the local geometry of deep learning

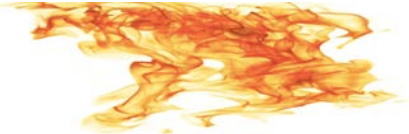
Richard Baraniuk
Rice University

models data

babylon

- 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

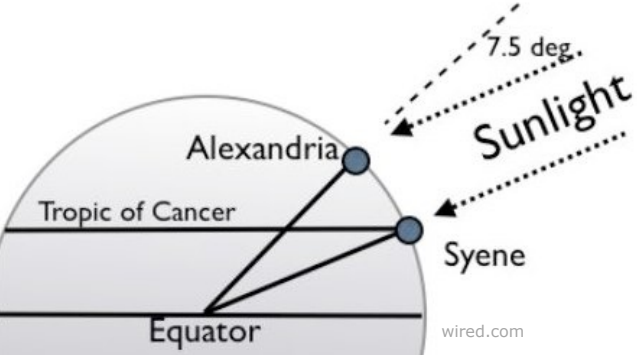
greece



- 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**

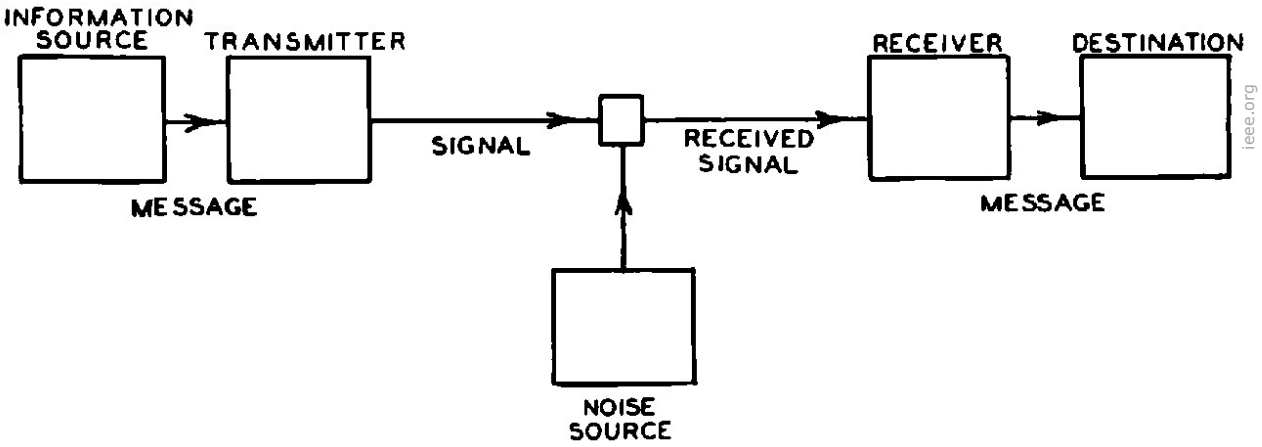
wikipedia.org



wired.com

[Feynman, Toulmin, Pearl]

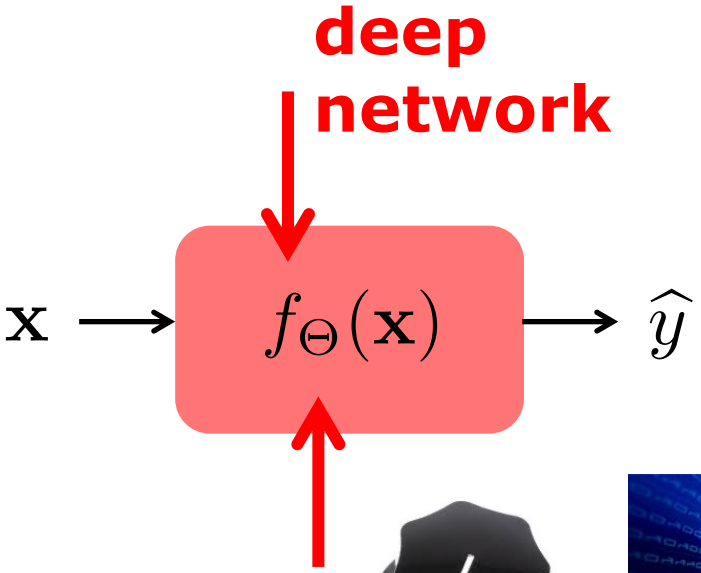
bell labs



- Shannon's **model** of a communication system inspired the development of **information theory**, which undergirds the present information age

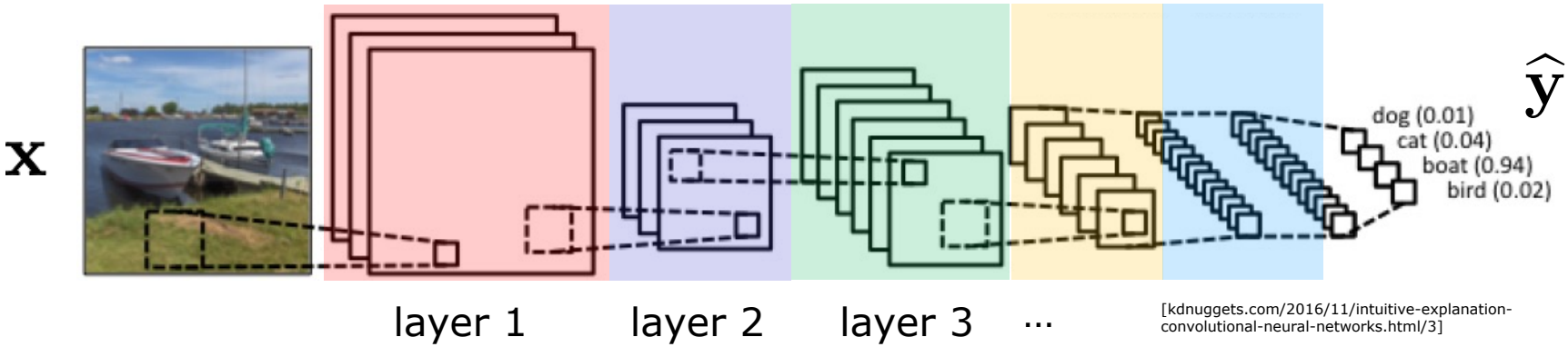
models data

signal processing / prediction using **deep learning**



deep network

Deep nets solve **prediction** tasks **hierarchically**



[kdnuggets.com/2016/11/intuitive-explanation-convolutional-neural-networks.html/3]

$$\widehat{\mathbf{y}} = f_\Theta(\mathbf{x}) = \left(f^{(L)}_{\theta^{(L)}} \circ \cdots \circ f^{(3)}_{\theta^{(3)}} \circ f^{(2)}_{\theta^{(2)}} \circ f^{(1)}_{\theta^{(1)}} \right)(\mathbf{x})$$

deep network

- Input, output: $\mathbf{x} =: \mathbf{z}_0$, $\widehat{\mathbf{y}}$

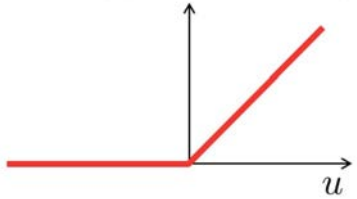
- Layer 1: $\mathbf{z}_1 = \sigma(\mathbf{W_1 z_0} + \mathbf{b_1})$

  - **Weight matrix**: $\mathbf{W}_\ell$
  - **Bias vector**: $\mathbf{b}_\ell$

  $\Big\}$ **deep net parameters**

  - **Activation** operator: σ
    Scalar activation function applied component-wise
    Ex: Rectified Linear Unit (**ReLU**), aka thresholding

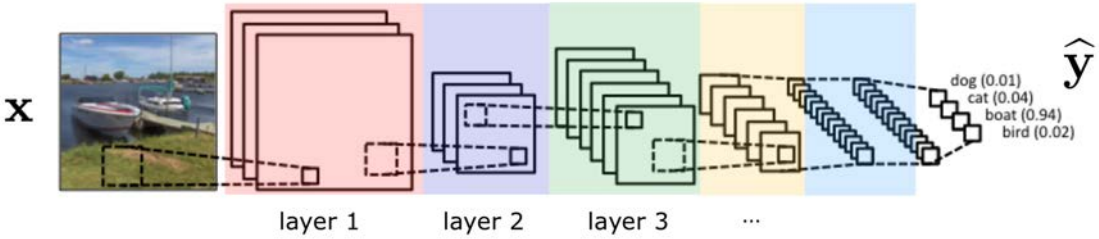
$\mathrm{ReLU}(u) = \max(u, 0)$



- Deep net with L layers

$$\widehat{\mathbf{y}} = \underline{\sigma^*(\mathbf{W}_L}\sigma(\cdots\underline{\sigma(\mathbf{W}_2}\underline{\sigma(\mathbf{W_1 z_0} + \mathbf{b_1})}\underline{+ \mathbf{b_2})}\cdots)\underline{+ \mathbf{b_L})}$$

learning



$$\{\mathbf{W}_\ell, \mathbf{b}_\ell, \ \ell = 1, \ldots, L\}$$

CAT ✗

cat

bird

boat

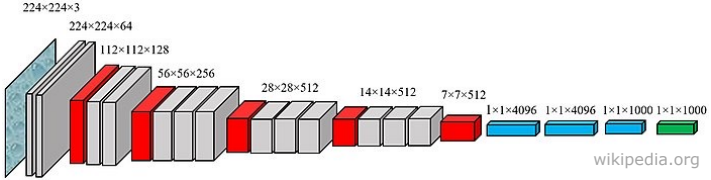
- Tune the parameters to **minimize** the **total prediction error** on the training data set using **gradient descent**

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



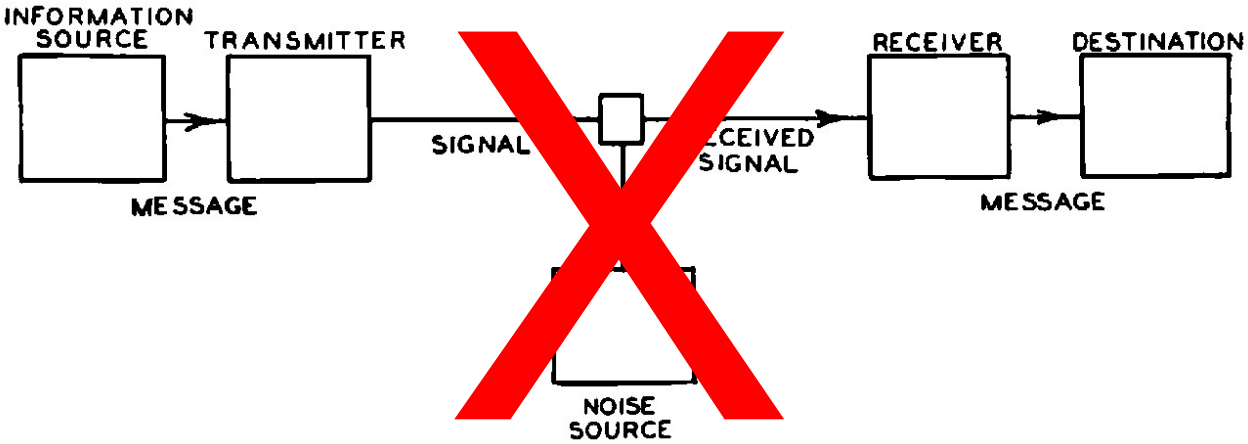
- **Big data**

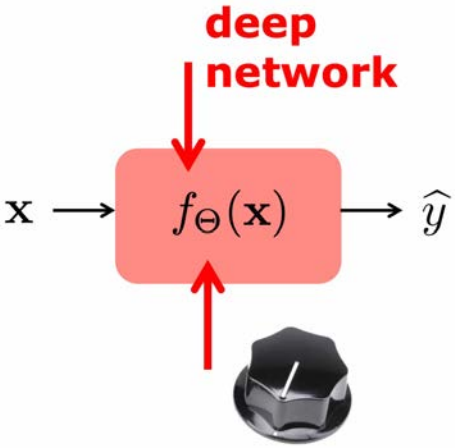
- **GPU** based computing

- Massive **alchemistic** trial and error

AI Researchers Left Disappointed As NIPS Sells Out In Under 12 Minutes (2019)

data trumping models



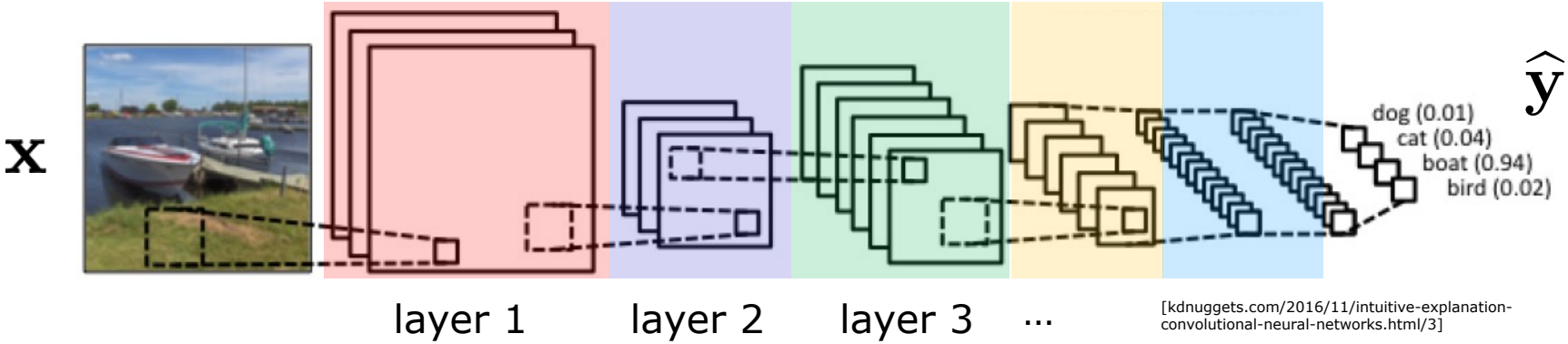

deep
network
$\mathbf{x}$
$f_\Theta(\mathbf{x})$
$\widehat{y}$


2023 IEEE International Conference on
Acoustics, Speech and Signal Processing
4 - 10 JUNE, RHODES ISLAND, GREECE
Signal Processing in the AI era

grand challenge

Deep nets are easy to describe **locally**

Not clear how to describe them **globally**



[kdnuggets.com/2016/11/intuitive-explanation-convolutional-neural-networks.html/3]

$$\widehat{\mathbf{y}} = f_\Theta(\mathbf{x}) = \left(f^{(L)}_{\theta^{(L)}} \circ \cdots \circ f^{(3)}_{\theta^{(3)}} \circ f^{(2)}_{\theta^{(2)}} \circ f^{(1)}_{\theta^{(1)}} \right)(\mathbf{x})$$

grand challenge

Deep nets are easy to describe **locally**

Not clear how to describe them **globally**

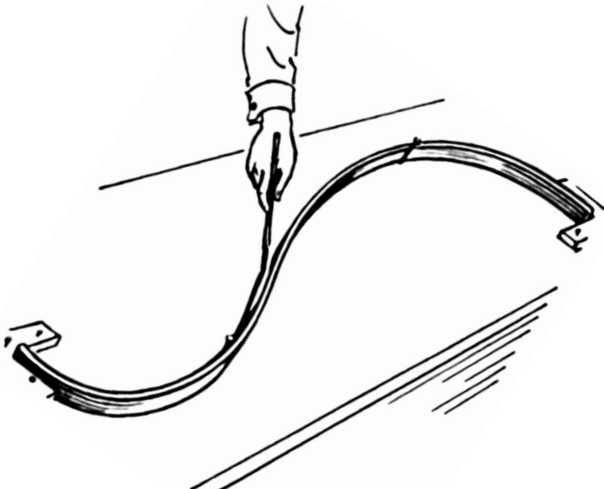
greek questions for the *babylonians*

- 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.
warnerbros.com

many deep networks are splines



Randall Balestriero

Rudolf Riedi

Imtiaz Humayun

Sébastien Paris

Romain Cosentino

deep nets and splines

A Spline Theory of Deep Learning

Randall Balestriero, baraniuk *Proceedings of the 35th International Conference on Machine Learning*, PMLR 80:374-383, 2018.

