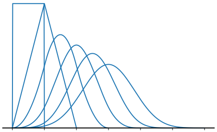
What Kinds of Functions do Neural Networks Learn?

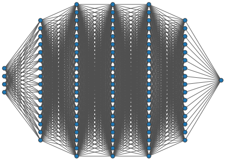
Rahul Parhi
Department of Electrical and Computer Engineering
University of Wisconsin–Madison

(joint work with Robert Nowak)

Informal Talk, Simons Institute

November 24th, 2021





What is Learning?

- Let $\{(\boldsymbol{x}_n, \boldsymbol{y}_n)\}_{n=1}^N \subset \mathbb{R}^d \times \mathbb{R}^D$ be a data set, and let $\mathcal{X}$ be a (Banach) space of functions mapping $\mathbb{R}^d \to \mathbb{R}^D$.
- The goal is to find $f \in \mathcal{X}$ such that $f(\boldsymbol{x}_n) \approx \boldsymbol{y}_n$.
- Consider the minimization

$$\min_{f \in \mathcal{X}} \sum_{n=1}^N \ell(\boldsymbol{y}_n, f(\boldsymbol{x}_n)).$$

$\Longrightarrow$ When $\mathcal{X}$ is an infinite-dimensional space, this problem is **ill-posed**.

Question

How do we make this problem **well-posed**?

Answer

Regularize!

(Explicit) Regularization

- Instead consider the minimization

$$\min_{f \in \mathcal{X}} \sum_{n=1}^{N} \ell(\boldsymbol{y}_n, f(\boldsymbol{x}_n)) + \lambda \|f\|_{\mathcal{X}}^p,$$

where $\|\cdot\|_{\mathcal{X}}$ is a (semi)norm, $\lambda > 0$, $1 \leq p < \infty$.

Three Remarkable Ideas

1. Smoothing Splines (1960s–1970s)
   $\Longrightarrow$ $\ell^2/L^2/$Tikhonov regularization
   $\Longrightarrow$ RKHS theory and kernel methods
2. Wavelet Thresholding (1990s)
   $\Longrightarrow$ $\ell^1/L^1/\mathrm{TV}$ regularization
   $\Longrightarrow$ Sparse signal and image processing
3. Neural Networks Trained with GD (1990s–present)
   $\Longrightarrow$ $\ell^1/L^1/\mathrm{TV}$ regularization
   $\Longrightarrow$ Everything

Comparing These Approaches

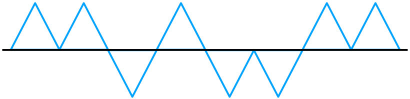
Consider the following function spaces defined in terms of the second (distributional) derivative of a function f, $\mathrm{D}^2 f$.

$$\dot{H}^2[0,1] \coloneqq \left\{ f : [0,1] \to \mathbb{R} : \ \mathrm{D}^2 f \in L^2[0,1] \right\},$$
$$\mathrm{BV}^2[0,1] \coloneqq \left\{ f : [0,1] \to \mathbb{R} : \ \mathrm{D}^2 f \in \mathcal{M}[0,1] \right\},$$

where $\mathcal{M}[0,1]$ is the space of finite (Radon) measures on $[0,1]$.

- $\dot{H}^2[0,1] \subset \mathrm{BV}^2[0,1] \subset L^2[0,1]$.

$\implies$ $\dot{H}^2[0,1]$ is a **Hilbert space**.
$\implies$ $\mathrm{BV}^2[0,1]$ is a (non-Hilbertian) **Banach space**.
$\implies$ Functions with discontinuous derivatives are in $\mathrm{BV}^2[0,1]$, but not in $\dot{H}^2[0,1]$.



Learning in $\dot{H}^2[0,1]$: Smoothing Splines

Suppose $f \in \dot{H}^2[0,1]$ and we observe

$$y_n = f(x_n) + \varepsilon_n,$$

where ε_n is i.i.d. white noise. The solution $\widehat{f}_{\mathsf{ss}}$ to

$$\min_{f \in \dot{H}^2[0,1]} \sum_{n=1}^{N} \ell(y_n, f(x_n)) + \lambda \|\mathrm{D}^2 f\|_{L^2}^2$$

is a **cubic smoothing spline**[1] and satisfies

$$\mathbb{E}\|f - \widehat{f}_{\mathsf{ss}}\|_{L^2}^2 \lesssim N^{-4/5},$$

which is the **minimax rate** for $\dot{H}^2[0,1]$.

