Deep Learning Meets Sparse Regularization

Rahul Parhi
Institute of Electrical and Micro Engineering
École polytechnique fédérale de Lausanne

UCSD ECE
29 February 2024

A Brief History of Neural Networks and AI

1943: McCulloch and Pitts had the vision to introduce artificial intelligence to the world.

BULLETIN OF
MATHEMATICAL BIOPHYSICS
VOLUME 5, 1943

A LOGICAL CALCULUS OF THE
IDEAS IMMANENT IN NERVOUS ACTIVITY

WARREN S. MCCULLOCH AND WALTER PITTS

FROM THE UNIVERSITY OF ILLINOIS, COLLEGE OF MEDICINE,
DEPARTMENT OF PSYCHIATRY AT THE ILLINOIS NEUROPSYCHIATRIC INSTITUTE,
AND THE UNIVERSITY OF CHICAGO

1958: Rosenblatt implemented the first perceptron for learning.

Psychological Review
Vol. 65, No. 6, 1958

THE PERCEPTRON: A PROBABILISTIC MODEL FOR
INFORMATION STORAGE AND ORGANIZATION
IN THE BRAIN [1]

F. ROSENBLATT

Cornell Aeronautical Laboratory

1986: Rumelhart, Hinton, and Williams studied backpropagation for training multilayer perceptrons.

**Learning representations
by back-propagating errors**

**David E. Rumelhart*, Geoffrey E. Hinton†
& Ronald J. Williams***

* Institute for Cognitive Science, C-015, University of California,
San Diego, La Jolla, California 92093, USA
† Department of Computer Science, Carnegie-Mellon University,
Pittsburgh, Philadelphia 15213, USA

What Is the Inductive Bias of 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.

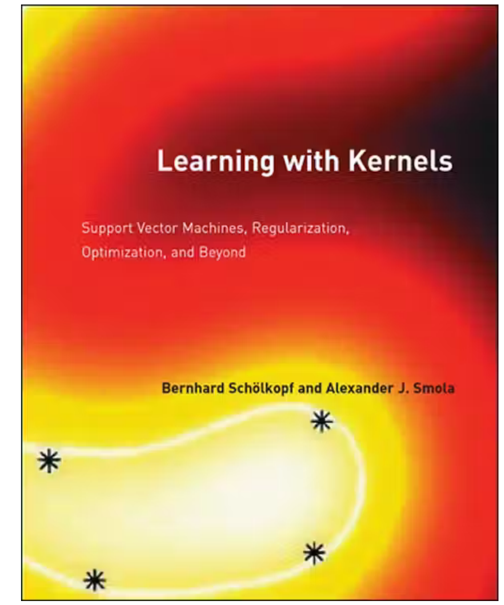
- Such functions can be approximated by a neural network with K neurons at a rate $K^{-\frac{1}{2}}$.
- Rates for classical function classes behave as $K^{-\frac{s}{d}}$ ← the curse

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

People Moved On From Neural Networks...

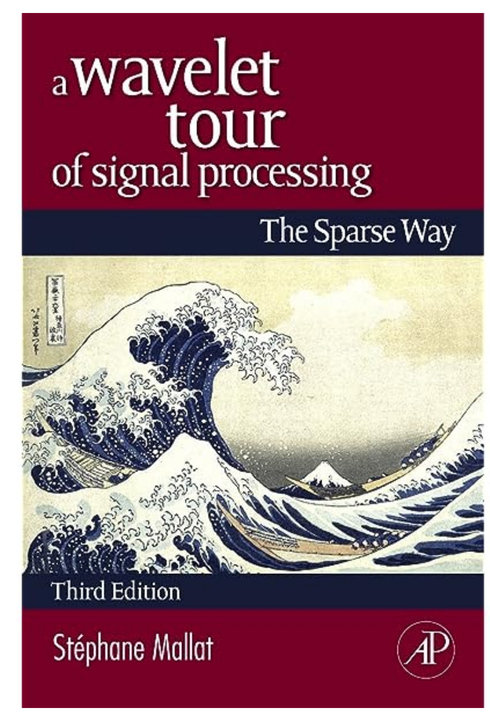
Support-vector networks
C Cortes, V Vapnik - Machine learning, 1995 - Springer
The support-vector network is a new learning machine for two-group classification problems. The machine conceptually implements the following idea: input vectors are non-linearly …
☆ Save 99 Cite Cited by 62558 Related articles

- Reproducing kernel Hilbert Spaces
- Representer theorem



Ideal spatial adaptation by wavelet shrinkage
DL Donoho, IM Johnstone - biometrika, 1994 - academic.oup.com
With ideal spatial adaptation, an oracle furnishes information about how best to adapt a spatially variable estimator, whether piecewise constant, piecewise polynomial, variable …
☆ Save 99 Cite Cited by 13135 Related articles

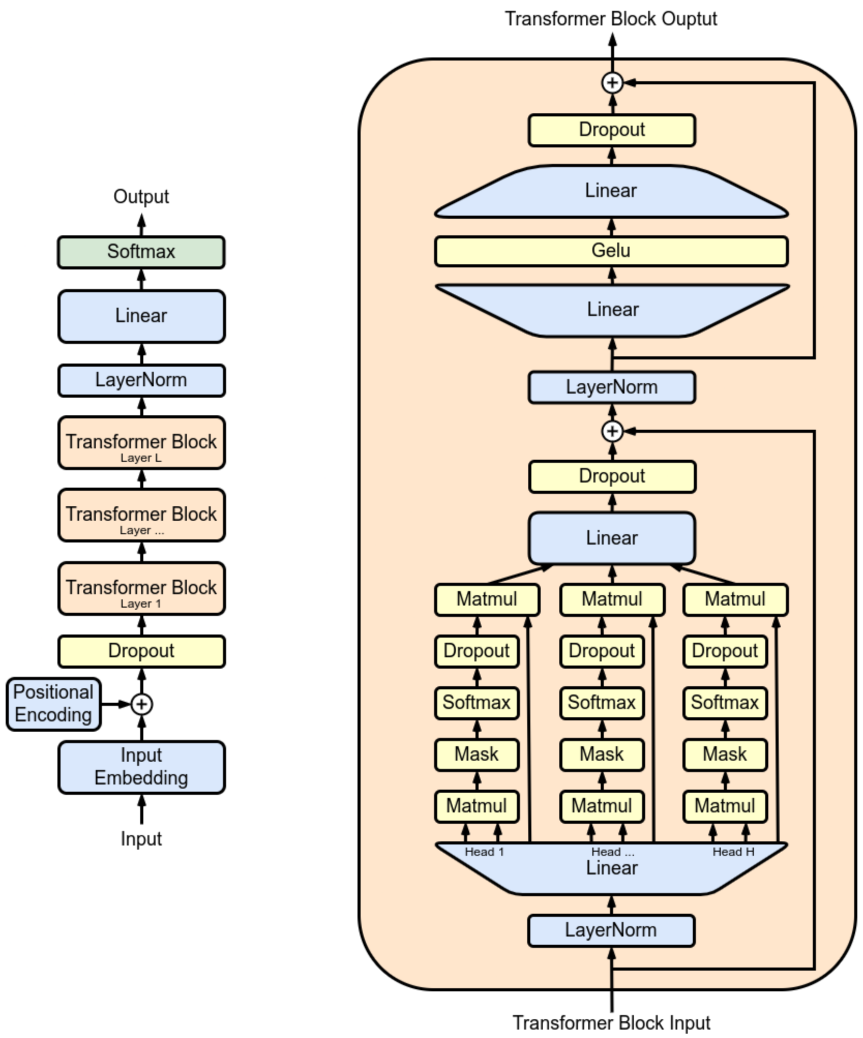
Nonlinear total variation based noise removal algorithms
LI Rudin, S Osher, E Fatemi - Physica D: nonlinear phenomena, 1992 - Elsevier
A constrained optimization type of numerical algorithm for removing noise from images is presented. The total variation of the image is minimized subject to constraints involving the …
☆ Save 99 Cite Cited by 18507 Related articles