Nonlinear Approximation and (Deep) ReLU Networks

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

Piecewise convexity of artificial neural networks

Blaine Rister [a,*], Daniel L. Rubin [b]

[a] *Stanford University, Department of Electrical Engineering, 1201 Welch Rd, Stanford, CA, 94305, USA*
[b] *Stanford University, Department of Radiology (Biomedical Informatics Research), 1201 Welch Rd Stanford, CA, 94305,*

A representer theorem for deep neural networks

Michael Unser

Neural network approximation

Ronald DeVore
Department of Mathematics, Texas A&M University, College Station, TX 77843, USA
E-mail: rdevore@math.tamu.edu

Boris Hanin
Department of Operations Research and Financial Engineering, Princeton University, NJ 08544, USA
E-mail: bhanin@princeton.edu

Guergana Petrova
Department of Mathematics, Texas A&M University, College Station, TX 77843, USA
E-mail: gpetrova@math.tamu.edu

On the Number of Linear Regions of Deep Neural Networks

Guido Montúfar
Max Planck Institute for Mathematics in the Sciences
montufar@mis.mpg.de

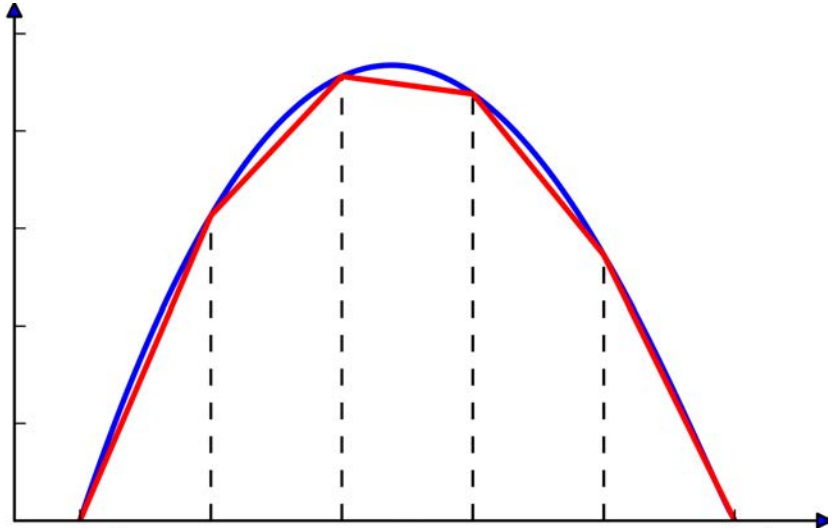
Razvan Pascanu
Université de Montréal
pascanur@iro.umontreal.ca

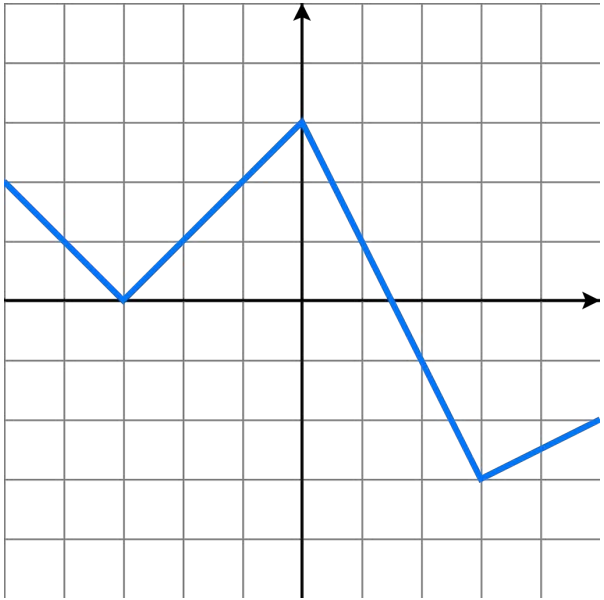
Kyunghyun Cho
Université de Montréal
kyunghyun.cho@umontreal.ca

Yoshua Bengio
Université de Montréal, CIFAR Fellow
yoshua.bengio@umontreal.ca

...

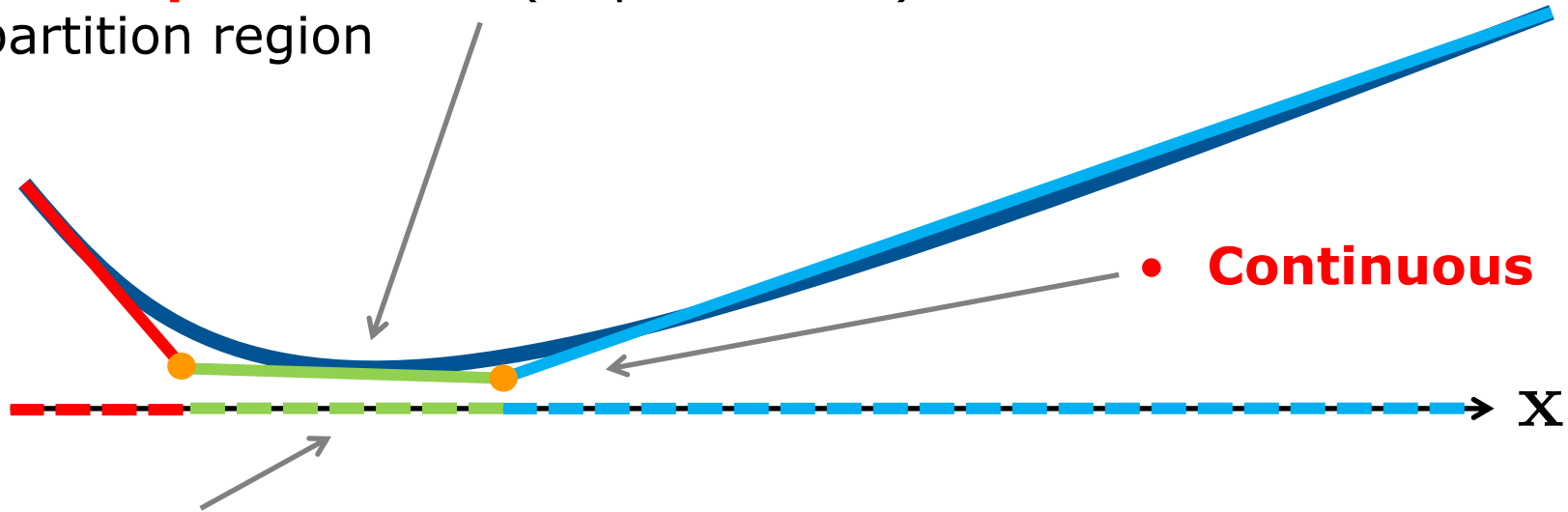
continuous piecewise affine (CPA) splines





CPA spline approximation

- **Affine parameters** (slope & offset) in each partition region



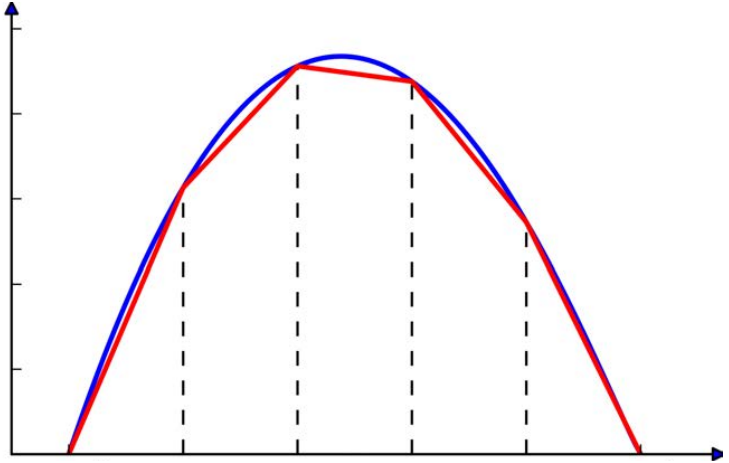
- **Continuous**

- **Piecewise: Partition** the domain (data) into **regions**
  - **polytopes** in high dimensions

kinds of splines

- A spline function approximation consists of
  - a **partition** Ω of the independent variable (input space)
  - a (simple) **local mapping** on each region of the partition

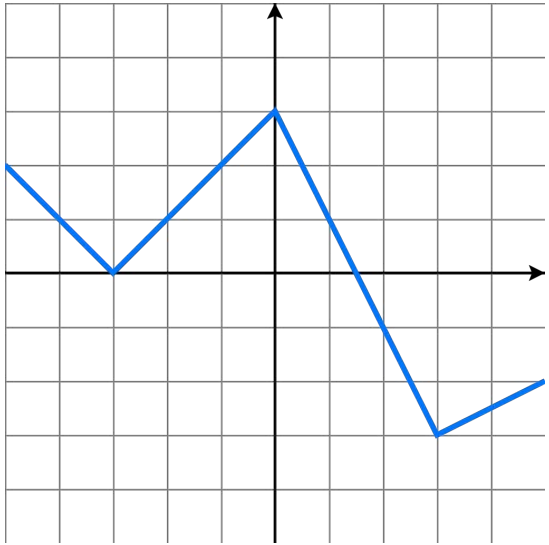
- **Powerful splines**
  - free, unconstrained partition Ω (ex: **"free-knot" splines**)
  - jointly optimize **both** the partition and local mappings (highly nonlinear, **computationally intractable**)



kinds of splines

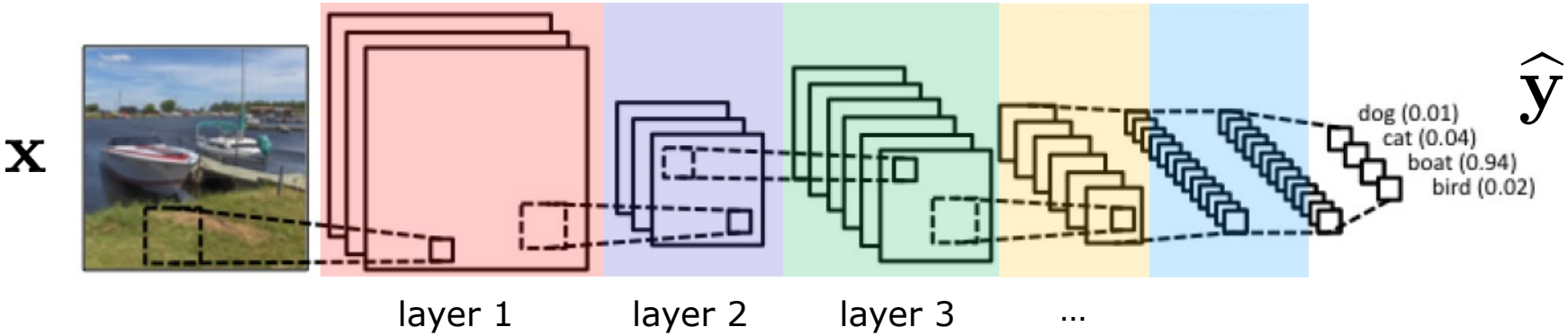
- A spline function approximation consists of
  - a **partition** Ω of the independent variable (input space)
  - a (simple) **local mapping** on each region of the partition

- **Easy splines**
  - fixed partition
    (ex: **uniform grid**, dyadic grid)
  - need only optimize the local mappings



the crux

- **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)



$$\widehat{\mathbf{y}} = f_\Theta(\mathbf{x}) = \left(f^{(L)}_{\theta^{(L)}} \circ \cdots \circ f^{(3)}_{\theta^{(3)}} \circ f^{(2)}_{\theta^{(2)}} \circ f^{(1)}_{\theta^{(1)}} \right)(\mathbf{x})$$

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

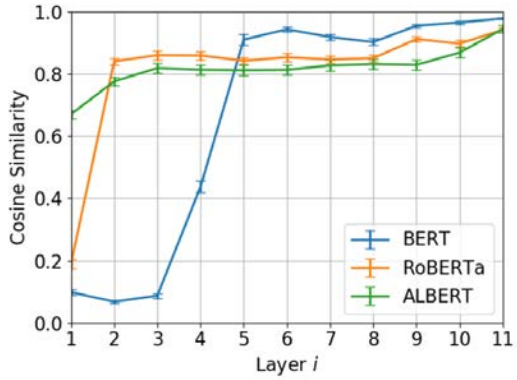


Figure 3: Consine similarity between the attention matrices $\hat{A}$'s at layer i and its next higher layer.