[1] de Boor & Lynch, 1966; Kimeldorf & Wahba, 1970

Learning in $\dot{H}^2[0,1]$: Wavelet Thresholding

The solution $\widehat{f}_{\mathsf{wav}}$ to

$$\min_{\boldsymbol{\alpha}} \sum_{n=1}^{N} \ell\left(y_n, \sum_{j,k} \alpha_{j,k}\psi_{j,k}(x_n)\right) + \lambda\|\boldsymbol{\alpha}\|_1$$

is a **wavelet thresholding estimator**[2] and satisfies

$$\mathbb{E}\|f - \widehat{f}_{\mathsf{wav}}\|_{L^2}^2 \lesssim N^{-4/5},$$

which is the **minimax rate** for $\dot{H}^2[0,1]$.

[2] Donoho, 1995

Learning in $\dot{H}^2[0,1]$: Locally Adaptive Splines

The solution $\widehat{f}_{\mathsf{las}}$ to

$$\min_{f \in \mathrm{BV}^2[0,1]} \sum_{n=1}^{N} \ell(y_n, f(x_n)) + \lambda \|\mathrm{D}^2 f\|_{\mathcal{M}}$$

is a **locally adaptive linear spline**[3] of the form

$$\widehat{f}_{\mathsf{las}}(x) = c_0 + c_1 x + \sum_{k=1}^{K} \alpha_k \, \rho(x - t_k),$$

where $\rho = \max\{0, \cdot\}$ is the ReLU.

- $\mathrm{D}^2 \left\{ c_0 + c_1 x + \sum_{k=1}^{K} \alpha_k \, \rho(x - t_k) \right\} = \sum_{k=1}^{K} \alpha_k \, \delta(\cdot - t_k).$
- The optimal coefficients $\boldsymbol{\alpha}$ minimize

$$\sum_{n=1}^{N} \ell(y_n, f(x_n)) + \lambda \|\boldsymbol{\alpha}\|_1.$$

[3] Fisher & Jerome, 1975; Mammen and van de Geer, 1997

Learning in $\dot{H}^2[0,1]$: Neural Networks

- A single-hidden layer ReLU network (with a skip connection):

$$f_{\boldsymbol{v},\boldsymbol{w},\boldsymbol{b},\boldsymbol{c}}(x) = c_0 + c_1 x + \sum_{k=1}^{K} v_k \, \rho(w_k x - b_k).$$

$\Longrightarrow$ Same form as a locally adaptive linear spline.

$$\mathrm{D}^2 \left\{ c_0 + c_1 x + \sum_{k=1}^{K} \alpha_k \, \rho(w_k x - b_k) \right\} = \sum_{k=1}^{K} v_k |w_k| \, \delta\left(\cdot - \frac{b_k}{w_k} \right).$$

- The solution to the neural network training problem

$$\min_{\boldsymbol{v},\boldsymbol{w},\boldsymbol{b},\boldsymbol{c}} \sum_{n=1}^{N} \ell(y_n, f_{\boldsymbol{v},\boldsymbol{w},\boldsymbol{b},\boldsymbol{c}}(x_n)) + \lambda \sum_{k=1}^{K} |v_k||w_k|$$

is a locally adaptive spline!

Learning in $\dot{H}^2[0,1]$: Neural Networks

- Let $\widehat{f}_{\mathsf{nn}}$ be the solution to

$$\min_{\boldsymbol{v},\boldsymbol{w},\boldsymbol{b},\boldsymbol{c}} \sum_{n=1}^{N} \ell(y_n, f_{\boldsymbol{v},\boldsymbol{w},\boldsymbol{b},\boldsymbol{c}}(x_n)) + \frac{\lambda}{2} \sum_{k=1}^{K} |v_k|^2 + |w_k|^2.$$

$\Longrightarrow$ Training a neural network with **weight decay**.

- For any $\gamma > 0$, $(v_k, w_k) \mapsto (v_k/\gamma, \gamma\, w_k)$ does not change $f_{\boldsymbol{v},\boldsymbol{w},\boldsymbol{b},\boldsymbol{c}}$.

$\Longrightarrow$ At the solution $|v_k| = |w_k|$. Grandvalet, 1998; Neyshabur, 2015

- The above problem is equivalent to

$$\min_{\boldsymbol{v},\boldsymbol{w},\boldsymbol{b},\boldsymbol{c}} \sum_{n=1}^{N} \ell(y_n, f_{\boldsymbol{v},\boldsymbol{w},\boldsymbol{b},\boldsymbol{c}}(x_n)) + \lambda \sum_{k=1}^{K} |v_k||w_k|$$

$\Longrightarrow$ $\widehat{f}_{\mathsf{nn}}$ is a locally adaptive linear spline! P. & Nowak, 2020

Learning in $\dot{H}^2[0,1]$: Neural Networks

- The locally adaptive linear spline satisfies

$$\mathbb{E}\|f - \widehat{f}_{\mathsf{las}}\|_{L^2}^2 \lesssim N^{-4/5}.$$

$\implies$ The neural network trained with weight decay satisfies

$$\mathbb{E}\|f - \widehat{f}_{\mathsf{nn}}\|_{L^2}^2 \lesssim N^{-4/5},$$

which is the minimax rate for $\dot{H}^2[0,1]$.

