the local geometry of deep learning

Richard Baraniuk
Rice University
models data
Babylonians were obsessed with data and calculating (polynomial curve fitting).

Knew about all the key theorems of the day and were extraordinary at predicting astronomical events.

Students learned math by working out large numbers of problems until they “understood” the general concept.

Incapable of scaffolding theorems together to create something larger.
• Ancient Greeks were **obsessed with models**
  - Ex: stars, sun, planets, moon are holes in a colossal cosmic colander that reveals the eternal fire beyond

• Such a (bad) model, inspired Eratosthenes to use geometry to deduce the **radius of the earth**

[Feynman, Toulmin, Pearl]
Shannon's model of a communication system inspired the development of information theory, which undergirds the present information age.
signal processing / prediction using deep learning

\[ x \xrightarrow{f_{\Theta}(x)} \hat{y} \]
Deep nets solve **prediction** tasks **hierarchically**

\[
\hat{y} = f_\Theta(x) = \left( f_{\theta(L)}^{(L)} \circ \cdots \circ f_{\theta(3)}^{(3)} \circ f_{\theta(2)}^{(2)} \circ f_{\theta(1)}^{(1)} \right)(x)
\]
**deep network**

- **Input, output:** \( x =: z_0, \ y \)

- **Layer 1:** \( z_1 = \sigma(W_1z_0 + b_1) \)
  - **Weight matrix:** \( W_\ell \)
  - **Bias vector:** \( b_\ell \)
  - **Activation operator:** \( \sigma \)
    
    - Scalar activation function applied component-wise
    - Ex: Rectified Linear Unit (ReLU), aka thresholding

- **Deep net with** \( L \) **layers**
  
  \[
  \hat{y} = \sigma^*(W_L \sigma(\cdots \sigma(W_2 \sigma(W_1 z_0 + b_1) + b_2) \cdots) + b_L)
  \]
• Tune the parameters to **minimize** the **total prediction error** on the training data set using **gradient descent**
throwing caution to the wind

- Solving problems with **data not models**!

- **Highly nonlinear** approximant!

- Highly **overparameterized**! (typically many more parameters than training data points)

- Highly **nonconvex** loss function (error) to optimize! (due to composition and nonlinear activation function)
a perfect storm

- Deeper network architectures
- Big data
- GPU based computing
- Massive *alchemistic* trial and error

AI Researchers Left Disappointed As NIPS Sells Out In Under 12 Minutes (2019)
data trumping models
2023 IEEE International Conference on Acoustics, Speech and Signal Processing
4 - 10 JUNE, RHODES ISLAND, GREECE

Signal Processing in the AI era
Deep nets are easy to describe \textbf{locally}.

\textbf{Not clear} how to describe them \textbf{globally}.

\[
\hat{y} = f_\Theta(x) = \left( f_{\theta(L)} \circ \cdots \circ f_{\theta(3)} \circ f_{\theta(2)} \circ f_{\theta(1)} \right)(x)
\]
grand challenge

Deep nets are easy to describe **locally**

**Not clear** how to describe them **globally**
Why is deep learning so effective?

Can we derive deep learning systems from first principles?

When and why does deep learning fail?

How can deep learning systems be improved and extended in a principled fashion?

Where is the foundational framework for theory?

See also DeVore, Daubechies, Mallat, Bruna, Soatto, Arora, Poggio, [growing community] ...
I'm sorry, Dave. I'm afraid I can't do that.
many deep networks are splines
Randall Balestriero

Rudolf Riedi

Imtiaz Humayun

Sébastien Paris

Romain Cosentino
Nonlinear Approximation and (Deep) ReLU Networks

I. Daubechies, R. DeVore, S. Foucart, B. Hanin, G. Petrova

Piecewise convexity of artificial neural networks

Blaine Rister\textsuperscript{a,\ast}, Daniel L. Rubin\textsuperscript{b}

\textsuperscript{a} Stanford University, Department of Electrical Engineering, 1201 Welch Rd, Stanford, CA, 94305, USA
\textsuperscript{b} Stanford University, Department of Radiology (Biomedical Informatics Research), 1201 Welch Rd Stanford, CA, 94305,

A representer theorem for deep neural networks

Michael Unser
continuous piecewise affine (CPA) splines
CPA spline approximation

- **Affine parameters** (slope & offset) in each partition region
- **Continuous**
- **Piecewise**: Partition the domain (data) into regions
  - polytopes in high dimensions
kinds of splines