Published as a conference paper at ICLR 2022

REVISITING OVER-SMOOTHING IN BERT FROM THE PERSPECTIVE OF GRAPH

Han Shi[1], Jiahui Gao[2], Hang Xu[3], Xiaodan Liang[4],
Zhenguo Li[3], Lingpeng Kong[2], Stephen M.S. Lee[2], James T. Kwok[1]
[1]Hong Kong University of Science and Technology, [2]The University of Hong Kong,
[3]Huawei Noah's Ark Lab, [4]Sun Yat-sen University

Published as a conference paper at ICLR 2023

A PRIMAL-DUAL FRAMEWORK FOR TRANSFORMERS AND NEURAL NETWORKS

Tan M. Nguyen*
Department of Mathematics
University of California, Los Angeles
tanmnguyen89@ucla.edu

Tam Nguyen*
Department of ECE
Rice University
nguyenminhtam9520@gmail.com

Nhat Ho
Department of Statistics & Data Sciences
University of Texas at Austin
minhnhat@utexas.edu

Andrea L. Bertozzi
Department of Mathematics
University of California, Los Angeles
bertozzi@math.ucla.edu

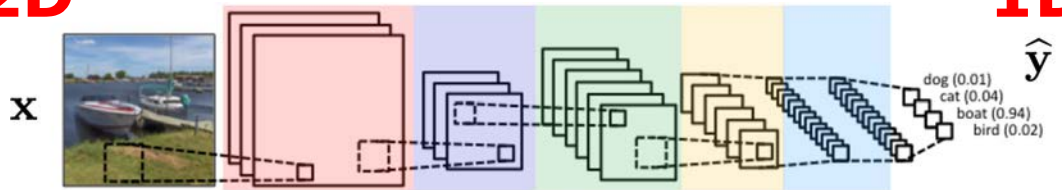
Richard G. Baraniuk
Department of ECE
Rice University
richb@rice.edu

Stanley J. Osher
Department of Mathematics
University of California, Los Angeles
sjo@math.ucla.edu

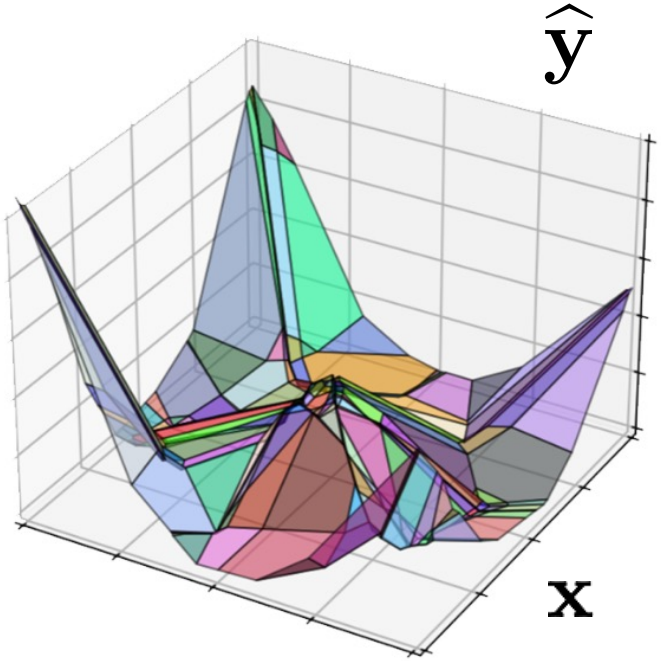
deep nets* are splines

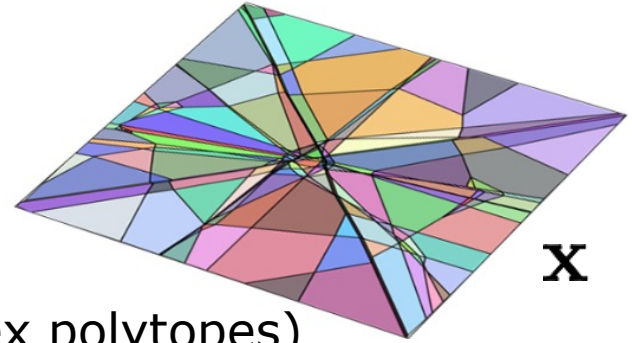
- A **multi-layer deep net** is a continuous piecewise affine (CPA) **spline** operator

2D



1D





input space partition

"power diagram" (convex polytopes)

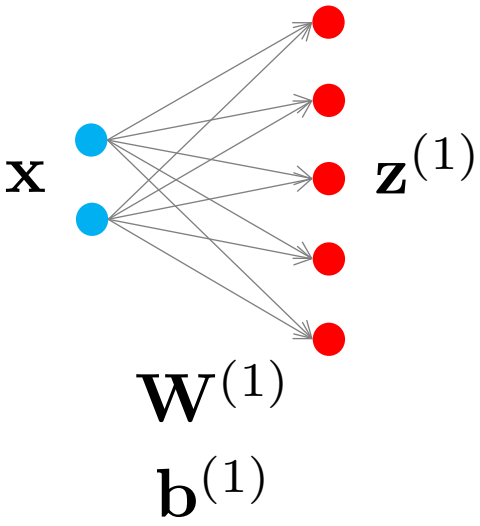
"Mad Max: Affine Spline Insights into Deep Learning," *Proceedings of the IEEE*, 2021

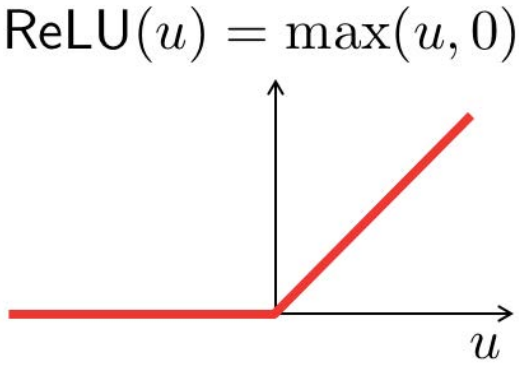
deep net spline partition geometry

nasa.gov

ReLU deep net units & layers

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





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

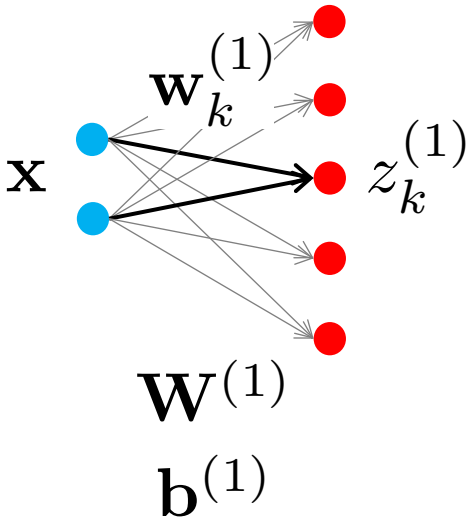
k-th output of layer 1

input to layer 1

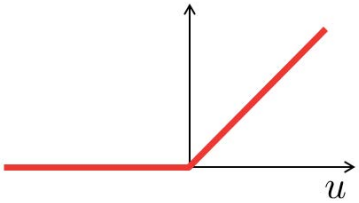
$$z_k^{(1)} = \text{ReLU}\left(\mathbf{w}_k^{(1)} \cdot \mathbf{x} + b_k^{(1)}\right)$$

bias k at layer 1

row k of weight matrix $\mathbf{W}^{(1)}$ at layer 1

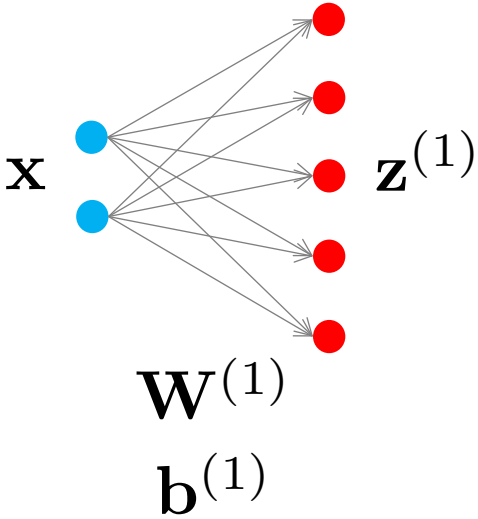


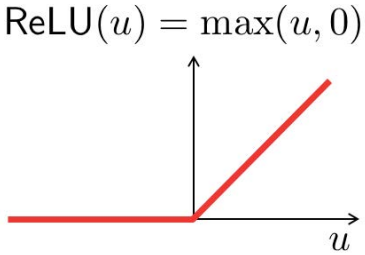
$\text{ReLU}(u) = \max(u, 0)$



layer spline operator

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





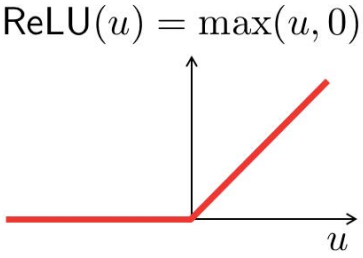
- What is the **spline partition** of the input space?

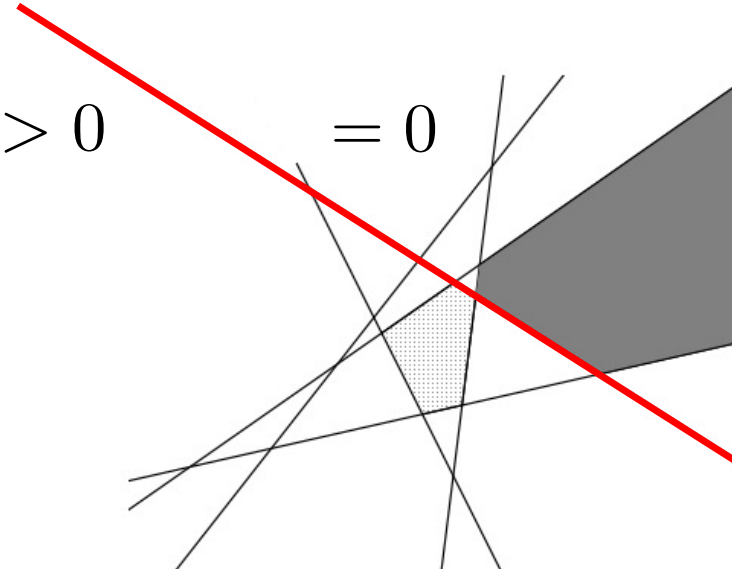
layer spline partition

- 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

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

equation of a **hyperplane** in the layer's input space



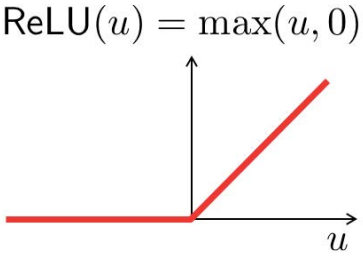


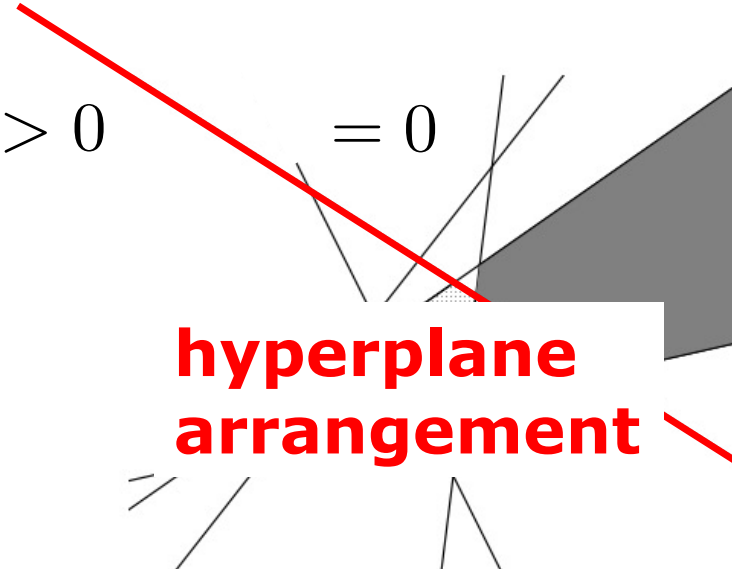
layer spline partition

- 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

$$z_k^{(1)} = \text{ReLU}\left(\underline{\mathbf{w}_k^{(1)} \cdot \mathbf{x} + b_k^{(1)}}\right) > 0 \qquad = 0$$

equation of a **hyperplane** in the layer's input space

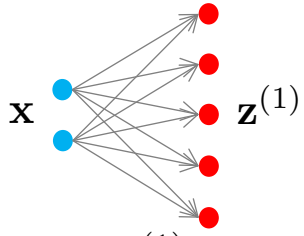


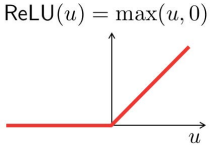


layer spline mapping

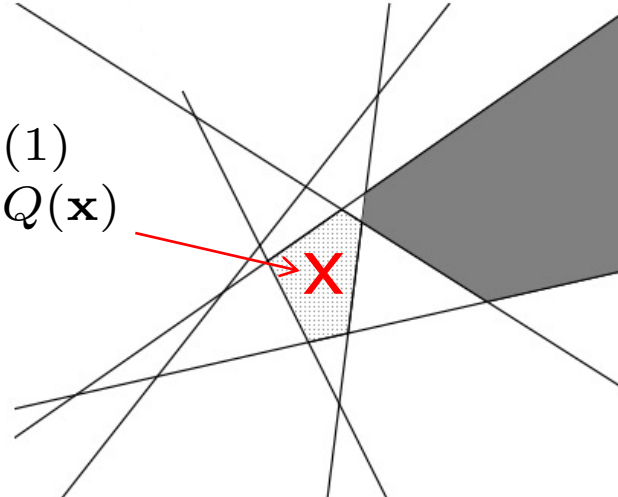
- Deep net layer spline operator
  - different fixed **affine transform** on each partition region $Q(\mathbf{x})$
  - **continuous** across partition regions

- Closed-form formulas are available for $\mathbf{A}^{(1)}_{Q(\mathbf{x})}, \mathbf{B}^{(1)}_{Q(\mathbf{x})}$



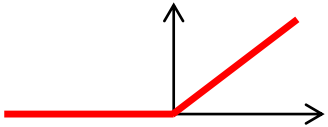


$$\mathbf{z}^{(1)} = \mathbf{A}^{(1)}_{Q(\mathbf{x})}\,\mathbf{x} + \mathbf{B}^{(1)}_{Q(\mathbf{x})}$$

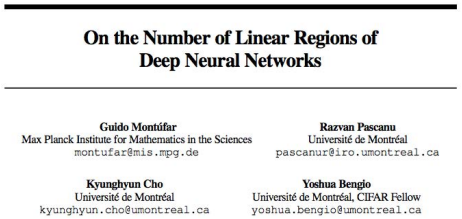


these are not your parents' splines!