- The (r)evolution of **sparsity**

  $\Longrightarrow$ Compressed sensing

And Here We Are Today



Large language models (LLMs) like generative pre-trained transformers (GPT) have taken the world by storm.

- DALL·E
- ChatGPT

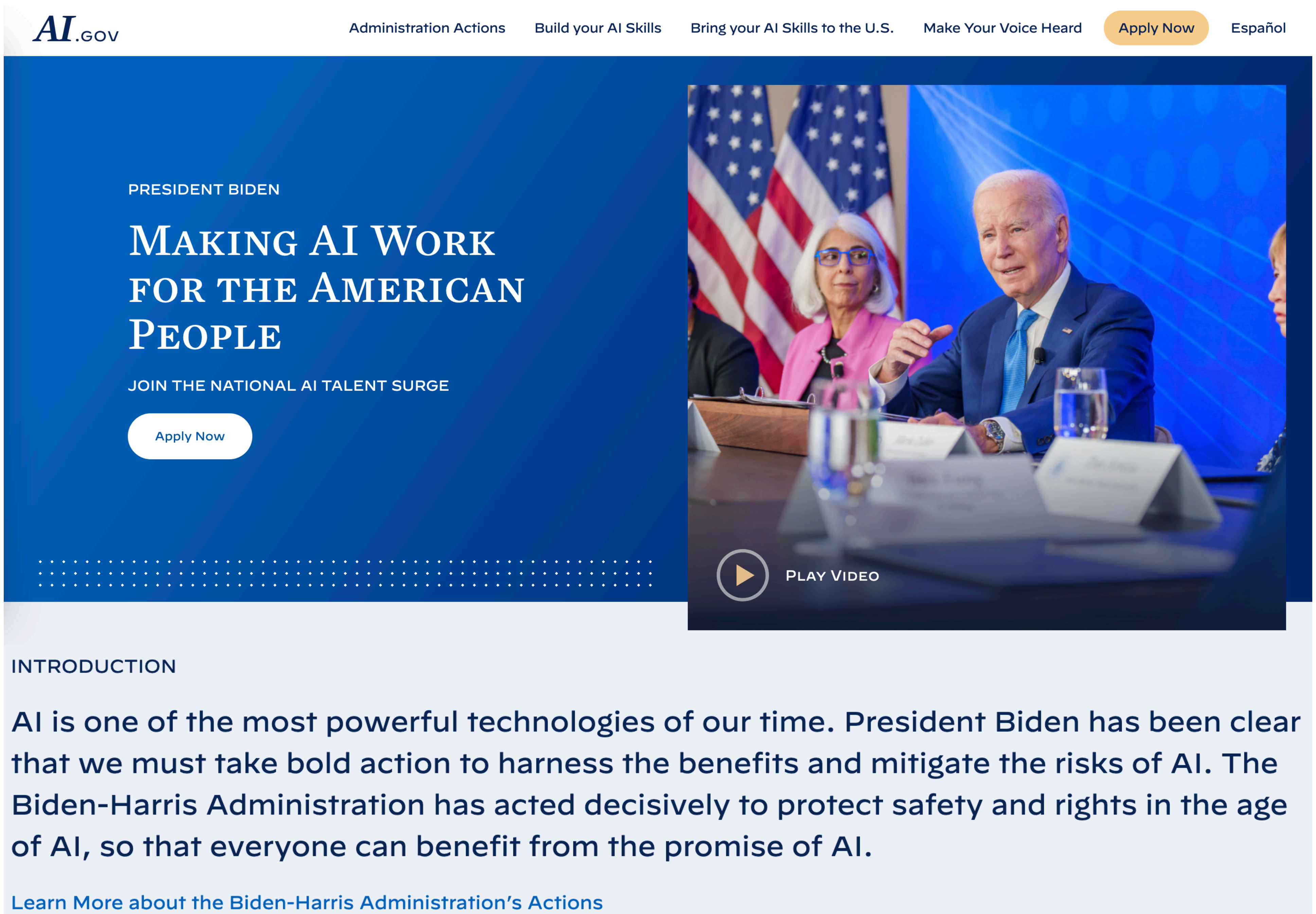
We have come full circle back to neural networks!

[PDF] Improving language understanding by generative pre-training

A Radford, K Narasimhan, T Salimans, I Sutskever

Natural language understanding comprises a wide range of diverse tasks such as textual entailment, question answering, semantic similarity assessment, and document …

And Here We Are Today



Develop standards, tools, and tests to ensure that AI systems are trustworthy and reliable.

Two Extremes of AI Research

First Extreme

Do we understand how it works?

Is it reliable and trustworthy?

Theoretical foundations

Rationalism

Plato

Second Extreme

Let's put it everywhere!

More interest in if it could work as opposed to if it could fail.

Trial and error

Empiricism

Aristotle

Scientific innovation needs both extremes.

Magnetic Resonance Imaging (MRI)

Accelerating MRI scans is one of the principal outstanding problems in the MRI research community.

- Early approaches were based on **compressed sensing**.

Magnetic Resonance in Medicine 58:1182–1195 (2007)

Sparse MRI: The Application of Compressed Sensing for Rapid MR Imaging

Michael Lustig,[1*] David Donoho,[2] and John M. Pauly[1]

⟹ Theoretical guarantees of **stability**.

Candès et al. (2006)
Donoho (2006)

- Modern approaches are based on **deep learning** and massive amounts of **data**.