• A spline function approximation consists of
  – a partition $\Omega$ of the independent variable (input space)
  – a (simple) **local mapping** on each region of the partition

• **Powerful splines**
  – free, unconstrained partition $\Omega$ (ex: “free-knot” splines)
  – jointly optimize **both** the partition and local mappings (highly nonlinear, computationally intractable)
kinds of splines

• A spline function approximation consists of
  – a partition $\Omega$ of the independent variable (input space)
  – a (simple) local mapping on each region of the partition

• Easy splines
  – fixed partition
    (ex: uniform grid, dyadic grid)
  – need only optimize the local mappings
• **Focus:** The lion-share* of today’s deep net architectures (convnets, resnets, skip-connection nets, inception nets, recurrent nets, ...) employ **continuous piecewise affine (CPA) layers** (fully connected, conv; (leaky) ReLU, abs value; max/mean/channel-pooling)

\[
\hat{y} = f_\Theta(x) = \left( f_{\theta(L)} \circ \cdots \circ f_{\theta(3)} \circ f_{\theta(2)} \circ f_{\theta(1)} \right)(x)
\]
**Transformers**

- **Transformer** networks have a more complicated structure than more neoclassical deep networks: **self-attention** mechanism

- At later layers, transformers can be closely approximated as CPA splines

---

**Figure 3:** Cosine similarity between the attention matrices $\hat{A}$'s at layer $i$ and its next higher layer.
deep nets* are splines

- A **multi-layer deep net** is a continuous piecewise affine (CPA) **spline** operator

---

deep net
spline partition
geometry
ReLU deep net units & layers

- A **deep net layer** implements a **CPA spline operator**
- Ex: Layer 1

\[
\begin{align*}
    \mathbf{z}^{(1)} &= \mathbf{W}^{(1)} \mathbf{x} + \mathbf{b}^{(1)} \\
    \text{ReLU}(u) &= \max(u, 0)
\end{align*}
\]
unit spline function

- Each **deep net unit/neuron** implements a CPA **spline function**
- Ex: At layer 1, the $k$-th output of a ReLU net is given by

\[
z_k^{(1)} = \text{ReLU} \left( w_k^{(1)} \cdot x + b_k^{(1)} \right)
\]

where $w_{k}^{(1)} \cdot x + b_{k}^{(1)}$ is the row $k$ of the weight matrix $W^{(1)}$ at layer 1.
layer spline operator

- The units in a **deep net layer** implement a CPA **spline operator**

$$\text{ReLU}(u) = \max(u, 0)$$

- What is the **spline partition** of the input space?
The spline partition of the input space is formed by intersecting a collection of half-spaces.
Ex: At layer 1, the $k$-th unit splits in layer input space in two

$z_{k}^{(1)} = \text{ReLU} \left( w_{k}^{(1)} \cdot x + b_{k}^{(1)} \right) > 0$

$\text{ReLU}(u) = \max(u, 0)$

equation of a hyperplane in the layer’s input space
The spline partition of the input space is formed by intersecting a collection of half-spaces.

Ex: At layer 1, the $k$-th unit splits in layer input space in two half-spaces.

The equation of a hyperplane in the layer’s input space is:

$$z_k^{(1)} = \text{ReLU} \left( w_k^{(1)} \cdot x + b_k^{(1)} \right) > 0$$

$$= 0$$

ReLU($u$) = max($u$, 0)
layer spline mapping

- Deep net layer spline operator
  - different fixed **affine transform** on each partition region $Q(x)$
  - **continuous** across partition regions

- Closed-form formulas are available for $A_{Q(x)}^{(1)}, B_{Q(x)}^{(1)}$

$$z^{(1)} = A_{Q(x)}^{(1)} x + B_{Q(x)}^{(1)}$$
these are not your parents’ splines!

- Even a **single layer** with $K$ units/neurons and piecewise linear activation function with $R$ pieces induces a spline partition with **exponentially many** partition regions $R^K$.

