x_{i+1}$ derived from the dataset $\boldsymbol{D}$. If the curvature estimates at $x_i$ and $x_{i+1}$ have different signs, then $z(x;\theta)$ must be linear on $(x_i, x_{i+1})$. If in contrast the curvature estimates at $x_i$ and $x_{i+1}$ are both positive (resp. negative), then $z(x;\theta)$ is convex (resp. concave) on $(x_i, x_{i+1})$. Our results show that ridgeless ReLU interpolants achieve the best possible generalization for learning $1d$ Lipschitz functions, up to universal constants.

## 1 INTRODUCTION

The ability of overparameterized neural networks to simultaneously fit data (i.e. interpolate) and generalize to unseen data (i.e. extrapolate) is a robust empirical finding that spans the use of deep learning in tasks from computer vision Krizhevsky et al. (2012); He et al. (2016), natural language processing Brown et al. (2020), and reinforcement learning Silver et al. (2016); Vinyals et al. (2019); Jumper et al. (2021). This observation is surprising when viewed from the lens of traditional learning theory Vapnik & Chervonenkis (1971); Bartlett & Mendelson (2002), which advocates for capacity control of model classes and strong regularization to avoid overfitting.

Part of the difficulty in explaining conceptually why neural networks are able to generalize is that it is unclear how to understand, concretely in terms of the network function, various forms of implicit and explicit regularization used in practice. For example, a well-chosen initialization for gradient-based optimizers can strongly impact for quality of the resulting learned network Mishkin & Matas (2015); He et al. (2015); Xiao et al. (2018). However, the specific geometric or analytic properties of the learned network ensured by a successful initialization scheme are hard to pin down.

In a similar vein, it is standard practice to experiment with (weak) explicit regularizers such as weight decay, obtained by adding an $\ell_2$ penalty on model parameters to the underlying empirical risk. While the effect of weight decay on parameters is transparent, it is typically challenging to reformulate this into properties of a learned non-linear model. In the simple setting of one layer ReLU networks this situation has recently become more clear. Specifically, starting with an observation in Neyshabur et al. (2014) the articles Savarese et al. (2019); Ongie et al. (2019); Parhi & Nowak (2020a;b; 2021) explore and develop the fact that $\ell_2$ regularization on parameters in this setting is provably equivalent to penalizing the total variation of a certain Radon transform of the network function (cf eg Theorem 3.2). While the results in these articles hold for any input dimension, in this article we consider the simplest case of input dimension 1. In this setting, our main contributions are:

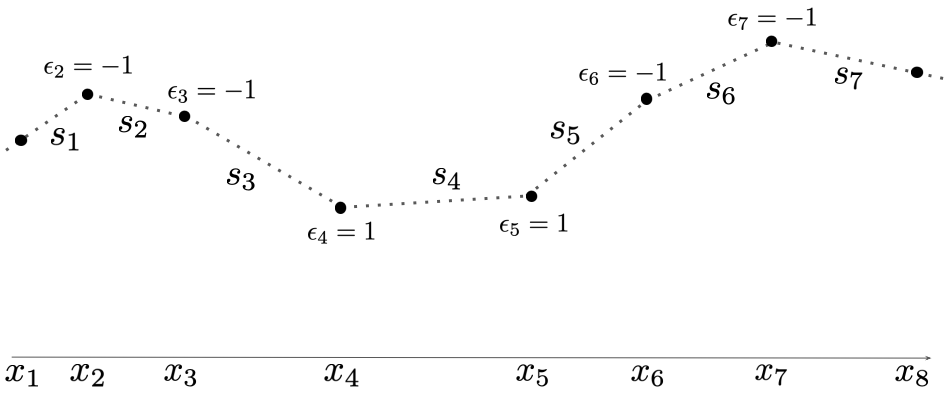



Figure 1: A dataset $\boldsymbol{D}$ with $m = 8$ points. Shown are the "connect the dots" interpolant $f_{\boldsymbol{D}}$ (dashed line), its slopes $s_i$ and the "discrete curvature" $\epsilon_i$ at each $x_i$.

1. Given a dataset $\boldsymbol{D} = \{(x_i, y_i)\}$ with scalar inputs and outputs, we obtain a complete characterization of all one layer ReLU networks with a single linear unit which fit the data and, among all such interpolating networks, do so with the minimal $\ell_2$ norm of the neuron weights. There are infinitely many such networks and, unlike in prior work, our characterization is phrased directly in terms of the behavior of the network function on intervals $(x_i, x_{i+1})$ between consecutive datapoints. Our description is purely geometric and can be summarized informally as follows (see Theorem 3.1 for the precise statement):

   - If we order $x_1 < \cdots < x_m$, then the data itself gives a discrete curvature estimate

   $$\epsilon_i := \operatorname{sgn}(s_i - s_{i-1}), \qquad s_i := \frac{y_{i+1} - y_i}{x_{i+1} - x_i}$$

   at $x_i$ of whatever function generated the data. See Figure 1.

   - If the curvature estimates $\epsilon_i$ and $\epsilon_{i+1}$ at $x_i$ and $x_{i+1}$ disagree (e.g. for $i = 3$ in Figure 1), then the network must be linear on $(x_i, x_{i+1})$. See Figures 2 and 3.

