A Function-Space Tour of Data Science

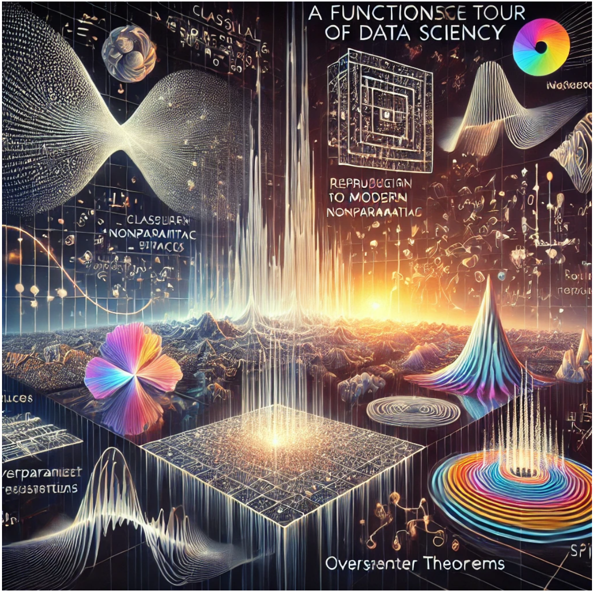
Rahul Parhi [1] Greg Ongie [2]

[1]UCSD [2]Marquette University

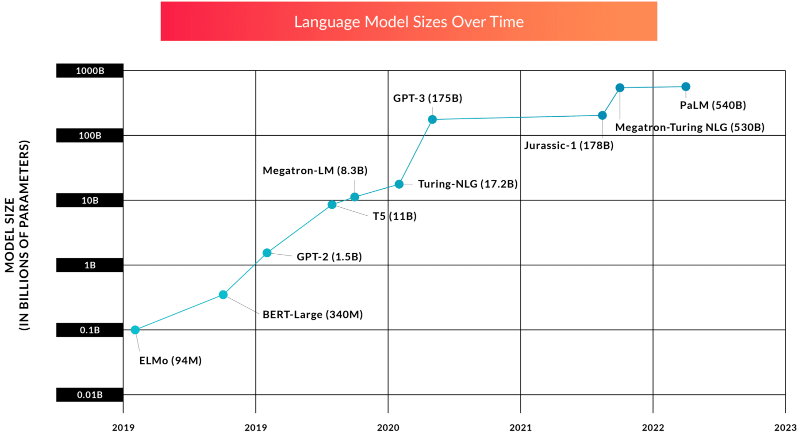
Conference on Parsimony and Learning (CPAL)

24 March 2025

Large Models Have Taken The World By Storm



SOTA Models are Getting Bigger and Bigger



Saracco "How much bigger can/should LLMs become?" IEEE Future Directions Blog

Modern Machine Learning is Overparameterized

Let $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n \subset \mathbb{R}^d \times \mathbb{R}$ be a training dataset and let $\mathcal{F}$ be a space of real-valued functions $f : \mathbb{R}^d \to \mathbb{R}$. Consider the learning problem

$$\min_{f \in \mathcal{F}} \sum_{i=1}^n (y_i - f(\boldsymbol{x}_i))^2 + \lambda R(f), \quad \lambda > 0.$$

If $\mathcal{F}$ is an infinite-dimensional space, then this problem is an "overparameterized" learning problem.

Example: Suppose $\{\varphi_k\}_{k \in \mathbb{Z}}$ is a basis for $\mathcal{F}$. Then, each $f \in \mathcal{F}$ can be represented as a model with an **infinite** parameters $\boldsymbol{\theta} = \{\theta_k\}_{k \in \mathbb{Z}}$ such that

$$f_{\boldsymbol{\theta}} = \sum_{k \in \mathbb{Z}} \theta_k \varphi_k.$$

The norm $R(f)$ reflects the "size" of the parameters $\boldsymbol{\theta}$.

The function-space view is a powerful tool to study the **infinite-parameter** limit of overparameterization.

Nonparametric methods as opposed to **parametric** methods.

Regularization is Necessary

Without regularization (either **implicit** from the optimization algorithm or **explicit** in the optimization problem), the overparameterized learning problem

$$\min_{f \in \mathcal{F}} \sum_{i=1}^{n} (y_i - f(\boldsymbol{x}_i))^2$$

is **ill-posed** since there are many interpolating (zero-loss solutions).

Which interpolating function (of the many possible) will be selected?

How does this choice affect performance/generalization?

Without regularization, it becomes challenging to answer these questions.

Today: A tour through (nonparametric) methods in data science through the unifying lens of **explicit regularization** in **function space**.

Goal: To provide sharp characterizations of the **inductive bias** of various data-fitting methods.

From Parametric to Nonparametric

Let Θ denote the **parameter space**. The associated **parametric model space** is

$$\mathcal{F}_\Theta = \{f_{\boldsymbol{\theta}} : \boldsymbol{\theta} \in \Theta\}$$

Let $C : \Theta \to \mathbb{R}_{\geq 0}$ denote a **parameter cost function**. Given any $f \in \mathcal{F}_\Theta$, its **parametric representation cost** is defined by

$$\mathring{R}(f) = \inf\{C(\boldsymbol{\theta}) : f = f_{\boldsymbol{\theta}}, \boldsymbol{\theta} \in \Theta\}$$

For an arbitrary (measurable) function $f : \mathbb{R}^d \to \mathbb{R}$, we can define its **nonparametric representation cost** as

$$R(f) = \begin{cases} \liminf_{f_k \to f} \mathring{R}(f), & \exists (f_k)_{k\in\mathbb{N}} \subset \mathcal{F}_\Theta \text{ that converges}^1 \text{ to } f \\ +\infty, & \text{else.} \end{cases}$$

The **native space** is given by

$$\mathcal{F} = \{f : \mathbb{R}^d \to \mathbb{R} \text{ measurable} : R(f) < +\infty\}.$$

[1] In an appropriate topology.

Function-Space Inductive Bias

Suppose that we have a **parametric method**.

- A parameter space Θ.
- A parametric model space $\mathcal{F}_\Theta$.

  $\Longrightarrow$ A subset of measurable functions $\mathbb{R}^d \to \mathbb{R}$.
- A parametric cost $C : \Theta \to \mathbb{R}_{\geq 0}$.

  $\Longrightarrow$ $C(\mathbf{0}) = 0$.

  $\Longrightarrow$ $\|\boldsymbol{\theta}\|_2 \leq \|\boldsymbol{\theta}'\|_2 \Leftrightarrow C(\boldsymbol{\theta}) \leq C(\boldsymbol{\theta}')$.

A parametric method **induces** a native space $\mathcal{F}$ and a corresponding nonparametric representation cost $R : \mathcal{F} \to \mathbb{R}_{\geq 0}$.

$$\min_{\boldsymbol{\theta} \in \Theta} \sum_{i=1}^{n} \mathcal{L}(y_i, f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)) + \lambda C(\boldsymbol{\theta}) \quad \Leftrightarrow \quad \min_{f \in \mathcal{F}} \sum_{i=1}^{n} \mathcal{L}(y_i, f(\boldsymbol{x}_i)) + \lambda R(f)$$

This is characterizes **function-space inductive bias** of $\mathcal{F}_\Theta$.

How Do Different Methods Look in Function Space?

- Can we characterize the nonparametric representation cost R?
- What are the properties of the native space $\mathcal{F}$?

$\Longrightarrow$ Is it a vector space?
$\Longrightarrow$ Is it a metric space?
$\Longrightarrow$ Is it a Banach space?
$\Longrightarrow$ Is it a Hilbert space?
$\Longrightarrow$ Does it have some other (topological) structure?
$\Longrightarrow$ How is $\mathcal{F}$ related to classical function spaces?

- How well does the parametric model space $\mathcal{F}_\Theta$ approximate $\mathcal{F}$?
- How well can we learn functions in $\mathcal{F}$ from data?

This is the **function-space view** of studying data-fitting methods.

The function-space view **unifies** classical and modern data-fitting methods.

Three Remarkable Ideas in Data Science

1. Kernel Methods
   - $\Longrightarrow$ ℓ^2-regularization of parameters
   - $\Longrightarrow$ Reproducing Kernel Hilbert Spaces
   - $\Longrightarrow$ Linear methods = not adaptive
2. Wavelet and Sparse Methods
   - $\Longrightarrow$ ℓ^1-regularization of parameters
   - $\Longrightarrow$ Besov Spaces and Bounded Variation (BV) Spaces
   - $\Longrightarrow$ Nonlinear methods = adaptive
3. Neural Networks
   - $\Longrightarrow$ ℓ^2-regularization of parameters
   - $\Longrightarrow$ Barron Spaces, Variation Spaces, and Radon BV Spaces
   - $\Longrightarrow$ Nonlinear methods = adaptive
   - $\Longrightarrow$ **Shallow vs. deep**

Classical methods were studied function space first.

Can we understand modern methods by characterizing their function spaces?

Kernel Methods

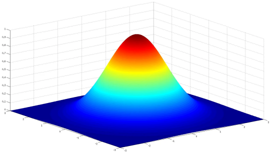
$$\left\{ f_{\boldsymbol{a}} = \sum_{i=1}^{n} a_i k(\cdot, \boldsymbol{x}_i) : \begin{array}{l} n \in \mathbb{N} \\ a_i \in \mathbb{R} \\ \boldsymbol{x}_i \in \mathbb{R}^d \end{array} \right\}$$

$$C(\boldsymbol{a}) = \boldsymbol{a}^{\mathsf{T}} \boldsymbol{K} \boldsymbol{a}$$

$\mathcal{F}$: RKHS induced by k

$$R(f) = \|f\|_{\mathcal{F}}^2$$
squared RKHS norm

$$\min_{\boldsymbol{a} \in \mathbb{R}^n} \sum_{i=1}^{n} \mathcal{L}(y_i, [\boldsymbol{K}\boldsymbol{a}]_i) + \lambda \boldsymbol{a}^{\mathsf{T}} \boldsymbol{K} \boldsymbol{a} \quad \Leftrightarrow \quad \min_{f \in \mathcal{F}} \sum_{i=1}^{n} \mathcal{L}(y_i, f(\boldsymbol{x}_i)) + \lambda \|f\|_{\mathcal{F}}^2$$



- The equivalence $\Leftrightarrow$ is understood via the **representer theorem**.
- There always exists a solution to the problem over $\mathcal{F}$ that lies in the span of shifted kernels.

Wavelet and Sparse Methods

$$\left\{ f_{\boldsymbol{\theta}} = \sum_{j,k} \theta_{j,k} \psi_{j,k} \, : \, \begin{array}{l} \psi_{j,k}(x) = \\ 2^{-j/2}\psi(2^j x - k) \end{array} \right\}$$

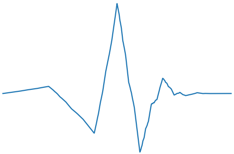
$$C(\boldsymbol{\theta}) = \|\boldsymbol{\theta}\|_{\ell^1}$$

$\mathcal{F}$: Besov space $B^1_{1,1}$

$$R(f) = \|f\|_{B^1_{1,1}}$$

Besov norm

$$\min_{\boldsymbol{\theta}} \sum_{i=1}^{n} \mathcal{L}(y_i, f_{\boldsymbol{\theta}}(x_i)) + \lambda \|\boldsymbol{\theta}\|_{\ell^1} \quad \Leftrightarrow \quad \min_{f \in B^1_{1,1}} \sum_{i=1}^{n} \mathcal{L}(y_i, f(x_i)) + \lambda \|f\|_{B^1_{1,1}}$$



- The equivalence $\Leftrightarrow$ is understood via the **wavelet shrinkage algorithm**.
- There always exists a solution to the problem over $B^1_{1,1}$ that is a sparse combination of wavelets.

Shallow Neural Networks

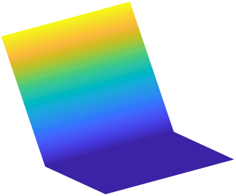
$$\left\{ f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \sum_{k=1}^{K} v_k \, \sigma(\boldsymbol{w}_k^\mathsf{T} \boldsymbol{x}) : \begin{array}{c} v_k \in \mathbb{R} \\ \boldsymbol{w}_k \in \mathbb{R}^d \end{array} \right\}$$

$\mathcal{F}$: Variation Space (Barron Space, Radon-BV Space)

$$C(\boldsymbol{\theta}) = \frac{1}{2} \sum_{k=1}^{K} |v_k|^2 + \|\boldsymbol{w}_k\|_2^2$$

$R(f) = \|f\|_{\mathcal{F}}$
variation norm

$$\min_{\boldsymbol{\theta}} \sum_{i=1}^{n} \mathcal{L}(y_i, f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)) + \frac{\lambda}{2} \sum_{k=1}^{K} |v_k|^2 + \|\boldsymbol{w}_k\|_2^2 \Leftrightarrow \min_{f \in \mathcal{F}} \sum_{i=1}^{n} \mathcal{L}(y_i, f(\boldsymbol{x}_i)) + \lambda \|f\|_{\mathcal{F}}$$



- The equivalence $\Leftrightarrow$ is understood via **Banach-space representer theorems**.
- There always exists a solution to the problem over $\mathcal{F}$ that is a **sparse** combination of neurons.

Deep Neural Networks

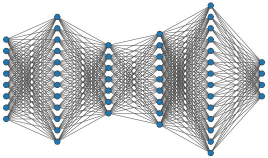
$$\{f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \boldsymbol{\sigma}(\boldsymbol{W}_L\boldsymbol{\sigma}(\boldsymbol{W}_{L-1}\boldsymbol{\sigma}(\cdots \boldsymbol{W}_1\boldsymbol{x})))\}$$

$$C(\boldsymbol{\theta}) = \frac{1}{L}\sum_{\ell=1}^{L}\|\boldsymbol{W}_\ell\|_F^2$$

$\mathcal{F}$: exists

$R(f)$: exists

$$\min_{\boldsymbol{\theta}}\sum_{i=1}^{n}\mathcal{L}(y_i, f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)) + \frac{\lambda}{L}\sum_{\ell=1}^{L}\|\boldsymbol{W}_\ell\|_F^2 \quad \Leftrightarrow \quad \min_{f\in\mathcal{F}}\sum_{i=1}^{n}\mathcal{L}(y_i, f(x_i)) + \lambda R(f)$$



- The equivalence $\Leftrightarrow$ is by construction.
- We currently do not know how to characterize $\mathcal{F}$ or $R(f)$.

$\Longrightarrow$ For $L > 2$, is it even a linear space?

A Note on Universal Approximation

A common heuristic to explain the success of deep learning is that neural networks are **universal approximators**. This heuristic is **meaningless** since any reasonable parametric model space is a universal approximator.

- polynomials
- kernel machines
- Fourier series
- wavelets
- shallow and deep neural networks

Theorem (Stone–Weierstraß)

Let $\Omega \subset \mathbb{R}^d$ be compact and $A \subset \mathrm{C}(\Omega)$ be a subalgebra (vector subspace closed under multiplication). Then, A is dense in $\mathrm{C}(\Omega)$ if and only if it separates points (for every $\boldsymbol{x}, \boldsymbol{x}' \in \Omega$ such that $\boldsymbol{x} \neq \boldsymbol{x}'$, there exists $p \in A$ such that $p(\boldsymbol{x}) \neq p(\boldsymbol{x}')$).

Pop Quiz: Does the closing procedure mean $\mathcal{F} = \mathrm{C}(\Omega)$?

When $R(f) < \infty$, it is (typically) the case that $\mathcal{F} \subsetneq \mathrm{C}(\Omega)$.

Outline

1. Hilbert Spaces $\Leftrightarrow$ Linear/Kernel Methods
2. Banach Spaces $\Leftrightarrow$ Nonlinear/Sparse Methods
3. Banach Spaces $\Leftrightarrow$ Shallow Neural Networks
4. Beyond(?) Banach Spaces $\Leftrightarrow$ Deep Neural Networks

Outline

1. Hilbert Spaces $\Leftrightarrow$ Linear/Kernel Methods
2. Banach Spaces $\Leftrightarrow$ Nonlinear/Sparse Methods
3. Banach Spaces $\Leftrightarrow$ Shallow Neural Networks
4. Beyond(?) Banach Spaces $\Leftrightarrow$ Deep Neural Networks

Hilbert Spaces: Basic Definition

Assume $\mathcal{F}$ is a **vector space of functions** defined on a domain $\Omega \subset \mathbb{R}^d$.

Will focus on functions with real outputs (scalar- or vector-valued).

We say $\langle \cdot, \cdot \rangle_{\mathcal{F}} : \mathcal{F} \times \mathcal{F} \to \mathbb{R}$ is an **inner product** on $\mathcal{F}$ if it is:

- *bilinear:* $\langle \alpha f + \beta g, h \rangle_{\mathcal{F}} = \alpha \langle f, h \rangle_{\mathcal{F}} + \beta \langle g, h \rangle_{\mathcal{F}}$ and
  $\langle f, \alpha g + \beta h \rangle_{\mathcal{F}} = \alpha \langle f, g \rangle_{\mathcal{F}} + \beta \langle f, h \rangle_{\mathcal{F}}$
- *symmetric:* $\langle f, g \rangle_{\mathcal{F}} = \langle g, f \rangle_{\mathcal{F}}$
- *positive definite:* $\langle f, f \rangle_{\mathcal{F}} \geq 0$ and $\langle f, f \rangle_{\mathcal{F}} = 0$ iff $f = 0$.