- Even a **single layer** with K units/neurons and piecewise linear activation function with R pieces



ReLU: R=2



On the Number of Linear Regions of Deep Neural Networks

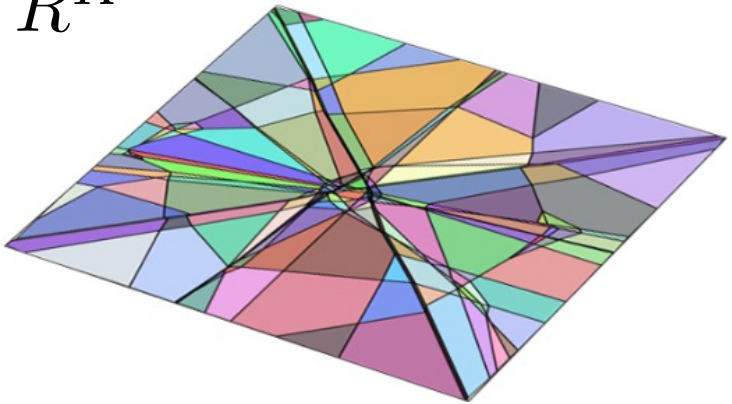
Guido Montúfar
Max Planck Institute for Mathematics in the Sciences
montufar@mis.mpg.de

Razvan Pascanu
Université de Montréal
pascanur@iro.umontreal.ca

Kyunghyun Cho
Université de Montréal
kyunghyun.cho@umontreal.ca

Yoshua Bengio
Université de Montréal, CIFAR Fellow
yoshua.bengio@umontreal.ca

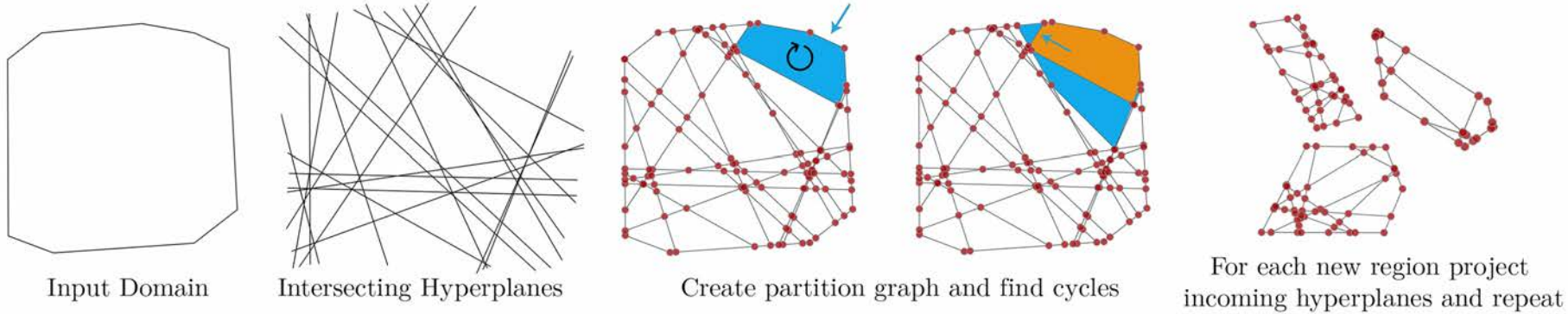
induces a spline partition with **exponentially many** partition regions R^K



- **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

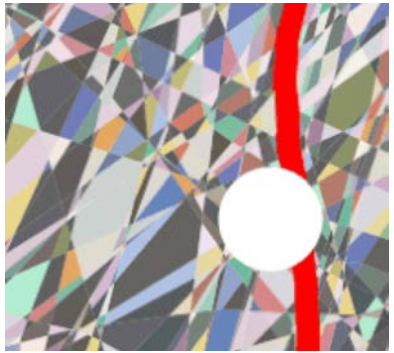


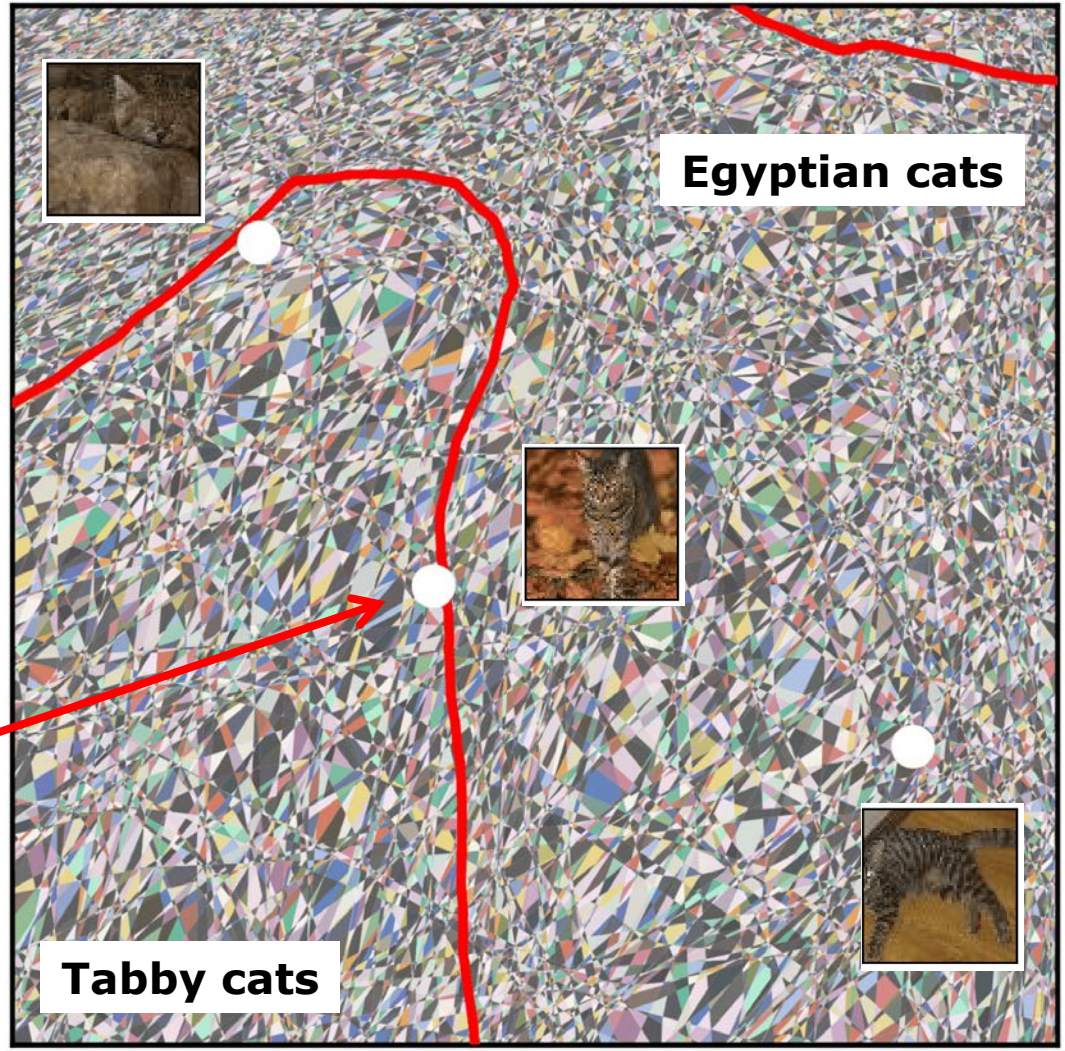
- Ex: VGG16 with **~138M parameters** trained on tiny-Imagenet
  ~7 minutes for ~1K regions

[imtiazhumayun.github.io/splinecam]

SplineCAM

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

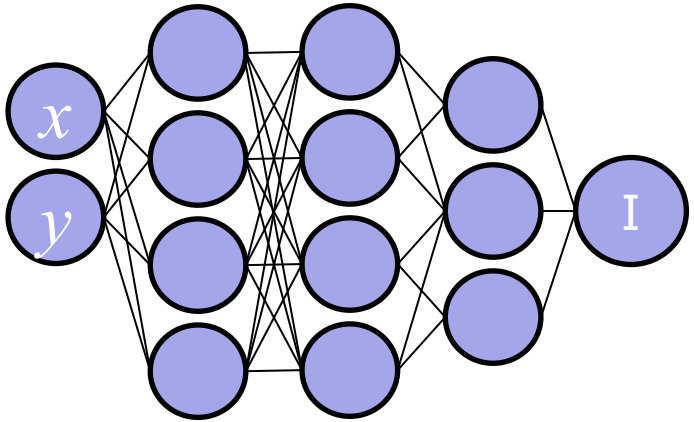




(8 min for 80k regions)

implicit neural representation (INR)

- Powerful deep-network based models for **2D images**



Input:
Spatial
Coordinates

Output:
2D Pixel Intensity

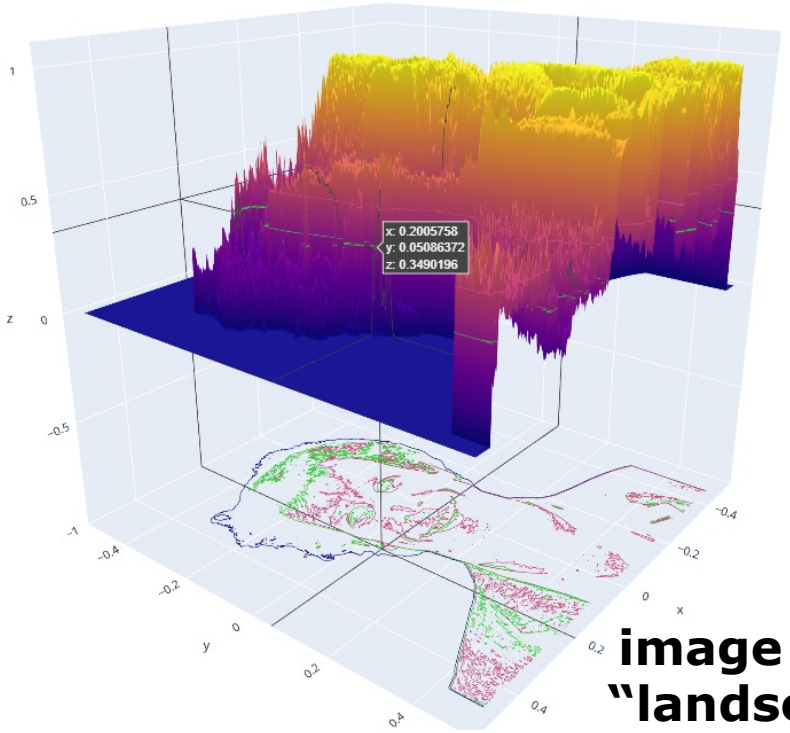
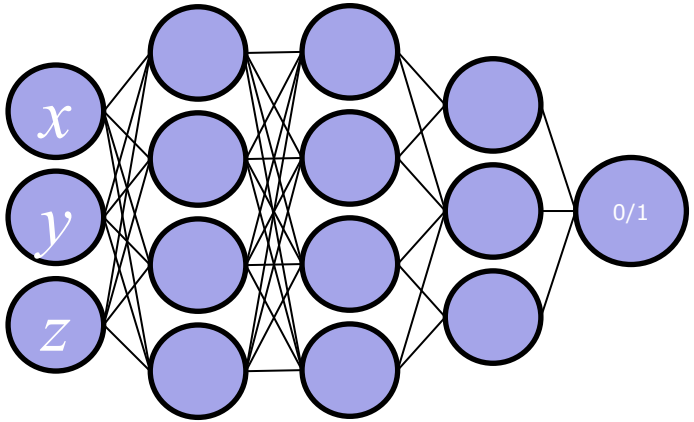


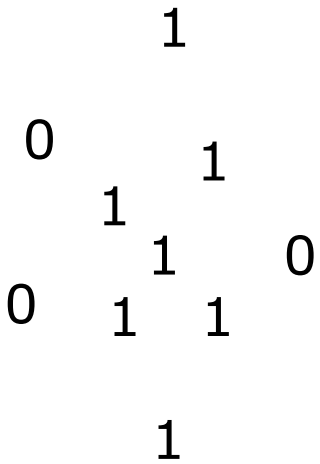
image as a "landscape"

implicit neural representation (INR)

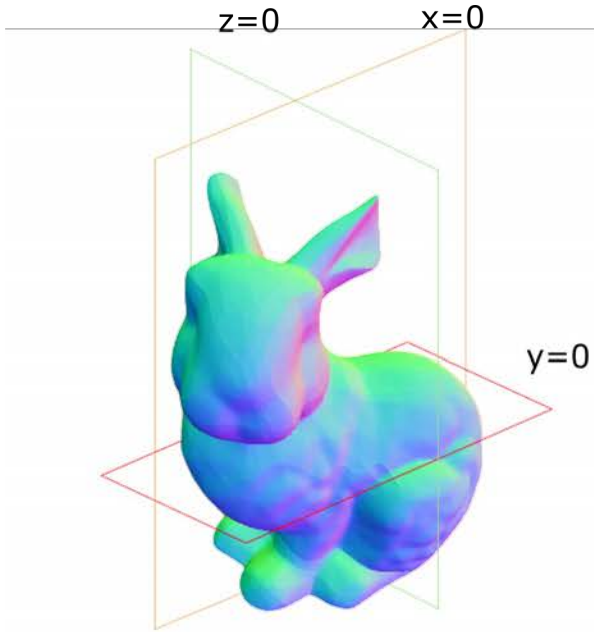
- Powerful deep-network based models for **3D objects**



Input:
Spatial
Coordinates

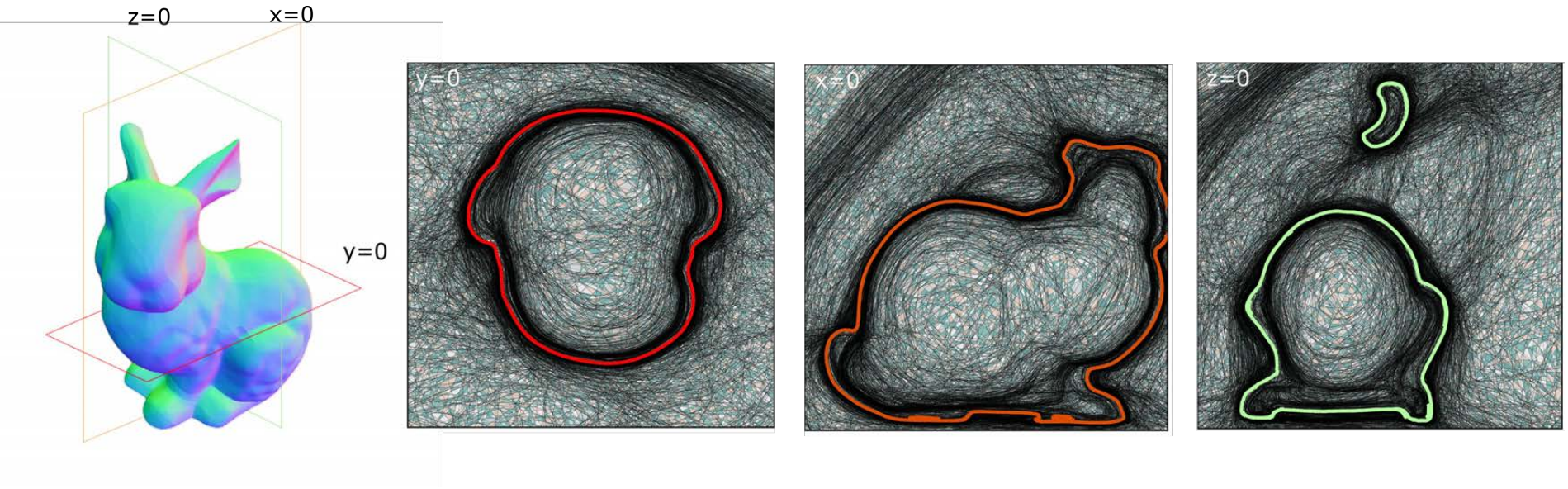


Output:
3D Voxel Occupancy



SplineCAM

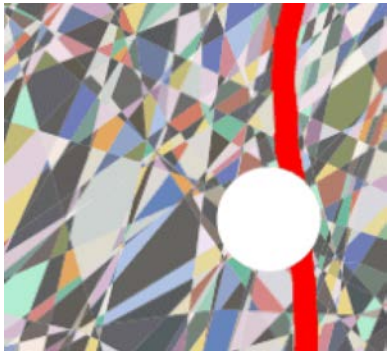
- Characterize the exact **object boundaries** in a deep net based **implicit neural representation (INR)** of a 3D object



