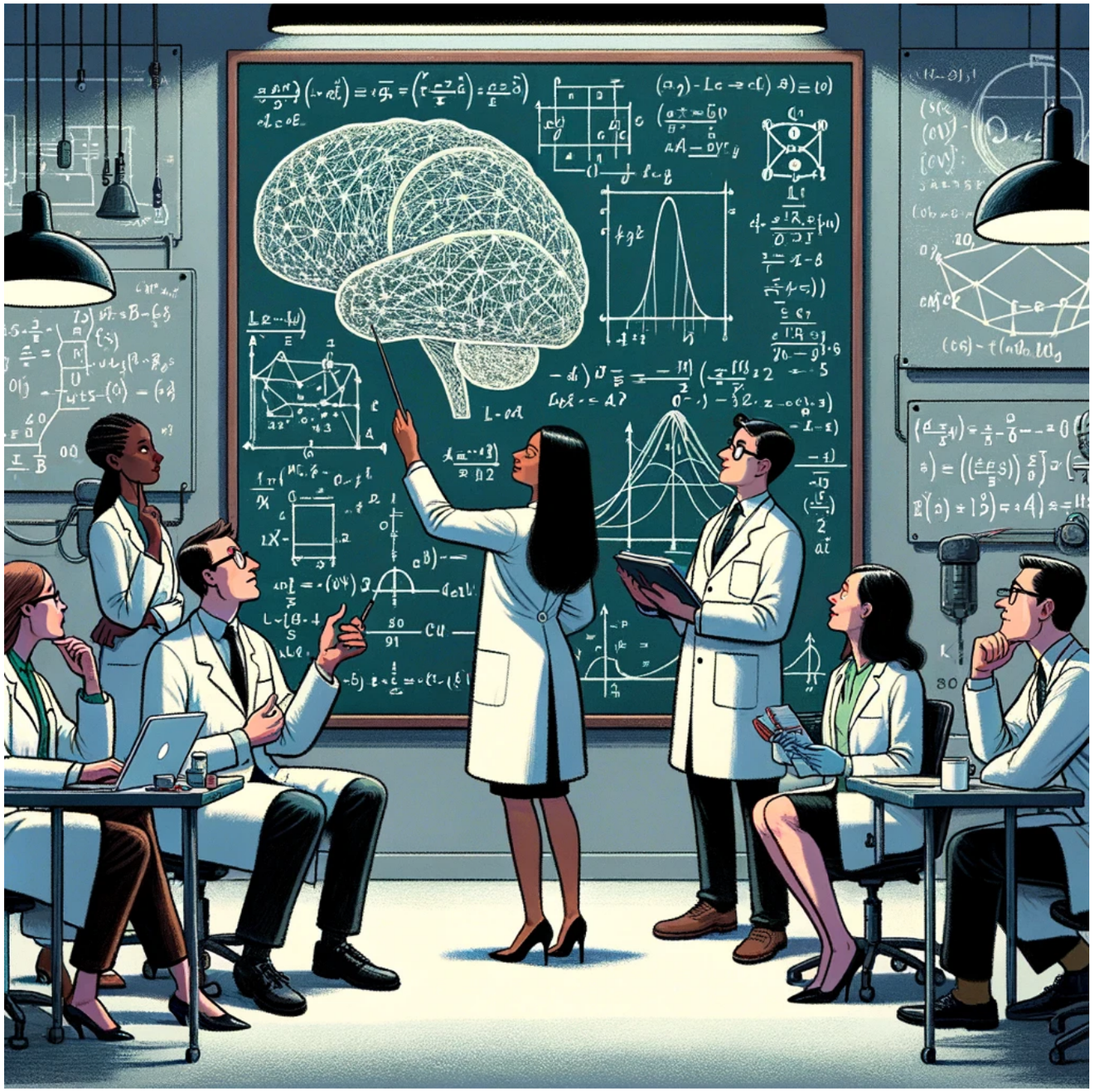
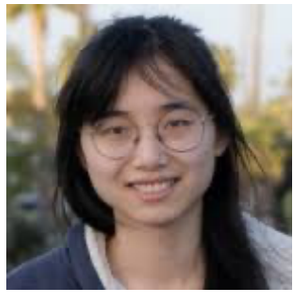
Deeper Understanding of Deep Learning: Functional Analysis for Neural Networks



Robert Nowak
University of Wisconsin-Madison
https://nowak.ece.wisc.edu

Joint work with

Rahul Parhi



Liu Yang

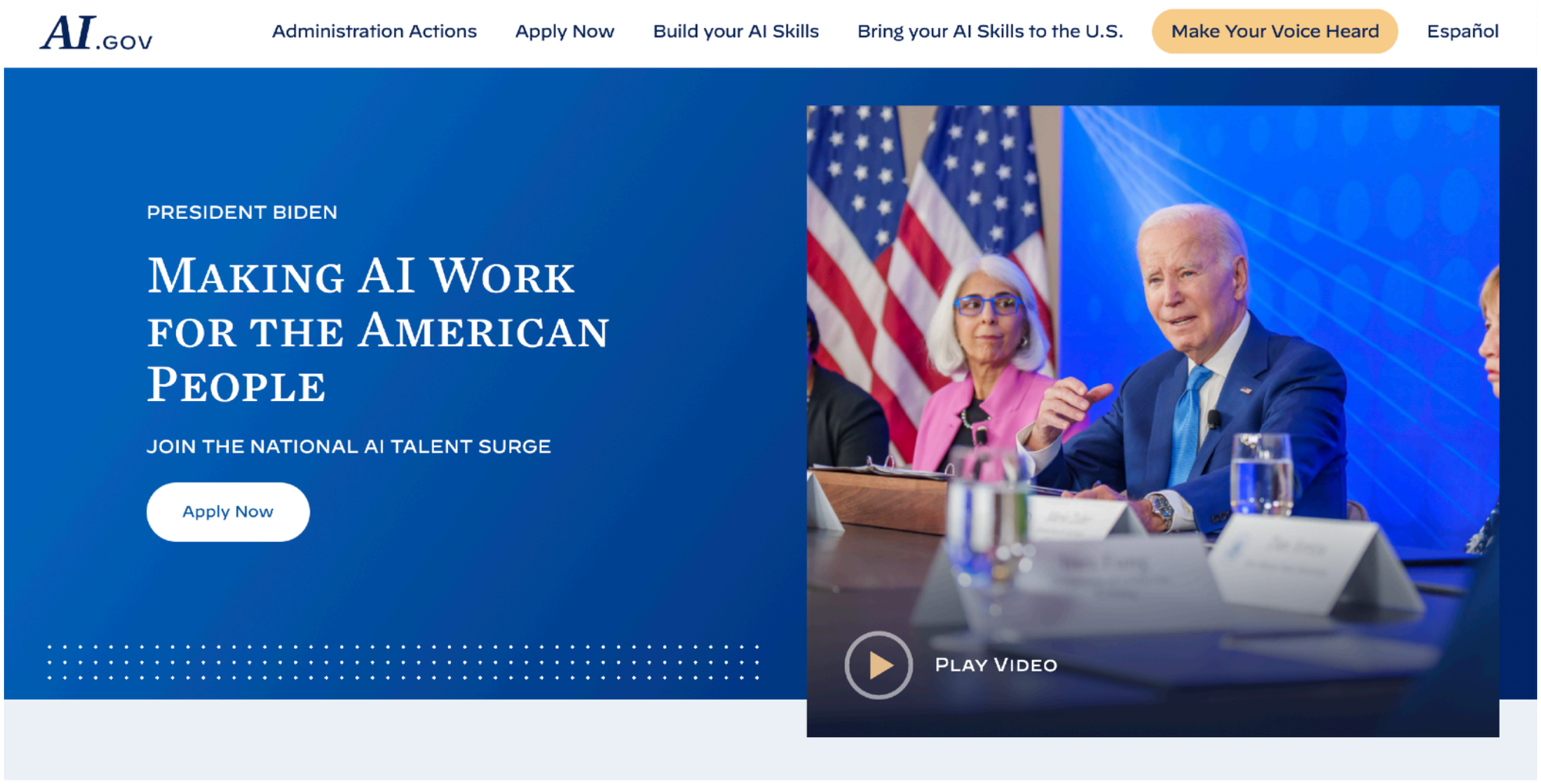
Joe Shenouda

TAMIDS Seminar & Tutorial Series
Fall 2023

TEXAS A&M Institute of Data Science

November 6, 2023

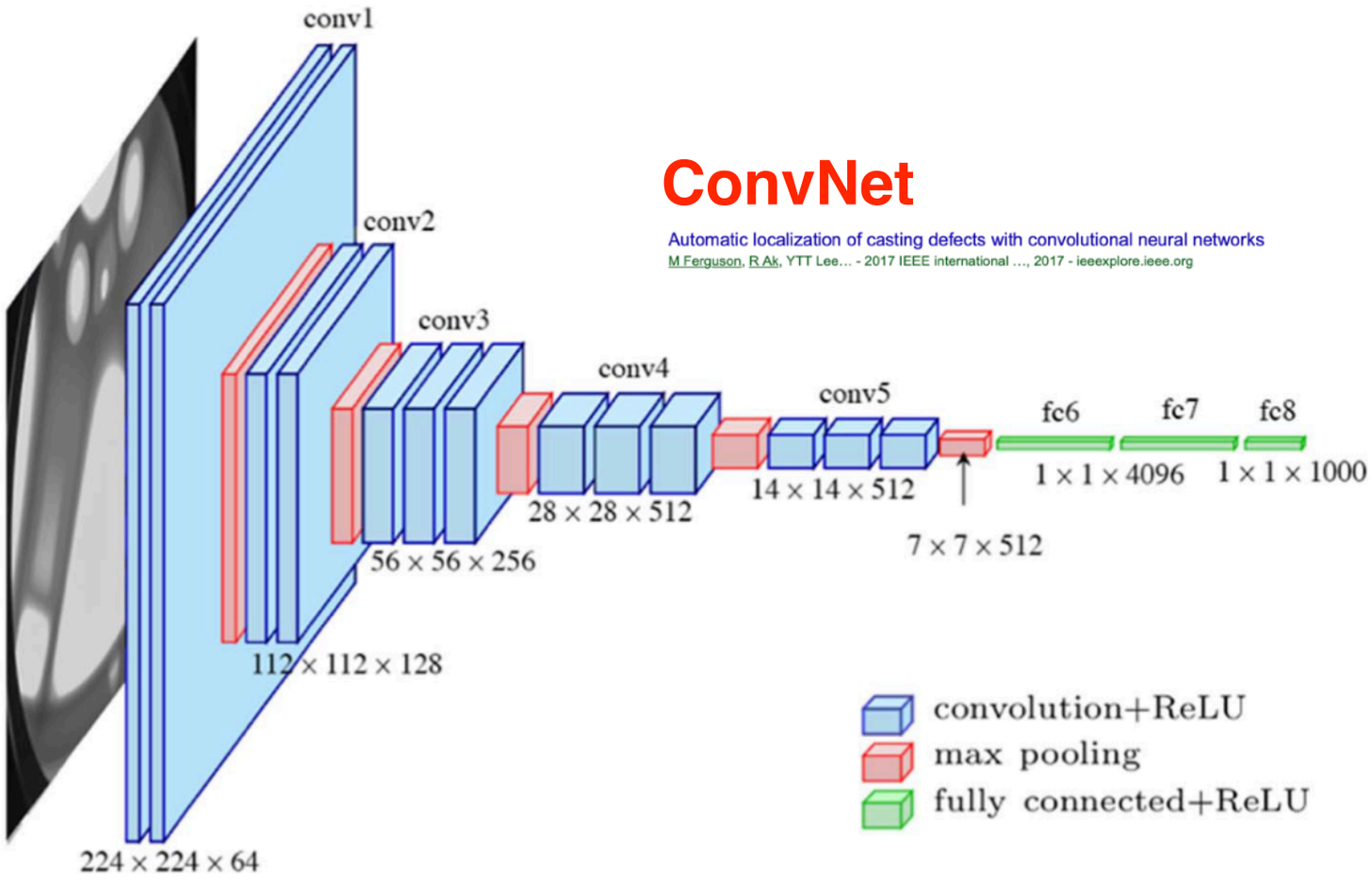
Here we are today: AI and Society



AI is one of the most powerful technologies of our time. President Biden has been clear that we must take bold action to harness the benefits and mitigate the risks of AI. The Biden-Harris Administration has acted decisively to protect safety and rights in the age of AI, so that everyone can benefit from the promise of AI.