   - If the curvature estimates $\epsilon_i$ and $\epsilon_{i+1}$ at $x_i$ and $x_{i+1}$ agree and are positive (resp. negative), then the network function is convex (resp. concave) on $(x_i, x_{i+1})$ and lies below (resp. above) the straight line interpolant of the data. See Figures 2 and 3.

2. The geometric description of the space of interpolants of $\boldsymbol{D}$ in the previous bullet yields sharp generalization bounds for learning $1d$ Lipschitz functions. This is stated in Corollary 3.3. Specifically, if the dataset $\boldsymbol{D}$ is generated by setting $y_i = f_*(x_i)$ for $f_* : \mathbb{R} \to \mathbb{R}$ a Lipschitz function, then any one layer ReLU network with a single linear unit which interpolates $\boldsymbol{D}$ but does so with minimal $\ell_2$-norm of the network parameters will generalize as well as possible to unseen data, up to a universal multiplicative constant. To the author's knowledge this is the first time such generalization guarantees have been obtained.

## 2 SETUP AND INFORMAL STATEMENT OF RESULTS

Consider a one layer ReLU network

$$z(x) = z(x;\theta) := ax + b + \sum_{j=1}^{n} W_j^{(2)} \left[ W_j^{(1)} x + b_i^{(1)} \right]_+, \quad [t]_+ := \mathrm{ReLU}(t) = \max\{0, t\} \tag{1}$$

with a single linear unit[1] and input/output dimensions equal to one. For a given dataset

$$\boldsymbol{D} = \{(x_i, y_i)\,,\ i = 1, \dots, m\}\,, \qquad -\infty < x_1 < \cdots < x_m < \infty, \quad y_i \in \mathbb{R},$$

if the number of datapoints $m$ is smaller than the network width $n$, there are infinitely many choices of the parameter vector $\theta$ for which $z(x;\theta)$ interpolates (i.e. fits) the data:

$$z(x_i;\theta) = y_i, \qquad \forall\, i = 1, \dots, m. \tag{2}$$

Without further information about $\theta$, little can be said about the function $z(x;\theta)$ for $x$ in intervals $(x_i, x_{i+1})$ between consecutive datapoints when $n$ is much larger than $m$. This precludes useful generalization guarantees uniformly over all $\theta$, subject only to the interpolation condition (2).

In practice interpolants are not chosen arbitrary. Instead, they are learned by some variant of gradient descent starting from a random initialization. For a given architecture, initialization, optimizer, regularizer, and so on, understanding how the learned network uses the known labels $\{y_i\}$ to assign values of $z(x;\theta)$ for $x$ not in the dataset is an important open problem. To make progress, a fruitful line of inquiry in prior work has been to search for additional complexity measures based on margins Wei et al. (2018), PAC-Bayes estimates Dziugaite & Roy (2017; 2018); Nagarajan & Kolter (2019), weight matrix norms Neyshabur et al. (2015); Bartlett et al. (2017), information theoretic compression estimates Arora et al. (2018), Rachemacher complexity Golowich et al. (2018), etc (see Jiang et al. (2019) for a review and comparison). While perhaps not explicitly regularized, these complexity measures are hopefully small in trained networks, giving additional capacity constrains.

In this article, we take a different approach. We do not seek results valid for any network architecture. Instead, our goal is to describe completely, in concrete geometrical terms, the properties of one layer ReLU networks $z(x;\theta)$ that interpolate a dataset $\boldsymbol{D}$ with the minimal possible $\ell_2$ penalty

$$C(\theta) = C(\theta, n) = \sum_{j=1}^{n} |W_j^{(1)}|^2 + |W_j^{(2)}|^2$$

on the neuron weights. More precisely, we study the space of ridgeless ReLU interpolants

$$\mathrm{RidgelessReLU}(\boldsymbol{D}) := \{z(x;\theta) \quad | \quad z(x_i;\theta) = y_i \ \ \forall (x_i, y_i) \in \boldsymbol{D}, \quad C(\theta) = C_*\}\,, \tag{3}$$

of a dataset $\boldsymbol{D}$, where

$$C_* := \inf_{\theta, n} \{C(\theta, n) \mid z(x_i; n, \theta) = y_i \ \ \forall (x_i, y_i) \in \boldsymbol{D}\}\,.$$

Intuitively, elements in $\mathrm{RidgelessReLU}(\boldsymbol{D})$ are ReLU nets that minimize a weakly penalized loss

$$\boldsymbol{L}(\theta; \boldsymbol{D}) + \lambda C(\theta), \qquad \lambda \ll 1, \tag{4}$$

where $\boldsymbol{L}$ is an empirical loss, such as the mean squared error over $\boldsymbol{D}$, and the strength $\lambda$ of the weight decay penalty $C(\theta)$ is infinitesimal. It it plausible but by no means obvious that, with high probability, gradient descent from a random initialization and a weight decay penalty whose strength decreases to zero over training converges to an element in $\mathrm{RidgelessReLU}(\boldsymbol{D})$. This article does not study optimization, and we therefore leave this as an interesting open problem. Our main result is simple description of $\mathrm{RidgelessReLU}(\boldsymbol{D})$ and can informally be stated as follows:

**Theorem 2.1** (Informal Statement of Theorem 3.1)**.** *Fix a dataset $\boldsymbol{D} = \{(x_i, y_i),\ i = 1, \dots, m\}$. Each datapoint $(x_i, y_i)$ gives an estimate*

$$\epsilon_i := \mathrm{sgn}\,(s_i - s_{i-1})\,, \qquad s_i := \frac{y_{i+1} - y_i}{x_{i+1} - x_i}$$

*for the local curvature of the data (Figure 1). Among all continuous and piecewise linear functions $f$ that fit $\boldsymbol{D}$ exactly, the ones in $\mathrm{RidgelessReLU}(\boldsymbol{D})$ are precisely those that:*

[1] The linear term $ax + b$ is not really standard in practice but as in prior work Savarese et al. (2019); Ongie et al. (2019); Parhi & Nowak (2020a) leads a cleaner mathematical formulation of results.