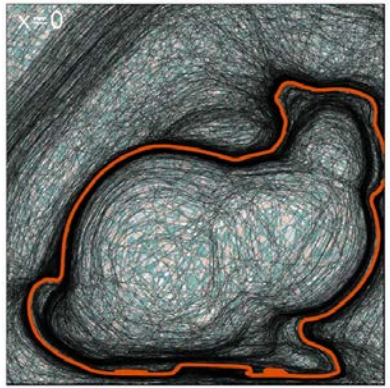
these are not your parents' splines

- Not necessarily **absurd**

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

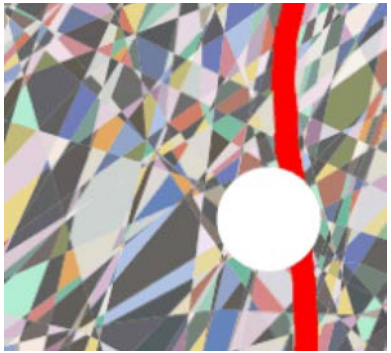


2. Most regions are **far from the action**

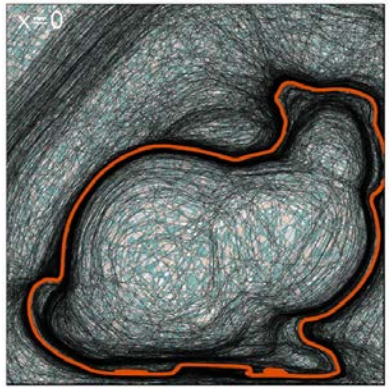


these are not your parents' splines

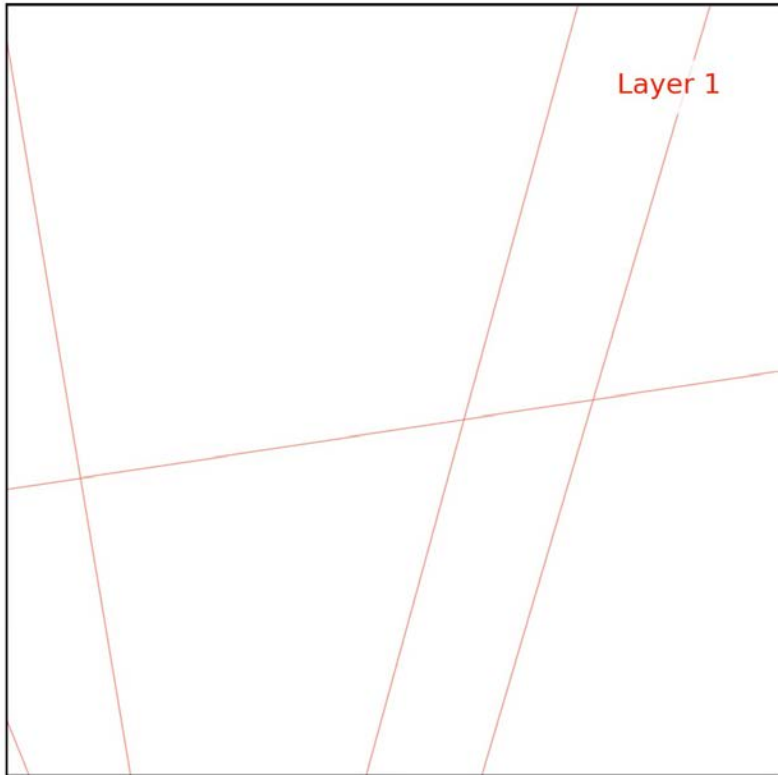
- Not necessarily **absurd**



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



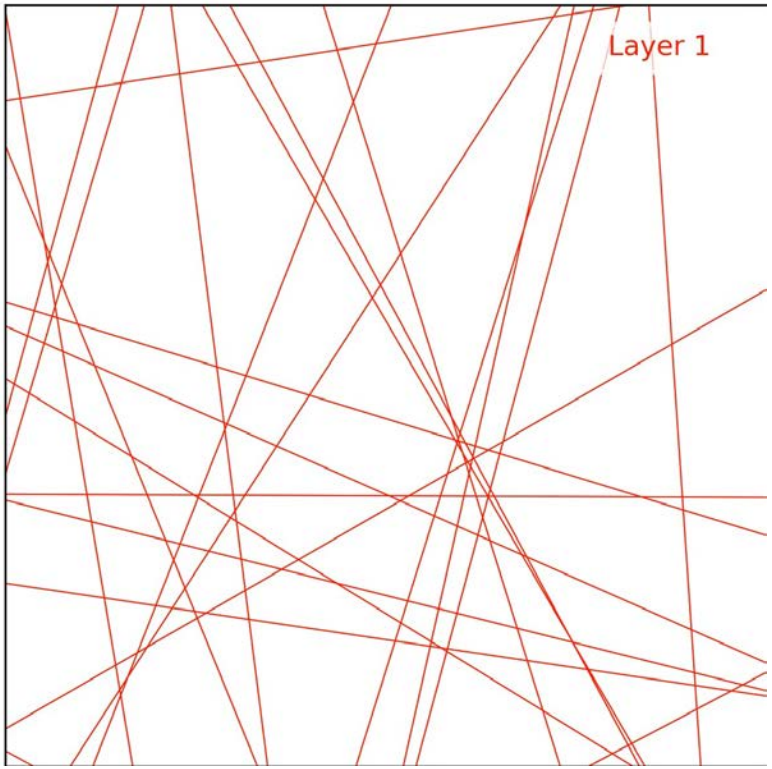
2D slice of input space

- **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

"The Geometry of Deep Networks: Power Diagram Subdivision," *NeurIPS*, 2019

multiscale partition subdivision



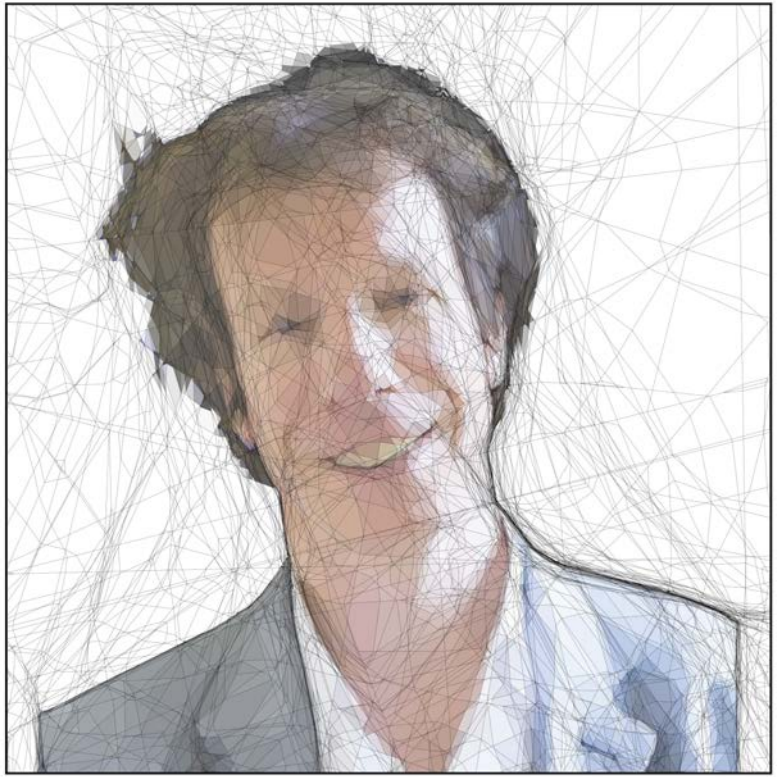
2D slice of input space

- **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



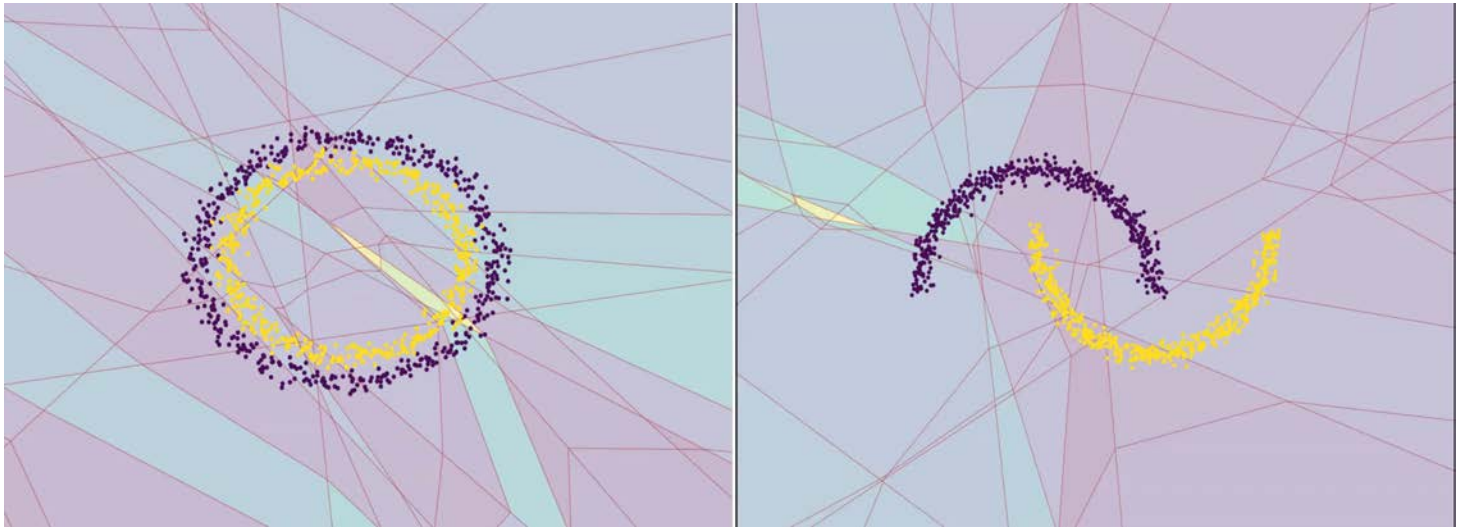
- Many interesting **multiresolution analysis** questions (think wavelets++)

- What are the analogues of
  - **sparsity?**
  - **tree structures?**
  - **fractals?**
  - **long-range dependence?**

applications: deep network learning

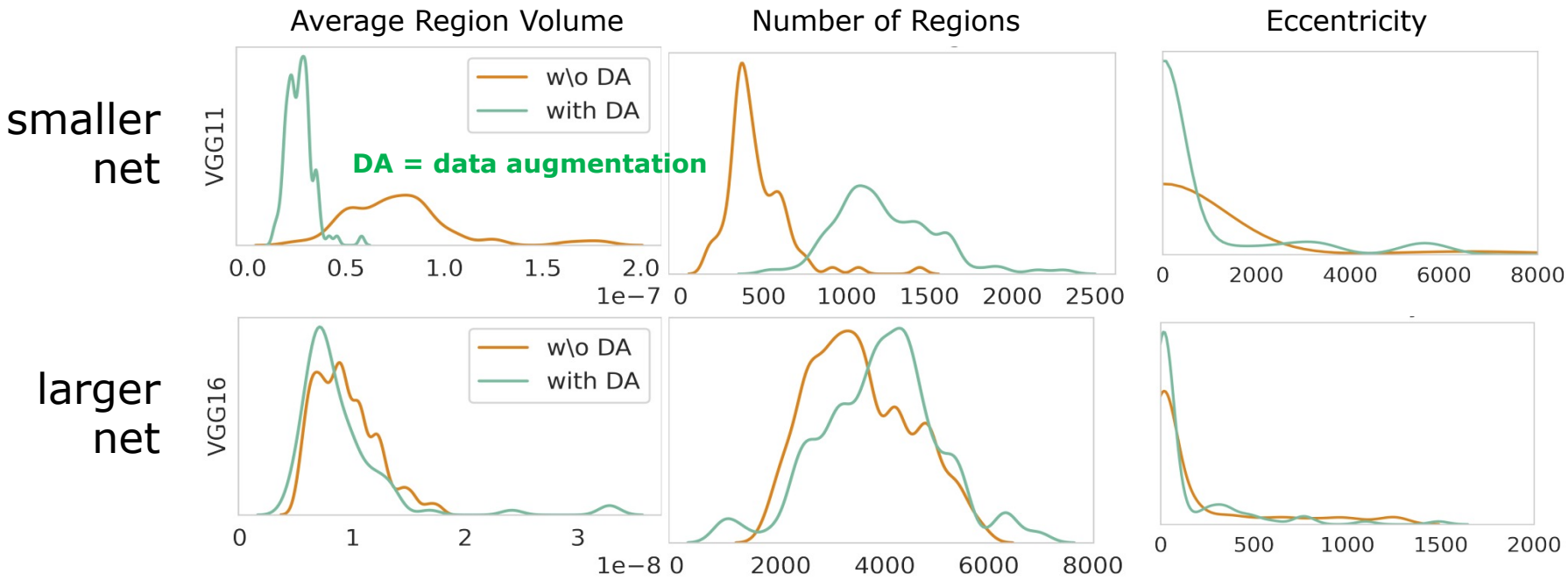
learning dynamics

- **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



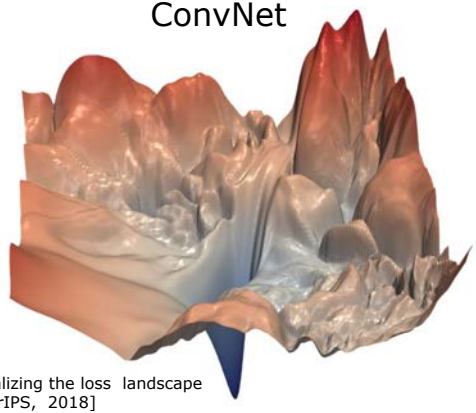
more learning applications

- Explaining why **batch normalization** improves learning

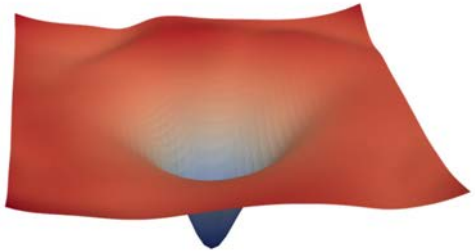
"Batch Normalization Explained," arXiv, 2022