Develop standards, tools, and tests to help ensure that AI systems are safe, secure, and trustworthy

Modern Neural Network Architectures



ConvNet

Automatic localization of casting defects with convolutional neural networks
M Ferguson, R Ak, YTT Lee... - 2017 IEEE international ..., 2017 - ieeexplore.ieee.org

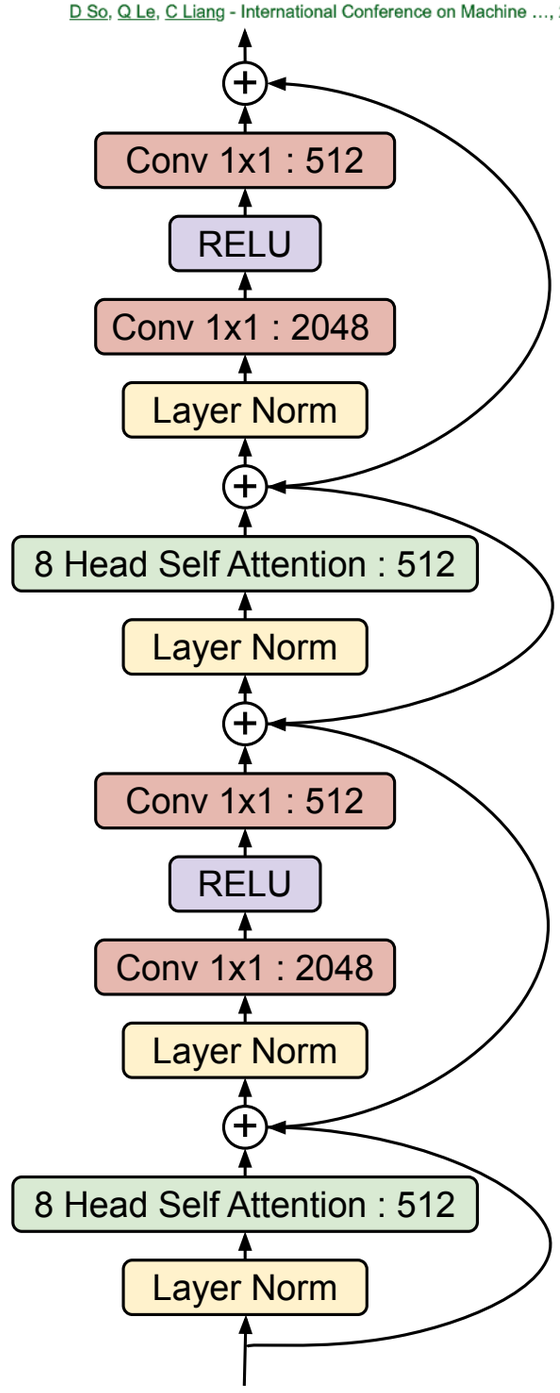
Rectified Linear Unit (ReLU)

$$(z)_+ = \max\{0, z\}$$

+ weight decay in training

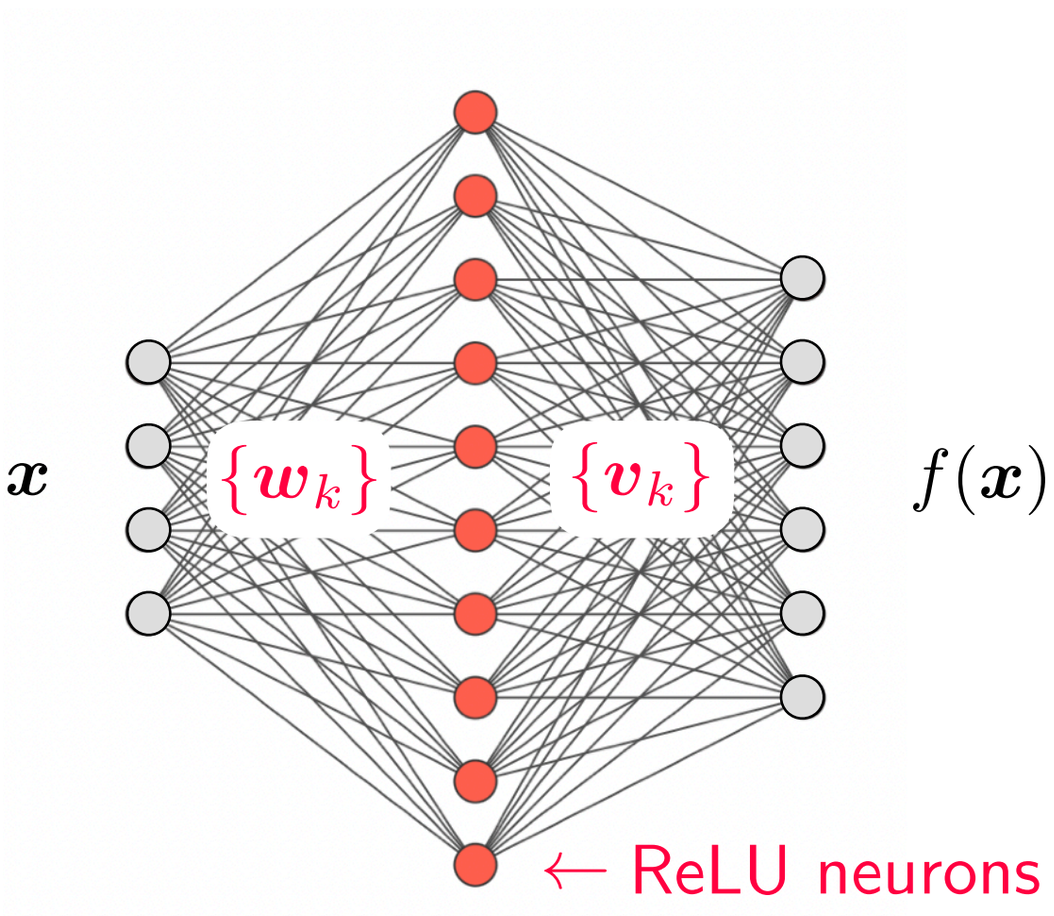
Transformer

The **evolved transformer**
D So, Q Le, C Liang - International Conference on Machine ..., 2019



Training Neural Networks

Neural Networks with Rectified Linear Units (ReLUs)



Rectified Linear Unit (ReLU)

$$(z)_+ = \max\{0, z\}$$

$$f(\boldsymbol{x}) \;=\; \sum_{j=1}^{K} \boldsymbol{v}_j \, (\boldsymbol{w}_j^T \boldsymbol{x})_+$$

"Training" Neural Networks

Consider the optimization

$$\min_{\boldsymbol{v},\boldsymbol{w}} \underbrace{\frac{1}{2}\sum_{i=1}^{n}\left\|\boldsymbol{y}_i - \sum_j \boldsymbol{v}_j\left(\boldsymbol{w}_j^T\boldsymbol{x}_i\right)_+\right\|_2^2}_{\text{data fitting term } \quad L(\boldsymbol{v},\boldsymbol{w})} + \underbrace{\frac{\lambda}{2}\sum_j \|\boldsymbol{v}_j\|_2^2 + \|\boldsymbol{w}_j\|_2^2}_{\text{regularize sum of squared weights}}$$

Gradient descent update for $\boldsymbol{v}_j$ (same for $\boldsymbol{w}_j$)

$$\begin{aligned}\boldsymbol{v}_j &\leftarrow \boldsymbol{v}_j - \mu\left(\nabla_{\boldsymbol{v}_j} L + \lambda\,\boldsymbol{v}_j\right)\\ &\leftarrow \boldsymbol{v}_j - \mu\,\nabla_{\boldsymbol{v}_j} L - \mu\lambda\,\boldsymbol{v}_j\end{aligned}$$

descent on L — weight decay

Standard approach when training neural networks (possibly with a different loss)

Neural Balance

because ReLU is piecewise linear

$$\boldsymbol{v}_j(\boldsymbol{w}_j^T\boldsymbol{x})_+ \;=\; \alpha^{-1}\,\boldsymbol{v}_j\left(\alpha\boldsymbol{w}_j^T\boldsymbol{x}\right)_+\,,\;\forall\alpha>0$$

"Neural Balance"

at the global minimum of weight decay objective $\|\boldsymbol{v}_j\|_2 = \|\boldsymbol{w}_j\|_2$

Proof: The solution to the optimization

$$\min_{\alpha>0}\|\alpha\,\boldsymbol{w}\|_2^2+\|\alpha^{-1}\boldsymbol{v}\|_2^2$$

is $\alpha=\sqrt{\|\boldsymbol{v}\|_2/\|\boldsymbol{w}\|_2}$

at the global minimum $\|\boldsymbol{v}_j\|_2 = \|\boldsymbol{w}_j\|_2$ and therefore

$$\frac{1}{2}\left(\|\boldsymbol{v}_j\|_2^2+\|\boldsymbol{w}_j\|_2^2\right) \;=\; \|\boldsymbol{v}_j\|_2\,\|\boldsymbol{w}_j\|_2$$

Grandvalet, Y. (1998). Least absolute shrinkage is equivalent to quadratic penalization. In *International Conference on Artificial Neural Networks* (pp. 201-206).

Neyshabur, B., Tomioka, R., & Srebro, N. (2015). In Search of the Real Inductive Bias: On the Role of Implicit Regularization in Deep Learning. In *ICLR (Workshop)*.