- *Are convex (resp. concave) on intervals $(x_i, x_{i+1})$ at which neighboring datapoints agree on the local curvature in the sense that $\epsilon_i = \epsilon_{i+1} = 1$ (resp. $\epsilon_i = \epsilon_{i+1} = -1$). On such intervals $f$ lies below (resp. above) the straight line interpolant of the data (Figs. 2 and 3).*
- *Are linear (or more precisely affine) on intervals $(x_i, x_{i+1})$ when neighboring datapoints disagree on the local curvature in the sense that $\epsilon_i \cdot \epsilon_{i+1} \neq 1$.*

Before giving a precise statement our results, we mention that, as described in detail below, the space $\mathrm{RidgelessReLU}(\boldsymbol{D})$ has been considered in a number of prior articles Savarese et al. (2019); Ongie et al. (2019); Parhi & Nowak (2020a). Our starting point will be the useful but abstract characterization of $\mathrm{RidgelessReLU}(\boldsymbol{D})$ they obtained in terms of the total variation of the derivative of $z(x;\theta)$ (see (5)).

We note also that the conclusions of Theorem 2.1 (and Theorem 3.1) also hold under seemingly very different hypotheses from ours. Namely, instead of $\ell_2$-regularization on the parameters, Blanc et al. (2020) considers SGD training for mean squared error with iid noise added to labels. Their Theorem 2 shows (modulo some assumptions about interpreting the derivative of the ReLU) that, among all ReLU networks a linear unit that interpolate a dataset $\boldsymbol{D}$, the only ones that minimize the implicit regularization induced by adding iid noise to SGD are precisely those that satisfy the conclusions of Theorem 2.1 and hence are exactly the networks in $\mathrm{RidgelessReLU}(\boldsymbol{D})$. This suggests that our results hold under much more general conditions.

Further, our characterization of $\mathrm{RidgelessReLU}(\boldsymbol{D})$ in Theorem 3.1 immediately implies strong generalization guarantees uniformly over $\mathrm{RidgelessReLU}(\boldsymbol{D})$. We give a representative example in Corollary 3.3, which shows that such ReLU networks achieve the best possible generalization error of Lipschitz functions, up to constants.

Finally, note that we allow networks $z(x;\theta)$ of any width but that if the width $n$ is too small relative to the dataset size $m$, then the interpolation condition (2) cannot be satisfied. Also, we point out that in our formulation of the cost $C(\theta)$ we have left both the linear term $ax + b$ and the neuron biases unregularized. This is not standard practice but seems to yield the cleanest results.

## 3 Statement of Results and Relation to Prior Work

Every ReLU network $z(x;\theta)$ is a continuous and piecewise linear function from $\mathbb{R}$ to $\mathbb{R}$ with a finite number of affine pieces. Let us denote by $\mathrm{PL}$ the space of all such functions and define

$$\mathrm{PL(D)} := \{f \in \mathrm{PL} |\ f(x_i) = y_i\ \forall i = 1, \dots, m\}$$

to be the space of piecewise linear interpolants of $\boldsymbol{D}$. Perhaps the most natural element in $\mathrm{PL(D)}$ is the "connect-the-dots interpolant" $f_{\boldsymbol{D}} : \mathbb{R} \to \mathbb{R}$ given by

$$f_{\boldsymbol{D}}(x) := \begin{cases} \ell_1(x), & x < x_2 \\ \ell_i(x), & x_i < x < x_{i+1}, \quad i = 2, \dots, m-2 \\ \ell_{m-1}(x), & x > x_{m-1} \end{cases},$$

where for $i = 1, \dots, m-1$, we've set

$$\ell_i(x) := (x - x_i)s_i + y_i, \qquad s_i := \frac{y_{i+1} - y_i}{x_{i+1} - x_i}.$$

See Figure 1. In addition to $f_{\boldsymbol{D}}$, there are many other elements in $\mathrm{RidgelessReLU}(\boldsymbol{D})$. Theorem 3.1 gives a complete description of all of them phrased in terms of how they may behave on intervals $(x_i, x_{i+1})$ between consecutive datapoints. Our description is based on the signs

$$\epsilon_i = \mathrm{sgn}\,(s_i - s_{i-1}), \qquad 2 \leq i \leq m$$

of the (discrete) second derivatives of $f_{\boldsymbol{D}}$ at the inputs $x_i$ from our dataset.