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

ConvNet



ResNet



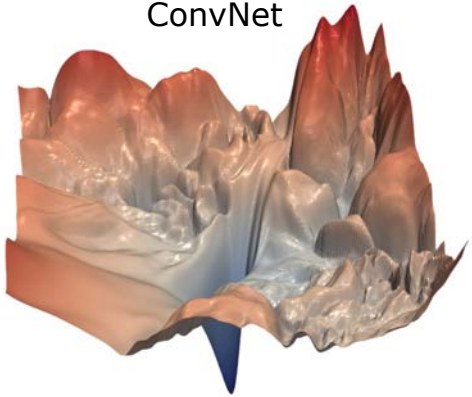
"Singular Value Perturbation and Deep Network Optimization," *Constructive Approximation,* 2022

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

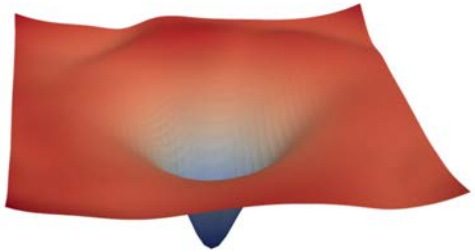
more learning applications

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

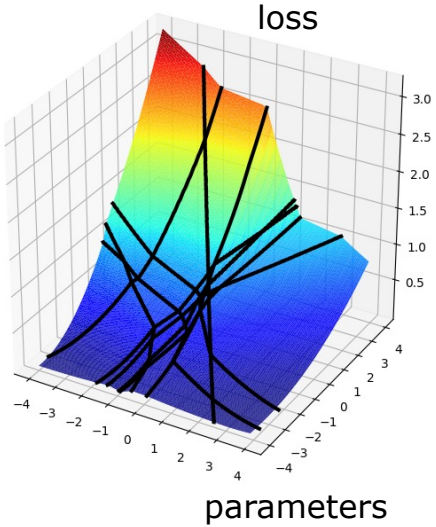
ConvNet



ResNet



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

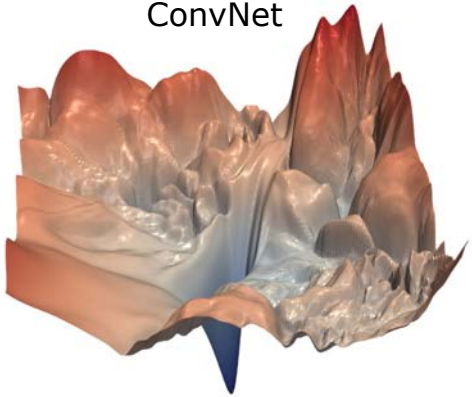


loss surface is **piecewise quadratic/cross-entropic**

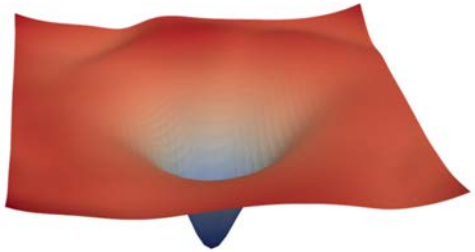
more learning applications

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

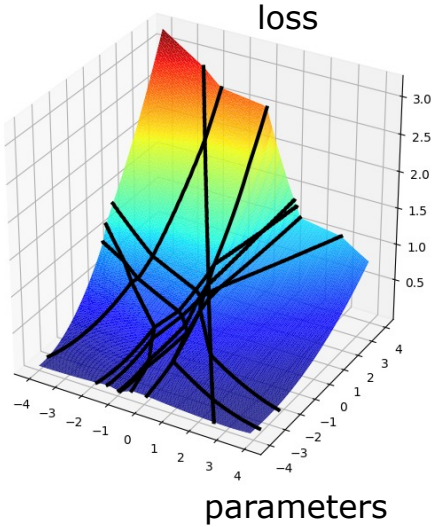
ConvNet



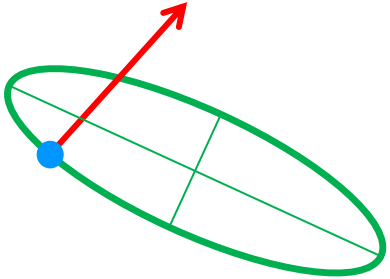
ResNet



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



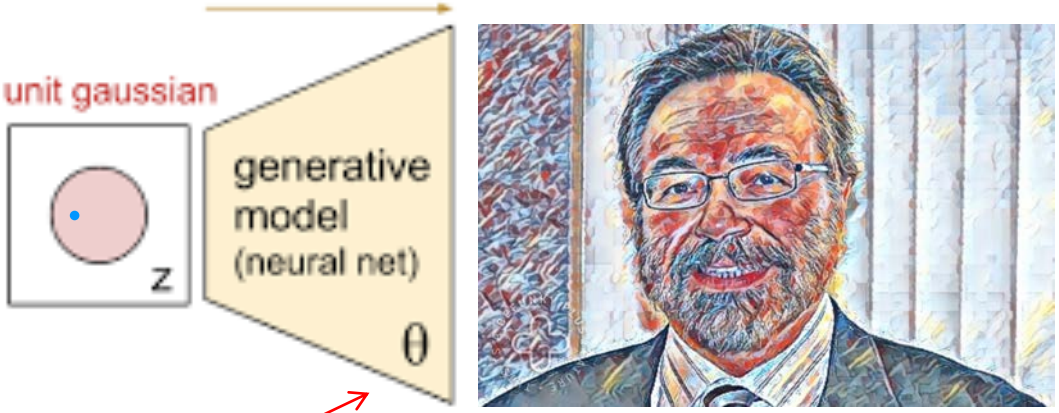
Singular values of local quadratic loss are better behaved for a resnet



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)

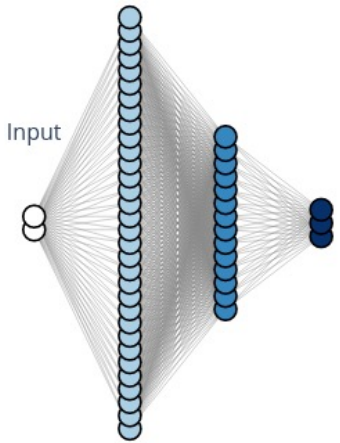


parameters learned from
a gazillion face and art images

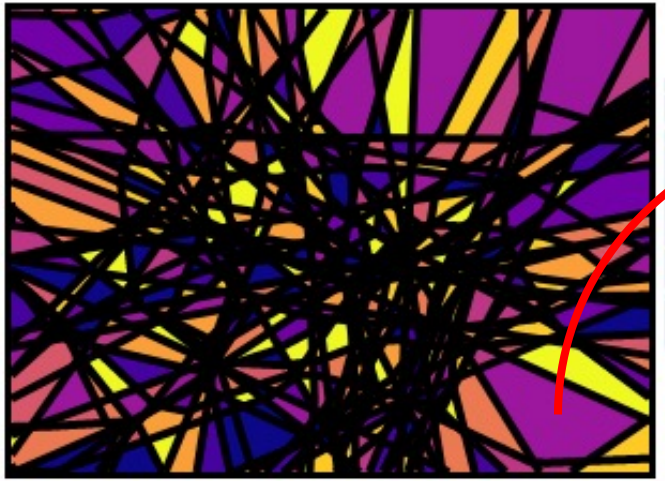
synthesized
Georgios
image

deep generative models

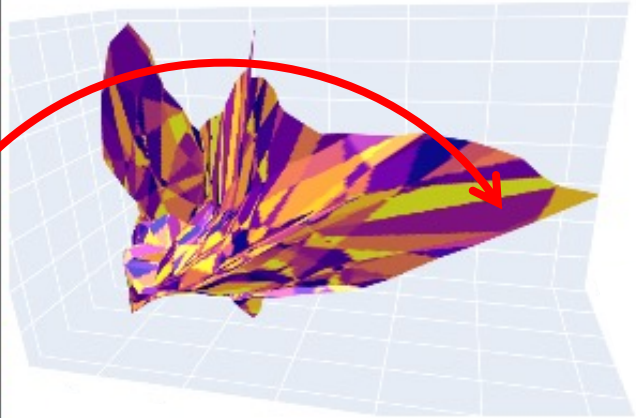
- Exploit fact that **DGN is a CPA spline**
  - DGN maps latent input space to a **CPA manifold** in higher-dimensional space



2D latent input space



3D output data space



Each latent space partition region is warped and placed into the output space by an affine transform

bias in machine learning

The New York Times

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

The Atlantic

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

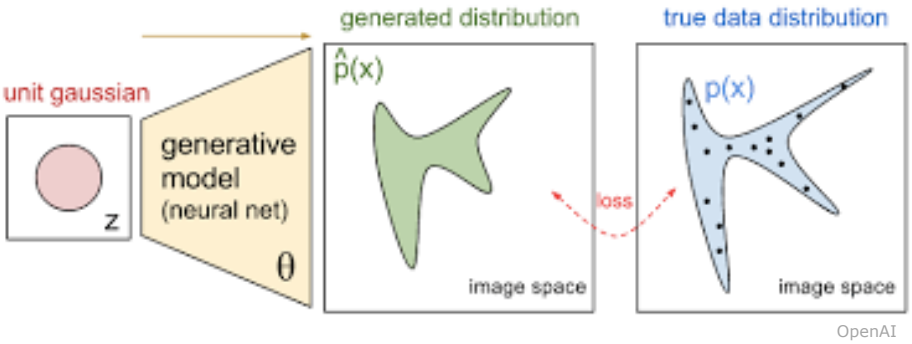
TechCrunch

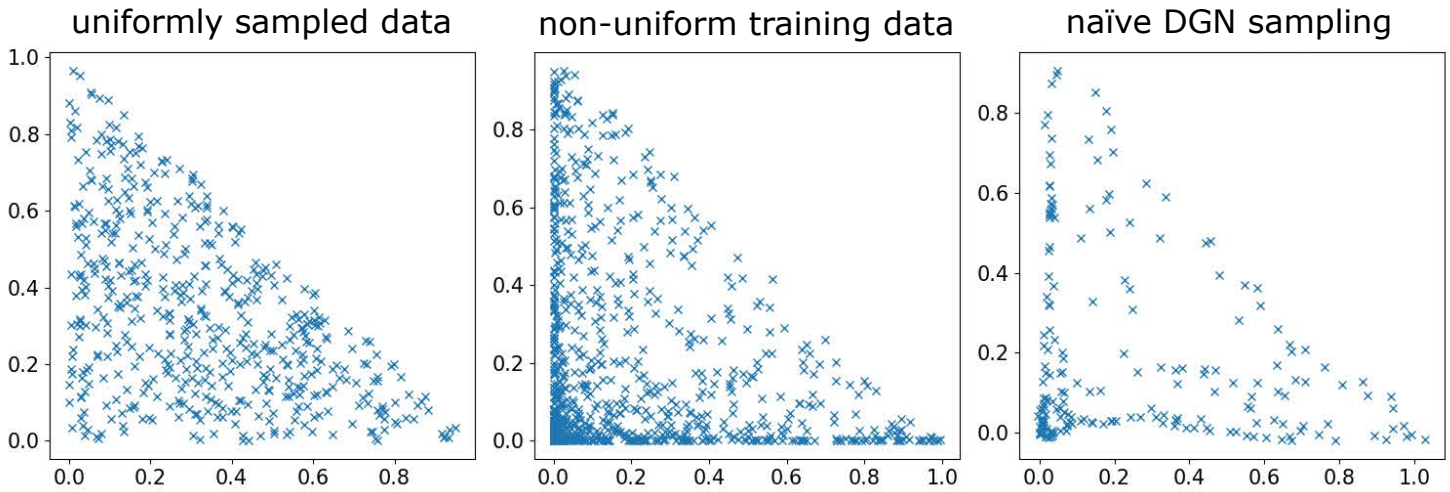
The New York Times

We Teach A.I. Systems Everything, Including Our Biases

biased sampling from DGNs

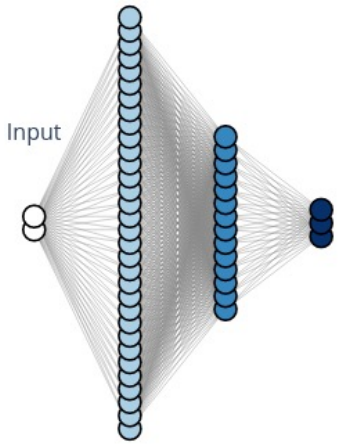
- Training data is often **non-uniformly sampled** from the data manifold
  - results in **biased** DGN samples





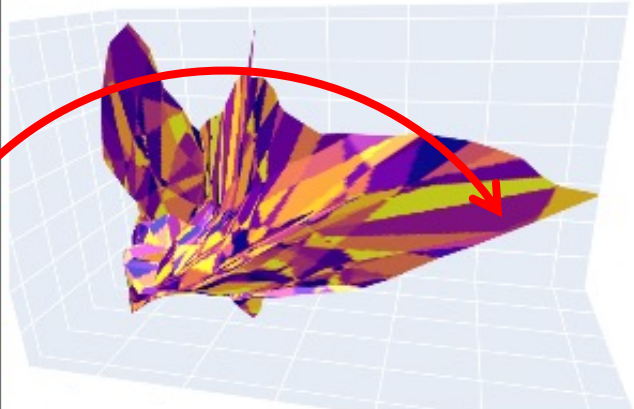
MaGNET (Maximum entropy Generative NETwork)

- 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



2D latent input space

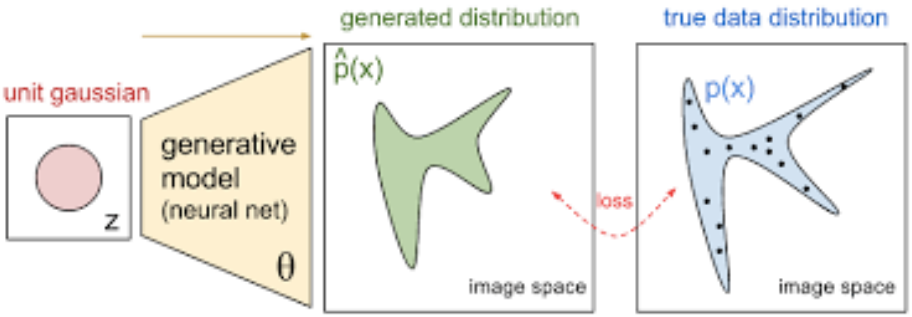
3D output data space



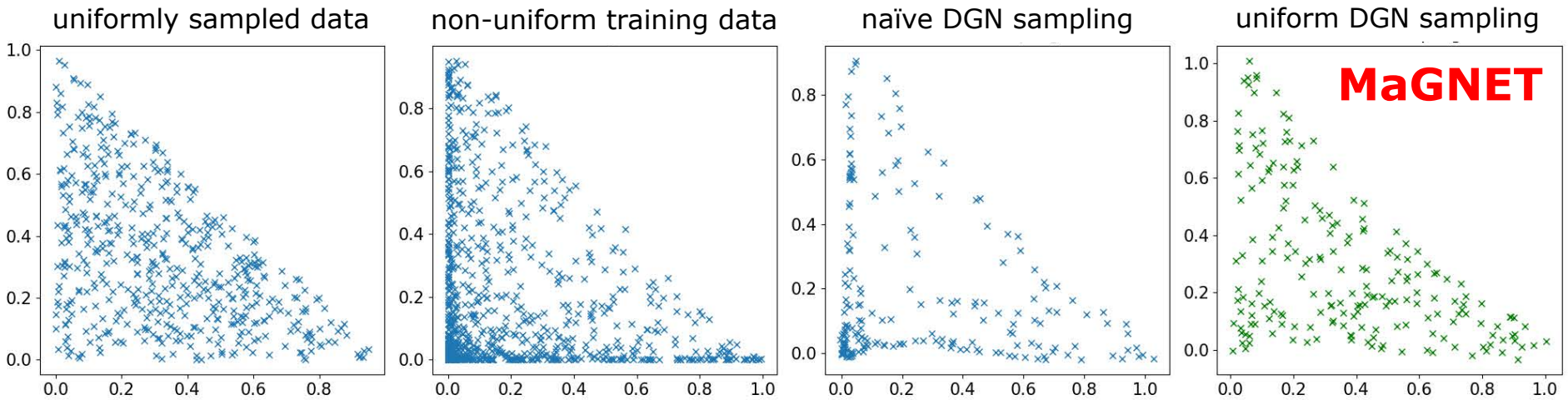
"MaGNET: Uniform Sampling from Deep Generative Network Manifolds Without Retraining," *ICLR*, 2022