Any inner product $\langle \cdot, \cdot \rangle_{\mathcal{F}}$ on $\mathcal{F}$ defines a **norm** on $\mathcal{F}$ by:

$$\|f\|_{\mathcal{F}} := \sqrt{\langle f, f \rangle_{\mathcal{F}}}$$

Definition

A **Hilbert space** is a vector space $\mathcal{F}$ equipped with an inner product $\langle \cdot, \cdot \rangle_{\mathcal{F}}$ that is *complete* with respect to the norm $\| \cdot \|_{\mathcal{F}}$

Example: Finite Collection of Basis Functions

Suppose $\mathcal{F}$ is the span of K linearly independent basis functions $\varphi_1, \dots, \varphi_K$:

$$f_{\boldsymbol{\theta}} = \sum_{k=1}^{K} \theta_k \varphi_k, \quad \theta_k \in \mathbb{R}, k = 1, \dots, K$$

equipped with the inner product and norm

$$\langle f_{\boldsymbol{\theta}}, f_{\boldsymbol{\beta}} \rangle_{\mathcal{F}} = \boldsymbol{\theta}^{\mathsf{T}} \boldsymbol{\beta} \implies \|f_{\boldsymbol{\theta}}\|_{\mathcal{F}}^2 = \|\boldsymbol{\theta}\|_2^2.$$

Then $\mathcal{F}$ is a finite-dimensional Hilbert space, and we have the equivalence

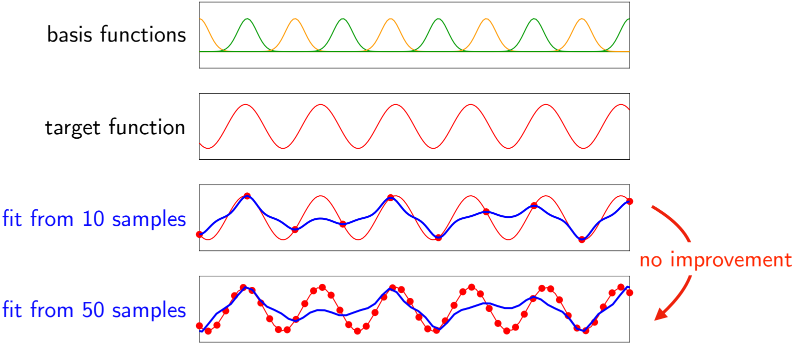
$$\min_{f \in \mathcal{F}} \sum_{i=1}^{n} (f(\boldsymbol{x}_i) - y_i)^2 + \lambda \|f\|_{\mathcal{F}}^2 \quad \Leftrightarrow \quad \min_{\boldsymbol{\theta} \in \mathbb{R}^K} \mathcal{L}(y_i, [\boldsymbol{V}\boldsymbol{\theta}]_i) + \lambda \|\boldsymbol{\theta}\|_2^2$$

where $\boldsymbol{V} \in \mathbb{R}^{n \times K}$ is such that $[\boldsymbol{V}]_{ik} = \varphi_k(\boldsymbol{x}_i)$.

Learning over $\mathcal{F}$ is a simple finite dimensional ℓ^2-regularized problem!

Limitations to Finite-Dimensional Hilbert Spaces

Finite-dimensional Hilbert spaces have **limited approximation capability**.



Could improve by adding basis functions (but how many? what type?)

Can we solve this issue with an **infinite-dimensional** Hilbert space?

L^2-space

One of the most fundamental infinite-dimensional Hilbert spaces is

$$L^2(\Omega) := \left\{ f : \Omega \to \mathbb{R} : \int_\Omega |f(\boldsymbol{x})|^2 d\boldsymbol{x} < +\infty \right\},$$

$$\langle f, g \rangle_{L^2(\Omega)} = \int_\Omega f(\boldsymbol{x}) g(\boldsymbol{x}) d\boldsymbol{x} \Rightarrow \|f\|_{L^2(\Omega)} = \sqrt{\int_\Omega |f(\boldsymbol{x})|^2 d\boldsymbol{x}}$$

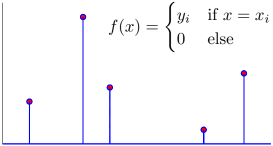
However, $L^2(\Omega)$ is “too big” of a space to be useful for learning.

Example: learning with L^2-norm regularization

$$\min_{f \in L^2(\Omega)} \sum_{i=1}^{n} (f(\boldsymbol{x}_i) - y_i)^2 + \lambda \|f\|^2_{L^2(\Omega)}$$

Pop Quiz: What functions minimize this loss?

Obtain **zero loss** by putting “spikes” at the datapoints:



Reproducing Kernel Hilbert Spaces

For learning to be possible in a infinite-dimensional Hilbert space, we need some additional regularity assumptions.

- Continuity seems to be a necessary requirement.
- But it is not sufficient – we also need to ensure that any sequence of functions approaching a "spike" cannot vanish in norm.

The largest class of Hilbert Spaces having precisely this property are known as **Reproducing Kernel Hilbert Spaces**

Definition

A Hilbert space of functions $\mathcal{F}$ is called a **Reproducing Kernel Hilbert Space (RKHS)** if for all $\boldsymbol{x} \in \Omega$ there exists a constant $C_{\boldsymbol{x}}$ such that

$$|f(\boldsymbol{x})| \leq C_{\boldsymbol{x}} \|f\|_{\mathcal{F}} \quad \text{for all } f \in \mathcal{F}.$$

Interpretation: if f is non-zero at any point, its norm is also non-zero.

Aronszajn 1950

Kernel Functions

In the language of functional analysis, an RKHS $\mathcal{F}$ is a Hilbert space where the evaluation functionals $f \mapsto f(\boldsymbol{x})$ are continuous.

By a result known as the **Riesz Representation Theorem**, this implies for all $\boldsymbol{x} \in \Omega$ there exists a function $K_{\boldsymbol{x}} \in \mathcal{F}$ such that

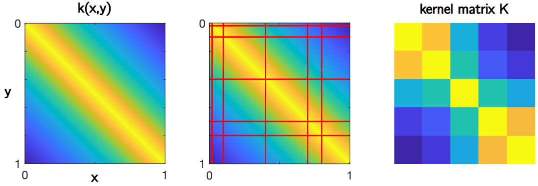
$$\langle K_{\boldsymbol{x}}, f \rangle_{\mathcal{F}} = f(\boldsymbol{x}) \quad \text{for all } f \in \mathcal{F}.$$

Define the associated **kernel** $k : \Omega \times \Omega \to \mathbb{R}$ by

$$k(\boldsymbol{x}, \boldsymbol{x'}) = \langle K_{\boldsymbol{x}}, K_{\boldsymbol{x'}} \rangle = K_{\boldsymbol{x}}(\boldsymbol{x}') = K_{\boldsymbol{x'}}(\boldsymbol{x}).$$

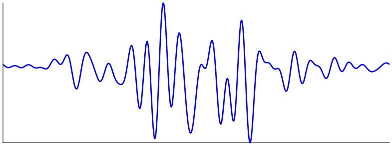
Two important properties: kernels are

1. **symmetric**: $k(\boldsymbol{x}, \boldsymbol{x}') = k(\boldsymbol{x}', \boldsymbol{x})$, for all $\boldsymbol{x}, \boldsymbol{x}' \in \Omega$ and
2. **positive definite**: for any finite set of points $\{\boldsymbol{x}_1, ..., \boldsymbol{x}_n\} \subset \Omega$, the *kernel matrix* $\boldsymbol{K} \in \mathbb{R}^{n \times n}$ with $K_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is a PSD matrix.



Example: Bandlimited Functions

Let $\mathcal{B} \subset L^2(\mathbb{R})$ be the space of functions $f : \mathbb{R} \to \mathbb{R}$ whose Fourier transform $\widehat{f}(\xi)$ vanishes for all frequencies $|\xi| > B$.



Pop Quiz: Why aren't "spike" functions allowed in this space?

Define the *sinc function* $s(x) := \mathcal{F}^{-1}(1_{[-B,B]})(x) = \sin(Bt)/\pi x$

Key property: $s * f = f$ for all $f \in \mathcal{B}$, since $\widehat{s * f} = \widehat{s} \cdot \widehat{f} = \widehat{f}$.

Put another way, we have

$$f(x) = (s * f)(x) = \int_{\mathbb{R}} f(x')s(x' - x)dx' = \langle f, s(\cdot - x)\rangle$$

Therefore, $\mathcal{B}$ is an RKHS with kernel function $k(x, x') = s(x - x')$.

Example: the Sobolev space $H^s(\Omega)$

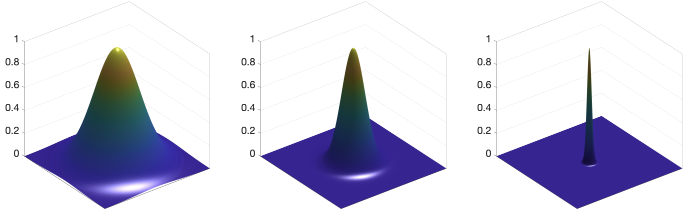
Let $H^s(\Omega)$ denote the space of functions $f : \Omega \to \mathbb{R}$ whose partial derivatives up to order s belong to $L^2(\Omega)$, equipped with the inner product

$$\langle f, g \rangle_{H^s(\Omega)} = \sum_{|\boldsymbol{\alpha}| \leq s} \int_\Omega D^{\boldsymbol{\alpha}} f(\boldsymbol{x}) D^{\boldsymbol{\alpha}} g(\boldsymbol{x}) \, d\boldsymbol{x} \;\Rightarrow\; \|f\|^2_{H^s(\Omega)} = \sum_{|\boldsymbol{\alpha}| \leq s} \|D^\alpha f\|^2_{L^2(\Omega)}$$

Fact: if $s > d/2$ then $H^s(\Omega)$ is an RKHS.

Smoothness $s \geq d/2$ is necessary, since otherwise arbitrarily thin "spike" functions would have vanishing norm:

$$f_\varepsilon(\boldsymbol{x}) := f(\boldsymbol{x}/\varepsilon) \Rightarrow \|\partial^s f_\varepsilon\|_{L^2} = \varepsilon^{d/2-s} \|\partial^s f\|_{L^2}$$



$\varepsilon = 1$ $\qquad$ $\varepsilon = 0.5$ $\qquad$ $\varepsilon = 0.1$

Building RKHSs from Kernels

Every RKHS induces a kernel $k(\cdot,\cdot)$ that is symmetric and positive definite.

On the flipside, given any function $k : \Omega \times \Omega \to \mathbb{R}$ that is symmetric and positive definite, we can **construct a RKHS** having k as its kernel.

1. Take the span of all kernel translates $k(\cdot, \boldsymbol{x})$:

$$\mathrm{span}\{k(\cdot,\boldsymbol{x}) : \boldsymbol{x} \in \Omega\} = \left\{ f_{\boldsymbol{a}} = \sum_{i=1}^{n} a_i k(\cdot, \boldsymbol{x}_i) : \begin{array}{l} n \in \mathbb{N} \\ a_i \in \mathbb{R} \\ \boldsymbol{x}_i \in \mathbb{R}^d \end{array} \right\}$$

2. Equip this space with the inner product:

$$f = \sum_{i=1}^{n} a_i k(\cdot, \boldsymbol{x}_i),\;\; g = \sum_{j=1}^{m} a'_j k(\cdot, \boldsymbol{x}'_j) \implies \langle f, g\rangle = \sum_{i=1}^{n}\sum_{j=1}^{m} a_i a'_j k(\boldsymbol{x}_i, \boldsymbol{x}_j)$$

3. Take the **closure** of the space (in the induced norm) to get the RKHS.

Theorem (Moore-Aronszajn)

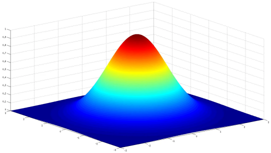
Every SPD function $k(\cdot,\cdot)$ defines a **unique** RKHS with k as its kernel.

Examples: Common Kernels

Linear Kernel $\quad k(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{x}^\mathsf{T}\boldsymbol{y},$

Polynomial Kernel $\quad k(\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{x}^\mathsf{T}\boldsymbol{y} + 1)^d,$ for some $k \in \mathbb{N}$

Gaussian Kernel $\quad k(\boldsymbol{x}, \boldsymbol{y}) = \exp\left(-\frac{1}{2\sigma^2}\|\boldsymbol{x} - \boldsymbol{y}\|_2^2\right),$ for some $\sigma > 0$



Another common way to create kernels is with a **feature map** $\varphi : \Omega \to \mathcal{H}$ where $\mathcal{H}$ is a Hilbert space (typically $\mathbb{R}^D$):

$$k(\boldsymbol{x}, \boldsymbol{y}) = \langle \varphi(\boldsymbol{x}), \varphi(\boldsymbol{y}) \rangle_{\mathcal{H}}$$

Example: $\varphi : \mathbb{R} \to \mathbb{R}^3$ given by $\varphi(x) = [1, \sqrt{2}x, x^2]^\mathsf{T}$ gives the polynomial kernel $k(x, y) = 1 + 2xy + x^2y^2 = (xy + 1)^2$

Example: Tangent Kernels

Given a parametric model $f_{\boldsymbol{\theta}}(\boldsymbol{x}) = f(\boldsymbol{\theta}; \boldsymbol{x})$, linearizing about $\boldsymbol{\theta} = \boldsymbol{\theta}_0$ gives

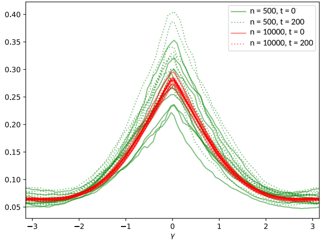
$$f(\boldsymbol{\theta}; \boldsymbol{x}) \approx f(\boldsymbol{\theta_0}; \boldsymbol{x}) + \nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta}_0; \boldsymbol{x})^{\mathsf{T}} (\boldsymbol{\theta} - \boldsymbol{\theta}_0).$$

For any fixed $\boldsymbol{\theta}$ define the **tangent kernel** $k_{\boldsymbol{\theta}}$

$$k_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{x}') := \nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta}; \boldsymbol{x})^{\mathsf{T}} \nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta}; \boldsymbol{x}'), \quad \text{for all} \quad \boldsymbol{x}, \boldsymbol{x}' \in \Omega$$

which is the kernel arising from the feature map $\phi(\boldsymbol{x}) = \nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta}, \boldsymbol{x})$.

Jacot et al. 2018 showed that when $f_{\boldsymbol{\theta}}$ is a **randomly initialized neural network architecture**, in the limit of the hidden-layer widths approaching infinity, $k_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{x}')$ converges to an explicit kernel that stays constant during training–the **neural tangent kernel**.



Infinitely-wide neural network architectures define kernels

Does this RKHS perspective explain the astounding success of neural networks?

RKHS Representer Theorem

Theorem

Let $\mathcal{F}$ be an RKHS with kernel $k : \Omega \times \Omega \to \mathbb{R}$. Fix $\lambda > 0$. Then

$$f^* \in \arg\min_{f \in \mathcal{F}} \sum_{i=1}^{n} \mathcal{L}(f(\boldsymbol{x}_i), y_i) + \lambda \|f\|_{\mathcal{F}}^2 \Leftrightarrow f^*(\boldsymbol{x}) = \sum_{i=1}^{n} a_i k(\boldsymbol{x}, \boldsymbol{x}_i).$$

for some $a_i \in \mathbb{R}$.

Proof sketch for square-loss:

$$\min_{f \in \mathcal{F}} \sum_{i=1}^{n} (f(\boldsymbol{x}_i) - y_i)^2 + \lambda \|f\|_{\mathcal{F}}^2 = \min_{f \in \mathcal{F}} \sum_{i=1}^{n} (\langle K_{\boldsymbol{x_i}}, f \rangle_{\mathcal{F}} - y_i)^2 + \lambda \langle f, f \rangle_{\mathcal{F}}$$

Set the "derivative" $\partial / \partial f$ of the loss to zero:

$$\sum_{i=1}^{n} 2(\langle K_{\boldsymbol{x_i}}, f^* \rangle_{\mathcal{F}} - y_i) K_{\boldsymbol{x}_i} + 2\lambda f^* = 0 \implies f^* = \sum_{i=1}^{n} a_i K_{\boldsymbol{x}_i}.$$

Wahba 1990; Schölkopf and Smola 2002

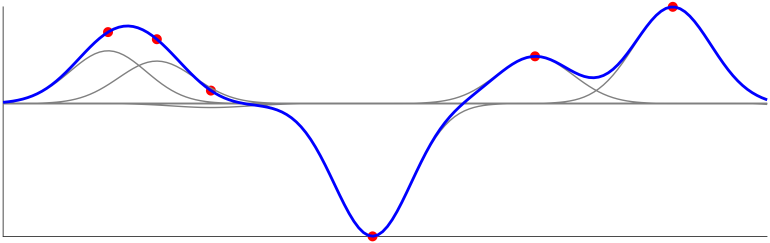
The "Kernel Trick"

Restricting to only functions of the form $f = \sum_{i=1}^n a_i k(\cdot, \boldsymbol{x}_i)$ we have

$$\min_{f \in \mathcal{F}} \sum_{i=1}^n \mathcal{L}(y_i, f(\boldsymbol{x}_i)) + \lambda \|f\|_{\mathcal{F}}^2 \quad \Leftrightarrow \quad \min_{\boldsymbol{a} \in \mathbb{R}^n} \sum_{i=1}^n \mathcal{L}(y_i, [\boldsymbol{K}\boldsymbol{a}]_i) + \lambda \boldsymbol{a}^\mathsf{T} \boldsymbol{K} \boldsymbol{a}$$

where $\boldsymbol{K} \in \mathbb{R}^{n \times n}$ is the kernel matrix $K_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$. This is now a finite dimensional optimization problem we can solve easily.