2306 IEEE TRANSACTIONS ON MEDICAL IMAGING, VOL. 40, NO. 9, SEPTEMBER 2021

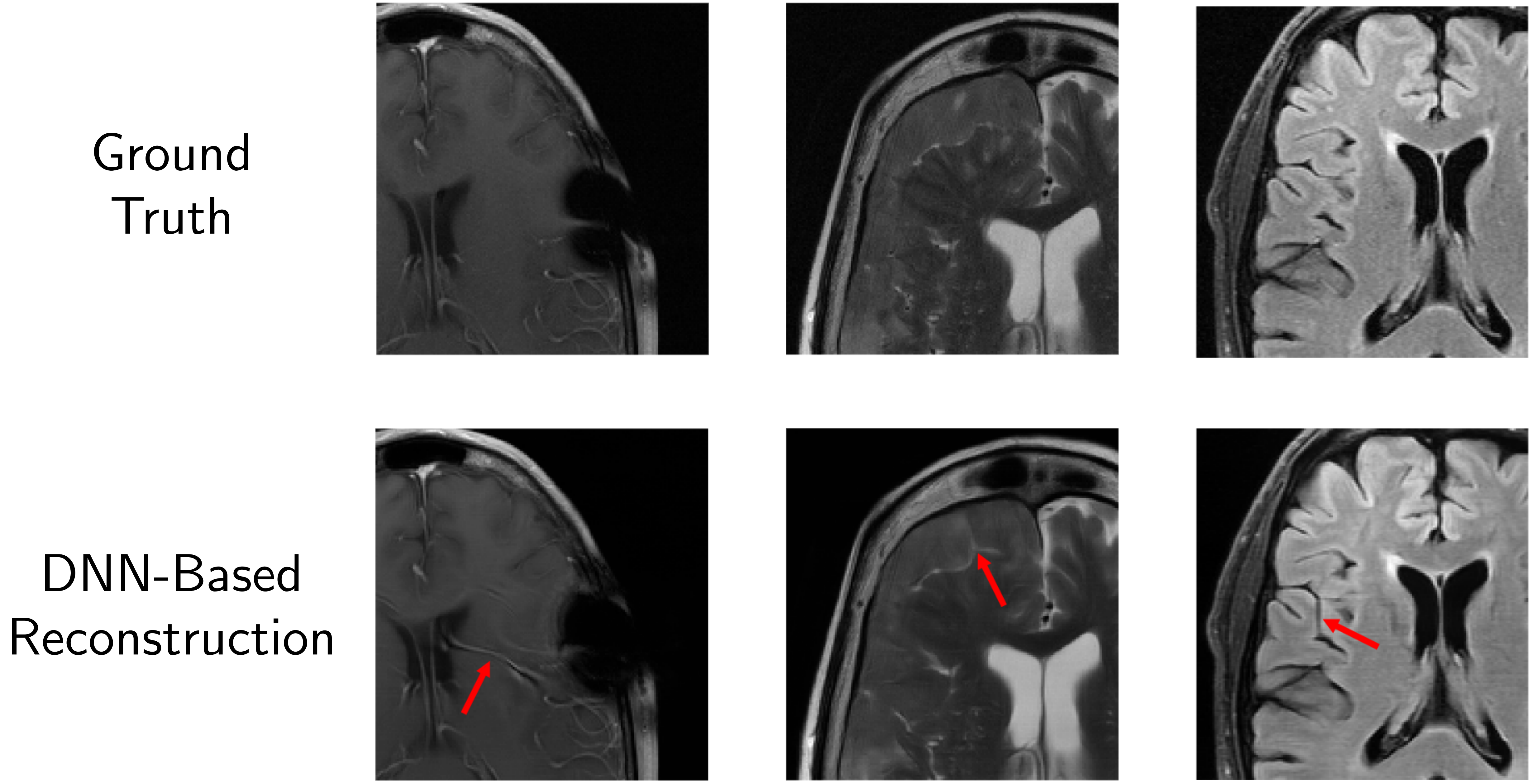
Results of the 2020 fastMRI Challenge for Machine Learning MR Image Reconstruction

Matthew J. Muckley, *Member, IEEE*, Bruno Riemenschneider, Alireza Radmanesh, Sunwoo Kim, *Member, IEEE*, Geunu Jeong, Jingyu Ko, Yohan Jun, Hyungseob Shin, Dosik Hwang, Mahmoud Mostapha, Simon Arberet, Dominik Nickel, Zaccharie Ramzi, *Student Member, IEEE*, Philippe Ciuciu, *Senior Member, IEEE*, Jean-Luc Starck, Jonas Teuwen, Dimitrios Karkalousos, Chaoping Zhang, Anuroop Sriram, Zhengnan Huang, Nafissa Yakubova, Yvonne W. Lui, and Florian Knoll, *Member, IEEE*

⟹ Almost no theoretical guarantees.

Can we trust deep-learning-based methods?

Results of the 2020 fastMRI Challenge



AI-generated hallucinations identified by radiologists as **false** vessels.

Muckley et al. (2021, IEEE Transactions on Medical Imaging)

Interpretability Crisis of AI and Deep Learning

We essentially understand the entire story
for kernel methods and wavelet/TV methods.

$\Longrightarrow$ These methods are (mathematically) interpretable.

Can we develop a similar story for neural networks and deep learning?

Rationalism

Plato

My Research

P. and Nowak (2020, IEEE Signal Process. Lett.)
P. and Nowak (2021, J. Mach. Learn. Res.)
P. and Nowak (2022, SIAM J. Math. Data Sci.)
P. and Nowak (2022, IEEE ICASSP)
P. and Nowak (2023, IEEE Trans. Inf. Theory)
P. and Nowak (2023, IEEE Signal Process. Mag.)
Shenouda, **P.**, and Nowak (2023, SAMPTA)
Shenouda, **P.**, Lee, and Nowak (2023, arXiv)
P. and Unser (2023, IEEE Signal Process. Lett.)
P. and Unser (2023, SAMPTA)
P. and Unser (2023, arXiv)
P. and Unser (2023, arXiv)
DeVore, Nowak, **P.**, and Siegel (2023, arXiv)

Lessons From Kernel Methods

A **representer theorem** designates a *finite-dimensional* parametric formula to solutions of an optimization problem posed over an *infinite-dimensional* function space.

Representer Theorem (*circa* 1970)

Let $\mathcal{H}$ be an RKHS with kernel $k(\cdot, \cdot)$. Then, for any data set $\{(\boldsymbol{x}_n, y_n)\}_{n=1}^N$, the solution to

$$\min_{f \in \mathcal{H}} \sum_{n=1}^{N} \mathcal{L}(y_n, f(\boldsymbol{x}_n)) + \lambda \|f\|_{\mathcal{H}}^2, \quad \lambda > 0,$$

admits a representation of the form

$$f_{\mathrm{RKHS}}(\boldsymbol{x}) = \sum_{n=1}^{N} a_n k(\boldsymbol{x}, \boldsymbol{x}_n).$$

Carl de Boor

Grace Wahba

Cubic Smoothing Splines

The solution to

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

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

is a cubic (smoothing) spline,

$$f_{\text{spline}}(x) = \sum_{n=1}^{N} a_n^\star \, k(x, x_n),$$

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 $\boldsymbol{y}$

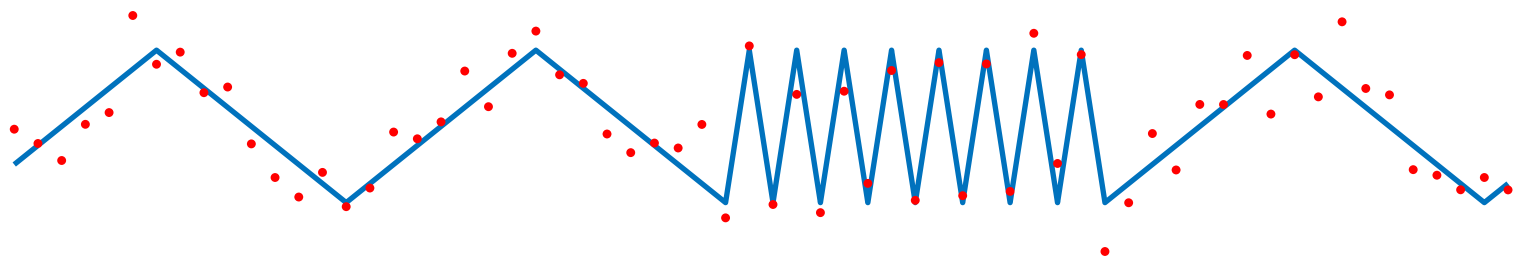
If $y_n = f^\star(x_n) + \varepsilon_n$ with $\|\mathrm{D}^2 f^\star\|_{L^2} < \infty$, then