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



[OpenAI]

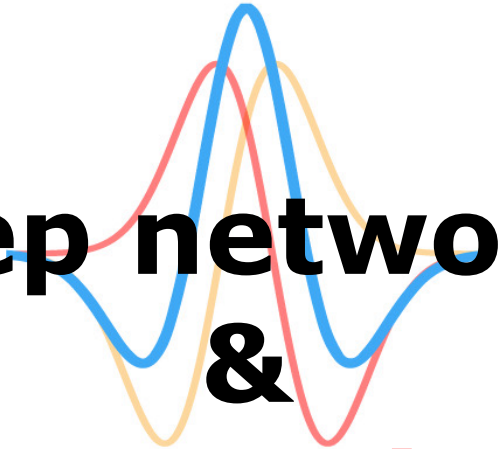


practical example

- Without labels or retraining, MaNET's uniform sampling **reduces gender bias** of FFHQ-StyleGAN2 by **41%**

standard latent sampling

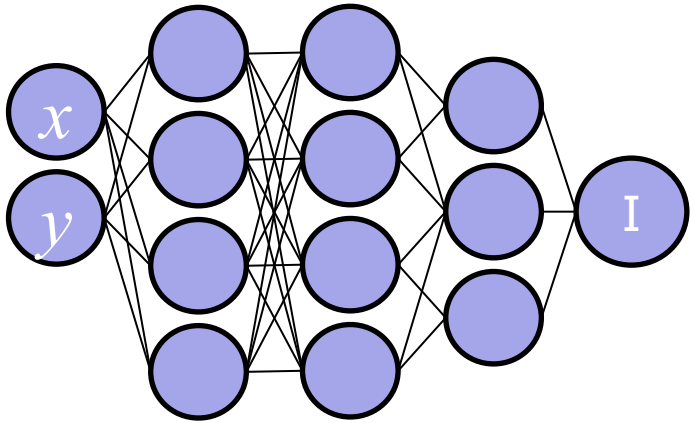
MaGNET latent sampling



deep networks & computational harmonic analysis

implicit neural representation (INR)

- Powerful deep-network based models for **2D images**



Input:
Spatial
Coordinates

Output:
2D Pixel
Intensity

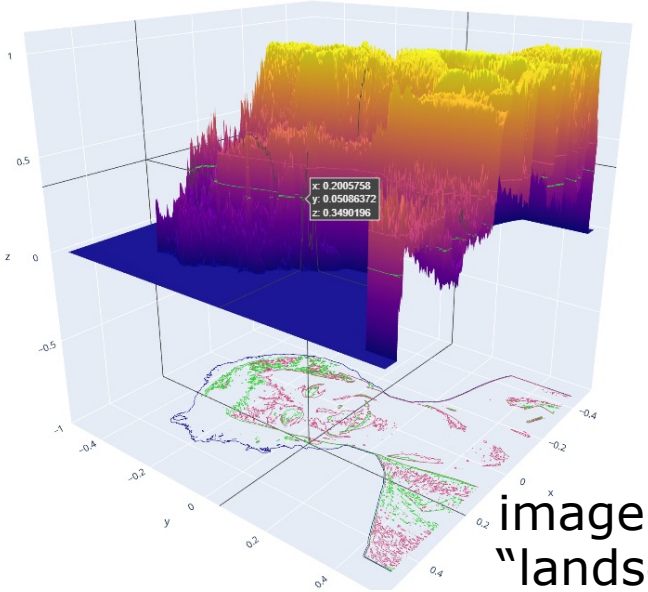


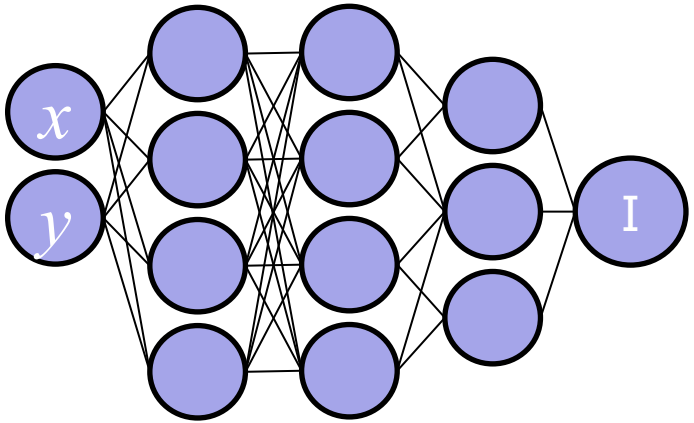
image as a
"landscape"

- Standard activation nonlinearities: ReLU, bumps, sinusoids

Wavelet Implicit neural REpresentations (WIRE)

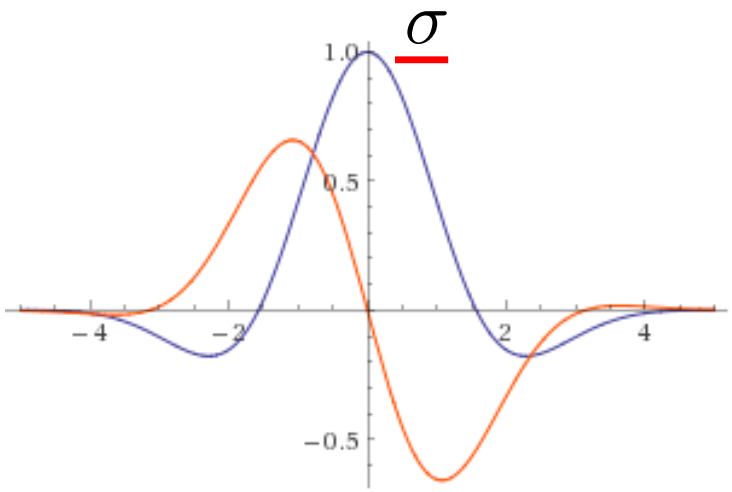
- 1D (complex) **wavelet activation** nonlinearity

$$\underline{\sigma}\left(W_3\,\underline{\sigma}(W_2\,\underline{\sigma}(W_1 u + b_1) + b_2) + b_3\right) \qquad u = \begin{bmatrix} x \\ y \end{bmatrix}$$



Input:
Spatial
Coordinates

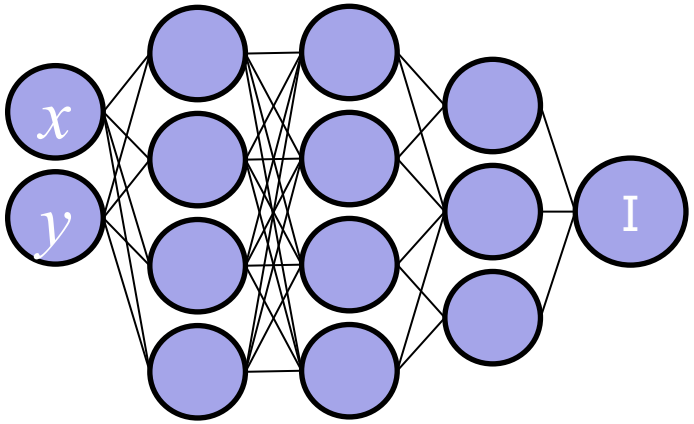
Output:
2D Pixel
Intensity



Wavelet Implicit neural REpresentations (WIRE)

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

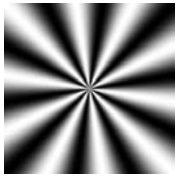
$$\sigma\left(W_3\,\sigma(W_2\,\underline{\sigma(W_1 u + b_1)} + b_2) + b_3\right)$$



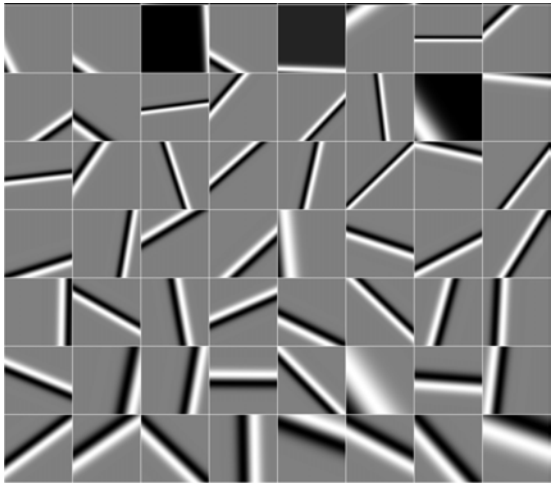
Input:
Spatial
Coordinates

Output:
2D Pixel
Intensity

Input
image



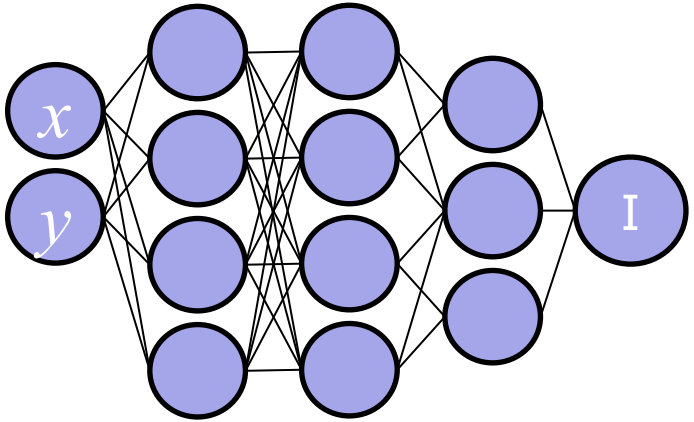
**Layer 1
output**



Wavelet Implicit neural REpresentations (WIRE)

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

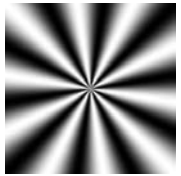
$$\sigma\left(W_3\,\sigma(W_2\,\underline{\sigma(W_1 u + b_1)} + b_2) + b_3\right)$$



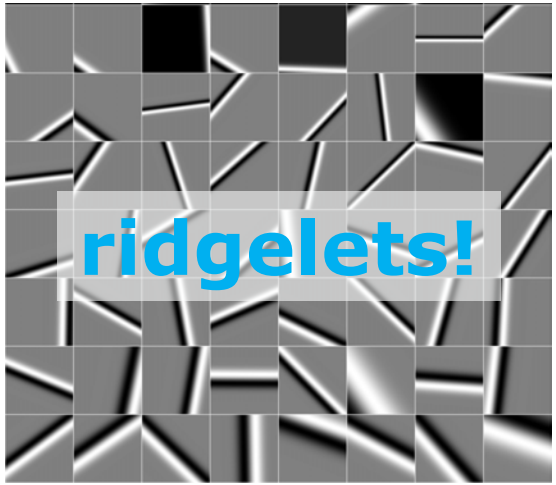
Input:
Spatial
Coordinates

Output:
2D Pixel
Intensity

Input
image

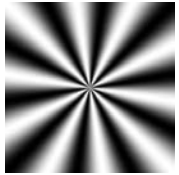


**Layer 1
output**



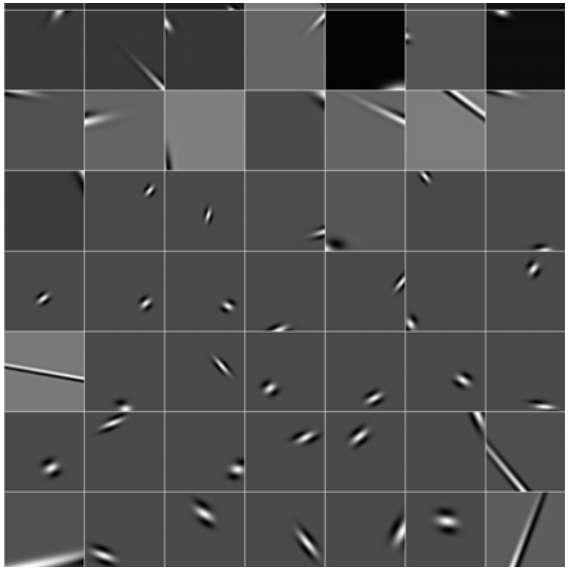
Wavelet Implicit neural REpresentations (WIRE)

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



Input image

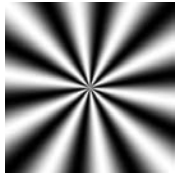
$$\sigma\left(W_3\,\sigma(W_2\,\underline{\sigma(W_1 u + b_1)} + b_2) + b_3\right)$$



Layer 1 output

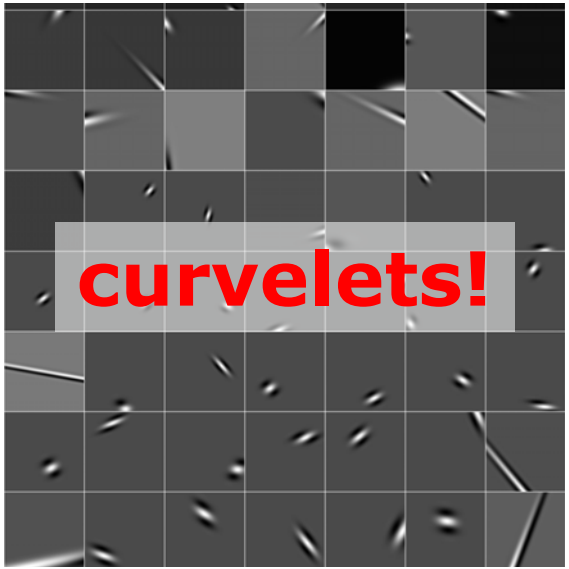
Wavelet Implicit neural REpresentations (WIRE)