Example: Gaussian RBF kernel $f(x) = \sum_{i=1}^n a_i \exp(-\frac{1}{2\sigma^2}(x - x_i)^2)$



Example: Smoothing Splines

The solution to

$$\min_f \sum_{i=1}^n (y_i - f(x_i))^2 + \lambda \int_0^1 |\mathrm{D}^2 f(x)|^2 \, \mathrm{d}x$$

$\|\mathrm{D}^2 f\|_{L^2}^2$

is a cubic (smoothing) spline,

$$f_{\text{spline}}(x) = \sum_{i=1}^n a_i^\star \, k(x, x_i),$$

where $\boldsymbol{a}^\star = \arg\min_{\boldsymbol{a} \in \mathbb{R}^n} \|\boldsymbol{y} - \mathbf{K}\boldsymbol{a}\|_2^2 + \lambda \boldsymbol{a}^\mathsf{T} \mathbf{K} \boldsymbol{a}$.

quadratic regularizer $\Rightarrow$
solution linear in data $\boldsymbol{y}$

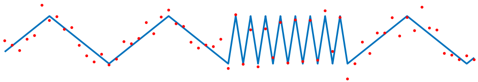
If $y_i = f^\star(x_i) + \varepsilon_i$ with $f^\star \in H^2$, then

$$\mathbf{E}\|f^\star - f_{\text{spline}}\|_{L^2}^2 = O(n^{-\frac{4}{5}}).$$

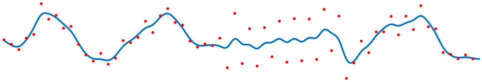
minimax rate

van de Geer 2000

Limitations of Linear/Kernel Methods

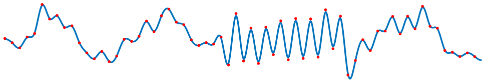


True function and noisy data



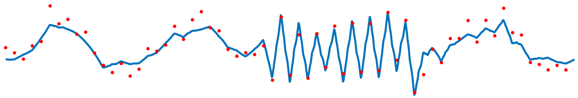
large λ:
oversmooths high variation portion of the data

small λ:
overfits low variation portion of the data

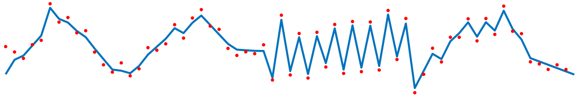


Linear methods cannot adapt to spatially varying smoothness.

Limitations of Linear/Kernel Methods

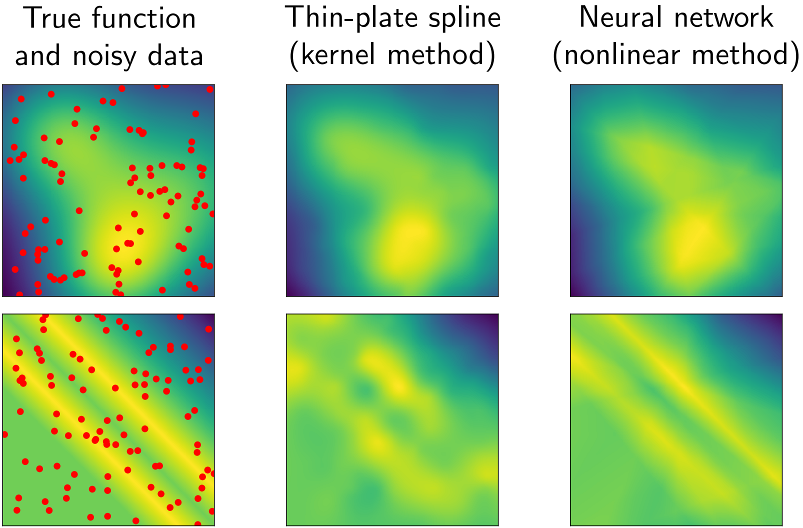


wavelet shrinkage



neural network

Limitations of Linear/Kernel Methods



Neural networks can adapt to **low-dimensional structure**.

Summary

- Kernel methods are well-understood from the function-space view.
  - $\Longrightarrow$ Essentially by construction.
  - $\Longrightarrow$ There is a one-to-one correspondance between a kernel $k(\cdot, \cdot)$ and their associated RKHS $\mathcal{H}_k$.
- We know when kernel methods work and how well they work.
  - $\Longrightarrow$ Kernel methods are "optimal" for learning functions in their associated RKHS.
- We know that there are situations where they do not work.
  - $\Longrightarrow$ There are fundamental drawbacks to linear methods.

Outline

1. Hilbert Spaces $\Leftrightarrow$ Linear/Kernel Methods
2. Banach Spaces $\Leftrightarrow$ Nonlinear/Sparse Methods
3. Banach Spaces $\Leftrightarrow$ Shallow Neural Networks
4. Beyond(?) Banach Spaces $\Leftrightarrow$ Deep Neural Networks

Banach Spaces - Basic Definition

Assume $\mathcal{F}$ is a **vector space of functions** defined on a domain $\Omega \subset \mathbb{R}^d$.

We will focus on functions with real outputs (scalar- or vector-valued).
We say that $\|\cdot\|_{\mathcal{F}} : \mathcal{F} \to \mathbb{R}_{\geq 0}$ is a **norm** if it is:

- *subadditive:* $\|f + g\|_{\mathcal{F}} \leq \|f\|_{\mathcal{F}} + \|g\|_{\mathcal{F}}$
- *homogeneous:* $\|\alpha f\|_{\mathcal{F}} = |\alpha| \|f\|_{\mathcal{F}}$
- *positive definite:* $\|f\|_{\mathcal{F}} = 0$ if and only if $f \equiv 0$

Remark: Every inner product $\langle \cdot, \cdot \rangle$ induces a valid norm: $\|f\|^2 \coloneqq \langle f, f \rangle$.

Definition

A **Banach space** is a vector space $\mathcal{F}$ equipped with a norm $\|\cdot\|_{\mathcal{F}}$ that is *complete* with respect to the norm $\|\cdot\|_{\mathcal{F}}$.

Reproducing Kernel Banach Spaces

Definition

A Banach space of functions $\mathcal{F}$ is called a **Reproducing Kernel Banach Space (RKBS)** if its norm $\|\cdot\|_{\mathcal{F}}$ is strictly convex, if its dual norm $\|\cdot\|_{\mathcal{F}'}$ is strictly convex, and for all $\boldsymbol{x} \in \Omega$ there exists a constant $C_{\boldsymbol{x}}$ such that

$$|f(\boldsymbol{x})| \leq C_{\boldsymbol{x}} \|f\|_{\mathcal{F}} \text{ for all } f \in \mathcal{F}.$$

- Strict convexity of the norm and dual norm ensures the existence of a **unique** reproducing kernel $k : \Omega \times \Omega \to \mathbb{R}$ with $k(\boldsymbol{x}, \cdot) \in \mathcal{F}$ and $k(\cdot, \boldsymbol{x}) \in \mathcal{F}'$ and

$$\begin{aligned} \langle k(\boldsymbol{x}, \cdot), f \rangle &= f(\boldsymbol{x}), \quad \text{for all } f \in \mathcal{F}. \\ \langle k(\cdot, \boldsymbol{x}), f \rangle &= f(\boldsymbol{x}), \quad \text{for all } f' \in \mathcal{F}'. \end{aligned}$$

- The RKBS shares many similarities to the RKHS framework, but reflexivity is too strong of a condition to capture important spaces related to **sparsity**.

Zhang et al. 2009; Lin et al. 2022; Bartolucci et al. 2023

Sparsity = Feature Learning?

In Hilbert-space methods, the learned models are **linear in parameters**.

Linear methods cannot adapt to spatially varying smoothness.

Linear methods do not **learn features**.

Early approaches to circumvent this issue were based on sparsity:

- lasso (Tibshirani 1996)
- sparse approximation (DeVore 1998)
- wavelet shrinkage/thresholding (Donoho and Johnstone 1998)
- compressed sensing (Candès et al. 2006; Donoho 2006)

Sparse methods are **nonlinear in parameters**.

Is sparsity key to understand **feature learning**?

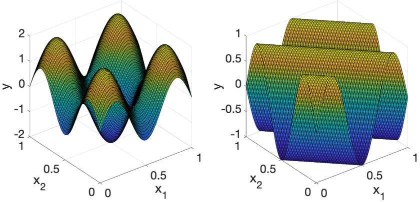
Sparsity and the Quest for Adaptivity

Latent Variable Perspective: The target function depends only on an r-dimensional projection ($r \ll d$) of the input.

Manifold Hypothesis: d-dimensional data sets that occur in the real world actually lie along r-dimensional latent manifolds ($r \ll d$).

Structured Smoothness: The target function has some unknown structured smoothness.

Can we design methods that **adapt** to the **unknown** structure?



Sparsity $\Rightarrow$ adaptation

Adaptation $=$ feature learning?

Bach 2017

Sparse Models: Finite-Dimensional Case

Returning to our simplest parametric model, where $\mathcal{F}$ is the linear span of a **dictionary** of finitely many functions $\varphi_1, \ldots, \varphi_K$.

$$f_{\boldsymbol{\theta}} = \sum_{k=1}^{K} \theta_k \varphi_k, \quad C(\boldsymbol{\theta}) = \|\boldsymbol{\theta}\|_1$$

$$\mathcal{F}_\Theta = \mathcal{F} = \operatorname{span}\{\varphi_k\}_{k=1}^{K}$$

$$R(f) = \inf_{\boldsymbol{\theta} : f = f_{\boldsymbol{\theta}}} \|\boldsymbol{\theta}\|_1$$

The learning problem is

$$\min_{\boldsymbol{\theta} \in \mathbb{R}^K} \sum_{i=1}^{n} \mathcal{L}(y_i, [\boldsymbol{V}\boldsymbol{\theta}]_i) + \lambda \|\boldsymbol{\theta}\|_1,$$

where the ith row of $\boldsymbol{V} \in \mathbb{R}^{n \times K}$ is

$$\boldsymbol{V} = [\varphi_1(x_i), \varphi_2(x_i), \ldots, \varphi_K(x_i)]$$

- Data-fitting over $\mathcal{F}$ is equivalent to a **finite-dimensional convex** optimization problem.
- There always exists a solution with at most K dictionary functions.

Sparse Models: Infinite-Dimensional Case

What if we had an infinitely large dictionary? $\{\varphi_k\}_{k\in\mathbb{Z}}$

$$f_{\boldsymbol{\theta}} = \sum_{k\in\mathbb{Z}} \theta_k \varphi_k, \quad C(\boldsymbol{\theta}) = \|\boldsymbol{\theta}\|_{\ell^1}$$

$$\mathcal{F}_\Theta \subset \mathcal{F} = \overline{\operatorname{span}}\{\varphi_k\}_{k\in\mathbb{Z}}$$

$$R(f) = \inf_{\boldsymbol{\theta}: f = f_{\boldsymbol{\theta}}} \|\boldsymbol{\theta}\|_1$$

The learning problem is

$$\min_{\boldsymbol{\theta}\in\ell^1(\mathbb{Z})} \sum_{i=1}^n \mathcal{L}(y_i, \mathrm{V}\{\boldsymbol{\theta}\}) + \lambda\|\boldsymbol{\theta}\|_{\ell^1},$$

where $\mathrm{V} : \ell^1(\mathbb{Z}) \to \mathbb{R}^n$.

- Data-fitting over $\mathcal{F}$ is equivalent to a **infinite-dimensional convex** optimization problem.
- Solutions have infinitely many dictionary functions? Are we screwed?

Luckily, we are not.

Intuition: Soft Thresholding

Consider the "denoising" problem

$$\min_{\boldsymbol{\theta} \in \ell^1(\mathbb{Z})} \|\boldsymbol{y} - \boldsymbol{\theta}\|_2^2 + \lambda \|\boldsymbol{\theta}\|_{\ell^1} = \min_{\boldsymbol{\theta} \in \ell^1(\mathbb{Z})} \sum_{k \in \mathbb{Z}} \left[(y_k - \theta_k)^2 + \lambda |\theta_k| \right]$$

Pop Quiz: What is the solution to this problem?

This problem can be "decoupled".

$$\min_{\theta_k \in \mathbb{R}} (y_k - \theta_k)^2 + \lambda |\theta_k| \quad \Rightarrow \quad \text{soft thresholding of } y_k$$

$$\Rightarrow \quad \widehat{\theta}_k = \operatorname{sgn}(y_k) \max\{0, |y| - \lambda/2\}$$

Since $\boldsymbol{\theta} \in \ell^1(\mathbb{Z})$, the sorted coefficients $|\theta_{(1)}| \geq |\theta_{(2)}| \geq \cdots$ must decay strictly faster than $1/k$.

For every $\lambda > 0$, only a finite number of coefficients will be nonzero.

Soft thresholding is **nonlinear in parameters**.

Donoho and Johnstone 1995

Representer Theorems for ℓ^1-Norm Regularization

The **sparsity** of solutions is related to the **convex geometry** of the optimization problem.

$$
\min_{\boldsymbol{\theta} \in \ell^1(\mathbb{Z})} \sum_{i=1}^{n} \mathcal{L}(y_i, \mathrm{V}\{\boldsymbol{\theta}\}) + \lambda \|\boldsymbol{\theta}\|_{\ell^1} \quad \Leftrightarrow \quad \min_{\boldsymbol{\theta} \in \ell^1(\mathbb{Z})} \|\boldsymbol{\theta}\|_{\ell^1} \quad \text{s.t.} \quad \sum_{i=1}^{n} \mathcal{L}(y_i, \mathrm{V}\{\boldsymbol{\theta}\}) \leq B
$$

$$
\Leftrightarrow \quad \min_{\boldsymbol{\theta} \in \ell^1(\mathbb{Z})} \|\boldsymbol{\theta}\|_{\ell^1} \text{ s.t. } \mathrm{V}\{\boldsymbol{\theta}\} \in \mathcal{C} \subset \mathbb{R}^n
$$

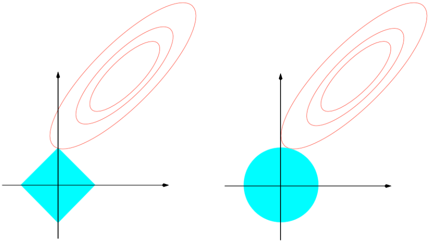
There exists a solution $\widehat{\boldsymbol{\theta}}$ that is n-sparse.

- Tightly linked to Carathéodory's theorem for convex hulls.
- This is an example of a **Banach-space representer theorem**.

$\Longrightarrow$ $\ell^1(\mathbb{Z})$ is a non-Hilbertian Banach space.

Chandrasekaran et al. 2012

Geometry of ℓ^1-Norm Regularization



The **extreme points** of the ℓ^1-ball are 1-sparse vectors.

1-sparse vectors are **Kronecker deltas**:

$$\mathrm{e}_k[n] = \delta_k[n] = \begin{cases} 1, & \text{if } n = k \\ 0, & \text{else.} \end{cases}$$

Pop Quiz: What are the extreme points of the ℓ^2-ball?

Convex Optimization in Infinite Dimensions?

$$\min_{\boldsymbol{\theta} \in \ell^1(\mathbb{Z})} \sum_{i=1}^{n} \mathcal{L}(y_i, \mathrm{V}\{\boldsymbol{\theta}\}) + \lambda \|\boldsymbol{\theta}\|_{\ell^1}$$

Convex problem, but infinite-dimensional.

BUT, we know there exists a solution $\widehat{\boldsymbol{\theta}}$ that is n-sparse:

$$f_{\widehat{\boldsymbol{\theta}}} = \sum_{j=1}^{n} \theta_{[j]} \varphi_{[j]}$$

Pop Quiz: Can we just optimize over n-sparse sequences?

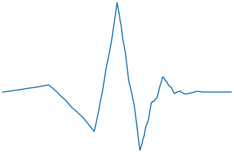
Yes, but the problem becomes **nonconvex**.