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

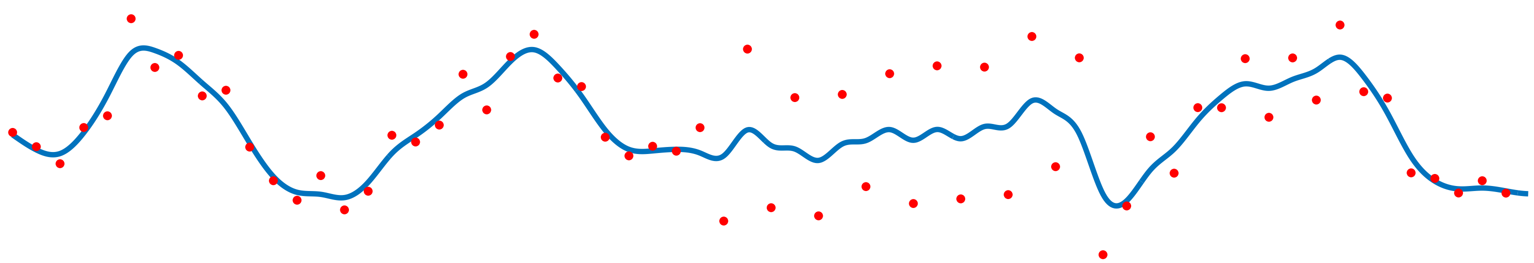
minimax rate

de Boor and Lynch (1966, Journal of Mathematics and Mechanics)
Kimeldorf and Wahba (1971, Journal of Mathematical Analysis and Applications)

Limitations of Linear/Kernel Methods

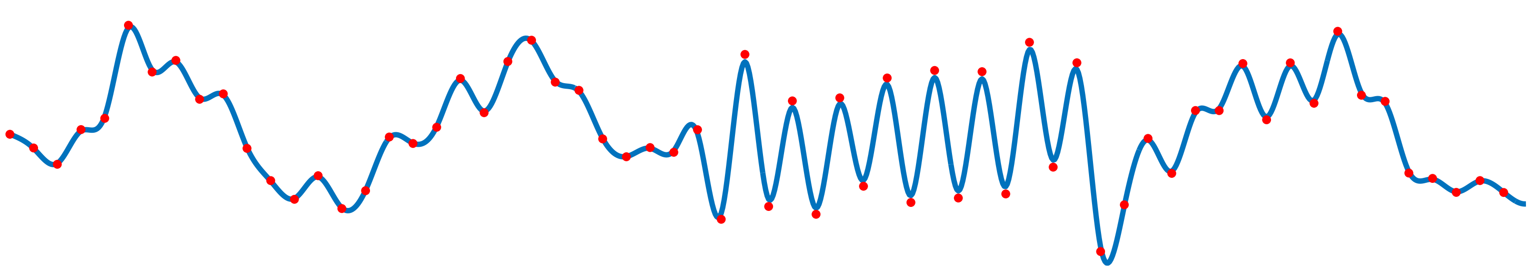


True function and noisy data



large λ:
oversmooths high variation portion of the data

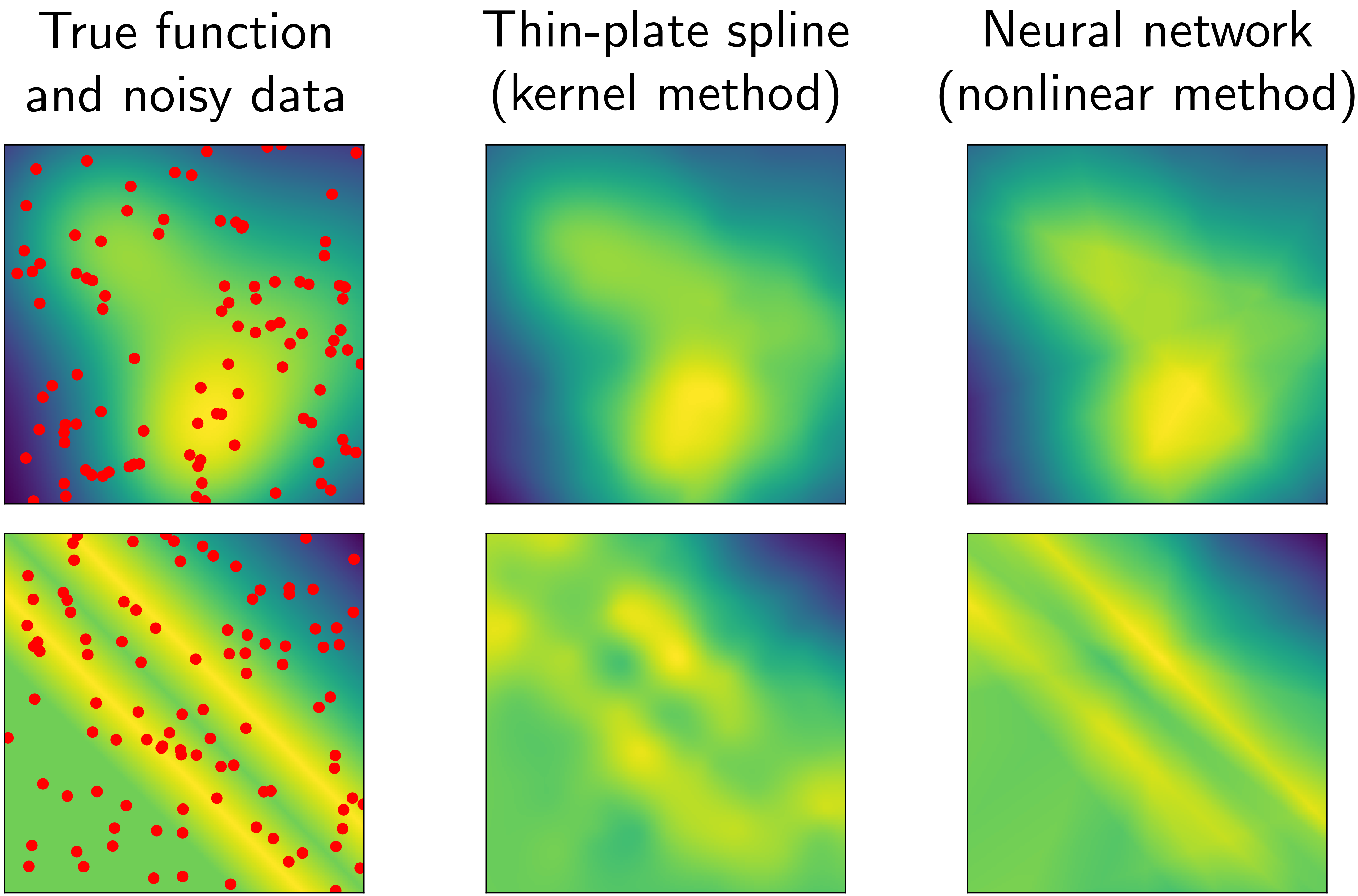
small λ:
overfits low variation portion of the data



Linear methods cannot adapt to spatially varying smoothness.