Secret Sparsity of Weight Decay

gradient descent with weight decay aims to minimize
sum of losses + sum of squared weights

$$\min_{\boldsymbol{v},\boldsymbol{w}} \frac{1}{2}\sum_{i=1}^{n}\left\|\boldsymbol{y}_i - \sum_j \boldsymbol{v}_j\left(\boldsymbol{w}_j^T\boldsymbol{x}_i\right)_+\right\|_2^2 + \frac{\lambda}{2}\sum_j \|\boldsymbol{v}_j\|_2^2 + \|\boldsymbol{w}_j\|_2^2$$

squared ℓ^2 norm

same global minima as "path-norm" regularization

$$\min_{\boldsymbol{v},\boldsymbol{w}} \frac{1}{2}\sum_{i=1}^{n}\left\|\boldsymbol{y}_i - \sum_j \boldsymbol{v}_j\left(\boldsymbol{w}_j^T\boldsymbol{x}_i\right)_+\right\|_2^2 + \lambda\sum_j \|\boldsymbol{v}_j\|_2\|\boldsymbol{w}_j\|_2$$

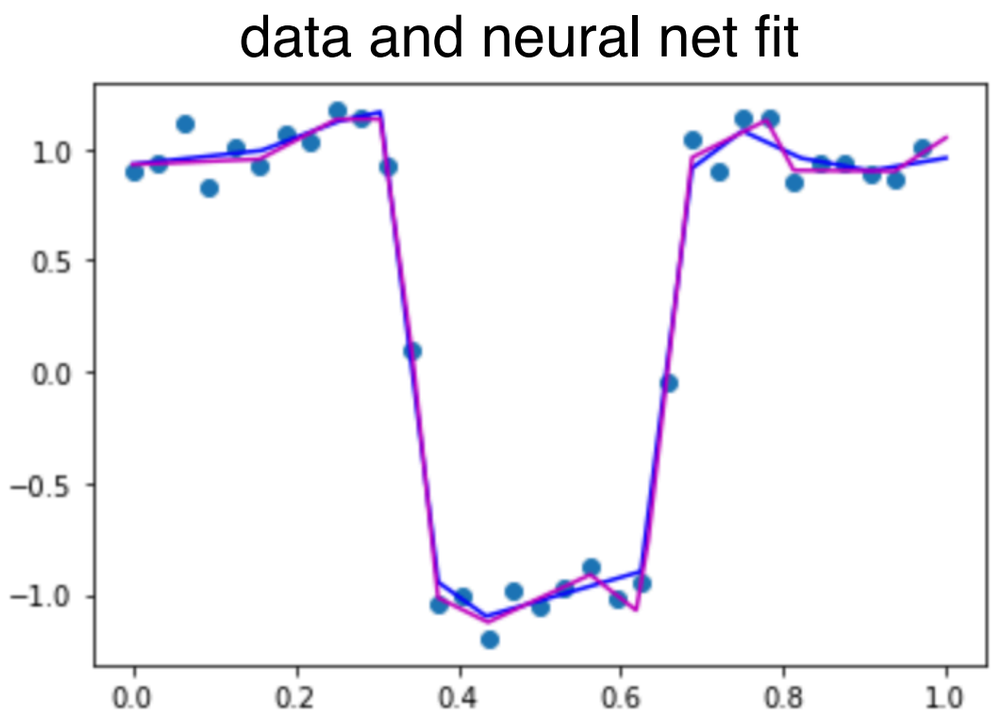
"path-norm"

same global minima as *constrained* ℓ^1 regularization

$$\min_{\boldsymbol{v},\boldsymbol{w}:\|\boldsymbol{w}_j\|_2=1} \frac{1}{2}\sum_{i=1}^{n}\left\|\boldsymbol{y}_i - \sum_j \boldsymbol{v}_j\left(\boldsymbol{w}_j^T\boldsymbol{x}_i\right)_+\right\|_2^2 + \lambda\sum_j \|\boldsymbol{v}_j\|_2$$

"multi-task" lasso

Weight Decay Produces Sparse Networks



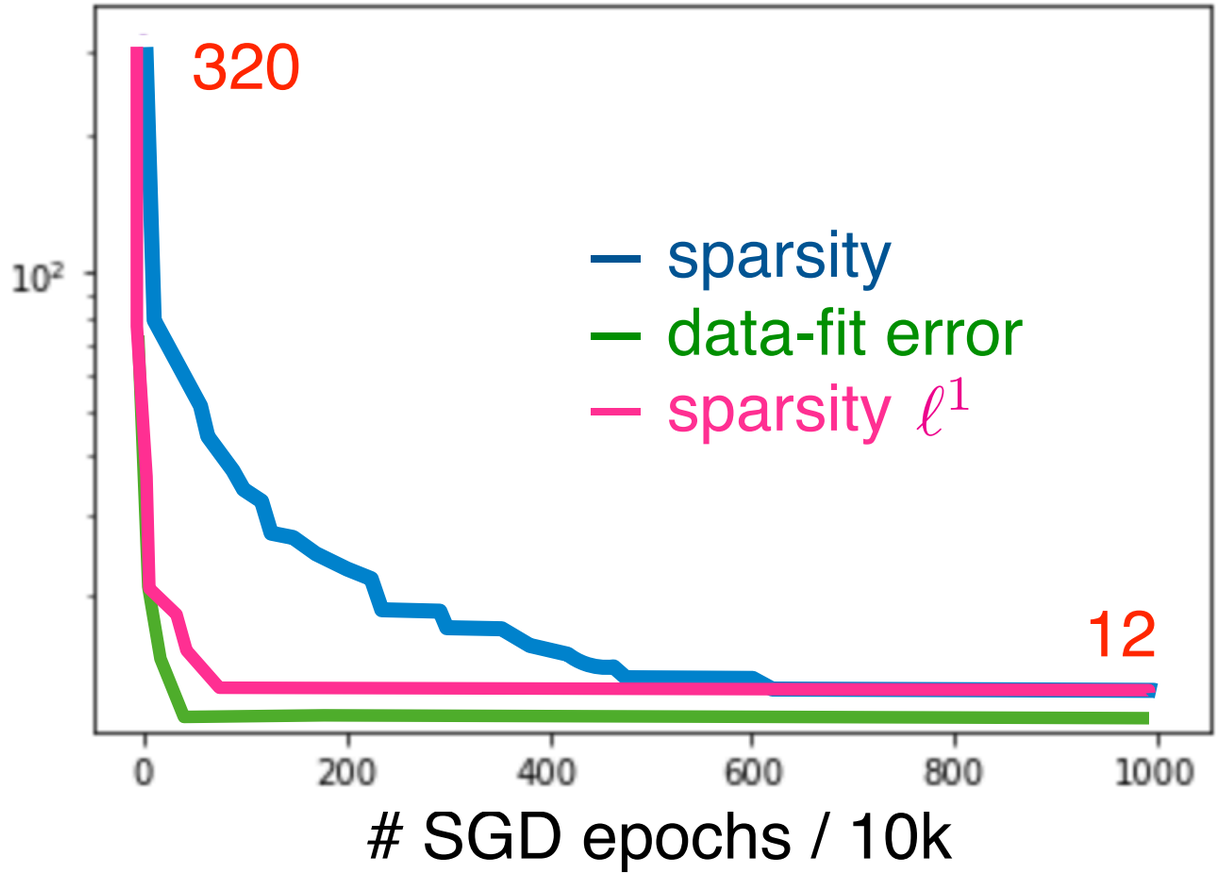
training a 320 neuron ReLU network to fit 32 data points

weight decay (eventually) produces a very sparse neural network …

after millions of GD steps!

equivalent ℓ^1 path-norm regularization (gradient descent + soft-thresholding) converges 15X faster

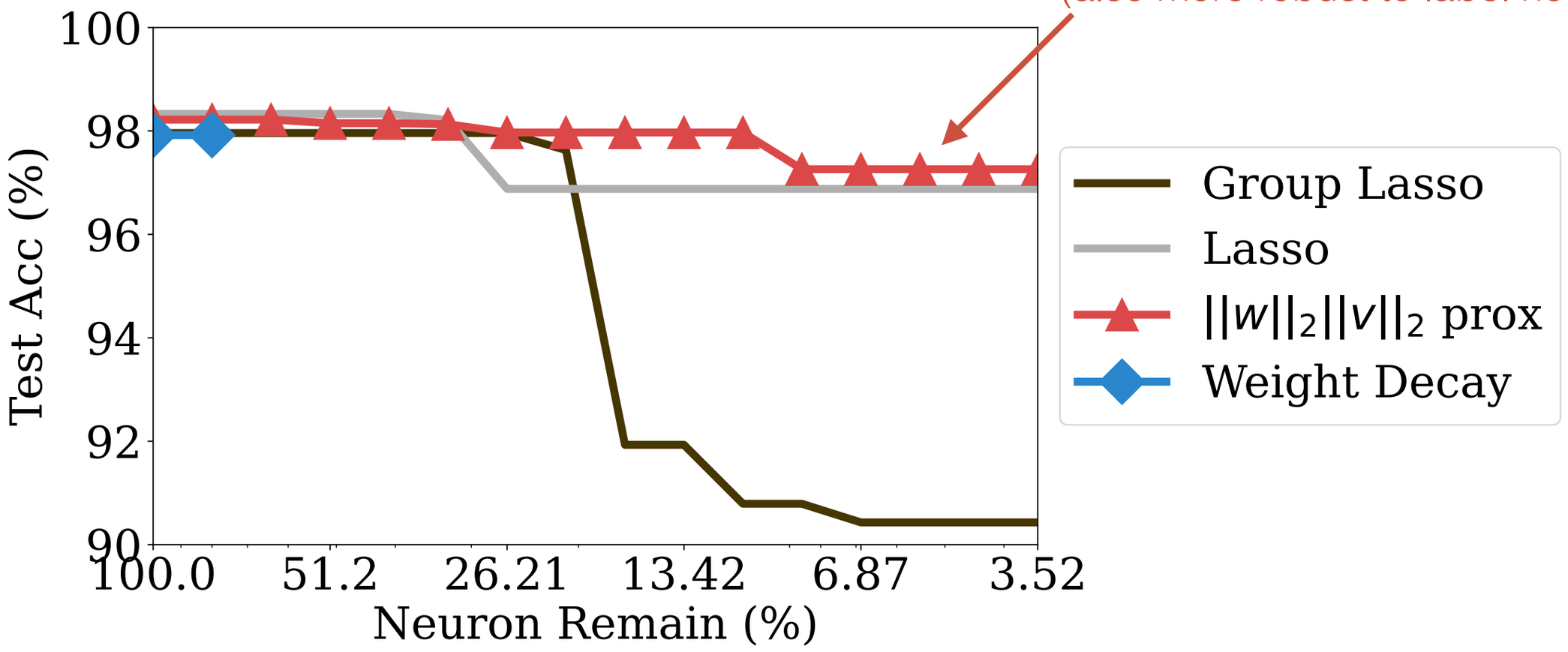
sparsity = # nonzero weights



Accurate, Robust, and Sparser Networks

prox gradient best at finding sparse and accurate models

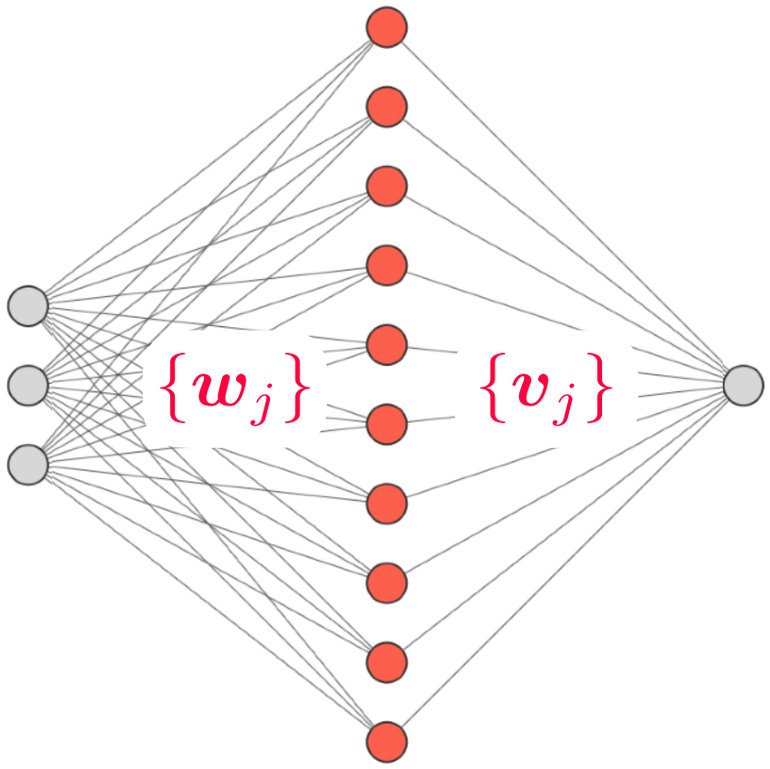
(also more robust to label noise)



architecture: three-layer, fully connected, dataset: MNIST

What Kinds of Functions Do Neural Networks Learn?