Wavelet Shrinkage

$$f_{\boldsymbol{\theta}}(x) = \sum_{j,k} \theta_{j,k}\, 2^{-j/2} \psi(2^j x - k)$$

$$C(\boldsymbol{\theta}) = \|\boldsymbol{\theta}\|_{\ell^1}$$

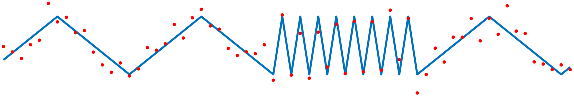
$$\min_{\boldsymbol{\theta}} \sum_{i=1}^{n} \mathcal{L}(y_i, f_{\boldsymbol{\theta}}(x_i)) + \lambda \|\boldsymbol{\theta}\|_{\ell^1}$$



- We can **efficiently** solve this optimization problem by thresholding the empirical wavelet coefficients.
- The resulting solution is able to **adapt** to intrinsic structure in the data-generating function.

Donoho and Johnstone 1994

Spatially Inhomogeneous Functions

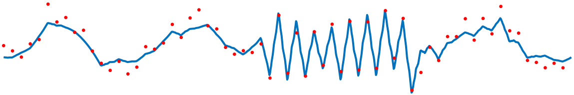


Spatially inhomogeneous functions have different kinds of **local regularity**.

Designing **locally adaptive** estimators was of great interest in the 1990s.

- Real-world signals (**low-dimensional** objects) are spatially inhomogeneous.
- Linear methods cannot adapt to spatial inhomogeneities while sparse/nonlinear methods can.

$\Longrightarrow$ Mathematical foundations for the success and popularity of wavelets.



What kinds of function spaces capture spatial inhomogeneities?

Donoho et al. 1990

Besov Spaces and Bounded Variation (BV) Spaces

Spatially inhomogeneous functions are well-captured by **Besov** and **Bounded Variation** (BV) spaces.

- The Besov space $B^s_{p,q}$ is the space of functions with s derivatives in L^p, where q allows for finer control of the regularity:

$$B^s_{p,q} \subset B^s_{p,q'} \quad \Leftrightarrow \quad q < q'$$

$\Longrightarrow$ When $p < 2$, $B^s_{p,q}$ contains functions that are spatially inhomogeneous.

- The (total) variation of a function is

$$\mathrm{TV}(f) = \sup \sum_{i=0}^{n-1} |f(x_{i+1}) - f(x_i)| \ \left(\approx \int |f'(x)| dx \right).$$

where the $\sup$ is over all partitions. BV is the space of functions with bounded (total) variation $\mathrm{TV}(f) < \infty$.

$\Longrightarrow$ $f \in \mathrm{BV}^k \Leftrightarrow f^{(k-1)} \in \mathrm{BV} \Leftrightarrow \mathrm{TV}^k(f) = \mathrm{TV}(f^{(k-1)})$.

- BV^k is morally a Besov space since we have the sandwich

$$B^k_{1,1} \subset \mathrm{BV}^k \subset B^k_{1,\infty}.$$

DeVore 1998

Native Space of Wavelet Soft Thresholding

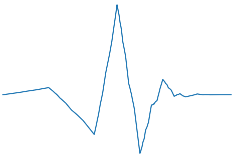
$$\left\{ f_{\boldsymbol{\theta}} = \sum_{j,k} \theta_{j,k}\psi_{j,k} : \begin{array}{l} \psi_{j,k}(x) = \\ 2^{-j/2}\psi(2^j x - k) \end{array} \right\}$$

$$C(\boldsymbol{\theta}) = \|\boldsymbol{\theta}\|_{\ell^1}$$

$\mathcal{F}$: Besov space $B^1_{1,1}$

$R(f) = \|f\|_{B^1_{1,1}}$
Besov norm

$$\min_{\boldsymbol{\theta}} \sum_{i=1}^{n} \mathcal{L}(y_i, f_{\boldsymbol{\theta}}(x_i)) + \lambda\|\boldsymbol{\theta}\|_{\ell^1} \quad \Leftrightarrow \quad \min_{f \in B^1_{1,1}} \sum_{i=1}^{n} \mathcal{L}(y_i, f(x_i)) + \lambda\|f\|_{B^1_{1,1}}$$



- ℓ^1-limits of wavelet coefficients converge to $B^1_{1,1}$ functions
  - $\Longrightarrow$ Other choices of sequence space norms give rise to **all** other Besov spaces $B^s_{p,q}$.
- There always exists a solution to the problem over $B^1_{1,1}$ that is a **sparse and finite** combination of wavelets.
  - $\Longrightarrow$ Soft-thresholding is the algorithm of choice.

Meyer 1992

Integral Representations of Functions

For smooth functions on $[0, 1]$, by the fundamental theorem of calculus, we have that

$$f(x) = f(0) + \int_0^x f'(t)\,\mathrm{d}t \quad \Rightarrow \quad f(x) = f(0) + \int_0^1 \mathrm{ReLU}^0(x - t) f'(t)\,\mathrm{d}t.$$

If we iterate this process...

$$f(x) = \sum_{j=0}^{k-1} f^{(j)}(0) x^j + \int_0^1 \mathrm{ReLU}^{k-1}(x - t) f^{(k)}(t)\,\mathrm{d}t$$

Pop Quiz: When is this quantity well-defined?

- $\|f^{(k)}\|_{L^1} < \infty$.
- Suppose $k = 1$ and $f(x) = \mathrm{ReLU}^0(x - 0.5)$.

$\implies \quad f'(x) = \delta_{0.5}$.

$\implies \quad f(x) = \int_0^1 \mathrm{ReLU}^0(x - t)\,\mathrm{d}\delta_{0.5}$.

- Identify $f^{(k)}$ with a measure $\nu \Rightarrow \|\nu\|_{\mathcal{M}} < \infty$.

Fisher and Jerome 1975

BV Spaces and Continuously-Indexed Dictionaries

For smooth functions, we have that

$$\mathrm{TV}(f) = \sup \sum_{i=0}^{n-1} |f(x_{i+1}) - f(x_i)| = \int_0^1 |f'(x)| \,\mathrm{d}x.$$

For nonsmooth functions,

$$\mathrm{TV}(f) = \|f'\|_{\mathcal{M}} \quad \Rightarrow \quad \mathrm{TV}^k(f) = \|\mathrm{D}^k f\|_{\mathcal{M}}.$$

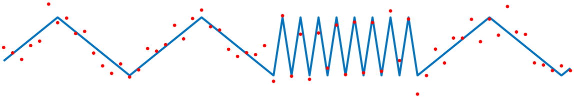
If $f \in \mathrm{BV}^k$ and $\nu = \mathrm{D}^k f$, then,

$$f(x) = \sum_{j=1}^{k-1} c_j x^j + \int_0^1 \mathrm{ReLU}^{k-1}(x - t) \,\mathrm{d}\nu(t)$$

- Peano kernel formula
- Infinite-width neural network?
- $f \in \mathrm{BV}^k \Rightarrow f$ can be build from the **continuously-indexed dictionary** $\{\mathrm{ReLU}^{k-1}(\cdot - t)\}_{t \in [0,1]}$.

Fisher and Jerome 1975

Minimax Optimality of Nonlinear Methods



Pop Quiz: What is the regularity of this function?

- This is a BV^2 function.
- The **minimax rate** for BV^2 is

$$\inf_{\widehat{f}} \sup_{\substack{f \in \mathrm{BV}^2 \\ \mathrm{TV}^2(f) \leq C}} \mathbf{E}\|f - \widehat{f}\|_{L^2}^2 \asymp n^{-4/5}$$

smoothing spline	wavelet shrinkage
$n^{-3/4}$	$n^{-4/5}$

- $n^{-3/4}$ is the **linear minimax rate**

Donoho and Johnstone 1998; Mammen and Geer 1997

Sparsity in the Continuum

Discrete sparse model:

$$\sum_{k\in\mathbb{Z}} \theta_k \varphi_k, \; C(\boldsymbol{\theta}) = \|\boldsymbol{\theta}\|_{\ell^1}$$

Continuous sparse model:

$$\int_{\Xi} \varphi_\xi \, \mathrm{d}\nu(\xi), \; C(\nu) = \|\nu\|_{\mathcal{M}}$$

The continuous model is "backwards compatible" with the discrete model.

$$\left\| \sum_{k\in\mathbb{Z}} \theta_k \delta_{\xi_k} \right\|_{\mathcal{M}} = \sum_{k\in\mathbb{Z}} |\theta_k| = \|\boldsymbol{\theta}\|_{\ell^1}$$

Pop Quiz: Why don't we use the L^1-norm for continuous sparsity?

- The extreme point of the unit ball of $\ell^1(\mathbb{Z})$ are the **Kronecker deltas**: $\{\mathrm{e}_k\}_{k\in\mathbb{Z}}$.
- The extreme points of the unit ball of $\mathcal{M}(\Xi)$ are the **Dirac deltas/measures**: $\{\delta_\xi\}_{\xi\in\Xi}$.
- The extreme points of the unit ball of $L^1(\Xi)$...**do not exist**.

Variation Spaces

Given a continuously-indexed dictionary $\mathcal{D} = \{\varphi_\xi\}_{\xi \in \Xi}$, the **variation space** of $\mathcal{D}$ is the space

$$\mathcal{V} = \mathcal{V}(\mathcal{D}) = \left\{ f_\nu = \int_\Xi \varphi_\xi \, \mathrm{d}\nu(\xi) \,:\, \nu \in \mathcal{M}(\Xi) \right\}.$$

This forms a Banach space when equipped with the norm/representation cost

$$R(f) = \|f\|_{\mathcal{V}} = \inf_{\substack{\nu \in \mathcal{M}(\Xi) \\ f = f_\nu}} \|\nu\|_{\mathcal{M}}$$

Example: BV^k (modulo polynomials) is a variation space with respect to the $\{\mathrm{ReLU}^{k-1}(\cdot - t)\}_{t \in [0,1]}$ dictionary.

The variation norm is the continuous counterpart of the **atomic norm**.

Barron 1993; Kůrková and Sanguineti 2001; Mhaskar 2004; Bach 2017

Variation Spaces as a Closure

$$f_{\boldsymbol{\theta}} = \sum_{k=1}^{K} \theta_k \varphi_{\xi_k}, \quad C(\boldsymbol{\theta}) = \|\boldsymbol{\theta}\|_{\ell^1}$$

$$\mathcal{F} = \overline{\operatorname{span}}\{\varphi_\xi\}_{\xi \in \Xi}$$

If the dictionary $\mathcal{D} = \{\varphi_\xi\}_{\xi \in \Xi}$ is sufficiently regular, then

$$\mathcal{V}(\mathcal{D}) = \mathcal{F}$$

Observation: Variation spaces are ℓ^1-limits of combinations from continuous dictionaries.

Donoho 2000; Siegel and Xu 2023

Meyer's Bump Algebra

Let $K_{\boldsymbol{x}}$ denote the Gaussian kernel centered at $\boldsymbol{x}$. Consider the dictionary $\mathcal{G} = \{K_{\boldsymbol{x}}\}_{\boldsymbol{x} \in \Omega}$, where $\Omega \subset \mathbb{R}^d$.

The variation space $\mathcal{V}(\mathcal{G})$ is called **Meyer's Bump Algebra**.

$$\mathcal{V}(\mathcal{G}) = B^d_{1,1}(\Omega)$$

On the other hand, the RKHS $\mathcal{H}$ generated by $\mathcal{G}$ is **fundamentally different**.

$$\mathcal{H} \subset H^s(\Omega) \text{ for any } s \geq 0$$

- $\mathcal{H}$ is an extremely small space.
- $\mathcal{G}$ is reasonably large.

$\implies \mathcal{H} \subsetneq \mathcal{G}$

Pop Quiz: Why did we get different spaces from the same parametric model space $\mathcal{F}_\Theta$?

This is alluding to a gap between Hilbert- and Banach-space methods.

Meyer 1992

Representer Theorems for $\mathcal{M}$-Norm Regularization

Theorem

Fix a dictionary $\mathcal{D} = \{\varphi_\xi\}_{\xi \in \Xi}$ of continuous functions. For any data set $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ and lower semicontinuous $\mathcal{L}(\cdot, \cdot)$, there exists a solution $\widehat{\nu}$ to

$$\min_{\nu \in \mathcal{M}(\Xi)} \sum_{i=1}^{n} \mathcal{L}(y_i, f_\nu(\boldsymbol{x}_i)) + \lambda \|\nu\|_{\mathcal{M}}, \quad \lambda > 0,$$

that admits a representation of the form

$$f_{\widehat{\nu}}(\boldsymbol{x}) = \sum_{k=1}^{K} v_k \varphi_{\xi_k}(\boldsymbol{x}), \quad K \leq n.$$

There always exists a solution that is a K-sparse sum of dictionary functions.

Fisher and Jerome 1975

Convex or Nonconvex?

$$\min_{\nu \in \mathcal{M}(\Xi)} \sum_{i=1}^{n} \mathcal{L}(y_i, f_\nu(\boldsymbol{x}_i)) + \lambda \|\nu\|_{\mathcal{M}}$$

Convex problem, but infinite-dimensional.

BUT, we know there exists a solution $\widehat{\nu}$ that is K-sparse:

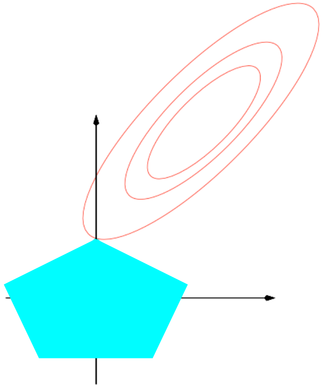
$$f_{\widehat{\nu}}(\boldsymbol{x}) = \sum_{k=1}^{K} v_k \varphi_{\xi_k}(\boldsymbol{x})$$

Pop Quiz: Can we just optimize over K-sparse sums?

Yes, but the problem becomes **nonconvex**.

$$\min_{\substack{v_1, \ldots, v_K \\ \xi_1, \ldots, \xi_K}} \sum_{i=1}^{n} \mathcal{L}\left(y_i, \sum_{k=1}^{K} v_k \varphi_{\xi_k}(\boldsymbol{x}_i)\right) + \lambda \|\boldsymbol{v}\|_1$$

Geometry of Convex Regularization



The "atoms" of the solution are the **extreme points** of the **regularization ball**.

Chandrasekaran et al. 2012

Abstract Representer Theorems

Theorem

Consider the learning problem over the function space $\mathcal{F}$

$$\inf_{f \in \mathcal{F}} \sum_{i=1}^{n} \mathcal{L}(y_i, f(\boldsymbol{x}_i)) + \lambda R(f).$$

Under appropriate hypotheses on $\mathcal{F}$ and R, there always exists a **sparse** solution of the form

$$\widehat{f} = \sum_{k=1}^{K} v_k \mathrm{e}_k, \quad K \leq n,$$

where $\mathrm{e}_k \in \mathrm{Ext}(\{f \in \mathcal{F} : \ R(f) \leq 1\})$.

Infinite-dimensional optimization problems with **finite data constraints** always admit **sparse solutions**.

Boyer et al. 2019; Bredies and Carioni 2020; Unser 2021

Summary

- Sparsity allows for methods to **adapt** to structure.
  - $\Longrightarrow$ Linear methods (including kernel methods) cannot.
  - $\Longrightarrow$ Quantification via minimax and linear minimax rates.
- Infinite-dimensional sparse models correspond to **convex problems**.
  - $\Longrightarrow$ These problems can be recast as **finite-dimenensional non-convex problems**.
  - $\Longrightarrow$ This will play a key role in understanding neural networks from the function-space view.
- The infinite-dimensional perspective reveals interesting aspects about the **geometry of convex regularization**.
  - $\Longrightarrow$ Abstract representer theorems and extreme points.

BREAK

Outline

What is the Inductive Bias Shallow Neural Networks?

What kinds of functions do neural networks prefer?

930 IEEE TRANSACTIONS ON INFORMATION THEORY, VOL. 39, NO. 3, MAY 1993

Universal Approximation Bounds for Superpositions of a Sigmoidal Function

Andrew R. Barron, *Member, IEEE*

Andrew Barron

Barron (1993) introduced a class of d-dimensional functions that can be approximated **extremely well** by neural networks.

- Such functions can be approximated by a neural network with K neurons at a rate $K^{-\frac{1}{2}}$.
- Rates for classical function classes behave as $K^{-\frac{s}{d}}$ the curse

$\implies$ Andrew Barron broke the curse of dimensionality!

Spectral Barron Spaces

Let $\mathscr{B}^s \subset L^1(\mathbb{R}^d)$ be the space of functions for which

$$\|f\|_{\mathcal{B}^s} := \int_{\mathbb{R}^d} |\widehat{f}(\boldsymbol{\omega})|(1 + \|\boldsymbol{\omega}\|)^s \, d\boldsymbol{\omega} < +\infty$$

Then $\mathscr{B}^s$ is a Banach space and is now referred to the sth order **spectral Barron space**.

- Barron 1993 proved that functions in $\mathscr{B}^1$ can be approximated by shallow **sigmoid** neural networks at a rate that does not grow with input dimension!

  $\Longrightarrow$ Klusowski and Barron 2018 extended this result and proved that functions in $\mathscr{B}^2$ can be approxiamted by shallow **ReLU** neural networks at a rate that does not grow with input dimension.