**Theorem 3.1.** *The space $\mathrm{RidgelessReLU}(\boldsymbol{D})$ consists of those $f \in \mathrm{PL(D)}$ satisfying:*

1. *$f$ coincides with $f_{\boldsymbol{D}}$ on the following intervals:*

*(1a) Near infinity, i.e. on the intervals* $(-\infty, x_2)$, $(x_{m-1}, \infty)$

*(1b) Near datapoints that have zero discrete curvature, i.e. on intervals* $(x_{i-1}, x_{i+1})$ *with* $i = 2, \ldots, m-1$ *such that* $\epsilon_i = 0$.

*(1c) Between datapoints with opposite discrete curvature, i.e. on intervals* $(x_i, x_{i+1})$ *with* $i = 2, \ldots, m-1$ *such that* $\epsilon_i \cdot \epsilon_{i+1} = -1$.

2. *$f$ is convex (resp. concave) and bounded above (resp. below) by $f_{\boldsymbol{D}}$ between any consecutive datapoints at which the discrete curvature is positive (resp. negative). Specifically, suppose for some $3 \leq i \leq i+q \leq m-2$ that $x_i$ and $x_{i+q}$ are consecutive discrete inflection points in the sense that*

$$\epsilon_{i-1} \neq \epsilon_i, \qquad \epsilon_i = \cdots = \epsilon_{i+q}, \qquad \epsilon_{i+q} \neq \epsilon_{i+q+1}.$$

*If $\epsilon_i = 1$ (resp. $\epsilon_i = -1$), then restricted to the interval $(x_i, x_{i+q})$, $f$ is convex (resp. concave) and lies above (resp. below) the incoming and outgoing support lines and below (resp. above) $f_{\boldsymbol{D}}$:*

$$\begin{aligned} \epsilon_i = 1 \quad &\Longrightarrow \quad \max\{\ell_{i-1}(x),\, \ell_{i+q}(x)\} \;\leq\; f(x) \leq f_{\boldsymbol{D}}(x) \\ \epsilon_i = -1 \quad &\Longrightarrow \quad \min\{\ell_{i-1}(x),\, \ell_{i+q}(x)\} \;\geq\; f(x) \;\geq\; f_{\boldsymbol{D}}(x) \end{aligned}$$

*for all $x \in (x_i, x_{i+q})$.*

We refer the reader to §A for a proof of Theorem 3.1. Before doing so, let us illustrate Theorem 3.1 as an algorithm that, given the dataset $\boldsymbol{D}$, describes all elements in $\mathrm{RidgelessReLU}(\boldsymbol{D})$ (see Figures 2 and 3):

Step 1 **Linearly interpolate the endpoints**: by property (1), $f \in \mathrm{RidgelessReLU}(\boldsymbol{D})$ must agree with $f_{\boldsymbol{D}}$ on $(-\infty, x_2)$ and $(x_{m-1}, \infty)$.

Step 2 **Compute discrete curvature**: for $i = 2, \ldots, m-1$ calculate the discrete curvature $\epsilon_i$ at the data point $x_i$.

Step 3 **Linearly interpolate on intervals with zero curvature**: for all $i = 2, \ldots, m-1$ at which $\epsilon_i = 0$ property (1) guarantees that $f$ coincides with the $f_{\boldsymbol{D}}$ on $(x_{i-1}, x_{i+1})$.

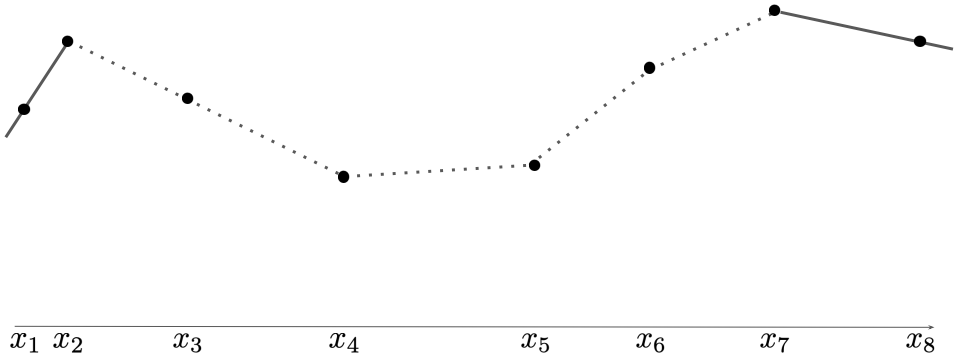
Step 4 **Linearly interpolate on intervals with ambiguous curvature**: for all $i = 2, \ldots, m-1$ at which $\epsilon_i \cdot \epsilon_{i+1} = -1$ property (1) guarantees that $f$ coincides with $f_{\boldsymbol{D}}$ on $(x_i, x_{i+1})$.

Step 5 **Determine convexity/concavity on remaining points**: all intervals $(x_i, x_{i+1})$ on which $f$ has not yet been determined occur in sequences $(x_i, x_{i+1}), \ldots, (x_{i+q-1}, x_{i+q})$ on which $\epsilon_{i+j} = 1$ or $\epsilon_{i+j} = 1$ for all $j = 0, \ldots, q$. If $\epsilon_i = 1$ (resp. $\epsilon_i = -1$), then $f$ is any convex (resp. concave) function bounded below (resp. above) by $f_{\boldsymbol{D}}$ and above (resp. below) the support lines $\ell_i(x)$, $\ell_{i+q}(x)$.