Remark

Training a neural network with weight decay **appears** to be ℓ^2-regularization but is actually ℓ^1-regularization.

Learning in $\mathrm{BV}^2[0,1]$

- $\widehat{f}_{\mathsf{ss}}$, $\widehat{f}_{\mathsf{wav}}$, $\widehat{f}_{\mathsf{nn}}$ are all **minimax optimal** when $f \in \dot{H}^2[0,1]$.
- What if $f \in \mathrm{BV}^2[0,1]$?

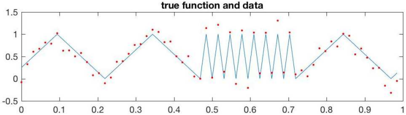
$\Longrightarrow$ The minimax rate for $\mathrm{BV}^2[0,1]$ is also $N^{-4/5}$.

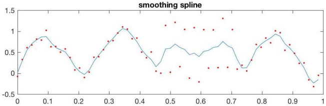
- We have

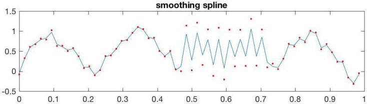
	$\widehat{f}_{\mathsf{ss}}$	$\widehat{f}_{\mathsf{wav}}$	$\widehat{f}_{\mathsf{nn}}$
$f \in \dot{H}^2[0,1]$	$N^{-4/5}$	$N^{-4/5}$	$N^{-4/5}$
$f \in \mathrm{BV}^2[0,1]$	$N^{-3/4}$	$N^{-4/5}$	$N^{-4/5}$

$\Longrightarrow$ The smoothing spline is suboptimal for $f \in \mathrm{BV}^2[0,1]$.

$\mathrm{BV}^2[0,1]$ Functions are Spatially Inhomogeneous

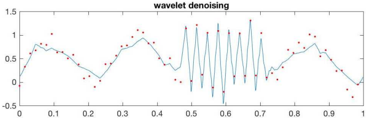


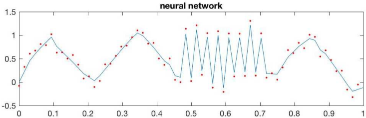




- Smoothing spline either oversmooths high variation portion of data or undersmooths low variation portion of data.

  $\Longrightarrow$ Drawback of kernel/Hilbert space methods in general.





- Wavelet and neural network approaches automatically **adapt** to the **local smoothness** of the data.

What Kinds of Functions do Neural Networks Learn?

Question

What kinds of functions do **shallow, univariate** neural networks learn?

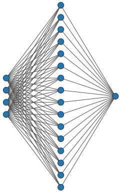
Answer

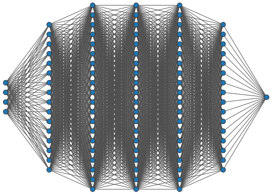
Functions in the **Banach space** $\mathrm{BV}^2[0,1]$.

Observation

Even the simplest (shallow, univariate) neural networks are **not** "fancy kernel machines".

What About Deep Neural Networks?





Shallow Multivariate Neural Networks

- In the univariate case, single-hidden layer neural networks solved a variational problem in

$$\mathrm{BV}^2(\mathbb{R}) = \left\{ f : \mathbb{R} \to \mathbb{R} : \ \|\mathrm{D}^2 f\|_{\mathcal{M}} < \infty \right\}$$

$\implies$ $\mathrm{TV}^2(f) \coloneqq \|\mathrm{D}^2 f\|_{\mathcal{M}}$

$\implies$ Key property is that D^2 sparsifies univariate neurons

$$\mathrm{D}^2\{\rho(wx - b)\} = |w|\delta(x - b/w)$$

- Multivariate neurons are $\boldsymbol{x} \mapsto \rho(\boldsymbol{w}^\mathsf{T}\boldsymbol{x} - b)$, $\boldsymbol{w} \in \mathbb{R}^d$, $b \in \mathbb{R}$.

$\implies$ These are "ridge functions".

Question

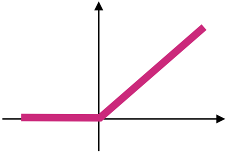
Is there an operator that sparsifies a multivariate neuron?

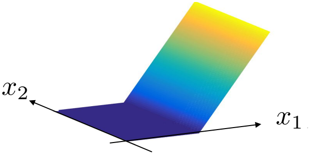
Answer

Yes, and it involves the Radon transform.