- Spectral Barron spaces are **variation spaces** for the dictionary

$$\{(1 + \|\boldsymbol{\omega}\|)^{-s} \mathrm{e}^{\mathrm{i}\boldsymbol{\omega}^\mathsf{T}\boldsymbol{x}}\}_{\boldsymbol{\omega} \in \mathbb{R}^d}$$

Barron 1993; Klusowski and Barron 2018; Siegel and Xu 2020

Maurey–Barron–Jones Lemma

Theorem

Fix a dictionary $\mathcal{D} = \{\varphi_\xi\}_{\xi \in \Xi}$ of bounded functions and let $\mathcal{V} = \mathcal{V}(\mathcal{D})$ be the associated variation space. Given $f \in \mathcal{V}$, there exists

$$f_K = \sum_{k=1}^{K} v_k \varphi_{\xi_k}$$

such that

$$\|f - f_K\|_{L^2(\Omega)} \leq C_0 C_\Omega \|f\|_{\mathcal{V}} K^{-1/2},$$

where C_0 is an absolute constant and

$$C_\Omega = \sup_{\xi \in \Xi} \|\varphi_\xi\|_{L^2(\Omega)}.$$

Variation spaces admit dimension-free approximation rates.

Limitations to Spectral Barron Spaces

- Spectral Barron spaces offer an incomplete description.
  - $\Longrightarrow$ $\mathscr{B}^2$ provides a **sufficient** condition for approximation by shallow ReLU networks with dimension-free rates.
  - $\Longrightarrow$ This kind of regularity is not necessary.

  Example: A single ReLU neuron does not lie in $\mathscr{B}^2$.

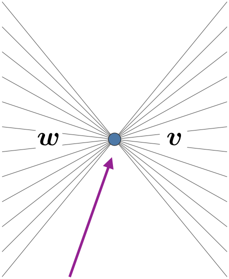
Question: What is the **largest space** of functions approximable by a shallow network (with a given activation function) at a particular estimation error rate?

This is a fundamental problem in approximation theory.

- Currently, there is not a complete characterizatio of the **approximation spaces** of shallow neural networks, but sufficient conditions are known. (DeVore et al. 2025)
  - $\Longrightarrow$ In 1D, these spaces are known and coincide with Besov spaces. (Petrushev 1988)

Neural Balance and Weight Decay

$\mathbb{R}^d \ni \boldsymbol{z}$



$\boldsymbol{v}(\boldsymbol{w}^\mathsf{T}\boldsymbol{z})_+ \in \mathbb{R}^D$

mathematical expression for a single ReLU neuron

ReLU activation

weight decay in training is equivalent to adding $\|\boldsymbol{w}\|_2^2 + \|\boldsymbol{v}\|_2^2$ to the training objective

Neural Balance Theorem

If a DNN is trained with weight decay, then the 2-norms of the input and output weights to each ReLU neuron must be **balanced**.

$$\|\boldsymbol{w}\|_2 = \|\boldsymbol{v}\|_2$$

Neural Balance

The ReLU activation is **homogeneous**

$$\boldsymbol{v}(\boldsymbol{w}^\mathsf{T}\boldsymbol{z})_+ = \gamma^{-1}\boldsymbol{v}(\gamma\boldsymbol{w}^\mathsf{T}\boldsymbol{z})_+, \quad \text{for any } \gamma > 0.$$

At a global minimizer of the weight decay objective, $\|\boldsymbol{v}\|_2 = \|\boldsymbol{w}\|_2$.

Proof. The solution to

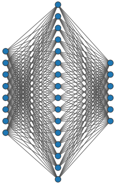
$$\min_{\gamma>0} \|\gamma^{-1}\boldsymbol{v}\|_2 + \|\gamma\boldsymbol{w}\|_2$$

is $\gamma = \sqrt{\|\boldsymbol{v}\|_2/\|\boldsymbol{w}\|_2}$. $\square$

At a global minimizer, $\dfrac{\|\boldsymbol{v}\|_2^2 + \|\boldsymbol{w}\|_2^2}{2} = \|\boldsymbol{v}\|_2\|\boldsymbol{w}\|_2$.

Grandvalet 1998

Secret Sparsity of Weight Decay



$$f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \sum_{k=1}^{K} \boldsymbol{v}_k (\boldsymbol{w}_k^{\mathsf{T}} \boldsymbol{x})_+$$

$$\boldsymbol{\theta} = \{(\boldsymbol{w}_k, \boldsymbol{v}_k)\}_{k=1}^{K}$$

weight decay

$$\min_{\boldsymbol{\theta}=\{(\boldsymbol{w}_k,\boldsymbol{v}_k)\}_{k=1}^{K}} \sum_{i=1}^{n} \mathcal{L}(\boldsymbol{y}_i, f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)) + \frac{\lambda}{2} \sum_{k=1}^{K} \|\boldsymbol{v}_k\|_2^2 + \|\boldsymbol{w}_k\|_2^2$$

path-norm

$$\min_{\boldsymbol{\theta}=\{(\boldsymbol{w}_k,\boldsymbol{v}_k)\}_{k=1}^{K}} \sum_{i=1}^{n} \mathcal{L}(\boldsymbol{y}_i, f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)) + \lambda \sum_{k=1}^{K} \|\boldsymbol{v}_k\|_2 \|\boldsymbol{w}_k\|_2$$

multitask lasso

$$\min_{\substack{\boldsymbol{\theta}=\{(\boldsymbol{w}_k,\boldsymbol{v}_k)\}_{k=1}^{K} \\ \|\boldsymbol{w}_k\|_2=1}} \sum_{i=1}^{n} \mathcal{L}(\boldsymbol{y}_i, f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)) + \lambda \sum_{k=1}^{K} \|\boldsymbol{v}_k\|_2$$

Rebalancing

Secret Sparsity of Weight Decay

weight decay $\iff$ $$\min_{\substack{\boldsymbol{\theta}=\{(\boldsymbol{w}_k,\boldsymbol{v}_k)\}_{k=1}^K \\ \|\boldsymbol{w}_k\|_2=1}} \sum_{i=1}^{n} \mathcal{L}(\boldsymbol{y}_i, f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)) + \lambda \sum_{k=1}^{K} \|\boldsymbol{v}_k\|_2$$

- Weight decay is equivalent to a **non-convex** multitask lasso.

$\Longrightarrow$ Convex reformulations of neural network training problems.

Ergen and Pilanci (2021, JMLR)
Sahiner et al. (2021, ICLR)

What Kinds of Functions Do Neural Networks Learn?

Why Do Neural Networks Work Well in High-Dimensional Problems?

Path-Norm and Representation Costs

$$\mathcal{F}_\Theta = \left\{ f(\boldsymbol{x}) = \sum_{k=1}^{K} v_k (\boldsymbol{w}_k^\mathsf{T} \boldsymbol{x})_+ \,:\, v_k \in \mathbb{R}, \boldsymbol{w}_k \in \mathbb{R}^d, \underbrace{K \in \mathbb{N}}_{\text{finite-width networks}} \right\}$$

The path-norm is a **valid norm** on $\mathcal{F}_\Theta$:

$$\boxed{\|f\|_{\mathcal{F}} = \sum_{k=1}^{K} |v_k| \|\boldsymbol{w}_k\|_2}$$

The "completion" of $\mathcal{F}_\Theta$ (in an appropriate sense) is a Banach space. It is the Banach space of all functions of the form

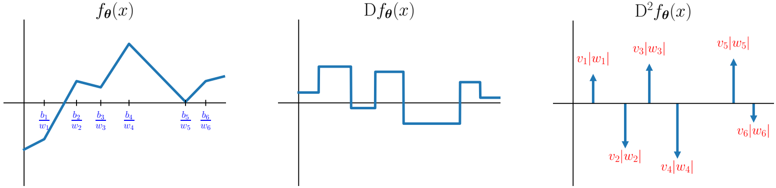
$$f(\boldsymbol{x}) = \int_{\mathbb{S}^{d-1}} (\boldsymbol{w}^\mathsf{T} \boldsymbol{x})_+ \, \boxed{\mathrm{d}\nu(\boldsymbol{w})}.$$

"output weights"

Barron (1993, IEEE Transactions on Information Theory)
Bach (2017, Journal of Machine Learning Research)
Siegel and Xu (2023, Constructive Approximation)

Path-Norm and Derivatives

$$f_{\boldsymbol{\theta}}(x) = \sum_{k=1}^{K} v_k (w_k x - b_k)_+$$



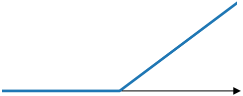
$$\text{path-norm}(f_{\boldsymbol{\theta}}) = \sum_{k=1}^{K} |v_k||w_k| = \int_{-\infty}^{\infty} |\mathrm{D}^2 f_{\boldsymbol{\theta}}(x)| \, \mathrm{d}x$$

More rigorously: total variation of $\mathrm{D} f_{\boldsymbol{\theta}}$

BV^2 is the native space for univariate shallow neural networks.

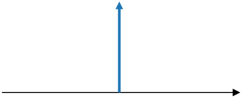
Savarese et al. 2019

What About the Multivariate Case?

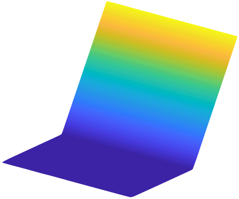


$(wx - b)_+$

$$\xrightarrow{\mathrm{D}^2}$$

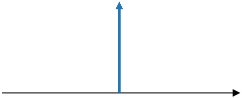


$|w|\delta(x - b/w)$



$(\boldsymbol{w}^\mathsf{T}\boldsymbol{x} - b)_+$

$$\xrightarrow{???}$$

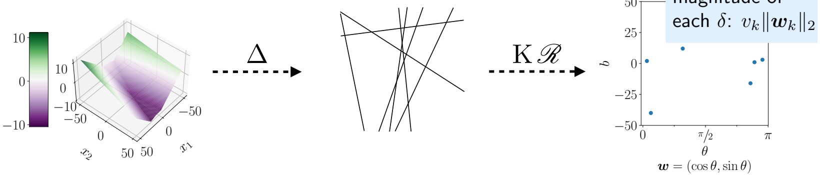


δ?

Multivariate Extension: The Radon Transform

$f_{\boldsymbol{\theta}}$ → differentiate twice → Dirac "lines" along activation thresholds → filtered Radon transform → δ at each neuron weight/bias

ReLU network



$$\text{path-norm}(f_{\boldsymbol{\theta}}) = \sum_{k=1}^{K} |v_k| \|\boldsymbol{w}_k\|_2 = \|\mathrm{K}\,\mathscr{R}\,\Delta f_{\boldsymbol{\theta}}\|_{\mathcal{M}}$$

Ongie et al. 2020

Functions of Radon Bounded Variation

Radon-domain TV^2: $\mathscr{R}\,\mathrm{TV}^2(f) := \|\mathrm{K}\,\mathscr{R}\Delta f\|_{\mathcal{M}}$

total variation
of the measure
$\mathrm{K}\,\mathscr{R}\Delta f$

$\mathrm{K}\,\mathscr{R}$ = filtered Radon transform

$\widehat{\mathrm{K}g}(\omega) \propto |\omega|^{d-1}\widehat{g}(\omega)$

$$\Delta = \sum_{k=1}^{d} \frac{\partial^2}{\partial x_k^2} = \text{Laplacian operator}$$

Average measure of **sparsity** of second derivatives along each **direction** in $\mathbb{R}^d$.

$\mathscr{R}\,\mathrm{BV}^2$ is the space of all functions on $\mathbb{R}^d$ with $\mathscr{R}\,\mathrm{TV}^2(f) < \infty$.

Banach, not Hilbert!

Ongie et al. 2020; Parhi and Nowak 2021

Representer Theorem

Neural Network Representer Theorem

For any data set $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ and lower semicontinuous $\mathcal{L}(\cdot, \cdot)$, there exists a solution to

$$\min_{f \in \mathscr{R}\,\mathrm{BV}^2} \sum_{i=1}^{n} \mathcal{L}(y_i, f(\boldsymbol{x}_i)) + \lambda \mathscr{R}\,\mathrm{TV}^2(f), \quad \lambda > 0,$$

that admits a representation of the form

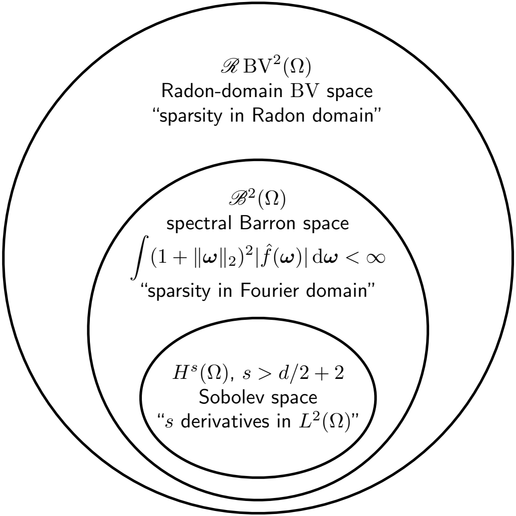
$$f_{\mathrm{ReLU}}(\boldsymbol{x}) = \sum_{k=1}^{K} v_k (\boldsymbol{w}_k^{\mathsf{T}} \boldsymbol{x} - b_k)_+ + \boldsymbol{w}_0^{\mathsf{T}} \boldsymbol{x} + b_0, \quad K < n.$$

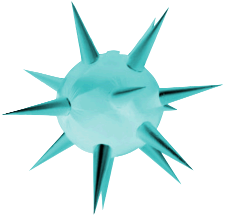
Training a **sufficiently parameterized** neural network ($K \geq N$) with weight decay (to a global minimizer) is a solution to the Banach space problem.

Neural networks learn $\mathscr{R}\,\mathrm{BV}^2$-functions.

Ongie et al. 2020; Parhi and Nowak 2021; Bartolucci et al. 2023; Unser 2023

Neural Spaces



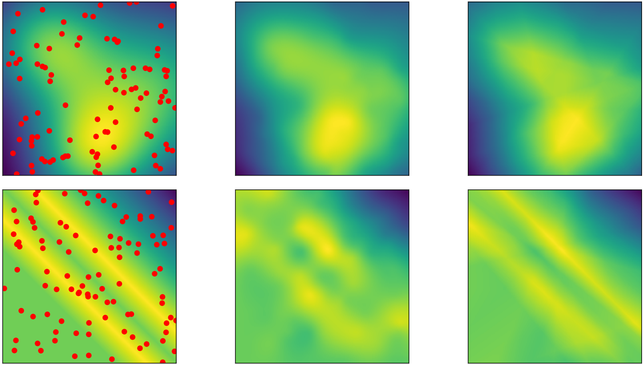


cartoon diagram of unit $\mathscr{R}\,\mathrm{BV}^2$-ball

Parhi and Nowak 2023

Adaptation to Directional Smoothness

True function and noisy data | Thin-plate spline (kernel method) | Neural network (nonlinear method)



Variation in only a **few directions** is a defining characteristic of $\mathscr{R}\,\mathrm{BV}^2$.

Breaking the Curse of Dimensionality?

Given $f \in \mathscr{R}\,\mathrm{BV}^2$, there exists a finite-width ReLU network f_K with K neurons such that

$$\|f - f_K\|_{L^\infty(\Omega)} = O\left(K^{\overset{-\alpha}{-\frac{1}{2}-\frac{3}{2d}}}\right) = O\left(K^{-\frac{1}{2}}\right).$$

Barron (1993)
Matoušek (1996)
Bach (2017)
Siegel (2023)

By the inequality of Carl (1981), this implies

$$\log \mathcal{N}(\delta, \underbrace{U(\mathscr{R}\,\mathrm{BV}^2)}_{\text{unit ball}}, \|\cdot\|_{L^\infty(\Omega)}) = \widetilde{O}\left(\delta^{\overset{-\frac{1}{\alpha}}{-\frac{2d}{d+3}}}\right) = \widetilde{O}(\delta^{-2}).$$

Approximation rates and metric entropies **do not grow** with the input dimension d.

Minimax Optimality of Shallow Neural Networks

Suppose that $\{\boldsymbol{x}_i\}_{i=1}^n$ are i.i.d. uniform on a bounded domain $\Omega \subset \mathbb{R}^d$. If $y_i = f^\star(\boldsymbol{x}_i) + \varepsilon_i$ with $\mathscr{R}\,\mathrm{TV}^2(f^\star) < \infty$, then any solution to

$$f_{\mathrm{ReLU}} \in \arg\min_{\boldsymbol{\theta}} \sum_{i=1}^n \mathcal{L}(y_i, f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)) + \frac{\lambda}{2} \sum_{k=1}^K |v_k|^2 + \|\boldsymbol{w}_k\|_2^2$$

weight decay objective

satisfies

no curse

$$\mathbf{E}\|f^\star - f_{\mathrm{ReLU}}\|_{L^2(\Omega)}^2 = \widetilde{O}(n^{-\frac{d+3}{2d+3}}) = \widetilde{O}(n^{-\frac{1}{2}}).$$

minimax rate

Linear methods (thin-plate splines, kernel methods, neural tangent kernels, etc.) **necessarily** suffer the curse of dimensionality.