Shallow Networks and Neural Balance



Equivalent optimizations:

$$\min_{\boldsymbol{v},\boldsymbol{w}} \frac{1}{2}\sum_{i=1}^{n}\left\|y_i - \sum_j v_j\left(\boldsymbol{w}_j^T\boldsymbol{x}_i\right)_+\right\|_2^2 + \frac{\lambda}{2}\sum_j |v_j|_2^2 + \|\boldsymbol{w}_j\|_2^2$$

weight decay regularizer

$$\min_{\boldsymbol{v},\boldsymbol{w}} \frac{1}{2}\sum_{i=1}^{n}\left\|y_i - \sum_j v_j\left(\boldsymbol{w}_j^T\boldsymbol{x}_i\right)_+\right\|_2^2 + \lambda\sum_j |v_j|\|\boldsymbol{w}_j\|_2$$

"path-norm"

Norm on the Space of Neural Network Functions

$$\mathcal{F} \;=\; \left\{ f : \mathbb{R}^d \to \mathbb{R} \;:\; f(\boldsymbol{x}) \;=\; \sum_{j=1}^{K} v_j \left(\boldsymbol{w}_j^T \boldsymbol{x}\right)_+ \;,\; \boldsymbol{w}_j \in \mathbb{R}^d \,, v_j \in \mathbb{R} \right\}$$

The **path-norm** is a norm on $\mathcal{F}$

$$\|f\| \;:=\; \sum_j |v_j| \|\boldsymbol{w}_j\|_2$$

Completion of $\mathcal{F}$ with respect to $\|\cdot\|$ is the Banach space of all functions generated by "infinite-width" neural networks. Each such function f is expressed as an integral over neurons with respect to a finite measure ν

$$f(\boldsymbol{x}) \;=\; \int_{\boldsymbol{w} \in \mathbb{S}^{d-1}} \left(\boldsymbol{w}^T \boldsymbol{x}\right)_+ \, d\nu(\boldsymbol{w})$$

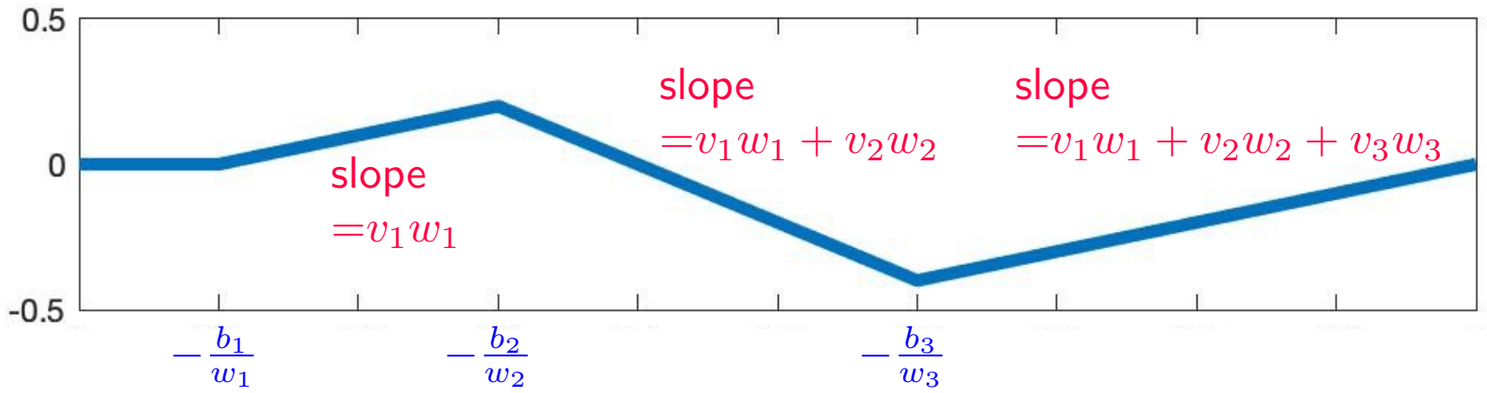
Universal approximation bounds for superpositions of a sigmoidal function
AR **Barron** - IEEE Transactions on Information theory, **1993** - ieeexplore.ieee.org

Breaking the curse of dimensionality with convex neural networks
F Bach - The Journal of Machine Learning Research, 2017 - dl.acm.org

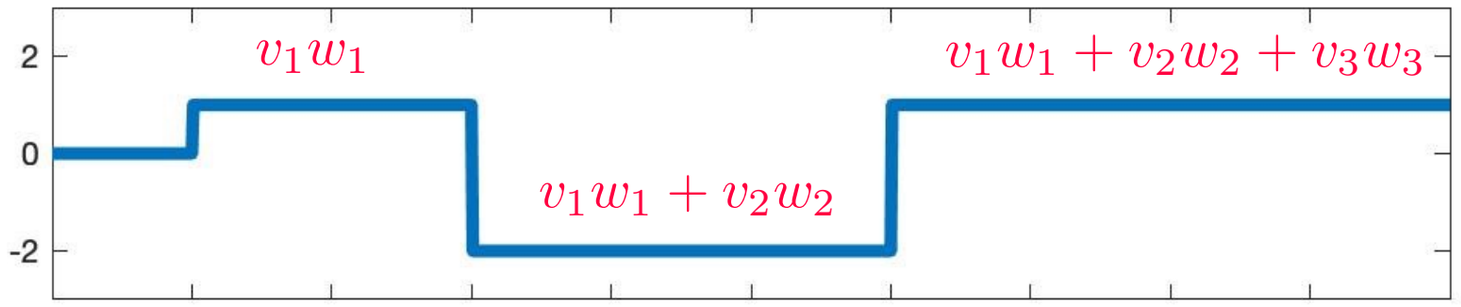
Path-Norm Related to Derivatives of f

$$f(x) \;=\; v_1\,(w_1 x + b_1)_+ \;+\; v_2\,(w_2 x + b_2)_+ \;+\; v_3\,(w_3 x + b_3)_+$$

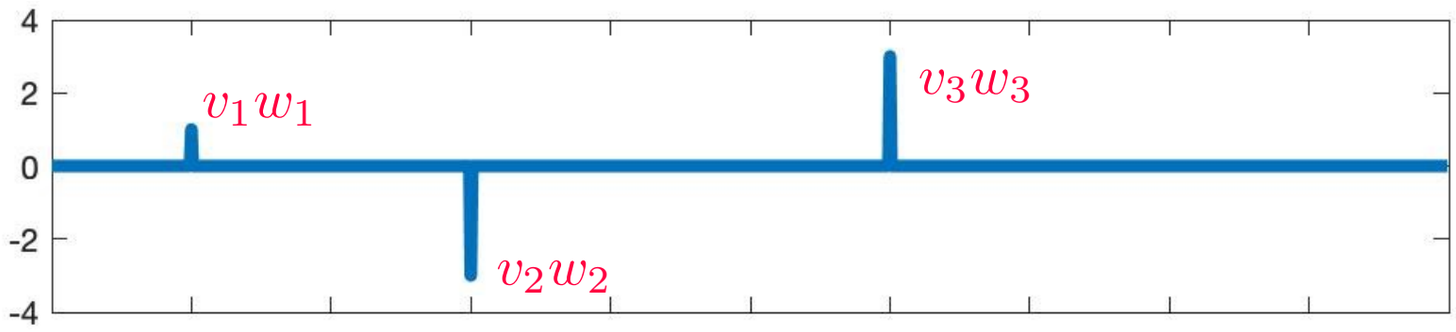
f



weak derivative f'



distributional (second) derivative f''



$$\text{path-norm}(f) \;=\; \sum |v_i w_i| \;=\; \int |f''(x)|\,dx$$

more precisely:
second-order
total variation of f,
denoted by $\text{TV}^2(f)$

Savarese, P., Evron, I., Soudry, D., & Srebro, N. (2019, June). How do infinite width bounded norm networks look in function space?. In *Conference on Learning Theory* (pp. 2667-2690). PMLR.

Weight Decay $\equiv \mathrm{TV}^2(f)$ Regularization

weight decay is gradient of squared ℓ^2 regularization

$$\min_{\boldsymbol{v},\boldsymbol{w},\boldsymbol{b}} \frac{1}{2}\sum_{i=1}^{n}\left(y_i - \sum_j v_j\left(w_j x_i + b_j\right)_+\right)^2 + \frac{\lambda}{2}\sum_j |v_j|^2 + |w_j|^2$$

equivalent to ℓ^1 regularization

$$\min_{\boldsymbol{v},\boldsymbol{w},\boldsymbol{b}} \frac{1}{2}\sum_{i=1}^{n}\left(y_i - \sum_j v_j\left(w_j x_i + b_j\right)_+\right)^2 + \lambda\sum_j |v_j w_j|$$

equivalent to $\mathrm{TV}^2(f)$ regularization

$$\min_{\boldsymbol{v},\boldsymbol{w},\boldsymbol{b}} \frac{1}{2}\sum_{i=1}^{n}\Bigg(y_i - \underbrace{\sum_j v_j\left(w_j x_i + b_j\right)_+}_{f(x_i)}\Bigg)^2 + \lambda\,\mathrm{TV}^2(f)$$

remember: just think of this as $\int |f''(x)|\,dx$

BV^2 is the space of all functions with $\mathrm{TV}^2(f) < \infty$

Neural Network Representer Theorem (Parhi & N, 2020): For any dataset $\{x_i, y_i\}$ and any lower semicontinous loss function ℓ, there exists a solution to

$$\min_{f \in \mathsf{BV}^2} \sum_{i=1}^{n} \ell(y_i, f(x_i)) \; + \; \lambda \, \mathsf{TV}^2(f)$$