Donoho, Liu, and MacGibbon (1990, Annals of Statistics)

Limitations of Linear/Kernel Methods



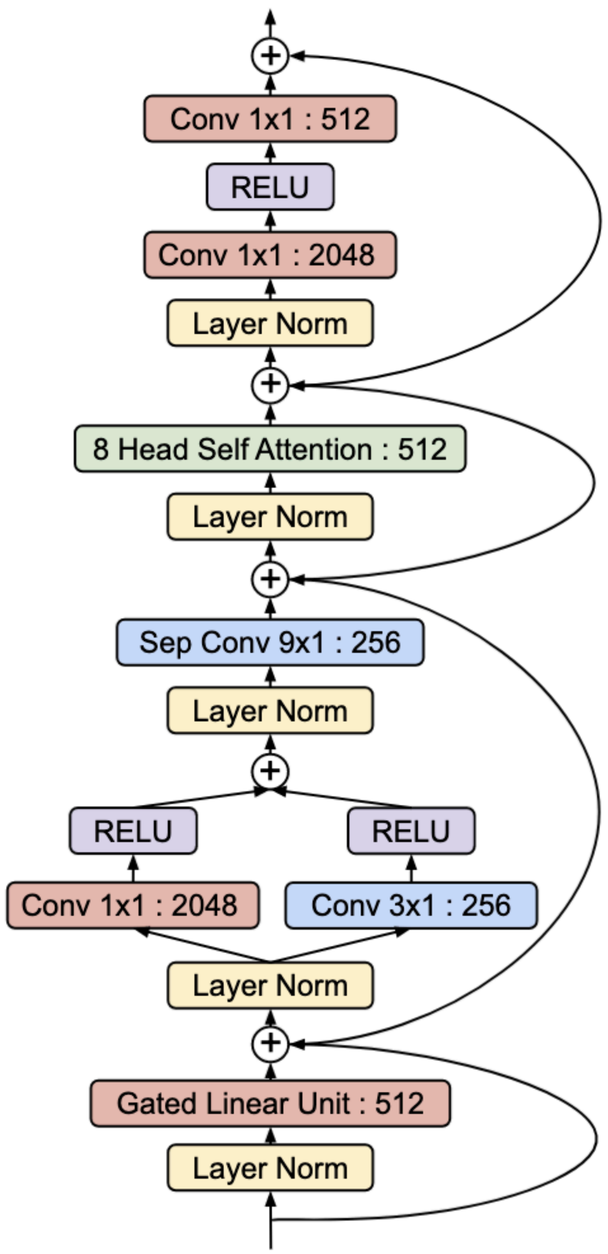
Neural networks can adapt to **low-dimensional structure**.

P. and Nowak (2023, IEEE Transactions on Information Theory)

Deep Neural Network Architectures

The Evolved Transformer

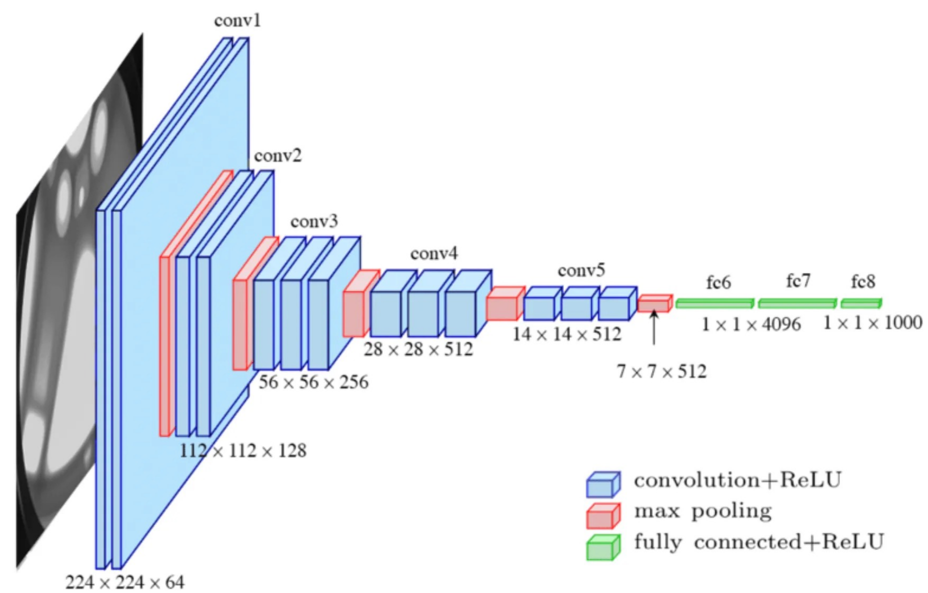
David So, Quoc Le, Chen Liang *Proceedings of the 36th International Conference on Machine Learning*, PMLR 97:5877-5886, 2019.



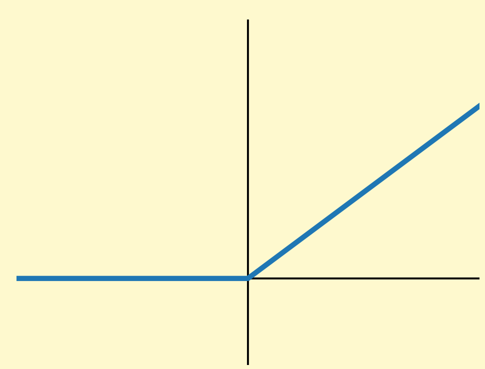
Google DeepMind 2023-10-26

ConvNets Match Vision Transformers at Scale

Samuel L Smith[1], Andrew Brock[1], Leonard Berrada[1] and Soham De[1]
[1]Google DeepMind



Very deep convolutional networks for large-scale image recognition
K Simonyan, A Zisserman
In this work we investigate the effect of the convolutional network depth on its accuracy in the large-scale image recognition setting. Our main contribution is a thorough evaluation of …
Save Cite Cited by 112161 Related articles