The Sparsifying Operator

- Ridge functions are univariate functions "extended" outward in all other dimenions.





- We can use the Radon transform of a function $f : \mathbb{R}^d \to \mathbb{R}$,

$$\mathscr{R}\{f\}(\boldsymbol{\gamma}, t) = \int_{\mathbb{R}^d} f(\boldsymbol{x}) \delta(\boldsymbol{\gamma}^\mathsf{T} \boldsymbol{x} - t) \, \mathrm{d}\boldsymbol{x}, \quad (\boldsymbol{\gamma}, t) \in \mathbb{S}^{d-1} \times \mathbb{R},$$

to "extract" the underlying univariate function to extend results for univariate functions to multivariate ridge functions.

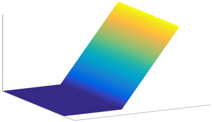
$\Longrightarrow$ The "ridge trick".

$\Longrightarrow$ Ridgelets (Candès, 1998)

The Sparsifying Operator

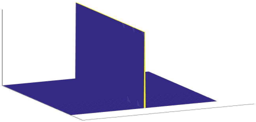
- Neuron

$$\implies \quad \rho(\boldsymbol{w}_0^\mathsf{T}(\cdot) - b_0), \; (\boldsymbol{w}_0, b_0) \in \mathbb{S}^{d-1} \times \mathbb{R}$$



- **Laplacian** of neuron

$$\implies \quad \Delta\big\{\rho(\boldsymbol{w}_0^\mathsf{T}(\cdot) - b_0)\big\} = \delta(\boldsymbol{w}_0^\mathsf{T}(\cdot) - b_0)$$



- **Filtered Radon transform** of Laplacian of neuron[4]

$$\implies \quad (\Lambda^{d-1} \mathscr{R} \Delta)\big\{\rho(\boldsymbol{w}_0^\mathsf{T}(\cdot) - b_0)\big\}(\boldsymbol{\gamma}, t) = \delta((\boldsymbol{\gamma}, t) - (\boldsymbol{w}_0, b_0)).$$

[4] Ongie et al., 2020; P & Nowak, 2021

Native Space for Shallow Neural Networks

Question

What would be the multivariate analogue of $\mathrm{BV}^2(\mathbb{R})$?

Answer

$\mathscr{R}\,\mathrm{BV}^2(\mathbb{R}^d)$.

$$\mathscr{R}\,\mathrm{BV}^2(\mathbb{R}^d) := \left\{ f : \mathbb{R}^d \to \mathbb{R} : \ \mathscr{R}\,\mathrm{TV}^2(f) < \infty \right\}$$

- $\mathscr{R}\,\mathrm{TV}^2(f) := \|\Lambda^{d-1}\,\mathscr{R}\,\Delta f\|_{\mathcal{M}}$
  $\implies \ \mathrm{TV}^2(f) = \|\mathrm{D}^2 f\|_{\mathcal{M}}$.
- When $d = 1$, $\mathscr{R}\,\mathrm{BV}^2(\mathbb{R}^d) = \mathrm{BV}^2(\mathbb{R})$ and $\mathscr{R}\,\mathrm{TV}^2(\cdot) = \mathrm{TV}^2(\cdot)$.

Representer Theorem

Theorem (P. & Nowak, 2021)

There exists a solution to the variational problem

$$\min_{f \in \mathscr{R}\,\mathrm{BV}^2(\mathbb{R}^d)} \sum_{n=1}^{N} \ell(y_n, f(\boldsymbol{x}_n)) + \lambda \mathscr{R}\,\mathrm{TV}^2(f)$$

of the form

$$\widehat{f}(\boldsymbol{x}) = \sum_{k=1}^{K} v_k \, \rho(\boldsymbol{w}_k^\mathsf{T} \boldsymbol{x} - b_k) + \boldsymbol{c}^\mathsf{T} \boldsymbol{x} + c_0, \quad K < N.$$

- $\widehat{f}$ is a **sparse** single-hidden layer ReLU network with a skip connection.

$\Longrightarrow$ Skip connection corresponds to null space of $\mathscr{R}\,\mathrm{TV}^2(\cdot)$.

Neural Network Training

- $\mathscr{R}\,\mathrm{TV}^2(\widehat{f}) = \sum_{k=1}^{K} |v_k| \|\boldsymbol{w}_k\|_2$.