- **Absurd!** 1M data points
  100M parameters
  1 zillion partition regions
SplineCAM

- Provably **exact** algorithm for computing a deep net’s **spline partition geometry** in a **2D slice**
  - exploit efficient graph-traversal method

- Ex: VGG16 with ~**138M parameters** trained on tiny-Imagenet
  ~7 minutes for ~1K regions

[imtiazhumayun.github.io/splinecam]
SplineCAM

- Visualize the **exact decision boundary** of an 8-layer **classification** net along a **2D slice** of the input space

(8 min for 80k regions)
implicit neural representation (INR)

- Powerful deep-network based models for 2D images

Input: Spatial Coordinates
Output: 2D Pixel Intensity

image as a “landscape”
implicit neural representation (INR)

- Powerful deep-network based models for **3D objects**

**Input:**
Spatial Coordinates

**Output:**
3D Voxel Occupancy
SplineCAM

- Characterize the exact **object boundaries** in a deep net based **implicit neural representation (INR)** of a 3D object
these are not your parents’ splines

• Not necessarily **absurd**

1. Most spline partition regions are **empty**

2. Most regions are **far from the action**
these are not your parents’ splines

- Not necessarily **absurd**

1. Most spline partition regions are **empty**

2. Most regions are **far from the action**

3. Regions have a rich **multiscale structure**
multiscale partition subdivision

- **Subdivision**
  Each unit (neuron) creates a **hyperplane** that cuts its layer input space into two half-spaces

- **Folding**
  With respect to the deep net’s input space, the hyperplanes are **folded** by the previous layers to maintain **continuity** of the CPA spline mapping

multiscale partition subdivision

- **Subdivision**
  Each unit (neuron) creates a **hyperplane** that cuts its layer input space into two half-spaces.

- **Folding**
  With respect to the deep net’s input space, the hyperplanes are **folded** by the previous layers to maintain **continuity** of the CPA spline mapping.

- Many interesting **multiresolution analysis** questions (think wavelets++)
multiscale partition subdivision

- Many interesting multiresolution analysis questions (think wavelets++)

- What are the analogues of
  - sparsity?
  - tree structures?
  - fractals?
  - long-range dependence?
applications:
deeplearning

deep network
learning
learning dynamics

- **Quantify** partition and decision boundary **during learning**

- Monitor **more than just end-to-end** train/test accuracy
compare training/design choices

- Quantify DNs based on the geometric properties of the partition, ex: via partition density around train/test points

![Graphs showing average region volume, number of regions, and eccentricity for smaller and larger nets with and without data augmentation (DA)].

DA = data augmentation
more learning applications

• Explaining why *batch normalization* improves learning
  “Batch Normalization Explained,” arXiv, 2022

• Proof that *residual networks* have a better behaved loss surface than convnets

“Singular Value Perturbation and Deep Network Optimization,” *Constructive Approximation*, 2022

[Hao Li et al., "Visualizing the loss landscape of neural nets," NeurIPS, 2018]
more learning applications

- Proof that **residual networks** have a better behaved loss surface than convnets


loss surface is **piecewise quadratic/cross-entropic**
more learning applications

- Proof that **residual networks** have a better behaved loss surface than convnets

*Singular values* of local quadratic loss are better behaved for a resnet

application: uniform sampling from generative networks
deep generative models

- Deep generative network (DGN) maps a low-dimensional latent space to an image manifold in high-dimensional data space (ex: GAN)
Exploit fact that **DGN is a CPA spline**
- DGN maps latent input space to a **CPA manifold** in higher-dimensional space

Each latent space partition region is warped and placed into the output space by an affine transform.
bias in machine learning

Facial Recognition Is Accurate, if You’re a White Guy

Misinformation Is About to Get So Much Worse
A conversation with the former Google CEO Eric Schmidt

AI tradeoffs: Balancing powerful models and potential biases

We Teach A.I. Systems Everything, Including Our Biases
biased sampling from DGNs

- Training data is often non-uniformly sampled from the data manifold
  - results in biased DGN samples
MaGNET (Maximum entropy Generative NETwork)

- Exploit the **analytical characterization of data distribution**
  - adapt sampling from DGN according to **local manifold properties**
  - account for the **change of volume** induced by each CPA mapping
  - reweight the latent sampling to obtain **uniformly distributed samples** on the generated manifold

"MaGNET: Uniform Sampling from Deep Generative Network Manifolds Without Retraining," *ICLR*, 2022
MaGNET