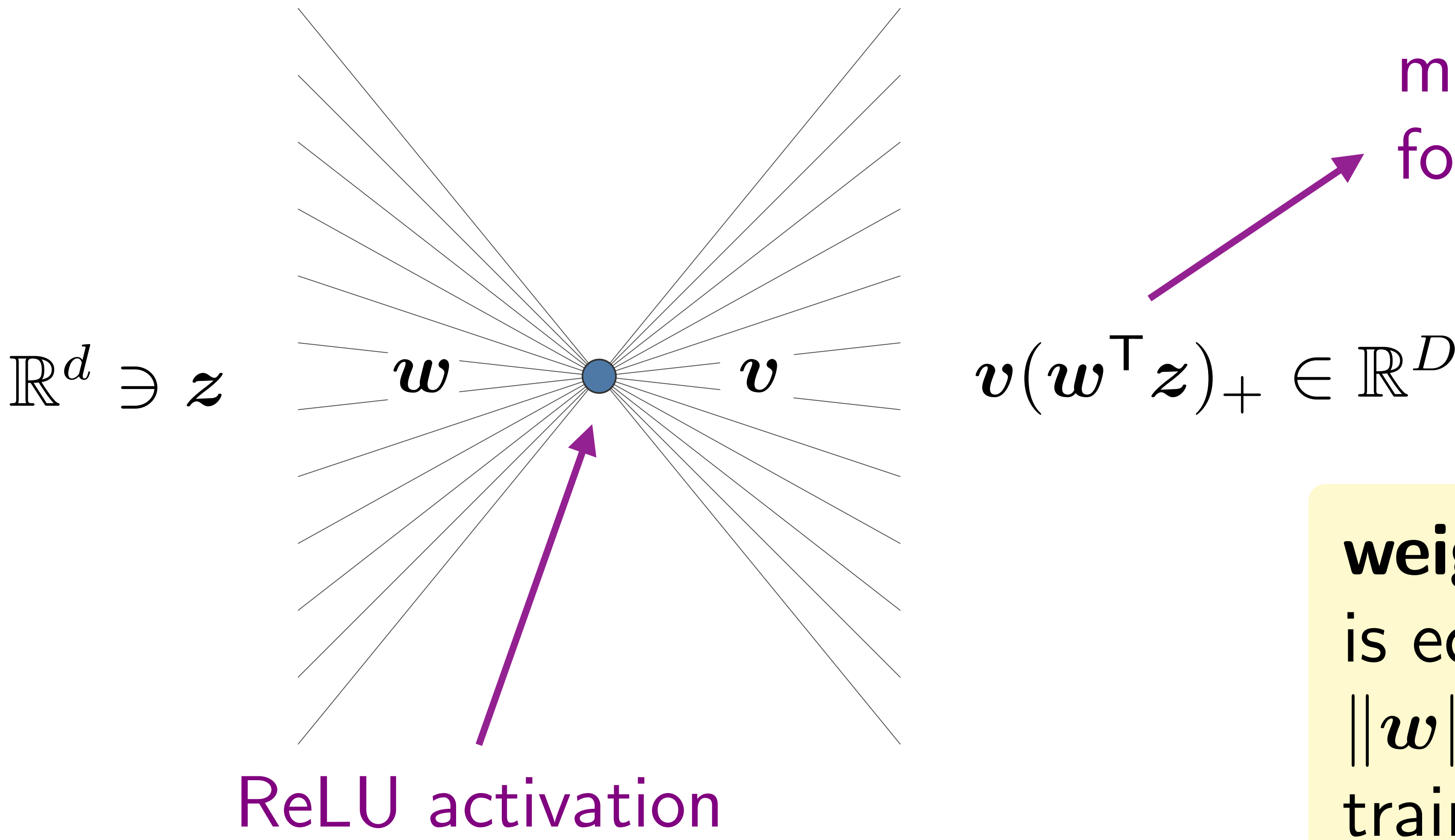


Rectified Linear Unit (ReLU)
$$\mathrm{ReLU}(t) = \max\{0, t\} = t_+$$

$+$ weight decay in training

What Is the Effect of Regularization in Deep Learning?

Neural Balance in Deep Neural Networks



mathematical expression for a single ReLU neuron

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

Neural Balance Theorem (P. and Nowak, 2023)

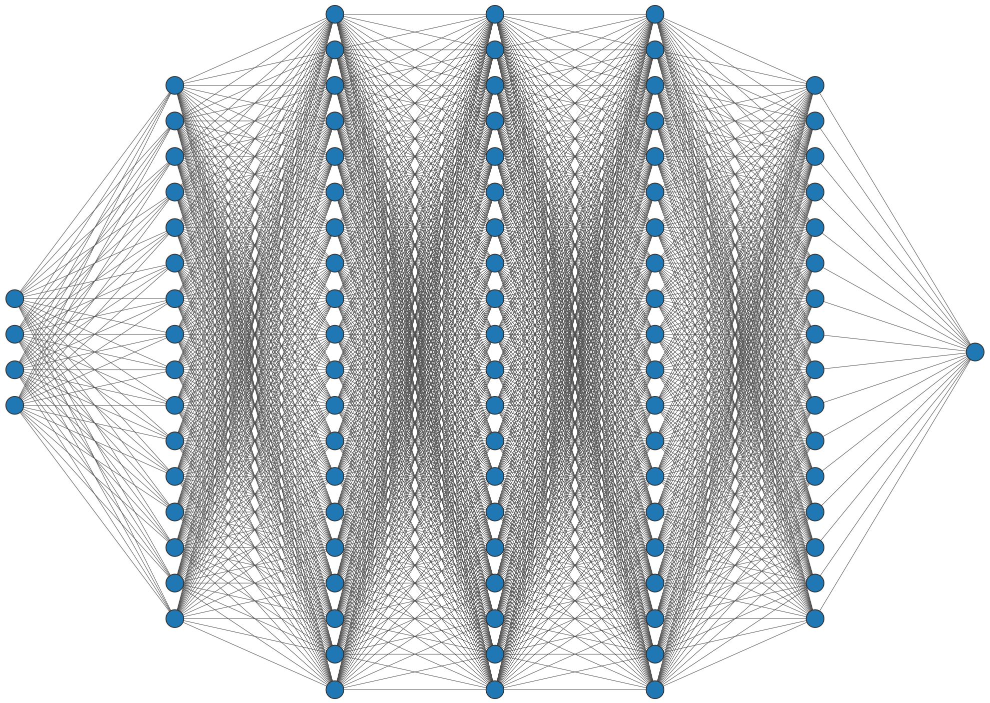
If a DNN is trained with weight decay, then the 2-norms of the input and output weights to each ReLU neuron must be **balanced**.