Linear minimax lower bound: $n^{-\frac{3}{d+3}}$

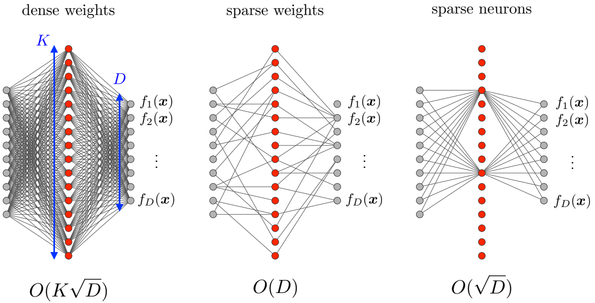
the curse

Parhi and Nowak 2023

Neuron Sharing

$$\min_{\substack{\boldsymbol{\theta}=\{(\boldsymbol{w}_k,\boldsymbol{v}_k)\}_{k=1}^K \\ \|\boldsymbol{w}_k\|_2=1}} \sum_{i=1}^{n} \mathcal{L}(\boldsymbol{y}_i, f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)) + \lambda \sum_{k=1}^{K} \|\boldsymbol{v}_k\|_2$$

weight decay
$\Longleftrightarrow$
non-convex multitask lasso



Weight decay favors variation in only a few directions (sparse weights)

Weight decay favors outputs that “share” neurons (sparse neurons)

Zeno et al. 2023; Shenouda et al. 2024; Varshney and Pilanci 2024

Depth Separation Result

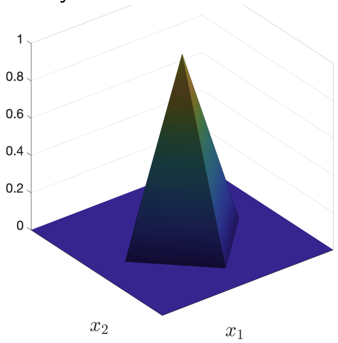
Are there fundamental differences in the native spaces described by (infinite-width) shallow ReLU networks versus deeper ReLU networks?

Put another way are there functions that have small three-layer representation costs but large, or even infinite, two-layer cost?

The answer is **yes**: Ongie et al. 2020 showed there exists a class of piecewise linear functions $f^* : \mathbb{R}^d \to \mathbb{R}$ such that $R(f^*)$ is infinite, yet they can be represented exactly by small three-layer networks.

Example: "pyramid function"

$$f^*(\boldsymbol{x}) = (1 - \|\boldsymbol{x}\|_1)_+$$



Depth Separation, Cont.

Recent work (McCarty 2023) proved that if $f : \mathbb{R}^d \to \mathbb{R}$ is a continuous piecewise linear function with finite pieces, then $R(f)$ is finite iff f is realizable as a finite-width shallow ReLU network.

Implication: If $f : \mathbb{R}^d \to \mathbb{R}$ for $d \geq 2$ is any continuous piecewise linear function with compact support, then $R(f) = +\infty$ (b/c a finite sum of ReLUs is never compactly supported).

What can be said about deep network representation costs?

Outline

1. Hilbert Spaces $\Leftrightarrow$ Linear/Kernel Methods
2. Banach Spaces $\Leftrightarrow$ Nonlinear/Sparse Methods
3. Banach Spaces $\Leftrightarrow$ Shallow Neural Networks
4. Beyond(?) Banach Spaces $\Leftrightarrow$ Deep Neural Networks

Previously we saw that shallow NN trained with parameter ℓ^2 cost are naturally associated with Banach spaces known as variation spaces.

The associated Banach space norm promotes functions that are a sparse linear combination of neurons.

Question: What are the representation costs associated with deep NNs? And what function space properties do they promote?

Question: What are the function spaces $\mathcal{F}$ associated with deep NNs? Are they fundmamentally different than variation spaces?

These are still mostly open questions!

Deep Newtorks - Overview, Cont.

Today: Highlight recent efforts to characterize the **representation costs** and **function spaces** associated with deep NNs.

1. Deep representation costs and non-linear notions of function rank.
   - $\Longrightarrow$ Warm-up: Deep Linear Networks
   - $\Longrightarrow$ Shallow ReLU nets w/multiple linear layers (Parkinson et al. 2023)
   - $\Longrightarrow$ Deep ReLU networks (Jacot 2023b; Jacot 2023a)
2. Deep compositions/hiearchies of function spaces, and representer theorems
   - $\Longrightarrow$ Compositions of Variation Spaces (Parhi and Nowak 2022)
   - $\Longrightarrow$ Compositions of RKBSs (Bartolucci et al. 2024)
   - $\Longrightarrow$ Deep Kernel Compositions (Chen 2024; Heeringa et al. 2025)

Warm-up: Deep Linear Networks

Let $\mathcal{F}_{\text{lin}}$ be the space of *linear functions* from $\mathbb{R}^d$ to $\mathbb{R}^k$. Then $f \in \mathcal{F}_{\text{lin}}$ iff

$$f(\boldsymbol{x}) = \boldsymbol{W}\boldsymbol{x}$$

for some matrix $\boldsymbol{W} \in \mathbb{R}^{k \times d}$.
Suppose we parameterize elements of $\mathcal{F}_{\text{lin}}$ as **two-layer linear networks**:

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \boldsymbol{U}\boldsymbol{V}\boldsymbol{x}$$

where $\boldsymbol{U}$ and $\boldsymbol{V}$ have inner dimension $r \geq \min\{k, d\}$, with associated parameter cost $C(\boldsymbol{\theta}) = \frac{1}{2}(\|\boldsymbol{U}\|_F^2 + \|\boldsymbol{V}\|_F^2)$.

Lemma (Burer and Monteiro 2003; Srebro et al. 2004)

$$\|\boldsymbol{W}\|_* = \min_{\boldsymbol{W}=\boldsymbol{U}\boldsymbol{V}} \frac{1}{2}\big(\|\boldsymbol{U}\|_F^2 + \|\boldsymbol{V}\|_F^2\big).$$

where $\|\boldsymbol{W}\|_*$ is the **nuclear norm** (the sum of all singular values of $\boldsymbol{W}$).

This shows the representation cost of a linear function $f(\boldsymbol{x}) = \boldsymbol{W}\boldsymbol{x}$ parametrized as a two-layer linear network is $\|\boldsymbol{W}\|_*$.

Warm-up: Deep Linear Networks

Now, suppose we parameterize elements of $\mathcal{F}_{\text{lin}}$ as L-layer linear networks:

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \boldsymbol{W}_L \cdots \boldsymbol{W}_2 \boldsymbol{W}_1 \boldsymbol{x}$$

with associated parameter cost:

$$C_L(\boldsymbol{\theta}) = \frac{1}{L} \sum_{\ell=1}^{L} \|\boldsymbol{W}_\ell\|_F^2.$$

Then if $f(\boldsymbol{x}) = \boldsymbol{W}\boldsymbol{x}$, one can show the L-layer representation cost is the **Schatten-$2/L$ quasi-norm** of $\boldsymbol{W}$ (Dai et al. 2021; Wang et al. 2023):

$$R_L(f) = \|\boldsymbol{W}\|_{S^{2/L}}^{2/L} = \sum_{i=1}^{\mathrm{rank}(\boldsymbol{W})} \sigma_i(\boldsymbol{W})^{2/L}.$$

This is a *non-convex* penalty when $L > 2$.

Generalizations to deep linear convolution networks and other structured matrix classes are considered in (Gunasekar et al. 2018; Dai et al. 2021).

Warm-up: Deep Linear Networks

For a linear function $f(\boldsymbol{x}) = \boldsymbol{W}\boldsymbol{x}$, define $\operatorname{rank}(f) = \operatorname{rank}(\boldsymbol{W})$. Then, in the limit as the number of linear layers L tends to infinity

$$\lim_{L\to\infty} R_L(f) = \lim_{L\to\infty} \|\boldsymbol{W}\|_{S^{2/L}}^{2/L} = \operatorname{rank}(\boldsymbol{W}) = \operatorname{rank}(f)$$

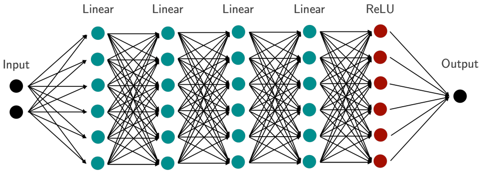
Therefore, (informally) we have:

$$\min_{\boldsymbol{\theta}} \sum_{i=1}^{n} \mathcal{L}(y_i, f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)) + \frac{\lambda}{L}\sum_{\ell=1}^{L}\|\boldsymbol{W}_\ell\|_F^2 \xrightarrow{L\to\infty} \min_{f\in\mathcal{F}_{\mathsf{lin}}} \sum_{i=1}^{n} \mathcal{L}(y_i, f(\boldsymbol{x}_i)) + \lambda \operatorname{rank}(f)$$

Deep linear networks trained with ℓ^2-regularization are biased toward **low-rank linear functions**.

Can we understand representation costs of deep *nonlinear* neural networks by alternative notions of function "rank"?

Shallow ReLU Nets with Added Linear Layers



Parameteric Model: L-layer network, first $L-1$ layers have linear activation, final layer has ReLU activation, scalar outputs.

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \boldsymbol{a}^\top \sigma(\boldsymbol{W}_{L-1} \cdots \boldsymbol{W}_1 \boldsymbol{x} + \boldsymbol{b}) + c$$

This is a **re-parameterization** of shallow ReLU networks.

Parameter Cost: $C_L(\boldsymbol{\theta}) = \frac{1}{L}\left(\|\boldsymbol{a}\|^2 + \sum_{\ell=1}^{L-1} \|\boldsymbol{W}_\ell\|_F^2\right)$.

Call the associated **representation cost** $R_L^{\text{lin}}(f)$.

How does the representation cost $R_L^{\text{lin}}(f)$ change (if at all) as the number of linear layers L increases?

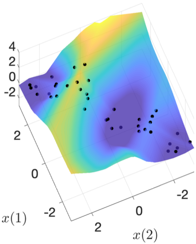
Parkinson et al. 2023

Unit Alignment Effect of Linear Layers

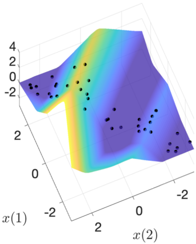
The addition of linear layers is equivalant to penalizing a non-convex Schatten quasi-norm on a single "virtual" inner-layer weight matrix:

$$R_L^{\mathsf{lin}}(f) = \min_{\boldsymbol{\theta}} \frac{1}{L}\|\boldsymbol{a}\|^2 + \frac{L-1}{L}\|\boldsymbol{W}\|_{S^{2/(L-1)}}^{2/(L-1)} \;\; s.t. \;\; f(\boldsymbol{x}) = \boldsymbol{a}^\mathsf{T}\sigma(\boldsymbol{W}\boldsymbol{x}+\boldsymbol{b})+c.$$

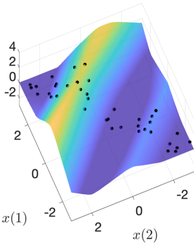
This implies $R_L^{\mathsf{lin}}(f)$ promotes functions that are realizable as a shallow ReLU network with **low-rank inner-layer weight matrix**. This can be thought of as a unit alignment effect:



(a) $L = 2$ layers



(b) $L = 3$ layers



(c) $L = 4$ layers

Can this effect be described in function space terms?

Relation to Single- and Multi-Index Models

If $f(\boldsymbol{x}) = \boldsymbol{a}^\mathsf{T}\sigma(\boldsymbol{W}\boldsymbol{x}+\boldsymbol{b})+c$ where $\boldsymbol{W}$ is rank-r, then there exists a matrix $\boldsymbol{V} \in \mathbb{R}^{d\times r}$ and function $g : \mathbb{R}^r \to \mathbb{R}$ such that

$$f(\boldsymbol{x}) = g(\boldsymbol{V}^\mathsf{T}\boldsymbol{x})$$

This is known as a **multi-index model** in the statistics literature.

The column space of $\boldsymbol{V}$ is often called the **index space**.

Estimating the index space from samples of f (and/or gradients of f) is a classical problem (Li 1991). The **expected gradient outer product (EGOP) matrix** is commonly used tool for this.

Definition

Given any weakly differentiable $f : \Omega \to \mathbb{R}$ and probability density function ρ defined over $\Omega \subset \mathbb{R}^d$, define its **EGOP matrix** $\boldsymbol{C}_f \in \mathbb{R}^{d\times d}$ by

$$\boldsymbol{C}_f := \mathbb{E}_X[\nabla f(X)\nabla f(X)^\mathsf{T}] = \int_\Omega \nabla f(\boldsymbol{x})\nabla f(\boldsymbol{x})^\mathsf{T}\rho(\boldsymbol{x})d\boldsymbol{x}$$

Index Rank of a Function

For a multi-index model $f(\boldsymbol{x}) = g(\boldsymbol{V}^\top \boldsymbol{x})$, the EGOP factors as

$$\boldsymbol{C}_f = \underbrace{\boldsymbol{V}}_{d\times r} \underbrace{E_X[\nabla g(\boldsymbol{V}^\top X)\nabla g(\boldsymbol{V}^\top X)^\top]}_{r\times r} \underbrace{\boldsymbol{V}^\top}_{r\times r}$$

Under general conditions on g, can show that $\mathrm{col}(\boldsymbol{C}_f) = \mathrm{col}(\boldsymbol{V})$.

This motivates the following definition:

Definition (Parkinson et al. 2023)

Define the **index rank** of a function f, $\mathrm{rank}_I(f)$, to be the rank of its EGOP matrix:

$$\mathrm{rank}_I(f) = \mathrm{rank}(\boldsymbol{C}_f).$$

In particular, for the multi-index model $f(\boldsymbol{x}) = g(\boldsymbol{V}^\top \boldsymbol{x})$

$$\mathrm{rank}_I(f) = \mathrm{rank}(\boldsymbol{V}).$$

under general conditions on g.

Bounds on the L-Linear-Layer Representation Cost

Theorem (Parkinson et al. 2023)

For all $f : \Omega \to \mathbb{R}$ realizable as a finite width two-layer ReLU network we have the bounds

$$\max\left\{R_2(f)^{2/L}, \|\boldsymbol{C}_f^{1/2}\|_{S^{2/L}}^{2/L}\right\} \leq R_L^{\mathsf{lin}}(f) \leq \operatorname{rank}_I(f)^{\frac{L-2}{L}} R_2(f)^{\frac{2}{L}}$$

Note: $\lim_{L\to\infty} \|\boldsymbol{C}_f^{1/2}\|_{S^{2/L}}^{2/L} = \operatorname{rank}(\boldsymbol{C}_f^{1/2}) = \operatorname{rank}(\boldsymbol{C}_f) = \operatorname{rank}_I(f)$.

Also: $\lim_{L\to\infty} \operatorname{rank}_I(f)^{\frac{L-2}{L}} R_2(f)^{\frac{2}{L}} = \operatorname{rank}_I(f)$

Therefore, as a corollary, we have

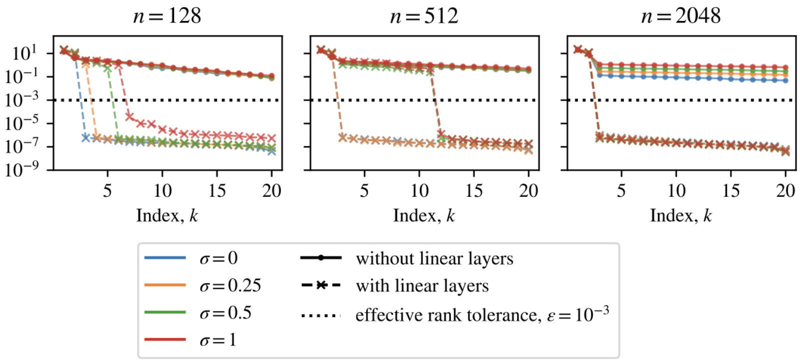
$$\lim_{L\to\infty} R_L^{\mathsf{lin}}(f) = \operatorname{rank}_I(f).$$

The R_L^{lin}-cost favors functions with **low index rank**,
i.e., functions well-approximated by multi-index models.

Numerical Example

Fitting n noisy samples from a index-rank 2 target function $f^* : \mathbb{R}^{20} \to \mathbb{R}$ using a shallow ReLU net with and without added linear layers.

Square-root of EGOP eigenvalues of trained networks



(σ is the noise standard deviation)

Parkinson et al. 2023

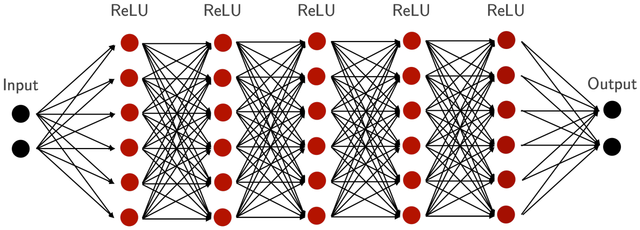
Related Work

- Bach 2017 gives generalization bounds for infinite-width shallow networks having bounded variation norm assuming the target function is a multi-index model.
- Recent line of work studies ability neural networks to provably learn low-index rank structure when trained via gradient methods:

  $\Longrightarrow$ Shallow networks (Damian et al. 2022; Bietti et al. 2022; Mousavi-Hosseini et al. 2022)

  $\Longrightarrow$ Three-layer networks (Nichani et al. 2023)
- EGOP analysis is central to the recently proposed "deep neural feature ansatz" (Radhakrishnan et al. 2024) as a means to explain feature learning in deep networks.