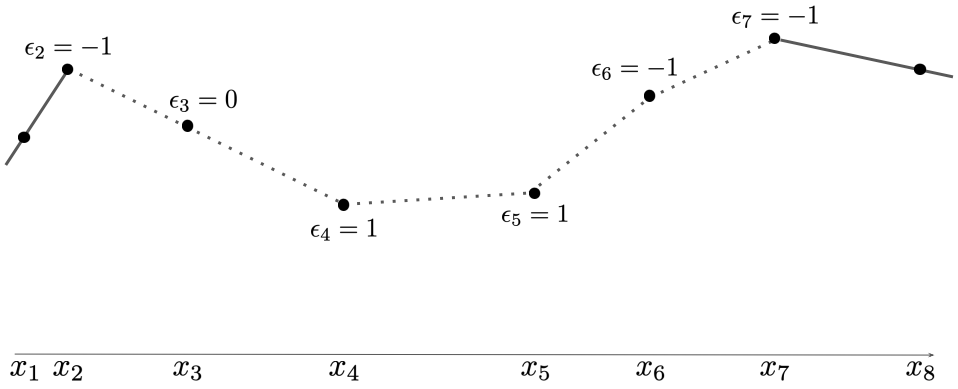
The starting point for the proof of Theorem 3.1 comes from the prior articles Neyshabur et al. (2014); Savarese et al. (2019); Ongie et al. (2019), which obtained an insightful "function space" interpretation of $\mathrm{RidgelessReLU}(\boldsymbol{D})$ as a subset of $\mathrm{PL(D)}$. Specifically, a simple computation (cf e.g. Theorem 3.3 in Savarese et al. (2019) and also Lemma A.14 below) shows that $f_{\boldsymbol{D}}$ achieves the smallest value of the total variation $||Df||_{TV}$ for the derivative $Df$ among all $f \in \mathrm{PL(D)}$. (The function $Df$ is piecewise constant and $||Df||_{TV}$ is the sum of absolute values of its jumps.) Part of the content of the prior work Neyshabur et al. (2014); Savarese et al. (2019); Ongie et al. (2019) is the following result

**Theorem 3.2** (cf Lemma 1 in Ongie et al. (2019) and around equation (17) in Savarese et al. (2019))**.** *For any dataset $\boldsymbol{D}$ we have*

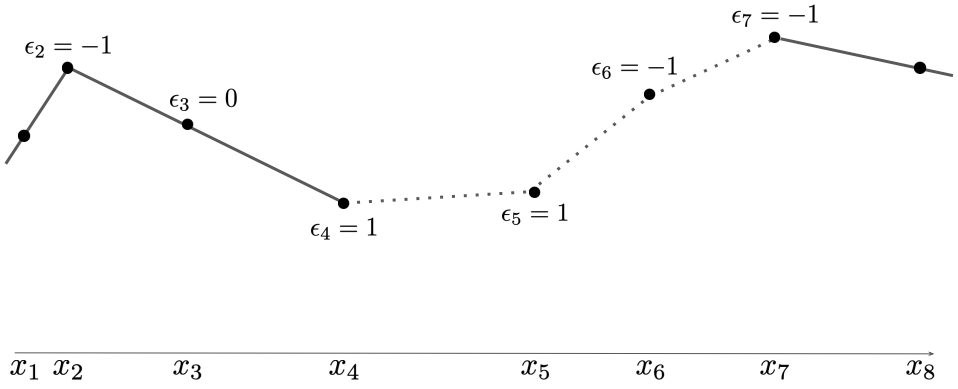
$$\mathrm{RidgelessReLU}(\boldsymbol{D}) = \{f \in \mathrm{PL(D)} \mid ||Df||_{TV} = ||Df_{\boldsymbol{D}}||_{TV}\}\,. \tag{5}$$