$\implies$ Can find a solution to

$$\min_{f \in \mathscr{R}\,\mathrm{BV}^2(\mathbb{R}^d)} \sum_{n=1}^{N} \ell(y_n, f(\boldsymbol{x}_n)) + \lambda \mathscr{R}\,\mathrm{TV}^2(f)$$

by training a ReLU network with "path-norm" regularization:

$$\min_{\boldsymbol{\theta}=(\boldsymbol{v},\mathbf{W},\boldsymbol{b},\boldsymbol{c},c_0)} \sum_{n=1}^{N} \ell(y_n, f_{\boldsymbol{\theta}}(\boldsymbol{x}_n)) + \lambda \sum_{k=1}^{K} |v_k| \|\boldsymbol{w}_k\|_2.$$

or, equivalently, with weight decay:

$$\min_{\boldsymbol{\theta}=(\boldsymbol{v},\mathbf{W},\boldsymbol{b},\boldsymbol{c},c_0)} \sum_{n=1}^{N} \ell(y_n, f_{\boldsymbol{\theta}}(\boldsymbol{x}_n)) + \lambda \sum_{k=1}^{K} |v_k|^2 + \|\boldsymbol{w}_k\|_2^2$$

- Shallow multivariate neural networks learn functions in the **Banach space** $\mathscr{R}\,\mathrm{BV}^2(\mathbb{R}^d)$.

What is $\mathscr{R}\,\mathrm{BV}^2(\mathbb{R}^d)$?

- $\mathscr{R}\,\mathrm{BV}^2(\mathbb{R}^d)$ is a **non-Hilbertian** Banach space.

$$\|f\|_{\mathscr{R}\,\mathrm{BV}^2(\mathbb{R}^d)} := \mathscr{R}\,\mathrm{TV}^2(f) + |f(\mathbf{0})| + \sum_{k=1}^{d} |f(\boldsymbol{e}_k) - f(\mathbf{0})|$$

$\Longrightarrow$ $\{\boldsymbol{e}_k\}_{k=1}^d$ is the canonical basis in $\mathbb{R}^d$.

- For $f \in \mathscr{R}\,\mathrm{BV}^2(\mathbb{R}^d)$, $\|f\|_{\mathscr{R}\,\mathrm{BV}^2(\mathbb{R}^d)}$ is an upper bound of its Lipschitz constant.
- Not a classically studied space in analysis.

What is $\mathscr{R}\,\mathrm{BV}^2(\Omega)$?

- Let $\Omega = \left\{\boldsymbol{x} \in \mathbb{R}^d : \ \|\boldsymbol{x}\| \leq 1\right\}$. Then,

$$\mathscr{R}\,\mathrm{BV}^2(\Omega) \coloneqq \{f : \Omega \to \mathbb{R} : \ \exists g \in \mathscr{R}\,\mathrm{BV}^2(\mathbb{R}^d) \text{ s.t. } g\big|_{\Omega} = f\}$$

- Every $f \in \mathscr{R}\,\mathrm{BV}^2(\Omega)$ admits an **integral representation**

$$f(\boldsymbol{x}) = \int_{\mathbb{S}^{d-1} \times [-1,1]} \rho(\boldsymbol{w}^\mathsf{T}\boldsymbol{x} - b)\,\mathrm{d}\mu(\boldsymbol{w}, b) + \boldsymbol{c}^\mathsf{T}\boldsymbol{x} + c_0$$

$\Longrightarrow$ We can approximate such integrals with K terms with $L^2(\Omega)$ error that scales like $\lesssim K^{-1/2}$, **breaking the curse of dimensionality**. Maurey/Pisier, 1981; Barron 1993

Approximation Properties of $\mathscr{R}\,\mathrm{BV}^2(\Omega)$

- Given $f \in \mathscr{R}\,\mathrm{BV}^2(\Omega)$, there exists

$$f_K(\boldsymbol{x}) = \sum_{k=1}^{K} v_k\, \rho(\boldsymbol{w}_k^\mathsf{T}\boldsymbol{x} - b_k) + \boldsymbol{c}^\mathsf{T}\boldsymbol{x} + c_0$$

  such that

$$\|f - f_K\|_{L^2(\Omega)} \lesssim K^{-\frac{1}{2}-\frac{3}{2d}} \lesssim K^{-\frac{1}{2}}.$$

  This is the **best** rate. Bach, 2017; Siegel & Xu, 2021; P. & Nowak, 2021

- Compare this to the **best** K-term approximation rates in $H^s[0,1]^d$, which scales as

$$\|f - f_K\|_{L^2} \lesssim K^{-\frac{s}{d}}$$

  and is achieved by truncated Fourier series approximation.

$\implies$ This rate **grows exponentially** with the input dimension d.

Estimation Properties of $\mathscr{R}\,\mathrm{BV}^2(\Omega)$

- Given $f \in \mathscr{R}\,\mathrm{BV}^2(\Omega)$, suppose we observe

$$y_n = f(\boldsymbol{x}_n) + \varepsilon_n,\ n = 1, \dots, N,$$

where $\{\boldsymbol{x}_n\}_{n=1}^N \subset \Omega$ are nicely distributed and $\{\varepsilon_n\}_{n=1}^N$ are i.i.d. white noise.