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

P. and Nowak (2023, IEEE Signal Processing Magazine)

Neural Network Training

$\boldsymbol{x}$



$f_{\boldsymbol{\theta}}(\boldsymbol{x})$

parameterized by a vector $\boldsymbol{\theta} \in \mathbb{R}^P$ of neural network **weights**

Neural network training problem for the data $\{(\boldsymbol{x}_n, y_n)\}_{n=1}^N$.

$$\min_{\boldsymbol{\theta} \in \mathbb{R}^P} \underbrace{\sum_{n=1}^{N} \mathcal{L}(y_n, f_{\boldsymbol{\theta}}(\boldsymbol{x}_n))}_{\text{data fidelity}} + \underbrace{\frac{\lambda}{2} \|\boldsymbol{\theta}\|_2^2}_{\text{regularization}}$$

Weight Decay in Neural Network Training

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^P} \underbrace{\sum_{n=1}^{N} \mathcal{L}(y_n, f_{\boldsymbol{\theta}}(\boldsymbol{x}_n))}_{\mathscr{L}(\boldsymbol{\theta})} + \frac{\lambda}{2}\|\boldsymbol{\theta}\|_2^2$$

weight decay objective

Gradient descent update on θ_i

weight decay

$$\theta_i^{t+1} = \theta_i^t - \tau \left(\frac{\partial \mathscr{L}}{\partial \theta_i}\bigg|_{\theta_i=\theta_i^t} + \lambda\theta_i^t \right) = \theta_i^t - \tau \frac{\partial \mathscr{L}}{\partial \theta_i}\bigg|_{\theta_i=\theta_i^t} - \tau\lambda\theta_i^t$$

step size
“learning rate”

GD update on $\mathscr{L}$

Hanson and Pratt (1988, NeurIPS)
Krogh and Hertz (1990, NeurIPS)

Neural Balance

The ReLU activation is **homogeneous**

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

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

Proof. The solution to

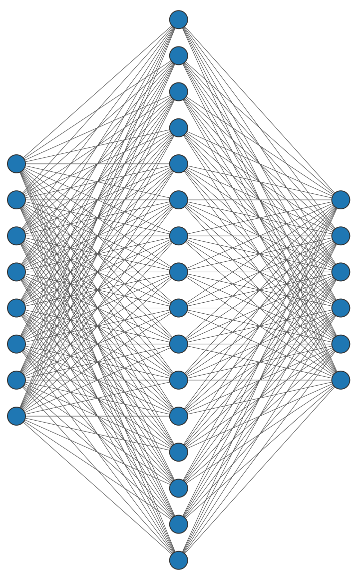
$$\min_{\gamma>0} \|\gamma^{-1}\boldsymbol{v}\|_2 + \|\gamma \boldsymbol{w}\|_2$$

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

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

Grandvalet (1998, ICANN)
Neyshabur et al. (2015, ICLR Workshop)

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

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

path-norm

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

multitask lasso

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

Rebalancing

Secret Sparsity of Weight Decay

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

- 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)

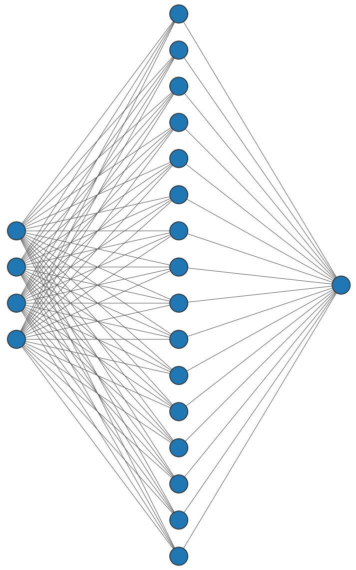
What Kinds of Functions Do Neural Networks Learn?

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

Practical Implications for Learning with Deep Neural Networks.

What Kinds of Functions Do Neural Networks Learn?

Shallow Neural Networks With Scalar Outputs



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

$$\min_{\boldsymbol{\theta}=\{(\boldsymbol{w}_k, v_k)\}_{k=1}^{K}} \sum_{n=1}^{N} \mathcal{L}(y_n, f_{\boldsymbol{\theta}}(\boldsymbol{x}_n)) + \frac{\lambda}{2} \sum_{k=1}^{K} |v_k|^2 + \|\boldsymbol{w}_k\|_2^2$$

$$\min_{\boldsymbol{\theta}=\{(\boldsymbol{w}_k, v_k)\}_{k=1}^{K}} \sum_{n=1}^{N} \mathcal{L}(y_n, f_{\boldsymbol{\theta}}(\boldsymbol{x}_n)) + \lambda \sum_{k=1}^{K} |v_k| \|\boldsymbol{w}_k\|_2$$

path-norm

Path-Norm and Neural Banach Spaces

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

finite-width networks

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

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

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

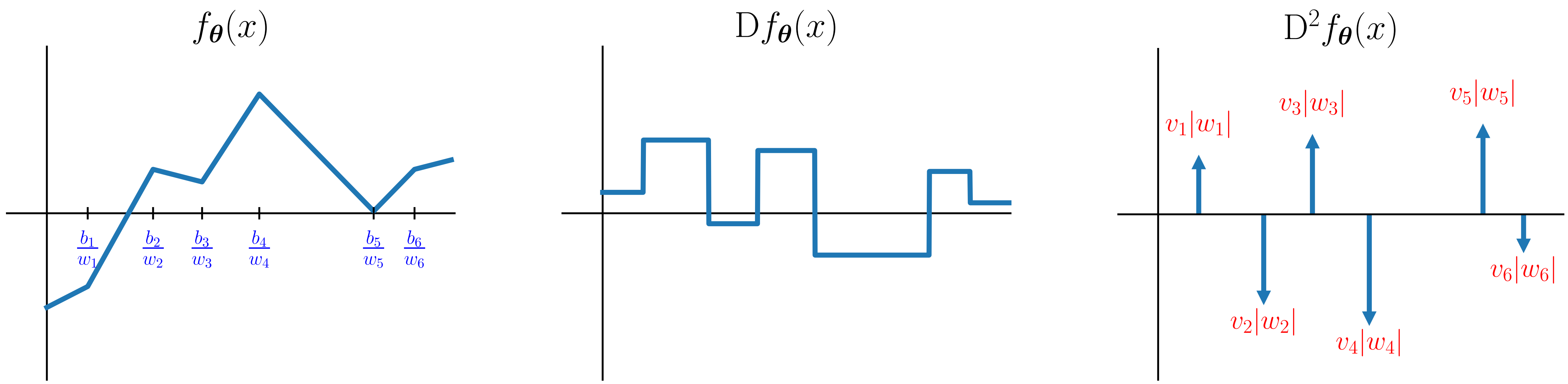
$$f(\boldsymbol{x}) = \int_{\mathbb{S}^{d-1}} (\boldsymbol{w}^\mathsf{T}\boldsymbol{x})_+ \, \mathrm{d}\nu(\boldsymbol{w}).$$

"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 $\mathrm{D} f_{\boldsymbol{\theta}}$

"How do infinite width bounded norm networks look in function space?"
Pedro Savarese, Itay Evron, Daniel Soudry, and Nathan Srebro
Conference on Learning Theory (2019)

Weight Decay $= \mathrm{TV}(\mathrm{D}f)$-Regularization

$$\min_{\boldsymbol{\theta}=\{(w_k,v_k)\}_{k=1}^K} \sum_{n=1}^N \mathcal{L}(y_n, f_{\boldsymbol{\theta}}(x_n)) + \frac{\lambda}{2}\sum_{k=1}^K |v_k|^2 + |w_k|^2$$