(a) Step 1

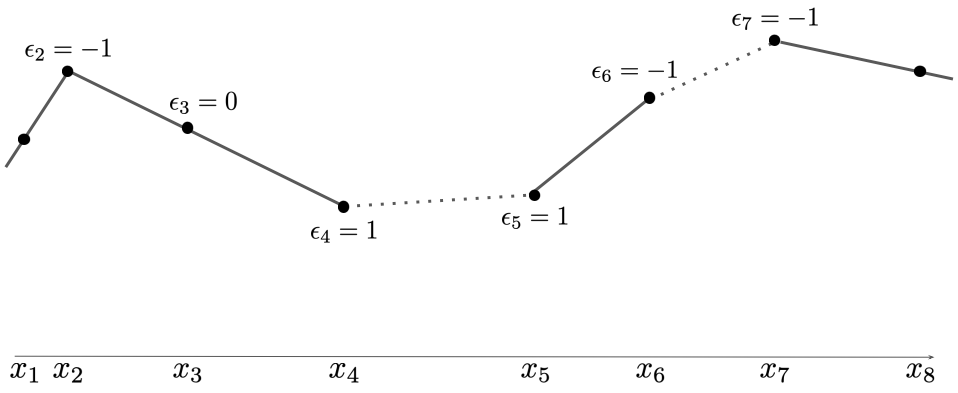



(b) Step 2

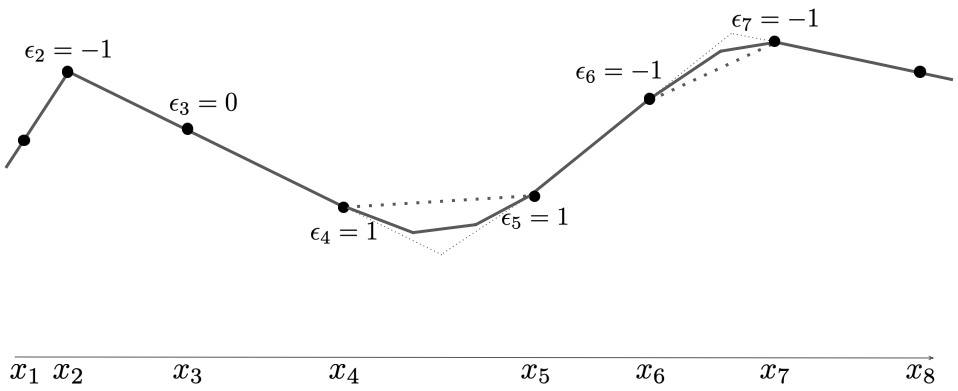



(c) Step 3

Figure 2: Steps 1 - 3 for generating $\mathrm{RidgelessReLU}(\boldsymbol{D})$ from the dataset $\boldsymbol{D}$.




(a) Step 4




(b) Step 5. One possible choice of a convex interpolant on $(x_4, x_5)$ and of a concave interpolant on $(x_6, x_7)$ is shown. Thin dashed lines are the supporting lines that bound all interpolants below on $(x_4, x_5)$ and above on $(x_6, x_7)$.

Figure 3: Steps 4 - 5 for generating $\mathrm{RidgelessReLU}(\boldsymbol{D})$ from the dataset $\boldsymbol{D}$.

Theorem 3.2 shows that $\mathrm{RidgelessReLU}(\boldsymbol{D})$ is precisely the space of functions in $\mathrm{PL(D)}$ that achieve the minimal possible total variation norm for the derivative. Thus, intuitively, functions in $\mathrm{RidgelessReLU}(\boldsymbol{D})$ are averse to oscillation in their slopes. The proof of this fact uses a simple idea introduced in Theorem 1 of Neyshabur et al. (2014) which leverages the homogeneity of the ReLU to translate between the regularizer $C(\theta)$ and the penalty $||Df||_{TV}$.

Theorem 3.1 yields strong generalization guarantees uniformly over $\mathrm{RidgelessReLU}(\boldsymbol{D})$. To state a representative example, suppose $\boldsymbol{D}$ is generated by a function $f_* : \mathbb{R} \to \mathbb{R}$:

$$y_j = f_*(x_j).$$

**Corollary 3.3** (Sharp generalization on Lipschitz Functions from Theorem 3.1)**.** *Fix a dataset* $\boldsymbol{D} = \{(x_i, y_i),\ i = 1, \dots, m\}$. *We have*

$$\sup_{f \in \mathrm{RidgelessReLU}(\boldsymbol{D})} ||f||_{\mathrm{Lip}} \leq ||f_*||_{\mathrm{Lip}}\,. \tag{6}$$

*Hence, if* $f_*$ *is* $L-$*Lipschitz and* $x_i = i/m$ *are uniformly spaced in* $[0, 1]$*, then*

$$\sup_{f \in \mathrm{RidgelessReLU}(\boldsymbol{D})} \sup_{x \in [0,1]} |f(x) - f_*(x)| \leq \frac{2L}{m}. \tag{7}$$

*Proof.* Observe that for any $i = 2, \dots, m-1$ and $x \in (x_i, x_{i+1})$ at which $Df(x)$ exists we have

$$\epsilon_i(s_{i-1} - s_i) \leq \epsilon_i(Df(x) - s_i) \leq \epsilon_i(s_{i+1} - s_i). \tag{8}$$

Indeed, when $\epsilon_i = 0$ the estimate (8) follows from property (1b) in Theorem 3.1. Otherwise, (8) follows immediately from the local convexity/concavity of $f$ in property (2). Hence, combining (8) with property (1a) shows that for each $i = 1, \dots, m-1$

$$||Df||_{L^\infty(x_i, x_{i+1})} \leq \max\left\{|s_{i-1}|\,,\ |s_i|\right\}.$$

Again using property (1a) and taking the maximum over $i = 2, \dots, m$ we find

$$||Df||_{L^\infty(\mathbb{R})} \leq \max_{1 \leq i \leq m-1} |s_i| = ||f_{\boldsymbol{D}}||_{\mathrm{Lip}}\,.$$

To complete the proof of (6) observe that for every $i = 1, \dots, m-1$

$$|s_i| = \left|\frac{y_{i+1} - y_i}{x_{i+1} - x_i}\right| = \left|\frac{f_*(x_{i+1}) - f_*(x_i)}{x_{i+1} - x_i}\right| \leq ||f_*||_{\mathrm{Lip}} \quad \Longrightarrow \quad ||f_{\boldsymbol{D}}||_{\mathrm{Lip}} \leq ||f_*||_{\mathrm{Lip}}\,.$$

Given any $x \in [0, 1]$, let us write $x'$ for its nearest neighbor in $\{i/m,\ i = 1, \dots, m\}$. We find

$$|f(x) - f_*(x)| \leq |f(x) - f(x')| + |f_*(x') - f_*(x)| \leq \left(||f||_{\mathrm{Lip}} + ||f_*||_{\mathrm{Lip}}\right)|x - x'| \leq \frac{2L}{m}.$$