- The solution to the neural network training problem

$$\widehat{f}_N = \operatorname*{arg\,min}_{\boldsymbol{\theta}=(\boldsymbol{v},\mathbf{W},\boldsymbol{b},\boldsymbol{c},c_0)} \sum_{n=1}^N \ell(y_n, f_{\boldsymbol{\theta}}(\boldsymbol{x}_n)) + \frac{\lambda}{2} \sum_{k=1}^K |v_k|^2 + \|\boldsymbol{w}_k\|_2^2$$

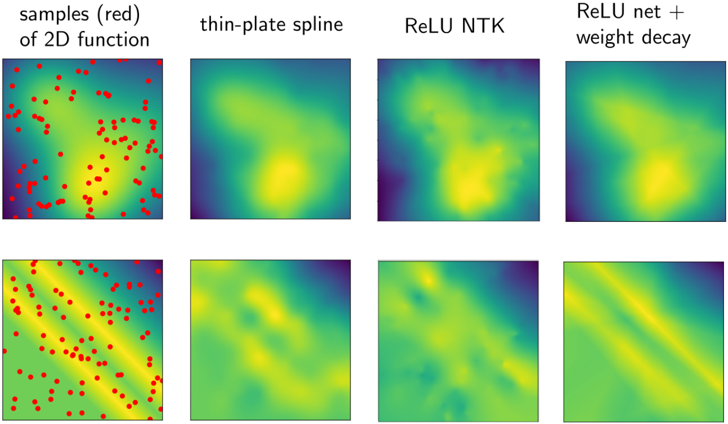
satisfies

$$\mathbb{E}\|f - \widehat{f}_N\|_{L^2}^2 \lesssim N^{-\frac{d+3}{2d+3}} \lesssim N^{-\frac{1}{2}}.$$

$\Longrightarrow$ This rate does not grow with the input dimension d.

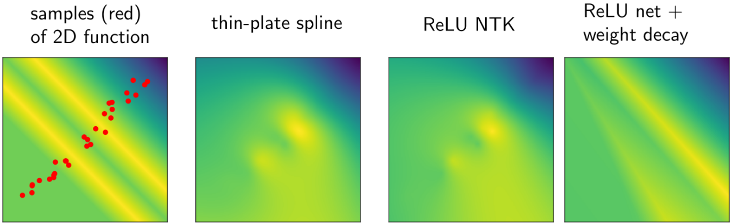
$\Longrightarrow$ This is the **minimax rate**. P. & Nowak, 2021

Data Fitting and Extrapolation



neural networks learn and extrapolate very differently than classical multivariate estimation techniques and kernel methods in general

Data Fitting and Extrapolation



neural networks learn and extrapolate very differently than classical multivariate estimation techniques and kernel methods in general

Other Neural Spaces

- The so-called second-order **spectral Barron space** is a Banach space when equipped with the norm

$$\|f\|_{\mathscr{B}^2(\mathbb{R}^d)} := \int_{\mathbb{R}^d} (1 + \|\boldsymbol{\omega}\|_2)^2 |F(\boldsymbol{\omega})| \, \mathrm{d}\boldsymbol{\omega}$$

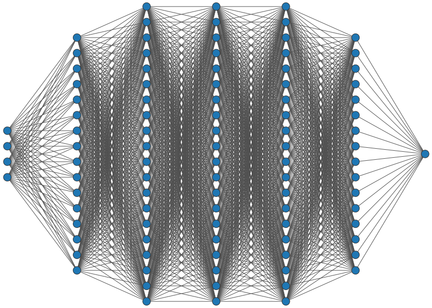
Barron, 1993

- $\mathscr{B}^2(\mathbb{R}^d) \subset \mathscr{R}\,\mathrm{BV}^2(\mathbb{R}^d)$ Siegel & Xu, 2021; P. & Nowak 2021

Deep Neural Networks

Question

What kinds of functions do deep neural networks learn?



Preliminaries for Learning with Deep Neural Networks

- $\mathscr{R}\,\mathrm{BV}^2(\mathbb{R}^d;\mathbb{R}^D) = \underbrace{\mathscr{R}\,\mathrm{BV}^2(\mathbb{R}^d) \times \cdots \times \mathscr{R}\,\mathrm{BV}^2(\mathbb{R}^d)}_{D \text{ times}}$.
- $\mathscr{R}\,\mathrm{BV}^2(\mathbb{R}^d;\mathbb{R}^D)$ is a **vector-valued** Banach space when equipped with the norm

$$\|f\|_{\mathscr{R}\,\mathrm{BV}^2(\mathbb{R}^d;\mathbb{R}^D)} = \sum_{m=1}^{D} \|f_m\|_{\mathscr{R}\,\mathrm{BV}^2(\mathbb{R}^d)},$$

where $f = (f_1, \ldots, f_D)$.