that is a sparse single hidden-layer ReLU neural network

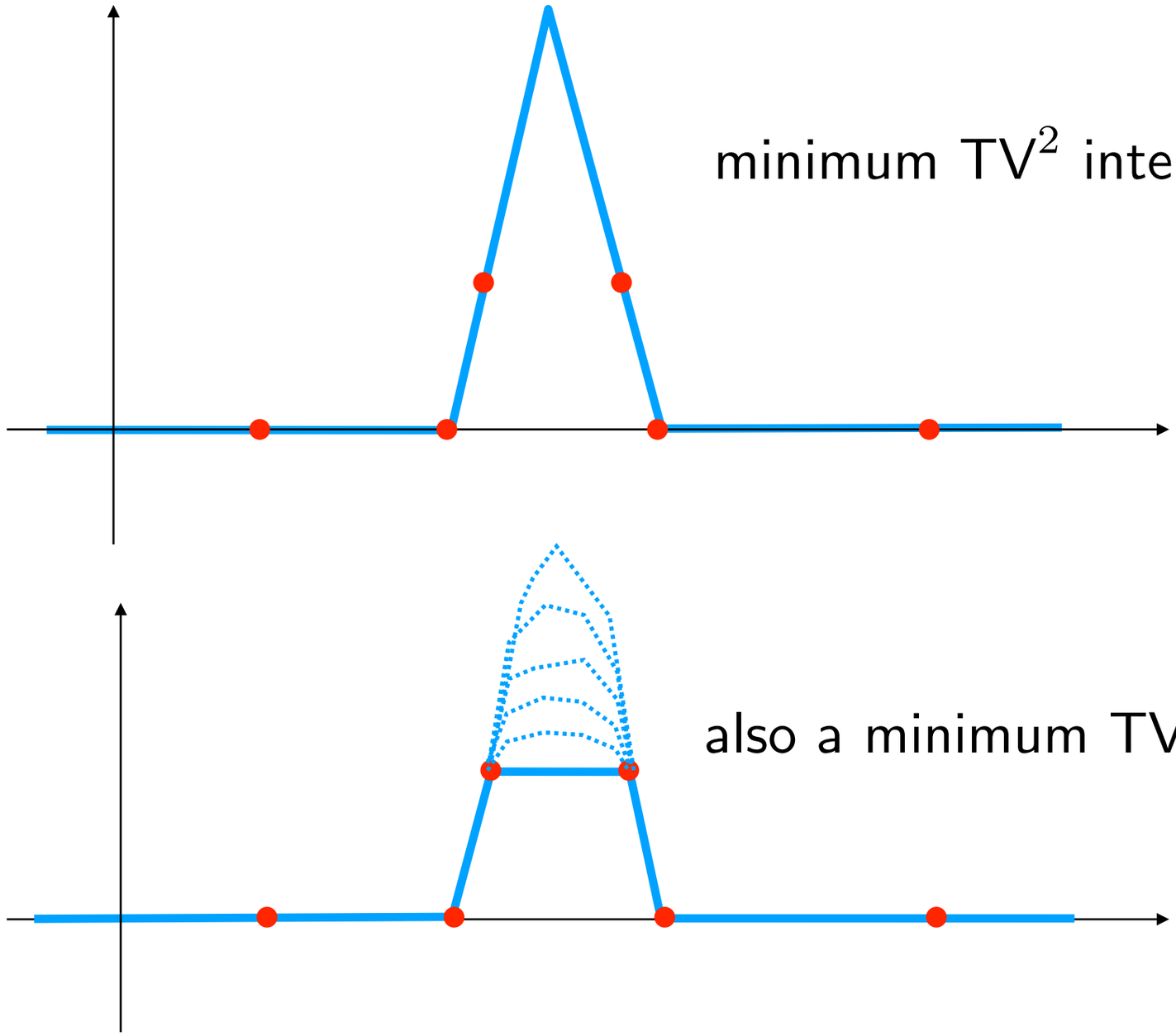
$$f(x) \; = \; \sum_{k=1}^{K} v_k \, (w_k x - b_k)_+ + w_0 x + b_0$$

ReLU neurons; skip connection; sparse solution: $K < n$

- properties of differential operator $D^2 := \frac{d^2}{dx^2}$: *Green's functions* are ReLUs and null space $= \{\text{linear functions}\}$
- each $f \in \mathsf{BV}^2$ can be expressed in terms of finite measure ν i.e., $D^2 f \; = \; \nu \; = \; \nu^+ - \nu^-$
- minimizing total variation $\|\nu\|_{\mathcal{M}} = \int d\nu^+ + \int d\nu^-$ subject to linear constraints is the *measure recovery problem* $\Rightarrow$ solution is sum of Dirac impulses

  Zuhovickii '48
- pseudoinverse (integrating twice) $\left(D^2\right)^+ \delta \; =$ ReLU activation function

Non-Uniqueness of Solution



minimum TV^2 interpolation of data

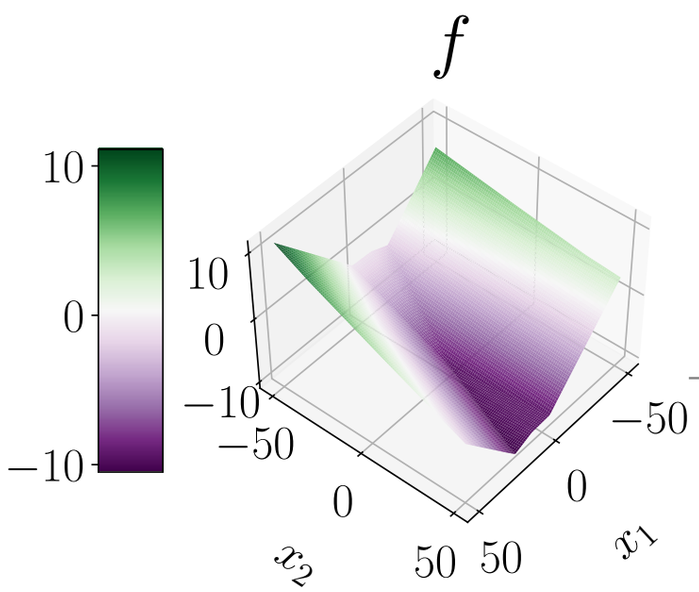
also a minimum TV^2 interpolation

there are infinite number of neural network interpolators that minimize TV^2

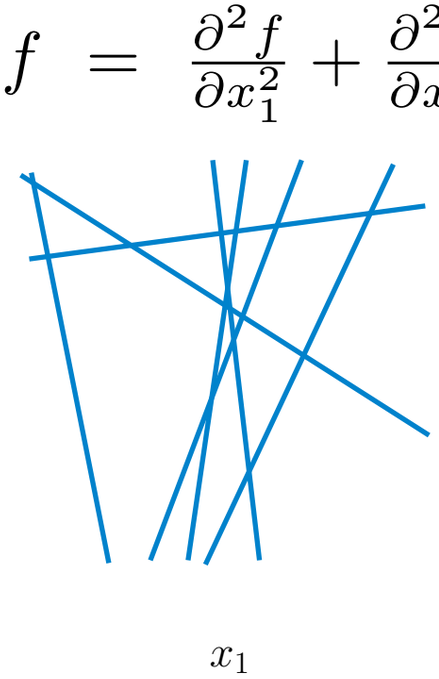
Extension to Multivariate Case: Radon Transform

f → differentiate twice → "lines" along activation thresholds → filtered Radon transform → Dirac δ at each neuron's angle/offset

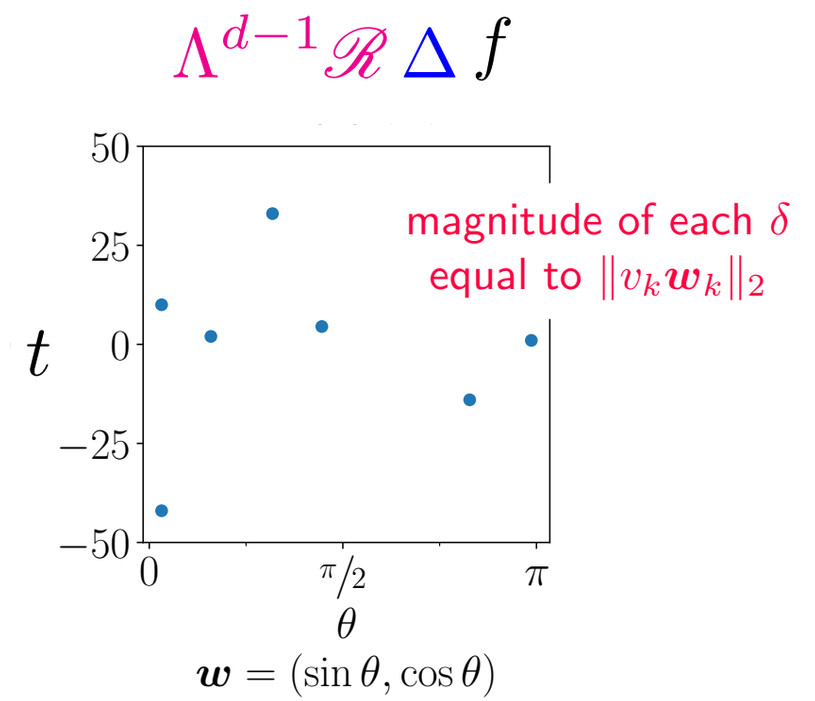
7 ReLU neuron network



Δ



$\Lambda^{d-1}\mathscr{R}$



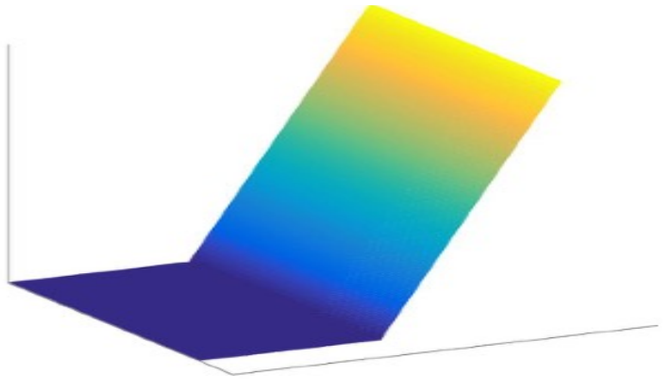
$$\text{path-norm}(f) = \sum |v_k| \, \|\boldsymbol{w}_k\|_2 = \text{total variation of } \Lambda^{d-1}\mathscr{R}\Delta f$$

Radon-domain second-order total variation

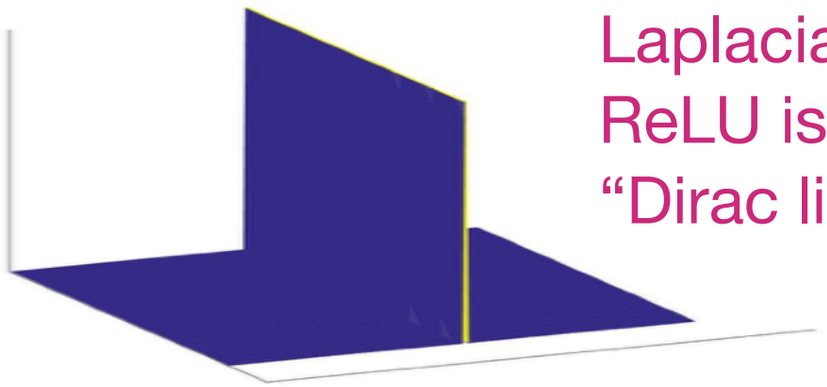
Ongie, G., Willett, R., Soudry, D., & Srebro, N. (2020). A Function Space View of Bounded Norm Infinite Width ReLU Nets: The Multivariate Case. In *International Conference on Learning Representations*.