Taking the supremum over $f \in \mathrm{RidgelessReLU}(\boldsymbol{D})$ and $x \in [0, 1]$ proves (7). □

Corollary 3.3 gives the best possible generalization error of Lipschitz functions, up to a universal multiplicative constant, in the sense that if all we knew about $f_*$ was that it was $L$-Lipschitz and were given its values on $\{i/m,\ i = 1, \dots, m\}$, then we cannot recover $f_*$ in $L^\infty$ to accuracy that is better than a constant times $L/m$. Further, the same kind of result holds with high probability if $x_i$ are drawn independently at random from $[0, 1]$, with the $2L/m$ on the right hand side replaced by $C\log(m)L/m$ for some universal constant $C > 0$. The appearance of the logarithm is due to the fact that among $m$ iid points in $[0, 1]$ the the largest spacing between consecutive points scales like $C\log(m)/m$ with high probability. Similar generalization results can easily be established, depending on the level of smoothness assumed for $f_*$ and the uniformity of the datapoints $x_i$.

In writing this article, it at first appeared to the author that the generalization bounds (7) cannot be directly obtained from the relation (5) of prior work. The issue is that a priori the relation (5) gives bounds only on the global value of $||Df||_{TV}$, suggesting perhaps that it does not provide strong constraints on local information about the behavior of ridgeless interpolants on small intervals $(x_i, x_{i+1})$. However, the relation (5) can actually be effectively *localized* to yield the estimates (6)

and (7) but with worse constants. The idea is the following. Fix $f \in \mathrm{RidgelessReLU}(\boldsymbol{D})$. For any $i_* = 3, \dots, m-2$ define the left, right and central portions of $\boldsymbol{D}$ as follows:

$$\boldsymbol{D}_L := \{(x_i, y_i),\ i < i_*\}\,, \quad \boldsymbol{D}_C := \{(x_i, y_i),\ i_* - 1 \leq i \leq i_* + 1\}\,, \quad \boldsymbol{D}_R := \{(x_i, y_i),\ i_* < i\}\,.$$

Consider further the left, right, and central versions of $f$, defined by

$$f_L(x) = \begin{cases} f(x), & x < x_{i_*} \\ \ell_{i_*}(x), & x > x_{i_*} \end{cases}, \qquad f_R(x) = \begin{cases} f(x), & x > x_{i_*} \\ \ell_{i_*}(x), & x < x_{i_*} \end{cases}$$

and

$$f_C(x) = \begin{cases} f(x), & x_{i*-1} < x < x_{i_*+1} \\ \ell_{i_*-1}(x), & x < x_{i_*-1} \\ \ell_{i_*}(x), & x > x_{i_*+1} \end{cases}.$$

Using (5), we have $||Df_{\boldsymbol{D}}||_{TV} = ||Df||_{TV}$. Further,

$$||Df||_{TV} \geq ||Df_L||_{TV} + ||Df_C||_{TV} + ||Df_R||_{TV}\,,$$

which, by again applying (5) but this time to $\boldsymbol{D}_L, \boldsymbol{D}_R$ and $f_L, f_R$, yields the bound

$$||Df||_{TV} \geq ||f_{\boldsymbol{D}_L}||_{TV} + ||Df_C||_{TV} + ||Df_{\boldsymbol{D}_R}||_{TV}\,.$$

Using that

$$||Df_{\boldsymbol{D}}||_{TV} = \sum_{i=2}^{m} |s_i - s_{i-1}|\,, \quad ||f_{\boldsymbol{D}_L}||_{TV} = \sum_{i=2}^{i_*-2} |s_i - s_{i-1}|\,, \quad ||Df_{\boldsymbol{D}_R}||_{TV} = \sum_{i=i_*+2}^{m-1} |s_i - s_{i-1}|$$

we derive the localized estimate

$$|s_{i_*+1} - s_{i_*}| + |s_{i_*} - s_{i_*-1}| + |s_{i_*-1} - s_{i_*-2}| \geq ||Df_C||_{TV}$$

Note further that

$$||Df_C||_{TV} \geq \max_{x \in (x_i, x_{i+1})} Df(x) - \min_{x \in (x_i, x_{i+1})} Df(x),$$

where the max and min are taken over those $x$ at which $Df(x)$ exists. The interpolation condition $f(x_i) = y_i$ and $f(x_{i+1}) = y_{i+1}$ yields that

$$\max_{x \in (x_i, x_{i+1})} Df(x) \geq s_i \qquad \text{and} \qquad \min_{x \in (x_i, x_{i+1})} Df(x) \leq s_i.$$

Putting together the previous three lines of inequalities (and checking the edge cases $i = 2, m-1$), we conclude that for any $i = 2, \dots, m-1$ we have

$$||Df(x) - s_i||_{L^\infty(x_i, x_{i+1})} \leq |s_{i+1} - s_i| + |s_i - s_{i-1}| + |s_{i-1} - s_{i-2}|\,,$$

where we set $s_0 = s_1$. Thus, as in the last few lines of the proof of Corollary 3.3, we conclude that

$$||f||_{\mathrm{Lip}} \leq 7\,||f_*||_{\mathrm{Lip}} \qquad \text{and} \qquad |f(x) - f_*(x)| \leq \frac{14L}{m}.$$