Representation Costs of Deep ReLU Networks



Model class: L-layer fully connected ReLU network, unbounded widths

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \boldsymbol{W}_L(\boldsymbol{W}_{L-1} \cdots \sigma(\boldsymbol{W}_2 \sigma(\boldsymbol{W}_1 \boldsymbol{x} + \boldsymbol{b}_1)_+ + \boldsymbol{b}_2)_+ \cdots + \boldsymbol{b}_{L-1})_+ + \boldsymbol{b}_L.$$

Focus on networks with *vector outputs*: $f_{\boldsymbol{\theta}} : \mathbb{R}^{d_{\text{in}}} \to \mathbb{R}^{d_{\text{out}}}$.

Parameter Cost: $C_L(\boldsymbol{\theta}) = \frac{1}{L}\|\boldsymbol{\theta}\|_F^2$ (sum-of-squares of all weights/biases)

Call the associated **representation cost** $R_L(f)$.

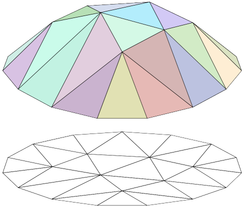
Jacot et al. 2022; Jacot 2023b; Jacot 2023a

CPWL Functions

Every ReLU net realizes a **continuous piecewise linear (CPWL) function**, in the following sense:

Definition

We say $f : \Omega \to \mathbb{R}^D$ is **CPWL** if f is continuous and there is a polyhedral decomposition of Ω such that f is affine on each polyhedra in the decomposition.



Conversely, every CPWL function over $\mathbb{R}^d$ can be represented by a ReLU NN with at most $\lceil \log_2(d+1) \rceil$ hidden layers (Arora et al. 2018).

The parametric model space of unbounded width L-layer ReLU nets coincides with all CPWL functions when $L \geq \lceil \log_2(d+1) \rceil$

The space of CPWL functions is a vector space, and is closed under compositions: if g and h are CPWL, then so is $f = h \circ g$.

However, it is not a closed space under any "reasonable" topology.

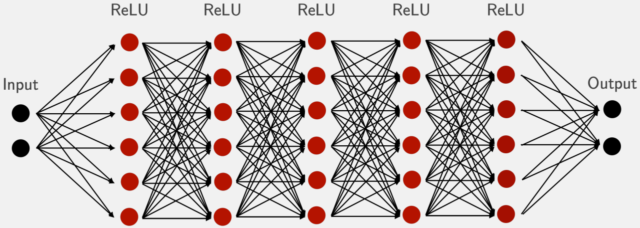
A Representer Theorem

Theorem (Jacot et al. 2022)

The data-fitting problem

$$\min_{f \in \mathsf{CPWL}} \sum_{i=1}^{n} \mathcal{L}(y_i, f(\boldsymbol{x}_i)) + \lambda R_L(f)$$

has a minimizer f realized by a depth-L ReLU network whose hidden-layer widths are upper bounded by $n(n+1)$ where n is the number of training samples.



Infinite-Depth Representation Cost

Define the **"infinite-depth" representation cost** of a CPWL function f:

$$R_\infty(f) = \lim_{L\to\infty} R_L(f).$$

R_∞ has the properties we would expect a "rank" on CPWL functions to have (Jacot 2023b):

- $R_\infty(f \circ g) \leq \min\{R_\infty(f), R_\infty(g)\}$
- $R_\infty(f + g) \leq R_\infty(f) + R_\infty(g)$
- if $f(\boldsymbol{x}) = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}$ then $R_\infty(f) = \mathrm{rank}(\boldsymbol{A})$.

Is there a function space description of R_∞?

A suggestive bound:

Lemma

Let f be CPWL, and suppose $\boldsymbol{x} \in \Omega$ is a point where f is differentiable. Then

$$\|Jf(\boldsymbol{x})\|_{S^{2/L}}^{2/L} \leq R_L(f)$$

where Jf is the Jacobian of f.

Jacobian Rank and Bottleneck Rank

Definition:

The **Jacobian rank** of a CPWL function $f : \Omega \to \mathbb{R}^{d_{\text{out}}}$ is

$$\operatorname{rank}_J(f) = \max_{\boldsymbol{x}} \operatorname{rank}(Jf(\boldsymbol{x})),$$

taking the max over points $\boldsymbol{x} \in \Omega$ where f is differentiable.

Definition:

The **bottleneck rank** of a CPWL function $f : \Omega \to \mathbb{R}^{d_{\text{out}}}$, denoted $\operatorname{rank}_{BN}(f)$, is the smallest integer $r \in \mathbb{N}$ such that $f|_\Omega = (g \circ h)|_\Omega$ where g and h are CPWL functions with inner dimension r.

If $f = h \circ g$ with inner dimension r, then by the chain rule

$$Jf(\boldsymbol{x}) = \underbrace{Jh(g(\boldsymbol{x}))}_{d_{\text{out}} \times r} \underbrace{Jg(\boldsymbol{x})}_{r \times d_{\text{in}}} \implies \operatorname{rank}_J(f) \leq r.$$

This shows $\operatorname{rank}_J(f) \leq \operatorname{rank}_{BN}(f)$ for any CPWL f.

But there are CPWL functions where strict inequality holds.

Jacot 2023b

Bounds on the "Infinite-Depth" Representation Cost

Theorem (Jacot 2023b)

For all CPWL functions $f : \Omega \to \mathbb{R}$

$$\mathrm{rank}_J(f) \leq R_\infty(f) \leq \mathrm{rank}_{\mathsf{BN}}(f).$$

Further, it is conjectured that for all CPWL functions f

$$R_\infty(f) = \mathrm{rank}_{\mathsf{BN}}(f).$$

Follow-up work (Jacot 2023a) characterizes a first-order "correction" $R_\infty^{(1)}(f)$ to the R_L cost

$$R_L(f) = R_\infty(f) + \frac{1}{L} R_\infty^{(1)}(f) + O(L^{-2})$$

Here R_∞ captures a "hard" notion of function rank, while the first-order correction $R_\infty^{(1)}$ gives regularity control of the composition factors.

Jacot 2023b; Jacot 2023a

Related Work

Recent work by Jacot *et al.* explores the implications of the bottleneck rank for learning:

- Emergent bottleneck rank structure CNNs (Wen and Jacot 2024)
- Neural collapse phenomenon (Jacot et al. 2024)
- Feature learning in Leaky ResNets (Jacot and Kaiser 2025) (CPAL)

Related nonlinear notions of function rank have been proposed to characterize **implicit regularization** in deep networks:

- Deep matrix factorization (Arora et al. 2019; Razin and Cohen 2020)
- Deep tensor factorization (Razin et al. 2021; Razin et al. 2022)
- Graph Neural Networks and Separation Rank (Razin et al. 2023).
- Rank minimization in deep ReLU networks (Timor et al. 2023)

Characterization of “Deep” Function Spaces

Rather than generating function spaces as the closure of some parametric model class, we can instead ask:

What function spaces give rise to **representer theorems** for deep NNs?

That is, are there function spaces $\mathcal{F}$ with an associated representation cost $R(\cdot)$ such that the data-fitting problem:

$$\min_{f \in \mathcal{F}} \sum_{i=1}^{n} \mathcal{L}(y_i, f(\boldsymbol{x}_i)) + \lambda R(f)$$

has a **deep neural net** as a solution? If so, can one guarantee bounds on hidden-layer widths or other constraints on the solution?

- Compositions of variation spaces/RKBSs (Parhi and Nowak 2022; Shenouda et al. 2024; Bartolucci et al. 2024)
- Neural Hilbert Ladders and Reproducing Kernel Chains (Chen 2024; Heeringa et al. 2025)

Compositions of Variation Spaces

Recall that shallow NNs are associated with a **variation space**, i.e., functions realized by an integral over all possible neurons.

A natural extension to deep NNs is to consider functions of the form $f^{(L)} \circ \cdots \circ f^{(1)}$ where each $f^{(\ell)}$ is in a **vector-valued variation space**

$$\mathcal{V}_\sigma(\mathbb{R}^{d_{\mathsf{in}}}; \mathbb{R}^{d_{\mathsf{out}}}) := \left\{ f(\boldsymbol{x}) = \int_{\mathbb{S}^{d_{\mathsf{in}}}} \sigma(\boldsymbol{w}^\mathsf{T} \boldsymbol{x}) \, d\boldsymbol{\nu}(\boldsymbol{w}) : \boldsymbol{x} \in \mathbb{R}^{d_{\mathsf{in}}}, \boldsymbol{\nu} \in \mathcal{M}(\mathbb{S}^{d_{\mathsf{in}}}; \mathbb{R}^{d_{\mathsf{out}}}) \right\},$$

where $\mathcal{M}(\mathbb{S}^{d_{\mathsf{in}}}; \mathbb{R}^{d_{\mathsf{out}}})$ is a space of **vector-valued measures**.

This space has the associated norm

$$\|f\|_{\mathcal{V}_\sigma(\mathbb{R}^{d_{\mathsf{in}}}; \mathbb{R}^{d_{\mathsf{out}}})} := \inf \left\{ \|\boldsymbol{\nu}\|_{2,\mathcal{M}} : f(\boldsymbol{x}) = \int_{\mathbb{S}^{d_{\mathsf{in}}}} \sigma(\boldsymbol{w}^\mathsf{T} \boldsymbol{x}) d\boldsymbol{\nu}(\boldsymbol{w}) \right\}$$

where $\|\boldsymbol{\nu}\|_{2,\mathcal{M}} = \int_{\mathbb{S}^{d_{\mathsf{in}}}} \mathrm{d}\|\boldsymbol{\nu}\|_2$ (analogous to the mixed ℓ^2-ℓ^1 matrix norm).

Parhi and Nowak 2022; Shenouda et al. 2024

Representer Theorem for Deep Variation Spaces

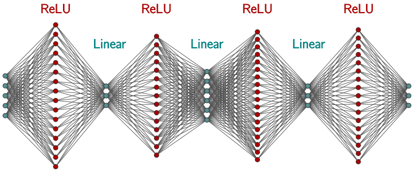
Theorem

Let $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^n \subset \mathbb{R}^{d_0} \times \mathbb{R}^{d_L}$ and $d_1, d_2, ..., d_{L-1} \in \mathbb{N}$. Then

$$\inf_{\substack{f^{(1)},...,f^{(L)} \\ f^{(\ell)} \in \mathcal{V}_\sigma(\mathbb{R}^{d_{\ell-1}}, \mathbb{R}^{d_\ell})}} \sum_{i=1}^n \mathcal{L}\big(\boldsymbol{y}_i, f^{(L)} \circ \cdots \circ f^{(1)}(\boldsymbol{x}_i)\big) + \lambda \sum_{\ell=1}^L \|f^{(\ell)}\|_{\mathcal{V}_\sigma(\mathbb{R}^{d_{\ell-1}}, \mathbb{R}^{d_\ell})},$$

has a minimizer $f_*^{(1)}, ..., f_*^{(L)}$ such that $f_* = f_*^{(L)} \circ \cdots \circ f_*^{(1)}$ is realzable as an L-layer deep ReLU network with **linear bottlenecks** of dimension d_ℓ between ReLU-layers, and whose ReLU-layer widths are at most nd_ℓ.

Ex: $L = 4$



This type of architecture was studied empirically in (Golubeva et al. 2021).

Parhi and Nowak 2022; Shenouda et al. 2024

Connection to Weight Decay Regularization

Solutions of the previous variational problem coincide with **weight decay regularized** training of deep ReLU networks with linear bottlenecks:

$$f_{\boldsymbol{\theta}} = f^{(L)}_{\boldsymbol{\theta}_L} \circ \cdots \circ f^{(1)}_{\boldsymbol{\theta}_1}$$

where $f^{(\ell)}_{\boldsymbol{\theta}_\ell}(\boldsymbol{x}) = \boldsymbol{U}_\ell(\boldsymbol{V}_\ell \boldsymbol{x} + \boldsymbol{b}_{\boldsymbol{\ell}})_+$ with $\boldsymbol{V}_\ell \in \mathbb{R}^{K_\ell \times d_{\ell-1}}$ and $\boldsymbol{U}_\ell \in \mathbb{R}^{d_\ell \times K_\ell}$, for $\ell = 1, \ldots, L-1$, and $f^{(L)}_{\boldsymbol{\theta}_L}(\boldsymbol{x}) = \boldsymbol{W}_L \boldsymbol{x} + \boldsymbol{b}_L$ with $\boldsymbol{W}_L \in \mathbb{R}^{d_\ell \times d_{\ell-1}}$.

Theorem

Assume the ReLU hidden-layer widths satisfy $K_\ell \geq n^2$. Then every minimizer $\boldsymbol{\theta}^* = (\boldsymbol{\theta}^*_1, \ldots, \boldsymbol{\theta}^*_L)$ of

$$\min_{\boldsymbol{\theta}} \sum_{i=1}^{n} \mathcal{L}\big(\boldsymbol{y}_i, f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)\big) + \frac{\lambda}{L}\|\boldsymbol{\theta}\|_2^2$$

is such that $f^{(1)}_{\boldsymbol{\theta}^*_1}, \ldots, f^{(L)}_{\boldsymbol{\theta}^*_L}$ is a minimizer of the previous variational problem.

Parhi and Nowak 2022; Shenouda et al. 2024

Eliminating Linear Bottlenecks

Bartolucci et al. 2024 extends the previous framework by eliminating linear bottlenecks between layers

Weight "matrices" $\boldsymbol{W}^{(\ell)}$ between layers are extended to be **bounded linear operators** from $\ell^2(\mathbb{N})$ to $\ell^2(\mathbb{N})$

$$f(\boldsymbol{x}) = \boldsymbol{x}^{(L)} \quad \text{where} \begin{cases} \boldsymbol{x}^{(0)} = \boldsymbol{x} & \in \mathbb{R}^{d_{\text{in}}} \\ \boldsymbol{x}^{(1)} = \boldsymbol{W}^{(1)}\boldsymbol{x}^{(0)} + \boldsymbol{b}^{(1)} & \in \ell^2(\mathbb{N}) \\ \boldsymbol{x}^{(2)} = \boldsymbol{W}^{(2)}\left(\sigma(\boldsymbol{x}^{(1)})\right) + \boldsymbol{b}^{(2)} & \in \ell^2(\mathbb{N}) \\ \quad\vdots & \quad\vdots \\ \boldsymbol{x}^{(L)} = \boldsymbol{W}^{(L)}\left(\sigma(\boldsymbol{x}^{(L)})\right) + \boldsymbol{b}^{(L+1)} & \in \mathbb{R}^{d_{\text{out}}}. \end{cases}$$

Can define **variation spaces** for operators $F : \ell^2(\mathbb{N}) \to \ell^2(\mathbb{N})$.

An analogous **representer theorem** holds: the associated variational problem has a deep NN as a minimizer whose hidden-layer widths $K_1^*, \ldots, K_L^*$ satisfy $K_L^* \leq n d_{\text{out}}$, and $K_{\ell-1}^* \leq n K_\ell^*$ for all other ℓ.

However, direct connection to weight decay regularization is lost.

Neural Hilbert Ladders/Reproducing Kernel Chains

Issue: Directly composing variation spaces/RKBSs does not allow for explicit characterization of the resulting function space.

Recent work considers a different approach to composing RKHSs/RKBSs in such a way that the resulting space has nice properties:

- Neural Hilbert Ladders (RKHSs) (Chen 2024)
- Reproducing kernel chains (RKBSs) (Heeringa et al. 2025)

Key idea is to build spaces by **compose kernels** rather than functions.

Ex: if $k_1 : \Omega \times \Omega \to \mathbb{R}$ is a reproducing kernel for an RKHS $\mathcal{H}_1$ and $\widetilde{k}_1 : \mathcal{H}_1 \times \mathcal{H}_1 \to \mathbb{R}$ *is a kernel defined on* $\mathcal{H}_1$, their composition

$$k_2(\boldsymbol{x}, \boldsymbol{y}) := \widetilde{k}_1(k_1(\boldsymbol{x}, \cdot), k_1(\cdot, \boldsymbol{y}))$$

defines a new RKHS $\mathcal{H}_2$ of functions over Ω. (Wilson et al. 2016)

Composing RKBS kernels associated with shallow NNs L-times yields an RKBS associated with L-layer deep NNs. Regularizing with the assoicated norm yields a **representer theorem** (Heeringa et al. 2025)

Comparison of Approaches

Frameworks giving **representer theorems** for deep neural networks:

Reference	Linear Bottlenecks	Native space Classification	Parameter Space Regularizer	Width Bounds
(Jacot et al. 2022)	No	???	ℓ^2	$n(n+1)$
(Shenouda et al. 2024)	Yes	???	ℓ^2	n^2
(Bartolucci et al. 2024)	No	???	None	n^ℓ
(Heeringa et al. 2025)	No	RKBS	None	n

Summary and Outlook

Function spaces give a unified perspective to learning with kernel methods, sparse methods, shallow NN, and (to some extent) deep NN.