Multidimensional ReLU Neurons

ReLU function: $\phi(w_1 x_1 + w_2 x_2 - b_0)$



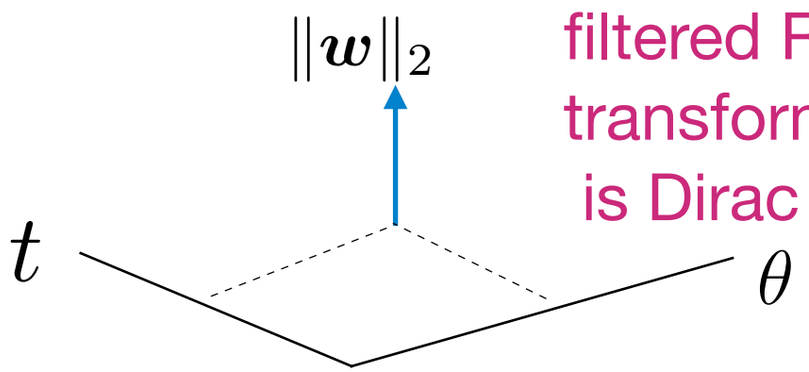
Laplacian of ReLU: $\Delta\phi \;=\; \frac{\partial^2 \phi}{\partial x_1^2} + \frac{\partial^2 \phi}{\partial x_2^2}$



Laplacian of ReLU is a "Dirac line"

filtered Radon transform:

$$\Lambda\mathscr{R}\,\Delta\phi \;=\; \|\boldsymbol{w}\|_2\,\delta(b - b_0)\,\delta\big(\theta - \tan^{-1}\,\tfrac{w_1}{w_2}\big)$$



filtered Radon transform of line is Dirac impulse

$$\|\boldsymbol{w}\|_2\,\delta(t - b_0)\,\delta\big(\theta - \tan^{-1}\,\tfrac{w_1}{w_2}\big)$$

indicates slope, orientation and offset of ridge

The Banach Space $\mathscr{R}\mathsf{BV}^2$

Radon domain TV^2: $\mathscr{R}\mathsf{TV}^2(f) := $ total variation of measure $\Lambda^{d-1}\mathscr{R}\Delta f$

$\Lambda^{d-1}\mathscr{R} =$ filtered Radon transform

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

measures "sparsity" (L^1-like norm) of second derivatives along each direction in $\mathbb{R}^d$

Second-Order Bounded Variation Space in Radon Domain:

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

Banach space representer theorems for neural networks and ridge splines

R **Parhi**, RD **Nowak** - The Journal of Machine Learning Research, 2021 - dl.acm.org

Neural Network Representer Theorem (Parhi & N, 2020): For any dataset $\{\boldsymbol{x}_i, y_i\}$ and any lower semicontinous loss function ℓ, there exists a solution to

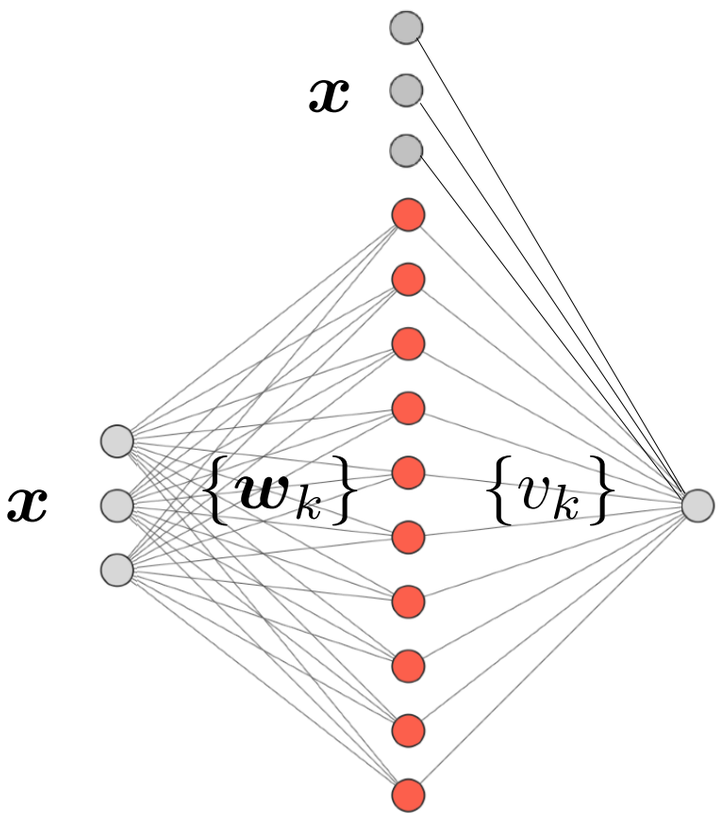
$$\min_{f \in \mathscr{R}\mathsf{BV}^2} \sum_{i=1}^{n} \ell(y_i, f(\boldsymbol{x}_i)) \; + \; \lambda \mathscr{R}\mathsf{TV}^2(f)$$

with a representation in the form of a single hidden-layer neural network

sparse solution

$$f(\boldsymbol{x}) \; = \; \sum_{k=1}^{K} v_k \, (\boldsymbol{w}_k^T \boldsymbol{x} - b_k)_+ \; + \; \boldsymbol{w}_0^T \boldsymbol{x} + b_0 \qquad K < n$$

ReLU neurons skip connection



implication: nonparametric problem reduces to training finite-dimensional neural network with weight decay

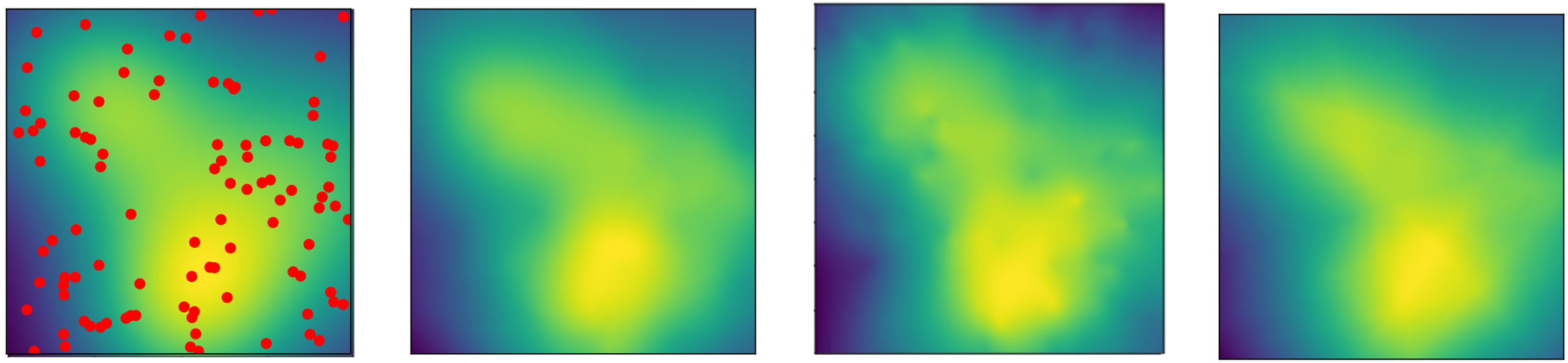
Neural Networks Adapt to Directional Smoothness

sample locations (red)

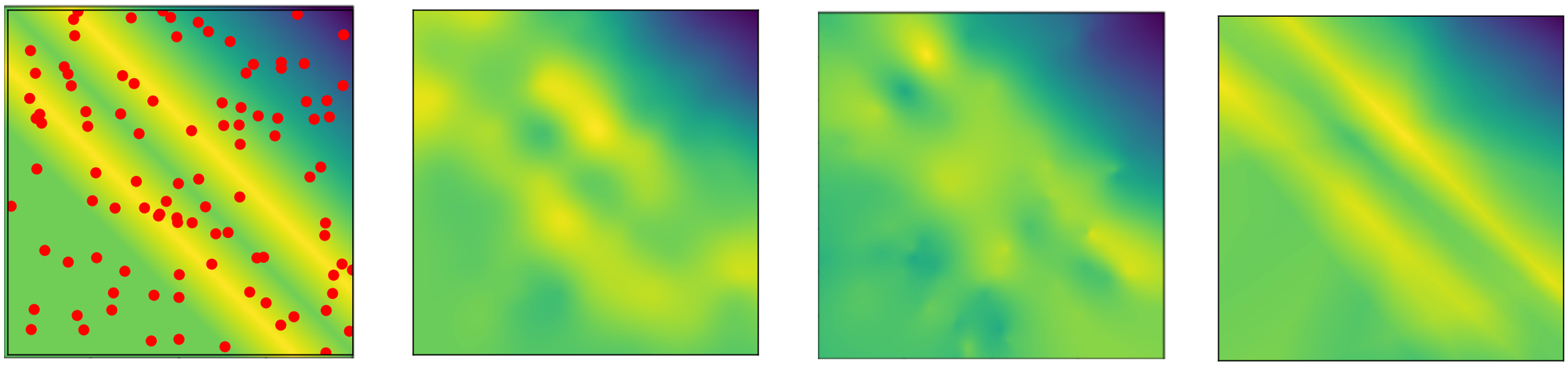
thin-plate spline

Neural Tangent Kernel

ReLU net



reconstructions from samples



linear methods cannot adapt to directional smoothness in function/data

Minimax Optimality of ReLU Networks

Let $\Omega \subset \mathbb{R}^d$ be a bounded domain. A function $f \in \mathscr{R}\mathrm{BV}^2(\Omega)$ can be approximated by a finite-width m ReLU neuron network f_m satisfying

$$\|f - f_m\|_{L^2}^2 \;=\; O\big(m^{-(1+3/d)}\big) \;=\; O\big(m^{-1}\big)$$

Estimating f in this class from n nicely distributed samples on $\Omega \subset \mathbb{R}^d$

$$\mathbb{E}\|f - f_m\|_{L^2(\Omega)}^2 \;=\; \widetilde{O}\left(m^{-(1+3/d)} + \frac{m}{n}\right) \;=\; \widetilde{O}(n^{-\frac{d+3}{2d+3}}) \;=\; O(n^{-1/2})$$

approx. error $+$ variance

no curse

this is the minimax rate for $\mathscr{R}\mathrm{BV}^2(\Omega)$

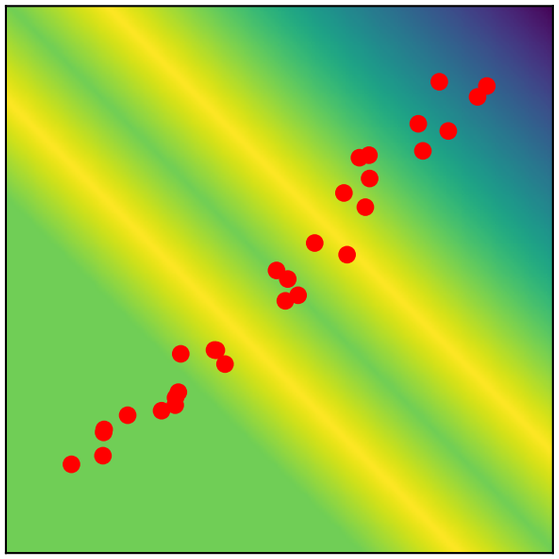
In contrast, for any linear method (thin-plate spline, RKHS, neural tangent kernel) the *linear* minimax rate is $n^{-\frac{3}{d+3}}$