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

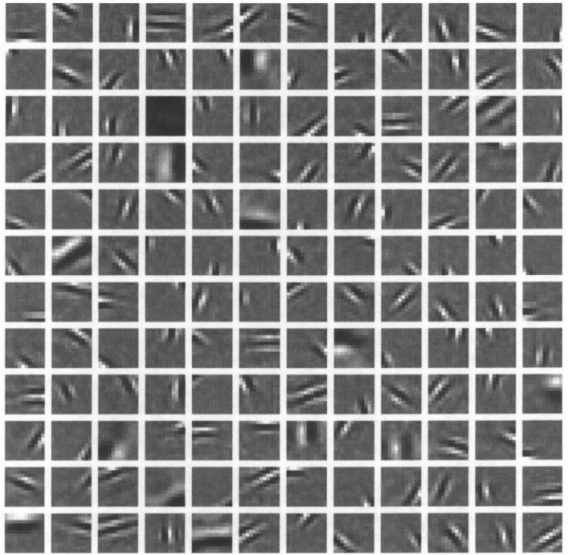


Input image

$$\sigma\left(W_3\,\sigma(W_2\,\underline{\sigma(W_1 u + b_1)} + b_2) + b_3\right)$$



Layer 1 output

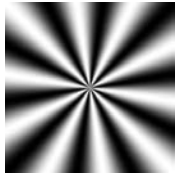


Olshausen & Field, 1996

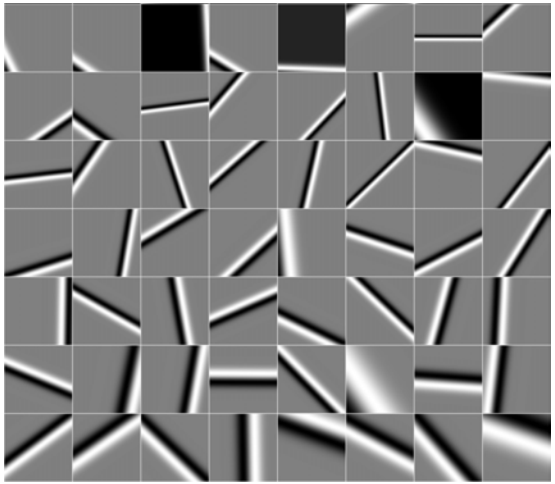
Wavelet Implicit neural REpresentations (WIRE)

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

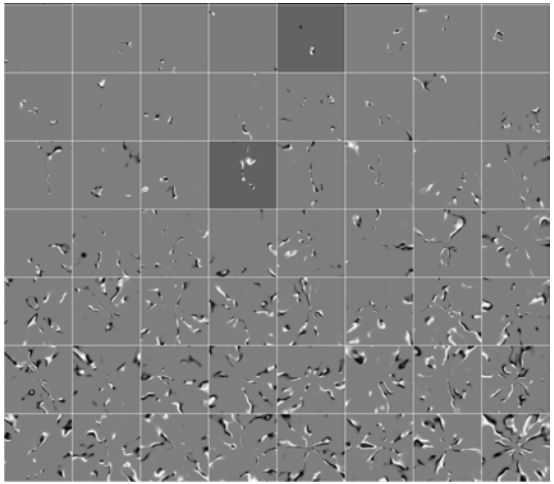
$$\sigma\left(W_3\,\underline{\sigma(W_2\,\sigma(W_1 u + b_1) + b_2)} + b_3\right)$$



Input image

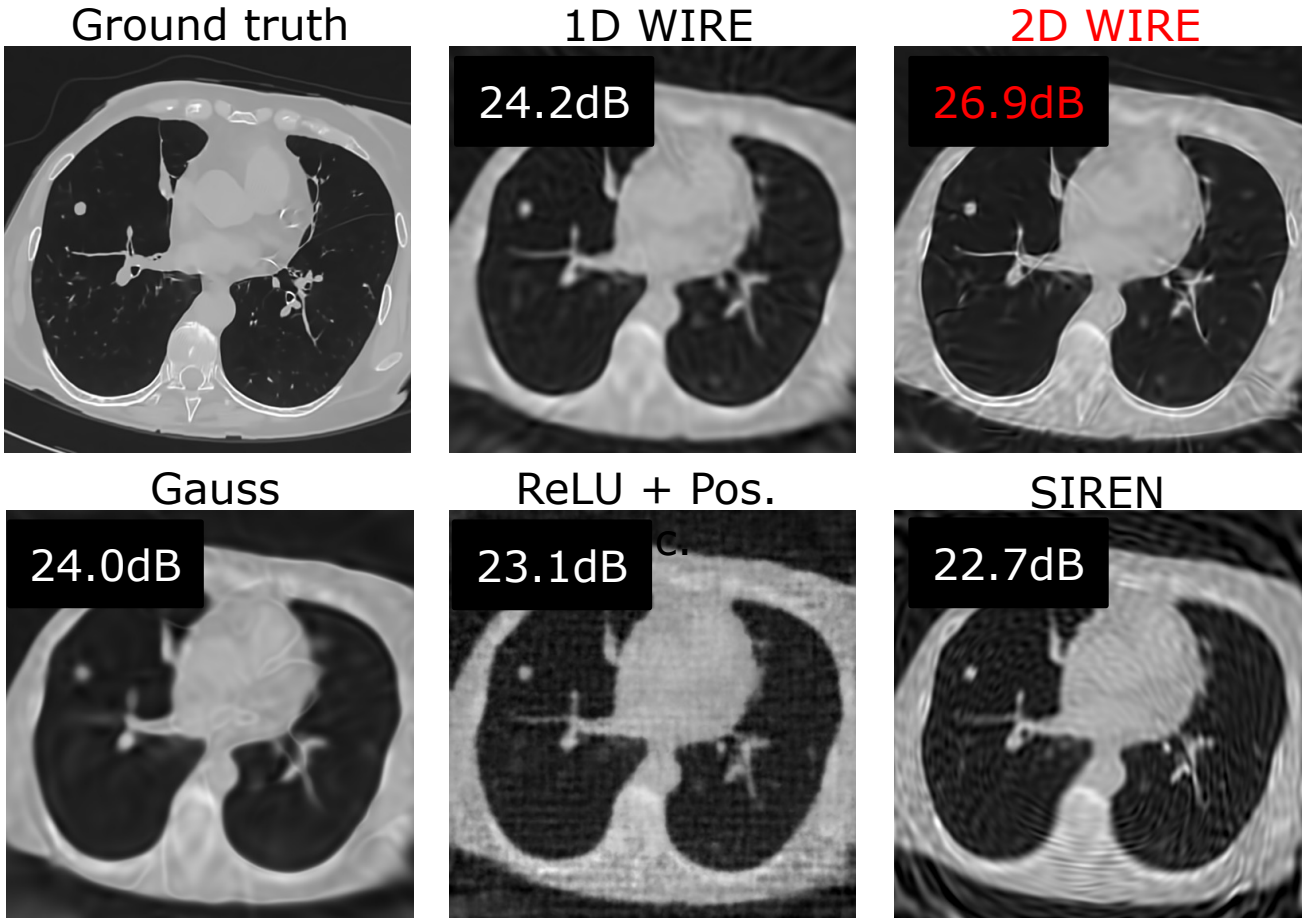


Layer 1 output



Layer 2 output

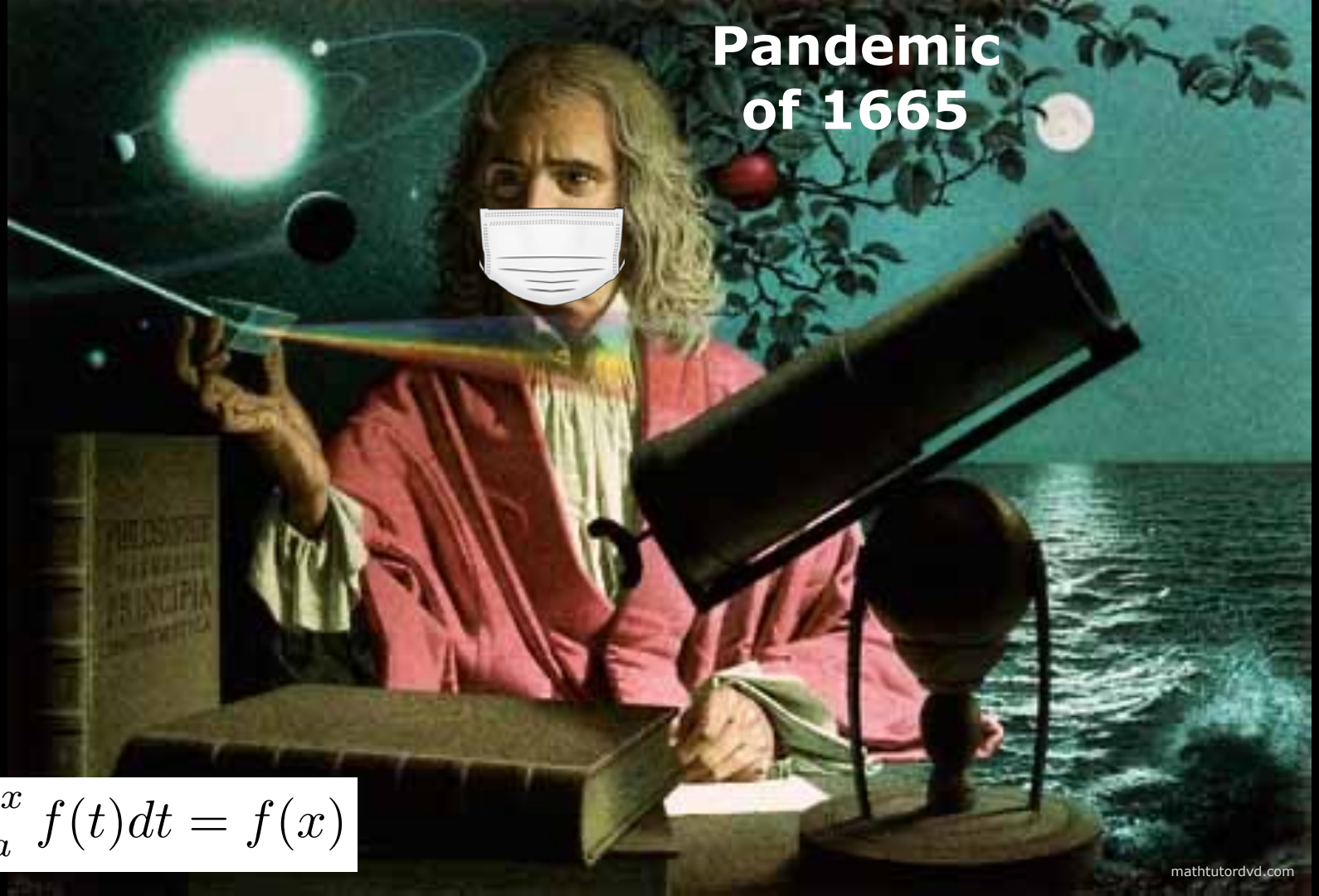
CT image recovery



wrap up

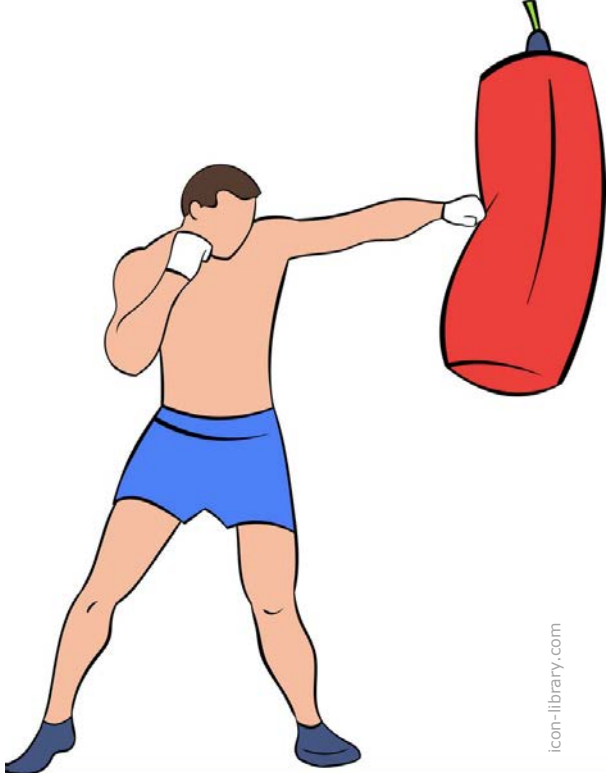
tremendous *alchemistical* progress

$$\frac{d}{dx}\int_a^x f(t)dt = f(x)$$



$$\frac{d}{dx}\int_a^x f(t)dt = f(x)$$

Here's the
punch line



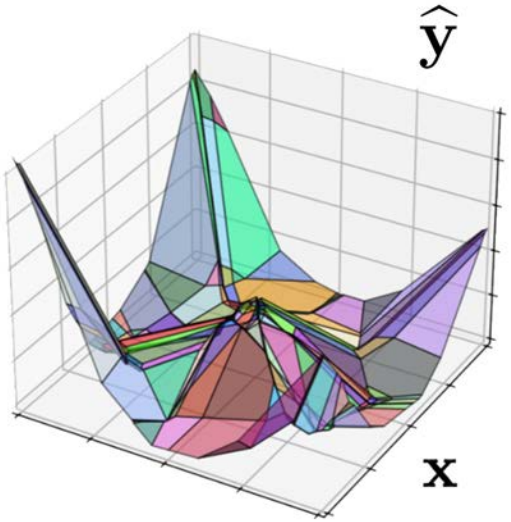
- **(CPA) Splines** provide a firm foundation for a **theory of deep learning**

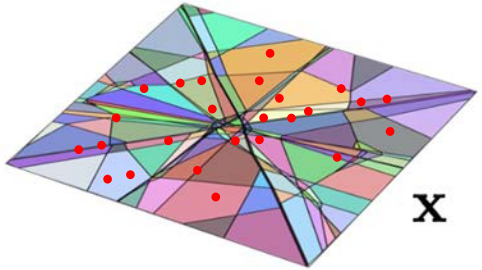
summary

- 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!**





Rice University DSP

dsp.rice.edu

splines & deep learning

"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
"DeepHull: Fast Convex Hull Approximation in High Dimensions," *ICASSP,* 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
"Analytical Probability Distributions and Exact EM for Deep Generative Networks," *NeurIPS,* 2020
"From Hard to Soft: Understanding Deep Network Nonlinearities…," *ICLR,* 2019
"A Max-Affine Spline Perspective of RNNs," *ICLR,* 2019
"The Geometry of Deep Networks: Power Diagram Subdivision," *NeurIPS*, 2019
"Spline Filters for End-to-End Deep Learning," *ICML,* 2018
"A Spline Theory of Deep Networks," *ICML,* 2018