## 4 Conclusion and Future Directions

In this article, we completely characterized all possible ReLU networks that interpolate a given dataset $\mathcal{D}$ in the simple setting of weakly $\ell_2$-regularized one layer ReLU networks with a single linear unit and input/output dimension 1. Moreover, our characterization shows that, to assign labels to unseen data such networks simply "look at the curvature of the nearest neighboring datapoints on each side," in a way made precise in Theorem 3.1. This simple geometric description led to sharp generalization results for learning $1d$ Lipschitz functions in Corollary 3.3. This opens many direction for future investigation. Theorem 3.1 shows, for instance, that there are infinitely many ridgeless ReLU interpolants of a given dataset $\mathcal{D}$. It would be interesting to understand which ones are actually learned by gradient descent from a random initialization and a weak (or even decaying) $\ell_2$-penalty in time. Further, as already pointed out after the Theorem 2.1, the conclusions of Theorem 3.1 appear to hold under very different kinds of regularization (e.g. Theorem 2 in Blanc et al. (2020)). This raises the question: what is the most general kind of regularizer that is equivalent to weight decay, at least in our simple setup? It would also be quite natural to extend the results in this article to ReLU networks with higher input dimension, for which weight decay is known to correspond to regularization of a certain weighted Radon transform of the network function Ongie et al. (2019); Parhi & Nowak (2020a;b; 2021). Finally, extending the results in this article to deeper networks and beyond fully connected architectures are fascinating directions left to future work.

## REFERENCES


Sanjeev Arora, Rong Ge, Behnam Neyshabur, and Yi Zhang. Stronger generalization bounds for deep nets via a compression approach. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pp. 254–263, 2018.

Peter L Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov):463–482, 2002.

Peter L Bartlett, Dylan J Foster, and Matus J Telgarsky. Spectrally-normalized margin bounds for neural networks. In *Advances in neural information processing systems*, pp. 6240–6249, 2017.

Guy Blanc, Neha Gupta, Gregory Valiant, and Paul Valiant. Implicit regularization for deep neural networks driven by an ornstein-uhlenbeck like process. In *Conference on learning theory*, pp. 483–513. PMLR, 2020.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

Gintare Karolina Dziugaite and Daniel M Roy. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. *Uncertainty in AI. 2017. arXiv:1703.11008*, 2017.

Gintare Karolina Dziugaite and Daniel M Roy. Data-dependent pac-bayes priors via differential privacy. *NIPS 2018. arXiv:1802.09583*, 2018.

Noah Golowich, Alexander Rakhlin, and Ohad Shamir. Size-independent sample complexity of neural networks. In *Proceedings of the 31st Conference On Learning Theory*, volume 75 of *Proceedings of Machine Learning Research*, pp. 297–299, 2018.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034, 2015.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Yiding Jiang, Behnam Neyshabur, Hossein Mobahi, Dilip Krishnan, and Samy Bengio. Fantastic generalization measures and where to find them. *ICLR 2020. arXiv:1912.02178*, 2019.

John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pp. 1097–1105, 2012.

Dmytro Mishkin and Jiri Matas. All you need is a good init. *ICLR. 2016. arXiv:1511.06422*, 2015.

Vaishnavh Nagarajan and J Zico Kolter. Deterministic pac-bayesian generalization bounds for deep networks via generalizing noise-resilience. *ICLR 2019. arXiv:1905.13344*, 2019.

Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. In search of the real inductive bias: On the role of implicit regularization in deep learning. *ICLR Workshop. arXiv:1412.6614*, 2014.

Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. Norm-based capacity control in neural networks. In *Conference on Learning Theory*, pp. 1376–1401. PMLR, 2015.

Greg Ongie, Rebecca Willett, Daniel Soudry, and Nathan Srebro. A function space view of bounded norm infinite width relu nets: The multivariate case. *ICRL 2020. arXiv:1910.01635*, 2019.

Rahul Parhi and Robert D Nowak. Banach space representer theorems for neural networks and ridge splines. *arXiv preprint arXiv:2006.05626*, 2020a.

Rahul Parhi and Robert D Nowak. Neural networks, ridge splines, and tv regularization in the radon domain. *arXiv e-prints*, pp. arXiv–2006, 2020b.

Rahul Parhi and Robert D Nowak. What kinds of functions do deep neural networks learn? insights from variational spline theory. *arXiv preprint arXiv:2105.03361*, 2021.

Pedro Savarese, Itay Evron, Daniel Soudry, and Nathan Srebro. How do infinite width bounded norm networks look in function space? *COLT arXiv:1902.05040*, 2019.

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

VN Vapnik and A Ya Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Measures of Complexity*, 16(2):11, 1971.

O Vinyals, I Babuschkin, J Chung, M Mathieu, M Jaderberg, W Czarnecki, A Dudzik, A Huang, P Georgiev, R Powell, et al. Alphastar: Mastering the real-time strategy game starcraft ii, 2019.

Colin Wei, Jason Lee, Qiang Liu, and Tengyu Ma. On the margin theory of feedforward neural networks. 2018.

Lechao Xiao, Yasaman Bahri, Jascha Sohl-Dickstein, Samuel S Schoenholz, and Jeffrey Pennington. Dynamical isometry and a mean field theory of cnns: How to train 10,000-layer vanilla convolutional neural networks. *ICML and arXiv:1806.05393*, 2018.