the curse

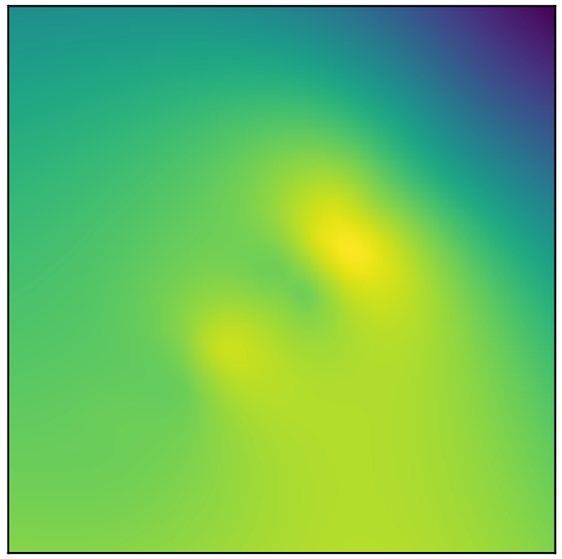
Parhi & N, '21

Data Fitting and Extrapolation

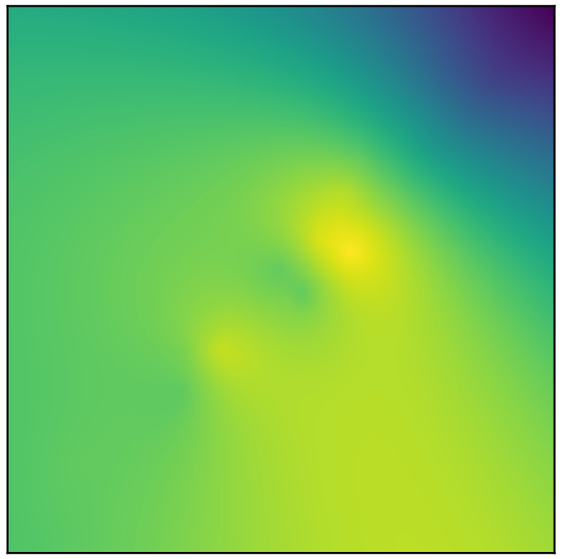
samples (red)
of 2D function



thin-plate spline



ReLU NTK

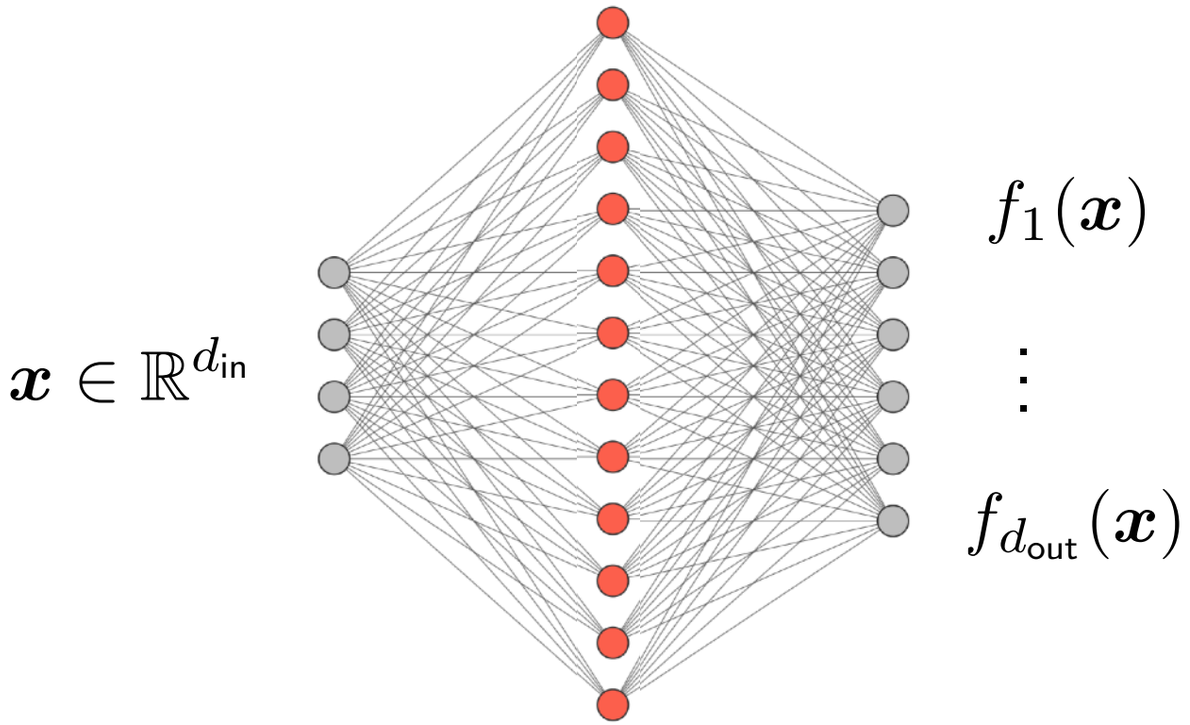


ReLU net +
weight decay

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

What Kinds of Functions Do Deep Neural Networks Learn?

Vector-Valued Networks



equivalent optimizations:

$$\min_{\boldsymbol{v},\boldsymbol{w}} \frac{1}{2}\sum_{i=1}^{n}\left\|\boldsymbol{y}_i - \sum_j \boldsymbol{v}_j\left(\boldsymbol{w}_j^T\boldsymbol{x}_i\right)_+\right\|_2^2 + \frac{\lambda}{2}\sum_j \|\boldsymbol{v}_j\|_2^2 + \|\boldsymbol{w}_j\|_2^2$$

$$\min_{\boldsymbol{v},\boldsymbol{w}:\|\boldsymbol{w}_j\|_2=1} \frac{1}{2}\sum_{i=1}^{n}\left\|\boldsymbol{y}_i - \sum_j \boldsymbol{v}_j\left(\boldsymbol{w}_j^T\boldsymbol{x}_i\right)_+\right\|_2^2 + \lambda\sum_j \|\boldsymbol{v}_j\|_2$$

natural norm for vector-valued neural network functions

Vector-Valued Function Spaces

$$\min_{\boldsymbol{v},\boldsymbol{w}:\|\boldsymbol{w}_j\|_2=1} \frac{1}{2}\sum_{i=1}^{n}\left\|\boldsymbol{y}_i - \sum_j \boldsymbol{v}_j\,(\boldsymbol{w}_j^T\boldsymbol{x}_i)_+\right\|_2^2 \;+\; \lambda\sum_j \|\boldsymbol{v}_j\|_2$$

continuum width networks:

$$f(\boldsymbol{x}) \;=\; \begin{bmatrix} f_1(\boldsymbol{x}) \\ f_2(\boldsymbol{x}) \\ \\ \vdots \\ \\ f_d(\boldsymbol{x}) \end{bmatrix} \;=\; \int_{\boldsymbol{w}\in\mathbb{S}^{d-1}} (\boldsymbol{w}^T\boldsymbol{x})_+ \; d\boldsymbol{\nu}(\boldsymbol{w})$$

finite vector-valued measure

$$\begin{aligned} \|f\| \;&:=\; \|\boldsymbol{\nu}\| \;=\; \text{total variation of } \ell^2\text{-norm of } \boldsymbol{\nu} \\ &=\; \sum_j \|\boldsymbol{v}_j\|_2 \text{ for finite width networks} \end{aligned}$$

weight decay encourages a small norm

Vector-Valued Variation Spaces and Bounds on Neural Network Widths
J. Shenouda, R. Parhi, K. Lee, RN (2023)

Deep Representer Theorem

Let $\mathscr{R}\mathsf{BV}^2_{\mathsf{deep}}$ denote space of functions generated by taking compositions of functions from the vector-valued variation space

Multilayer Representer Theorem (Parhi & N 2021, Shenouda et al 2023): For any dataset $\{\boldsymbol{x}_i, y_i\}$ and any lower semicontinous loss function ℓ, there exists a solution to

$$\min_{f \in \mathscr{R}\mathsf{BV}^2_{\mathsf{deep}}} \sum_{i=1}^{n} \ell(y_i, f(\boldsymbol{x}_i)) \; + \; \lambda \mathscr{R}\mathsf{TV}^2_{\mathsf{deep}}(f)$$

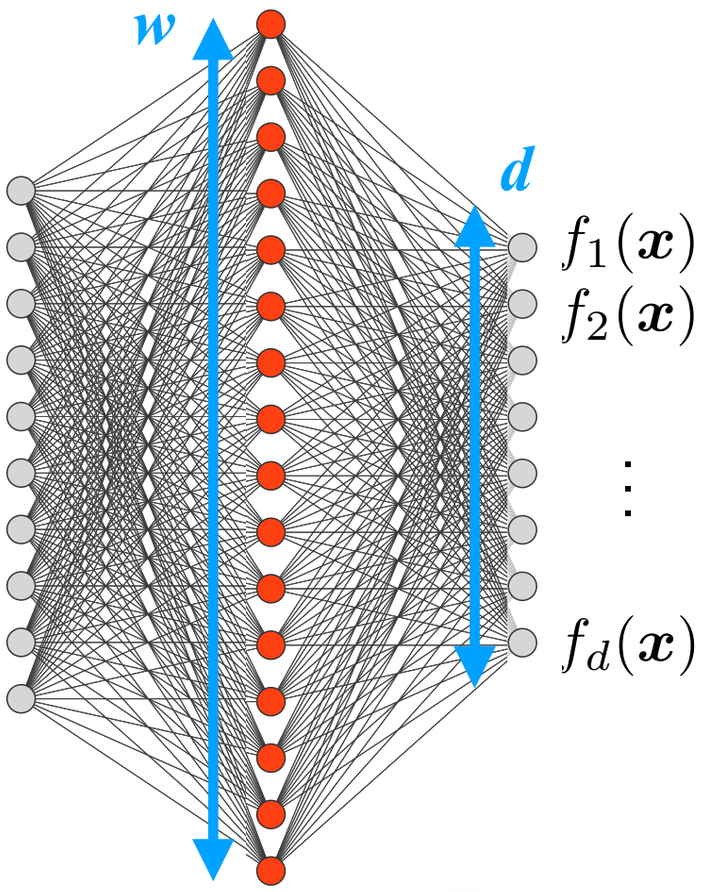
with a representation in the form of a sparse multilayer ReLU neural network with $< nd_l$ neurons in l-th layer

implication: nonparametric problem reduces to training multi-layer neural network with weight decay