Compositional or "Deep" $\mathscr{R}\,\mathrm{BV}^2$ space

- Consider the space

$$\mathscr{R}\,\mathrm{BV}^2_{\mathsf{deep}}(L) \coloneqq \left\{ f = f^{(L)} \circ \cdots \circ f^{(1)} \;\middle|\; \begin{array}{l} f^{(\ell)} \in \mathscr{R}\,\mathrm{BV}^2(\mathbb{R}^{d_{\ell-1}}; \mathbb{R}^{d_\ell}), \\ \ell = 1, \ldots, L \end{array} \right\}.$$

- This definition captures two architectural specifications of deep neural networks.
  1. L, the number of hidden layers.
  2. d_ℓ, the functional "width" of each layer.

Deep ReLU Network Representer Theorem

Representer Theorem (P. & Nowak 2021)

There **exists** a solution to the variational problem

$$\min_{f \in \mathscr{R}\mathrm{BV}^2_{\mathsf{deep}}(L)} \sum_{n=1}^{N} \ell(\boldsymbol{y}_n, f(\boldsymbol{x}_n)) + \lambda \sum_{\ell=1}^{L} \|f^{(\ell)}\|_{\mathscr{R}\mathrm{BV}^2(\mathbb{R}^{d_{\ell-1}};\mathbb{R}^{d_\ell})}$$

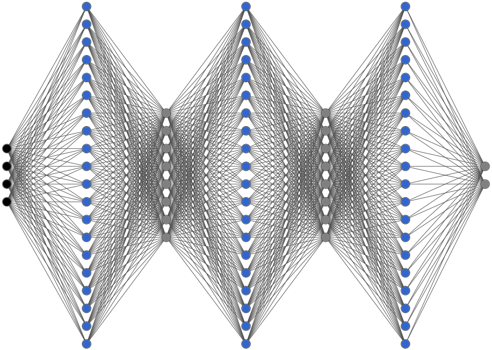
of the form $\boldsymbol{x}^{(L)}$, where

$$\begin{cases} \boldsymbol{x}^{(0)} := \boldsymbol{x}, \\ \boldsymbol{x}^{(\ell)} := \mathbf{V}^{(\ell)} \boldsymbol{\rho}(\mathbf{W}^{(\ell)} \boldsymbol{x}^{(\ell-1)} - \boldsymbol{b}^{(\ell)}) + \mathbf{C}^{(\ell)} \boldsymbol{x}^{(\ell-1)} + \boldsymbol{c}_0^{(\ell)}, \; \ell = 1, \dots, L. \end{cases}$$

Let $\widehat{f}(\boldsymbol{x}) := \boldsymbol{x}^{(L)}$.

Deep ReLU Network Representer Theorem

- $\widehat{f}$ is a deep ReLU network with **skip connections** and **rank bounded weight matrices**.



- The width of the ℓth ReLU layer is $\leq N d_\ell$.
- The weight matrix between ReLU layers is $\mathbf{A}^{(\ell)} \coloneqq \mathbf{W}^{(\ell+1)}\mathbf{V}^{(\ell)}$.
  $\mathbf{A}^{(\ell)}$ satsifies $\operatorname{rank}(\mathbf{A}^{(\ell)}) \leq d_\ell$.
  $\Longrightarrow$ d_ℓ is the "functional width" of layer ℓ.

Learning with Deep Neural Networks

Our representer theorem implies the neural network training problem

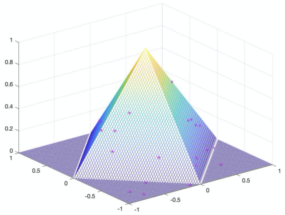
$$\min_{\boldsymbol{\theta} \in \Theta} \sum_{n=1}^{N} \ell(\boldsymbol{y}_n, f_{\boldsymbol{\theta}}(\boldsymbol{x}_n)) + \lambda \sum_{\ell=1}^{L} \left(\frac{1}{2} \sum_{k=1}^{K^{(\ell)}} \|\boldsymbol{v}_k^{(\ell)}\|_1^2 + \|\boldsymbol{w}_k^{(\ell)}\|_2^2 + \sum_{j=0}^{d_\ell} \|\boldsymbol{c}_j^{(\ell)}\|_1 \right).$$

- "Modified" weight decay

Benefits of Depth

- There exist functions in $\mathscr{R}\,\mathrm{BV}^2_{\mathsf{deep}}(L)$ with $L \geq 2$ that are not in $\mathscr{R}\,\mathrm{BV}^2(\mathbb{R}^d)$ (Ongie et al., 2020)

$\implies \quad f(\boldsymbol{x}) = \max\{0, 1 - \|\boldsymbol{x}\|_1\}$ "pyramid function"



$\implies \quad f \notin \mathscr{R}\,\mathrm{BV}^2(\mathbb{R}^d)$

$\implies \quad f \in \mathscr{R}\,\mathrm{BV}^2_{\mathsf{deep}}(L = 2)$.