Powerful tool for characterizing **approximation**, **estimation**, and **generalization** capabilities of neural networks. (Bach 2017; E and Wojtowytsch 2020; Siegel and Xu 2020; Schmidt-Hieber 2020; Zhang and Wang 2023; E et al. 2022; Siegel and Xu 2023; DeVore et al. 2025)

Practical implications for **efficient optimization** and **compression** of neural network models. (Ergen and Pilanci 2021; Yang et al. 2022; Varshney and Pilanci 2024; Shenouda et al. 2024)

Open Problems

Several Key Open Problems Remain

- Is there a function space characterization of R_L, the representation cost associated w/ L-layer ReLU networks, for $L > 2$?
- What is the native space associated with R_L-cost? How does it change with input dimension d?
- Prove the conjecture $R_\infty(f) = \mathrm{rank}_{\mathsf{BN}}(f)$. (Jacot 2023b)
- What connections can be drawn between function space perspectives of explicit regularization versus *implicit regularization* arising from practical training with gradient methods?

References I

Aronszajn, Nachman (1950). "Theory of reproducing kernels". In: **Transactions of the American Mathematical Society** 68.3, pp. 337–404.

Arora, Raman, Amitabh Basu, Poorya Mianjy, and Anirbit Mukherjee (2018). "Understanding Deep Neural Networks with Rectified Linear Units". In: **International Conference on Learning Representations (ICLR)**.

Arora, Sanjeev, Nadav Cohen, Wei Hu, and Yuping Luo (2019). "Implicit regularization in deep matrix factorization". In: **Advances in neural information processing systems (NeurIPS)** 32.

Bach, Francis (2017). "Breaking the curse of dimensionality with convex neural networks". In: **Journal of Machine Learning Research** 18.1, pp. 629–681.

Barron, Andrew R. (1993). "Universal approximation bounds for superpositions of a sigmoidal function". In: **IEEE Transactions on Information theory** 39.3, pp. 930–945.

Bartolucci, Francesca, Ernesto De Vito, Lorenzo Rosasco, and Stefano Vigogna (2023). "Understanding neural networks with reproducing kernel Banach spaces". In: **Applied and Computational Harmonic Analysis** 62, pp. 194–236.

— (2024). "Neural reproducing kernel Banach spaces and representer theorems for deep networks". In: **arXiv preprint arXiv:2403.08750**.

References II

Bietti, Alberto, Joan Bruna, Clayton Sanford, and Min Jae Song (2022). "Learning single-index models with shallow neural networks". In: **Advances in Neural Information Processing Systems**.

Boyer, Claire, Antonin Chambolle, Yohann De Castro, Vincent Duval, Frédéric De Gournay, and Pierre Weiss (2019). "On representer theorems and convex regularization". In: **SIAM Journal on Optimization** 29.2, pp. 1260–1281.

Bredies, Kristian and Marcello Carioni (2020). "Sparsity of solutions for variational inverse problems with finite-dimensional data". In: **Calculus of Variations and Partial Differential Equations** 59.1, pp. 1–26.

Burer, Samuel and Renato DC Monteiro (2003). "A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization". In: **Mathematical programming** 95.2, pp. 329–357.

Candès, Emmanuel J., Justin Romberg, and Terence Tao (2006). "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information". In: **IEEE Transactions on Information Theory** 52.2, pp. 489–509.

Carl, Bernd (1981). "Entropy numbers, s-numbers, and eigenvalue problems". In: **Journal of Functional Analysis** 41.3, pp. 290–306.

Chandrasekaran, Venkat, Benjamin Recht, Pablo A. Parrilo, and Alan S. Willsky (2012). "The convex geometry of linear inverse problems". In: **Foundations of Computational Mathematics** 12.6, pp. 805–849.

References III

Chen, Zhengdao (2024). "Neural Hilbert ladders: Multi-layer neural networks in function space". In: **Journal of Machine Learning Research** 25.109, pp. 1–65.

Dai, Zhen, Mina Karzand, and Nathan Srebro (2021). "Representation costs of linear neural networks: Analysis and design". In: **Advances in Neural Information Processing Systems** 34, pp. 26884–26896.

Damian, Alexandru, Jason Lee, and Mahdi Soltanolkotabi (2022). "Neural networks can learn representations with gradient descent". In: **Conference on Learning Theory**. PMLR, pp. 5413–5452.

DeVore, Ronald, Robert D. Nowak, Rahul Parhi, and Jonathan W. Siegel (2025). "Weighted Variation Spaces and Approximation by Shallow ReLU Networks". In: **Applied and Computational Harmonic Analysis** 74.101713. DOI: 10.1016/j.acha.2024.101713.

DeVore, Ronald A. (1998). "Nonlinear approximation". In: **Acta Numerica** 7, pp. 51–150.

Donoho, David L. (2000). "High-dimensional data analysis: The curses and blessings of dimensionality". In: **AMS Math Challenges Lecture** 1.2000, p. 32.

— (2006). "Compressed sensing". In: **IEEE Transactions on Information Theory** 52.4, pp. 1289–1306.

Donoho, David L. and Iain M. Johnstone (1994). "Ideal spatial adaptation by wavelet shrinkage". In: **Biometrika** 81.3, pp. 425–455.

References IV

Donoho, David L. and Iain M. Johnstone (1995). "Adapting to unknown smoothness via wavelet shrinkage". In: **Journal of the American Statistical Association** 90.432, pp. 1200–1224.

— (1998). "Minimax estimation via wavelet shrinkage". In: **The Annals of Statistics** 26.3, pp. 879–921.

Donoho, David L., Richard C. Liu, and Brenda MacGibbon (1990). "Minimax risk over hyperrectangles, and implications". In: **Annals of Statistics**, pp. 1416–1437.

E, Weinan, Chao Ma, and Lei Wu (2022). "The Barron space and the flow-induced function spaces for neural network models". In: **Constructive Approximation** 55.1, pp. 369–406.

E, Weinan and Stephan Wojtowytsch (2020). "On the Banach Spaces Associated with Multi-Layer ReLU Networks: Function Representation, Approximation Theory and Gradient Descent Dynamics". In: **CSIAM Transactions on Applied Mathematics** 1.3, pp. 387–440.

Ergen, Tolga and Mert Pilanci (2021). "Convex geometry and duality of over-parameterized neural networks". In: **Journal of Machine Learning Research**.

Fisher, Stephen 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.

References V

- Golubeva, Anna, Guy Gur-Ari, and Behnam Neyshabur (2021). "Are wider nets better given the same number of parameters?" In: **International Conference on Learning Representations**.
- Grandvalet, Yves (1998). "Least absolute shrinkage is equivalent to quadratic penalization". In: **International Conference on Artificial Neural Networks**. Springer, pp. 201–206.
- Gunasekar, Suriya, Jason D Lee, Daniel Soudry, and Nati Srebro (2018). "Implicit bias of gradient descent on linear convolutional networks". In: **Advances in neural information processing systems** 31.
- Heeringa, Tjeerd Jan, Len Spek, and Christoph Brune (2025). "Deep Networks are Reproducing Kernel Chains". In: **arXiv preprint arXiv:2501.03697**.
- Jacot, Arthur (Dec. 2023a). "Bottleneck structure in learned features: Low-dimension vs regularity tradeoff". In: **Advances in Neural Information Processing Systems** 36, pp. 23607–23629.
- — (June 2023b). "Implicit Bias of Large Depth Networks: a Notion of Rank for Nonlinear Functions". In: **International Conference on Learning Representations (ICLR)**.
- Jacot, Arthur, Franck Gabriel, and Clement Hongler (2018). "Neural Tangent Kernel: Convergence and Generalization in Neural Networks". In: **Advances in Neural Information Processing Systems**. Vol. 31.

References VI

- Jacot, Arthur, Eugene Golikov, Clément Hongler, and Franck Gabriel (2022). "Feature Learning in L_2-regularized DNNs: Attraction/Repulsion and Sparsity". In: **Advances in Neural Information Processing Systems** 35, pp. 6763–6774.
- Jacot, Arthur and Alexandre Kaiser (2025). "Hamiltonian Mechanics of Feature Learning: Bottleneck Structure in Leaky ResNets". In: **Conference on Parsimony and Learning (CPAL)**.
- Jacot, Arthur, Peter Súkeník, Zihan Wang, and Marco Mondelli (2024). "Wide neural networks trained with weight decay provably exhibit neural collapse". In: **arXiv preprint arXiv:2410.04887**.
- Klusowski, Jason M and Andrew R Barron (2018). "Approximation by Combinations of ReLU and Squared ReLU Ridge Functions With ℓ^1 and ℓ^0 Controls". In: **IEEE Transactions on Information Theory** 64.12, pp. 7649–7656.
- Kůrková, Věra and Marcello Sanguineti (2001). "Bounds on rates of variable-basis and neural-network approximation". In: **IEEE Transactions on Information Theory** 47.6, pp. 2659–2665.
- Li, Ker-Chau (1991). "Sliced inverse regression for dimension reduction". In: **Journal of the American Statistical Association** 86.414, pp. 316–327.
- Lin, Rong Rong, Hai Zhang Zhang, and Jun Zhang (2022). "On reproducing kernel Banach spaces: Generic definitions and unified framework of constructions". In: **Acta Mathematica Sinica, English Series** 38.8, pp. 1459–1483.

References VII

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

Matoušek, Jiří (1996). "Improved upper bounds for approximation by zonotopes". In: **Acta Mathematica** 177.1, pp. 55–73.

McCarty, Sarah (2023). "Piecewise linear functions representable with infinite width shallow ReLU neural networks". In: **Proceedings of the American Mathematical Society, Series B** 10.27, pp. 296–310.

Meyer, Yves (1992). **Wavelets and Operators**. 37. Cambridge University Oress.

Mhaskar, Hrushikesh N. (2004). "On the tractability of multivariate integration and approximation by neural networks". In: **Journal of Complexity** 20.4, pp. 561–590.

Mousavi-Hosseini, Alireza, Sejun Park, Manuela Girotti, Ioannis Mitliagkas, and Murat A Erdogdu (2022). "Neural Networks Efficiently Learn Low-Dimensional Representations with SGD". In: **The Eleventh International Conference on Learning Representations**.

Nichani, Eshaan, Alex Damian, and Jason D Lee (2023). "Provable guarantees for nonlinear feature learning in three-layer neural networks". In: **Advances in Neural Information Processing Systems** 36, pp. 10828–10875.

References VIII

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: **International Conference on Learning Representations**.

Parhi, Rahul and Robert D. Nowak (2021). "Banach Space Representer Theorems for Neural Networks and Ridge Splines". In: **Journal of Machine Learning Research** 22.43, pp. 1–40. URL: https://jmlr.org/papers/v22/20-583.html.

— (2022). "What Kinds of Functions Do Deep Neural Networks Learn? Insights from Variational Spline Theory". In: **SIAM Journal on Mathematics of Data Science** 4.2, pp. 464–489. DOI: 10.1137/21M1418642.

— (2023). "Near-Minimax Optimal Estimation With Shallow ReLU Neural Networks". In: **IEEE Transactions on Information Theory** 69.2, pp. 1125–1140. DOI: 10.1109/TIT.2022.3208653.

Parkinson, Suzanna, Greg Ongie, and Rebecca Willett (2023). "ReLU neural networks with linear layers are biased towards single-and multi-index models". In: **arXiv preprint arXiv:2305.15598**.

Petrushev, Pencho P. (1988). "Direct and converse theorems for spline and rational approximation and Besov spaces". In: **Function Spaces and Applications: Proceedings of the US-Swedish Seminar held in Lund, Sweden, June 15–21, 1986**. Springer, pp. 363–377.

References IX

- Radhakrishnan, Adityanarayanan, Daniel Beaglehole, Parthe Pandit, and Mikhail Belkin (2024). "Mechanism for feature learning in neural networks and backpropagation-free machine learning models". In: **Science** 383.6690, pp. 1461–1467.
- Razin, Noam and Nadav Cohen (2020). "Implicit regularization in deep learning may not be explainable by norms". In: **Advances in neural information processing systems** 33, pp. 21174–21187.
- Razin, Noam, Asaf Maman, and Nadav Cohen (2021). "Implicit regularization in tensor factorization". In: **International Conference on Machine Learning (ICML)**, pp. 8913–8924.
- — (2022). "Implicit regularization in hierarchical tensor factorization and deep convolutional neural networks". In: **International Conference on Machine Learning**. PMLR, pp. 18422–18462.
- Razin, Noam, Tom Verbin, and Nadav Cohen (2023). "On the ability of graph neural networks to model interactions between vertices". In: **Advances in Neural Information Processing Systems** 36, pp. 26501–26545.
- Sahiner, Arda, Tolga Ergen, John M. Pauly, and Mert Pilanci (2021). "Vector-output ReLU Neural Network Problems are Copositive Programs: Convex Analysis of Two Layer Networks and Polynomial-time Algorithms". In: **International Conference on Learning Representations**.

Savarese, Pedro, Itay Evron, Daniel Soudry, and Nathan Srebro (2019). "How do infinite width bounded norm networks look in function space?" In: **Conference on Learning Theory (COLT)**. PMLR, pp. 2667–2690.

Schmidt-Hieber, Johannes (2020). "Nonparametric regression using deep neural networks with ReLU activation function". In: **Annals of Statistics** 48.4, pp. 1875–1897.

Schölkopf, Bernhard and Alexander J. Smola (2002). **Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond**. Adaptive computation and machine learning. MIT Press. ISBN: 9780262194754.

Shenouda, Joseph, Rahul Parhi, Kangwook Lee, and Robert D Nowak (2024). "Variation spaces for multi-output neural networks: Insights on multi-task learning and network compression". In: **Journal of Machine Learning Research** 25.231, pp. 1–40.

Siegel, Jonathan W. (2023). "Optimal approximation of zonoids and uniform approximation by shallow neural networks". In: **arXiv preprint arXiv:2307.15285**.

Siegel, Jonathan W and Jinchao Xu (2020). "Approximation rates for neural networks with general activation functions". In: **Neural Networks** 128, pp. 313–321.

— (2023). "Characterization of the variation spaces corresponding to shallow neural networks". In: **Constructive Approximation** 57.3, pp. 1109–1132.

References XI

- Srebro, Nathan, Jason Rennie, and Tommi Jaakkola (2004). "Maximum-margin matrix factorization". In: **Advances in Neural Information Processing Systems** 17.
- Tibshirani, Robert (1996). "Regression shrinkage and selection via the lasso". In: **Journal of the Royal Statistical Society Series B: Statistical Methodology** 58.1, pp. 267–288.
- Timor, Nadav, Gal Vardi, and Ohad Shamir (2023). "Implicit regularization towards rank minimization in ReLU networks". In: **International Conference on Algorithmic Learning Theory**. PMLR, pp. 1429–1459.
- Unser, Michael (2021). "A unifying representer theorem for inverse problems and machine learning". In: **Foundations of Computational Mathematics** 21.4, pp. 941–960.
- — (2023). "Ridges, neural networks, and the Radon transform". In: **Journal of Machine Learning Research** 24.37, pp. 1–33.
- van de Geer, Sara (2000). **Empirical Processes in M-estimation**. Vol. 6. Cambridge University Press.
- Varshney, Prateek and Mert Pilanci (2024). "Convex Distillation: Efficient Compression of Deep Networks via Convex Optimization". In: **arXiv preprint arXiv:2410.06567**.
- Wahba, Grace (1990). **Spline models for observational data**. Vol. 59. SIAM.

References XII

- Wang, Yifei, Tolga Ergen, and Mert Pilanci (2023). "Parallel Deep Neural Networks Have Zero Duality Gap". In: **International Conference on Learning Representations**.
- Wen, Yuxiao and Arthur Jacot (2024). "Which Frequencies do CNNs Need? Emergent Bottleneck Structure in Feature Learning". In: **International Conference on Machine Learning (ICML)**.
- Wilson, Andrew Gordon, Zhiting Hu, Ruslan Salakhutdinov, and Eric P Xing (2016). "Deep kernel learning". In: **Artificial Intelligence and Statistics**. PMLR, pp. 370–378.
- Yang, Liu, Jifan Zhang, Joseph Shenouda, Dimitris Papailiopoulos, Kangwook Lee, and Robert D Nowak (2022). "A better way to decay: Proximal gradient training algorithms for neural nets". In: **OPT 2022: Optimization for Machine Learning (NeurIPS 2022 Workshop)**.
- Zeno, Chen, Greg Ongie, Yaniv Blumenfeld, Nir Weinberger, and Daniel Soudry (2023). "How do minimum-norm shallow denoisers look in function space?" In: **Advances in Neural Information Processing Systems** 36, pp. 57520–57557.
- Zhang, Haizhang, Yuesheng Xu, and Jun Zhang (2009). "Reproducing Kernel Banach Spaces for Machine Learning.". In: **Journal of Machine Learning Research** 10.12.

References XIII

Zhang, Kaiqi and Yu-Xiang Wang (2023). "Deep Learning meets Nonparametric Regression: Are Weight-Decayed DNNs Locally Adaptive?" In: **International Conference on Learning Representations**.