The Effect of Weight Decay

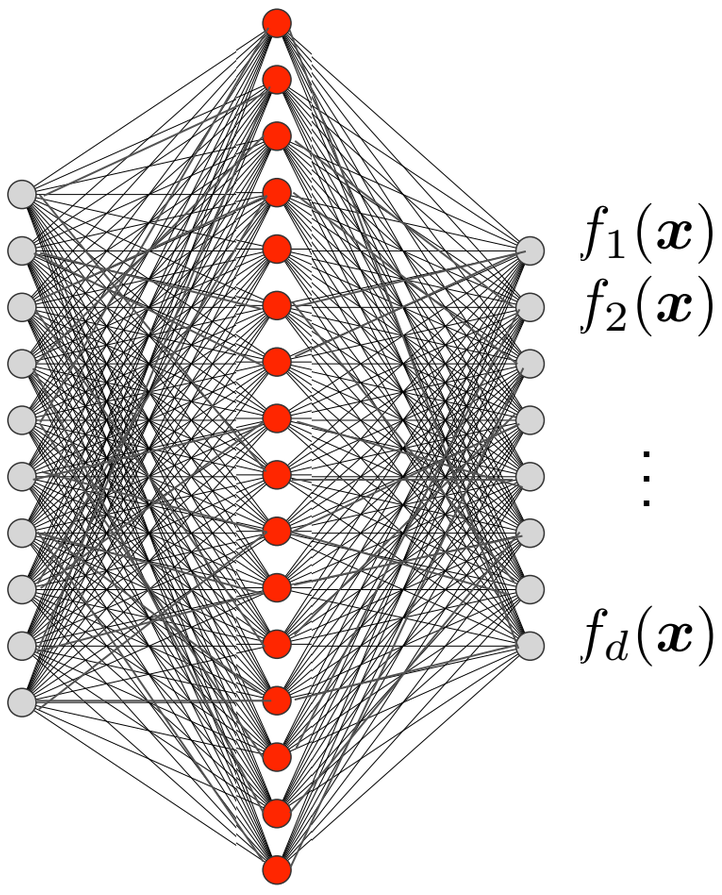
$$\|f\| \quad := \quad \sum \|\boldsymbol{v}_j\|_2 \text{ for finite width networks}$$

dense weights



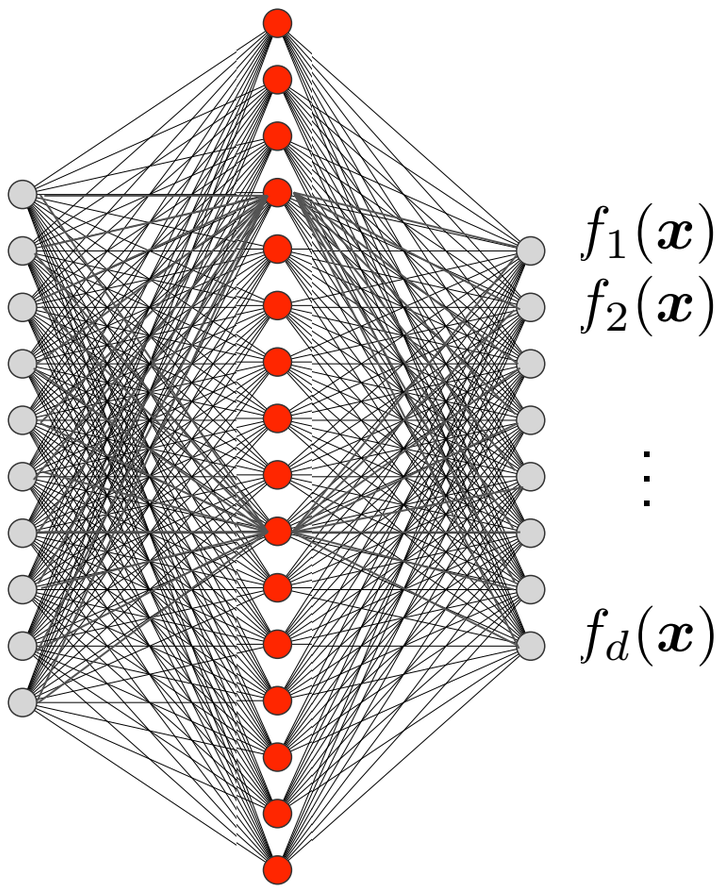
$\|f\| = O(w\sqrt{d})$

sparse weights



$\|f\| = O(d)$

sparse neurons



$\|f\| = O(\sqrt{d})$

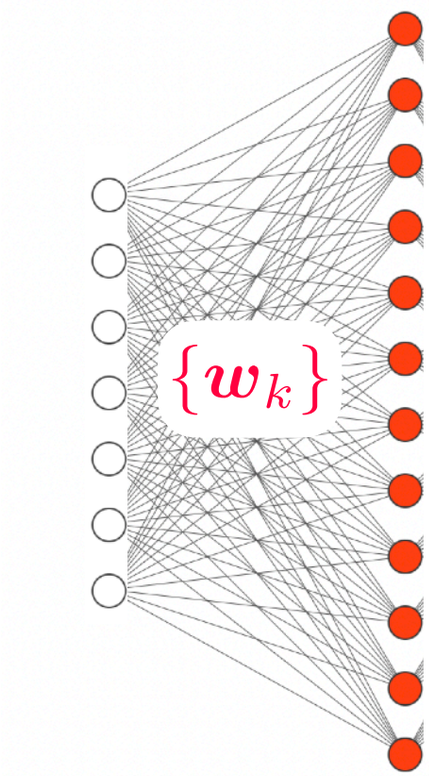
weight decay encourages functions with strong variations in only a few directions (sparse weights)

weight decay encourages output functions to "share" neurons

Another Consequence: Tight Bounds on Widths

consider one ReLU layer within a deep network

n training patterns



neuron outputs

$$\boldsymbol{\Phi} = \{\boldsymbol{\phi}_i\}_{i=1}^n$$

layer outputs

$$\boldsymbol{\Psi} = \{\boldsymbol{\psi}_i\}_{i=1}^n \subset \mathbb{R}^d$$

Layer Width Theorem : Let $\boldsymbol{\Phi}$ and $\boldsymbol{\Psi}$ be matrices composed of the post-activation features and the outputs of a ReLU layer in a deep net, trained to minimize the weight decay objective. Then there exists a representation of this layer with K neurons where K satisfies

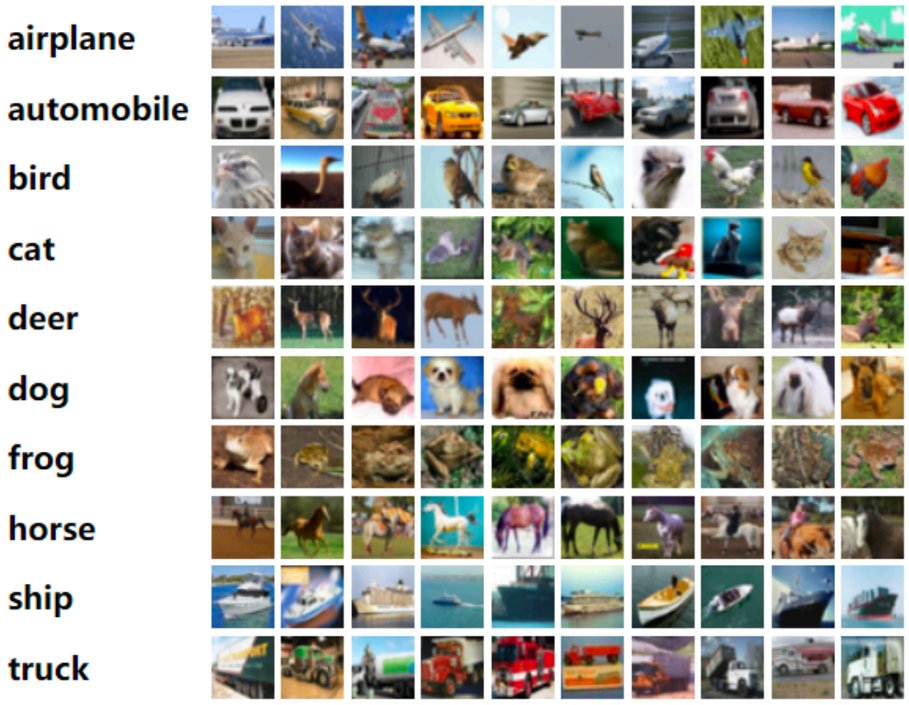
$$K \leq \text{rank}(\boldsymbol{\Phi}) \times \text{rank}(\boldsymbol{\Psi}) \leq n\, d$$

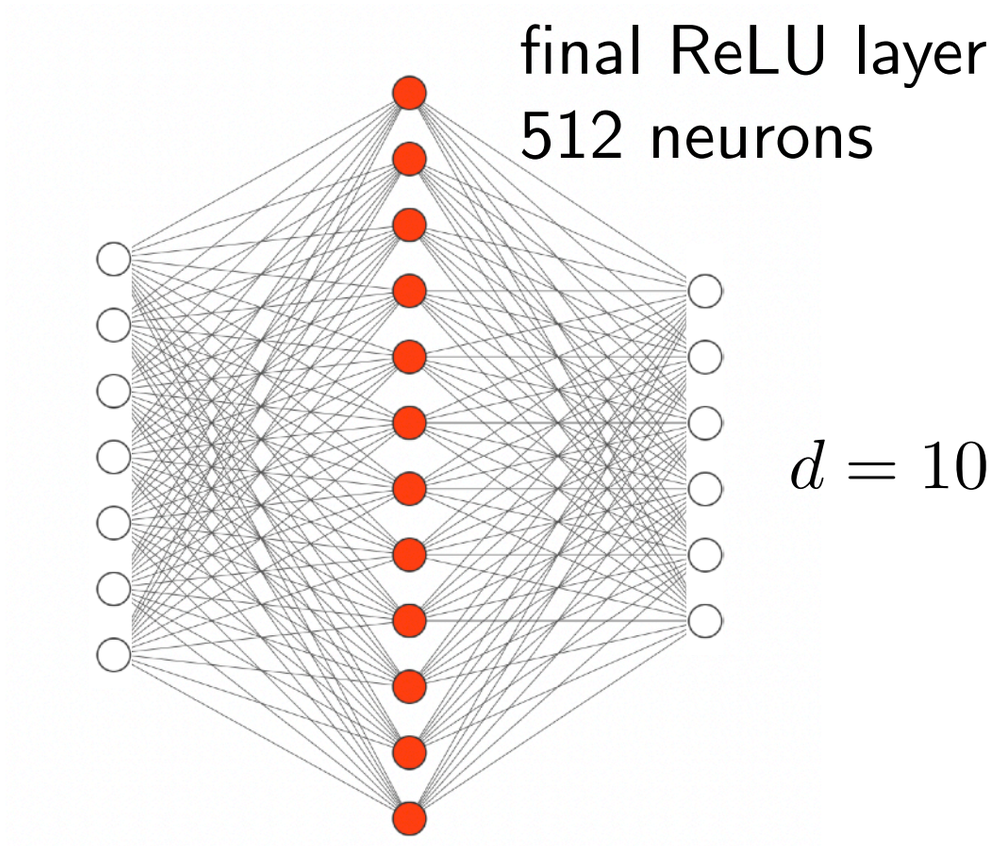
can be a significant improvement on best prior bound of n^2 (Jacot et al '22)

relevant because in practice the ranks often become small at deeper layers

Sparsifying Trained Networks

VGG-19 trained to minimize cross-entropy loss + weight decay on CIFAR10





Theory: $K \leq \text{rank}(\mathbf{\Phi}) \times \text{rank}(\mathbf{\Psi}) \approx 10 \times 10$ neurons are sufficient

	$\mathbf{V}$	$\widehat{\mathbf{V}}$
Active Neurons	512	47
Test Accuracy	93.92%	93.88%
Train Loss	0.0104	0.0112

multi-task lasso finds an equivalent solution (wrt training data) with just 47 neurons!

Summary

ReLU neural networks are optimal solutions to data-fitting problems in new function spaces:

- Radon-domain bounded variation space
- Banach space, not Hilbert space
- immune to curse-of-dimensionality
- solutions are sparse / narrow
- adaptive to spatial and directional varying smoothness

Weight Decay is equivalent to "multi-task lasso" regularization and thus promotes solutions that have strong variation in limited directions and encourages outputs to share neurons

Thanks!

This work was supported by the following agencies