- Fitting data from the pyramid function with a shallow network will result in **large** network weight norm.
- Fitting data from the pyramid function with a deep network will result in **small** network weight norm.

Takeaway Messages

- ReLU networks trained with **variants of weight decay** are optimal solutions to learning in $\mathscr{R}\,\mathrm{BV}^2_{\mathsf{deep}}(L)$.
  - $\Longrightarrow$ This space includes spatially inhomogeneous functions.
  - $\Longrightarrow$ ReLU networks learn spatially inhomogeneous functions.
- The $\mathscr{R}\,\mathrm{BV}^2_{\mathsf{deep}}(L)$ framework provides new rationale for **skip connections** in network architectures.
- The $\mathscr{R}\,\mathrm{BV}^2_{\mathsf{deep}}(L)$ framework suggests considering architectures with explicit **low-rank weight matrices**.
- ReLU networks learn functions in $\mathscr{R}\,\mathrm{BV}^2$-type function spaces
  - $\Longrightarrow$ These are **new, not classical** function spaces.
  - $\Longrightarrow$ $\mathscr{R}\,\mathrm{BV}^2(\mathbb{R}^d)$ is a **non-reflexive Banach space** with a sparsity-promoting norm.

References I

Bach, Francis (2017). "Breaking the curse of dimensionality with convex neural networks". In: **Journal of Machine Learning Research**.

Barron, Andrew R. (1993). "Universal approximation bounds for superpositions of a sigmoidal function". In: **IEEE Transactions on Information theory**.

Candès, Emmanuel J. (1998). "Ridgelets: theory and applications". PhD thesis. Stanford University Stanford.

de Boor, Carl and Robert E. Lynch (1966). "On splines and their minimum properties". In: **Journal of Mathematics and Mechanics**.

Donoho, David L. (1995). "De-noising by soft-thresholding". In: **IEEE transactions on information theory** 41.3, pp. 613–627.

Fisher, S. D. and Joseph W. Jerome (1975). "Spline solutions to L^1 extremal problems in one and several variables". In: **Journal of Approximation Theory** 13.1, pp. 73–83.

Grandvalet, Yves (1998). "Least absolute shrinkage is equivalent to quadratic penalization". In: **International Conference on Artificial Neural Networks**. Springer, pp. 201–206.

Kimeldorf, George S. and Grace Wahba (1970). "A correspondence between Bayesian estimation on stochastic processes and smoothing by splines". In: **The Annals of Mathematical Statistics** 41.2, pp. 495–502.

Mammen, Enno and Sara van de Geer (1997). "Locally adaptive regression splines". In: **Annals of Statistics** 25.1, pp. 387–413.

Neyshabur, Behnam, Ryota Tomioka, and Nathan Srebro (2015). "In Search of the Real Inductive Bias: On the Role of Implicit Regularization in Deep Learning". In: **3rd International Conference on Learning Representations, Workshop Track Proceedings**.

References II

Ongie, Greg, Rebecca Willett, Daniel Soudry, and Nathan Srebro (2020). "A Function Space View of Bounded Norm Infinite Width ReLU Nets: The Multivariate Case". In: **8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020**.

Parhi, Rahul and Robert D. Nowak (2020). "The role of neural network activation functions". In: **IEEE Signal Processing Letters** 27, pp. 1779–1783.

— (2021a). "Banach space representer theorems for neural networks and ridge splines". In: **Journal of Machine Learning Research** 22.43, pp. 1–40.

— (2021b). "Near-Minimax Optimal Estimation With Shallow ReLU Neural Networks". In: **arXiv preprint arXiv:2109.08844**.

— (2021c). "What Kinds of Functions do Deep Neural Networks Learn? Insights from Variational Spline Theory". In: **arXiv preprint arXiv:2105.03361**.

Pisier, Gilles (1981). "Remarques sur un résultat non publié de B. Maurey". In: **Séminaire Analyse fonctionnelle (dit**, pp. 1–12.

Siegel, Jonathan W. and Jinchao Xu (2021). "Characterization of the Variation Spaces Corresponding to Shallow Neural Networks". In: **arXiv preprint arXiv:2106.15002**.