- Training data is often **non-uniformly sampled** from the data manifold
  - results in **biased** DGN samples
- MaGNET does **not** require re-training nor labels
practical example

- Without labels or retraining, MaNET’s uniform sampling reduces gender bias of FFHQ-StyleGAN2 by 41%
deep networks & computational harmonic analysis
implicit neural representation (INR)

- Powerful deep-network based models for 2D images

![Diagram](image.png)

- Standard activation nonlinearities: ReLU, bumps, sinusoids
Wavelet Implicit neural REpresentations (WIRE)

- 1D (complex) **wavelet activation** nonlinearity

\[
\sigma \left( W_3 \sigma \left( W_2 \sigma \left( W_1 u + b_1 \right) + b_2 \right) + b_3 \right)
\]

\[
u = \begin{bmatrix} x \\ y \end{bmatrix}
\]
Wavelet Implicit neural REpresentations (WIRE)

- 1D (complex) **wavelet activation** nonlinearity

\[
\sigma (W_3 \sigma (W_2 \sigma (W_1 u + b_1) + b_2) + b_3)
\]

Input:
Spatial Coordinates

Output:
2D Pixel Intensity

Layer 1 output

Input image
Wavelet Implicit neural REpresentations (WIRE)

- 1D (complex) **wavelet activation** nonlinearity

\[
\sigma(W_3 \sigma(W_2 \sigma(W_1 u + b_1) + b_2) + b_3)
\]

**Input:** Spatial Coordinates

**Output:** 2D Pixel Intensity

```
Input image
```

**Layer 1 output**

```
ridgelets!
```
Wavelet Implicit neural REpresentations (WIRE)

- **2D (complex) wavelet activation** nonlinearity

\[
\sigma \left( W_3 \sigma \left( W_2 \sigma \left( W_1 u + b_1 \right) + b_2 \right) + b_3 \right)
\]
Wavelet Implicit neural REpresentations (WIRE)

- **2D (complex) wavelet activation nonlinearity**

\[ \sigma \left( W_3 \sigma \left( W_2 \sigma (W_1 u + b_1) + b_2 \right) + b_3 \right) \]

Layer 1 output

Olshausen & Field, 1996

Input image

curvelets!
Wavelet Implicit neural REpresentations (WIRE)

- 1D (complex) wavelet activation nonlinearity

\[ \sigma (W_3 \sigma (W_2 \sigma (W_1 u + b_1) + b_2) + b_3) \]
CT image recovery

Ground truth 1D WIRE 2D WIRE

Gauss ReLU + Pos. SIREN

24.2dB 26.9dB 24.0dB 23.1dB 22.7dB
wrap up

tremendous *alchemistical* progress
\[
\frac{d}{dx} \int_a^x f(t) \, dt = f(x)
\]
\[ \frac{d}{dx} \int_a^x f(t) \, dt = f(x) \]

Pandemic of 1665
• (CPA) Splines provide a firm foundation for a theory of deep learning
summary

- Driving concepts of **deep learning** are here to stay:
  - Overparameterization, nonconvex optimization, big data
  - The field needs foundational theory to guide experimentation

- Modern deep nets are a (composition of) **CPA splines** with rich structure
  - power diagram
  - multiscale subdivision

- Spline viewpoint enables
  - analysis and improvement of learning
  - improved generative modeling
  - so much more!
“Singular Value Perturbation and Deep Network Optimization,” *Constructive Approximation*, 2022
“Batch Normalization Explained,” arXiv, 2022
“A Farewell to the Bias-Variance Tradeoff? An Overview of the Theory of Overparameterized Machine Learning,” 2022
“SplineCAM,” imtiazhumayun.github.io/splinecam, 2022
“Polarity Sampling: Quality and Diversity Control of Pre-Trained Generative Networks via Singular Values,” *CVPR*, 2022
“MaGNET: Uniform Sampling from Deep Generative Network Manifolds Without Retraining,” *ICLR*, 2022
“Mad Max: Affine Spline Insights into Deep Learning,” *Proceedings of the IEEE*, 2021
“From Hard to Soft: Understanding Deep Network Nonlinearities...,” *ICLR*, 2019
“A Max-Affine Spline Perspective of RNNs,” *ICLR*, 2019
“Spline Filters for End-to-End Deep Learning,” *ICML*, 2018
“A Spline Theory of Deep Networks,” *ICML*, 2018