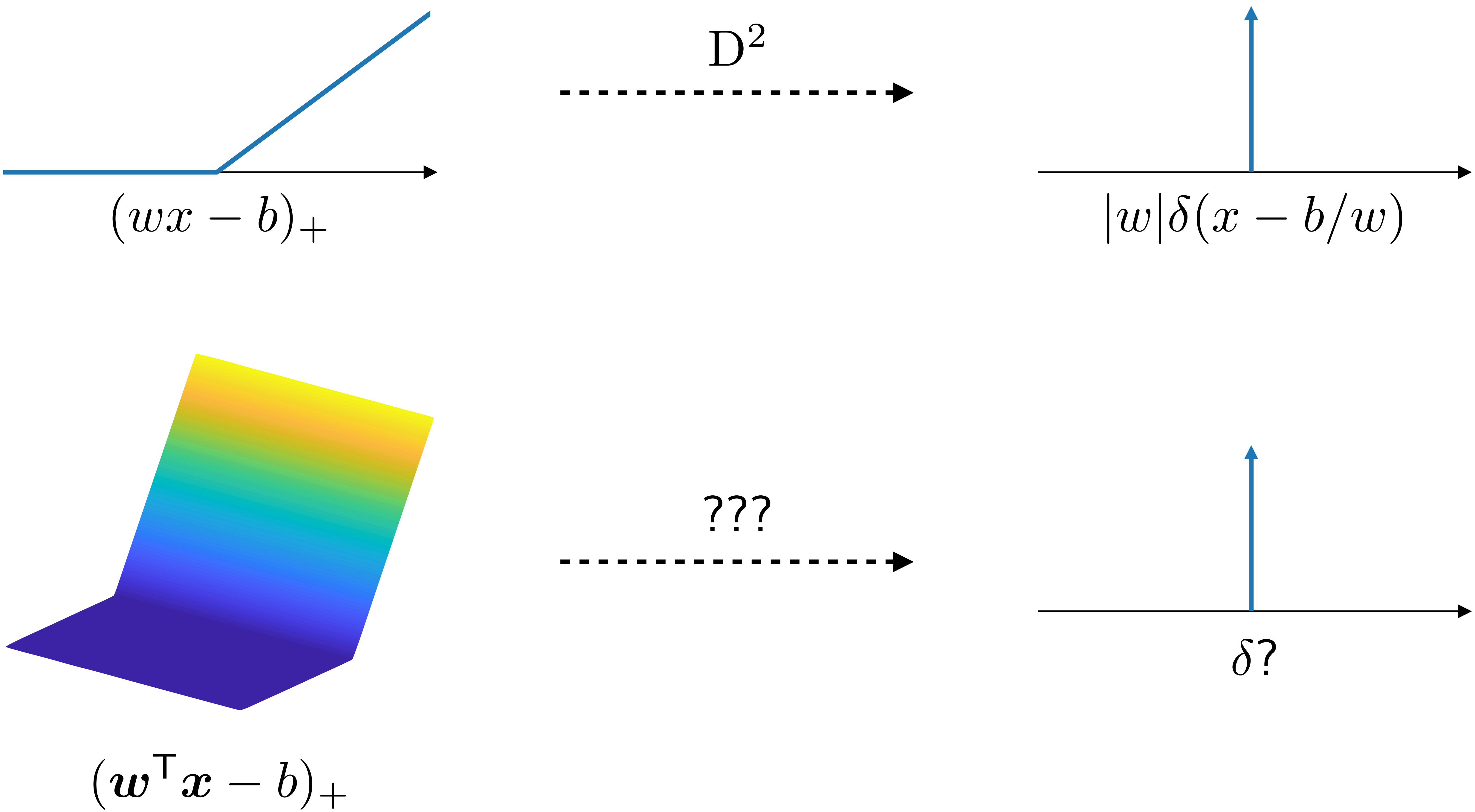
$$\min_{\boldsymbol{\theta}=\{(w_k,v_k)\}_{k=1}^K} \sum_{n=1}^N \mathcal{L}(y_n, f_{\boldsymbol{\theta}}(x_n)) + \lambda\sum_{k=1}^K |v_k||w_k|$$

$$\min_{\boldsymbol{\theta}=\{(w_k,v_k)\}_{k=1}^K} \sum_{n=1}^N \mathcal{L}(y_n, f_{\boldsymbol{\theta}}(x_n)) + \lambda\,\mathrm{TV}(\mathrm{D}\, f_{\boldsymbol{\theta}})$$

→ $\mathrm{TV}^2(f_{\boldsymbol{\theta}})$

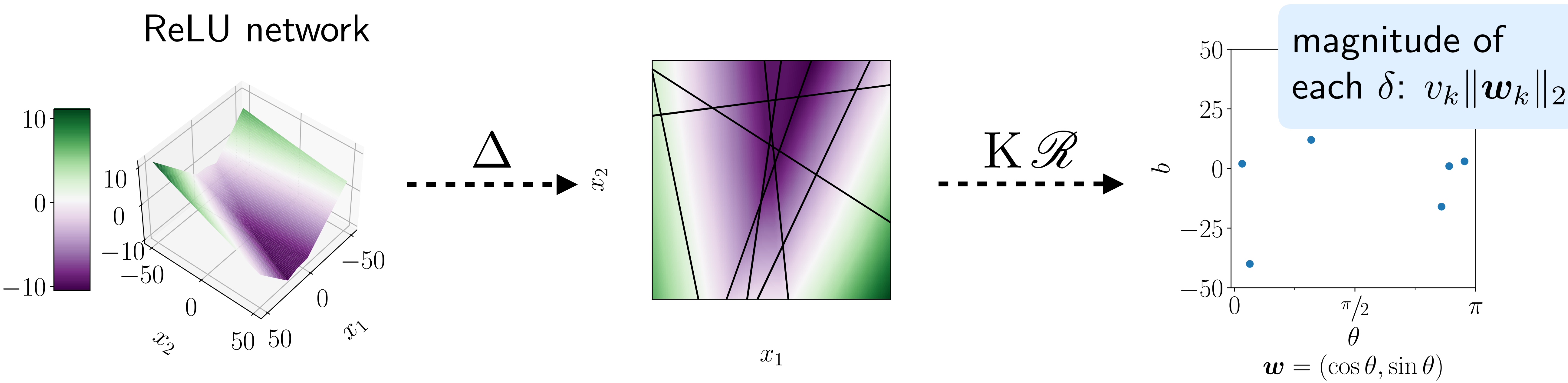
BV^2 is the space of all functions with $\mathrm{TV}^2(f) = \|\mathrm{D}^2 f\|_{\mathcal{M}} < \infty$.

What About the Multivariate Case?



Multivariate Extension: The Radon Transform

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



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

second-order Radon-domain total variation

"A function space view of bounded norm infinite width ReLU nets: The multivariate case"
Greg Ongie, Rebecca Willett, Daniel Soudry, and Nathan Srebro
International Conference on Learning Representations (2020)

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

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!

P. and Nowak (2021, Journal of Machine Learning Research)

A Banach Space Representer Theorem

Neural Network Representer Theorem (**P.** and Nowak 2021)

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

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

that admits a representation of the form

$$f_{\mathrm{ReLU}}(\boldsymbol{x}) = \sum_{k=1}^K v_k \underbrace{(\boldsymbol{w}_k^\mathsf{T}\boldsymbol{x} - b_k)_+}_{\text{ReLU neurons}} + \underbrace{\boldsymbol{w}_0^\mathsf{T}\boldsymbol{x} + b_0,}_{\text{skip connection}} \quad \underbrace{K < N.}_{\text{sparse solution}}$$

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

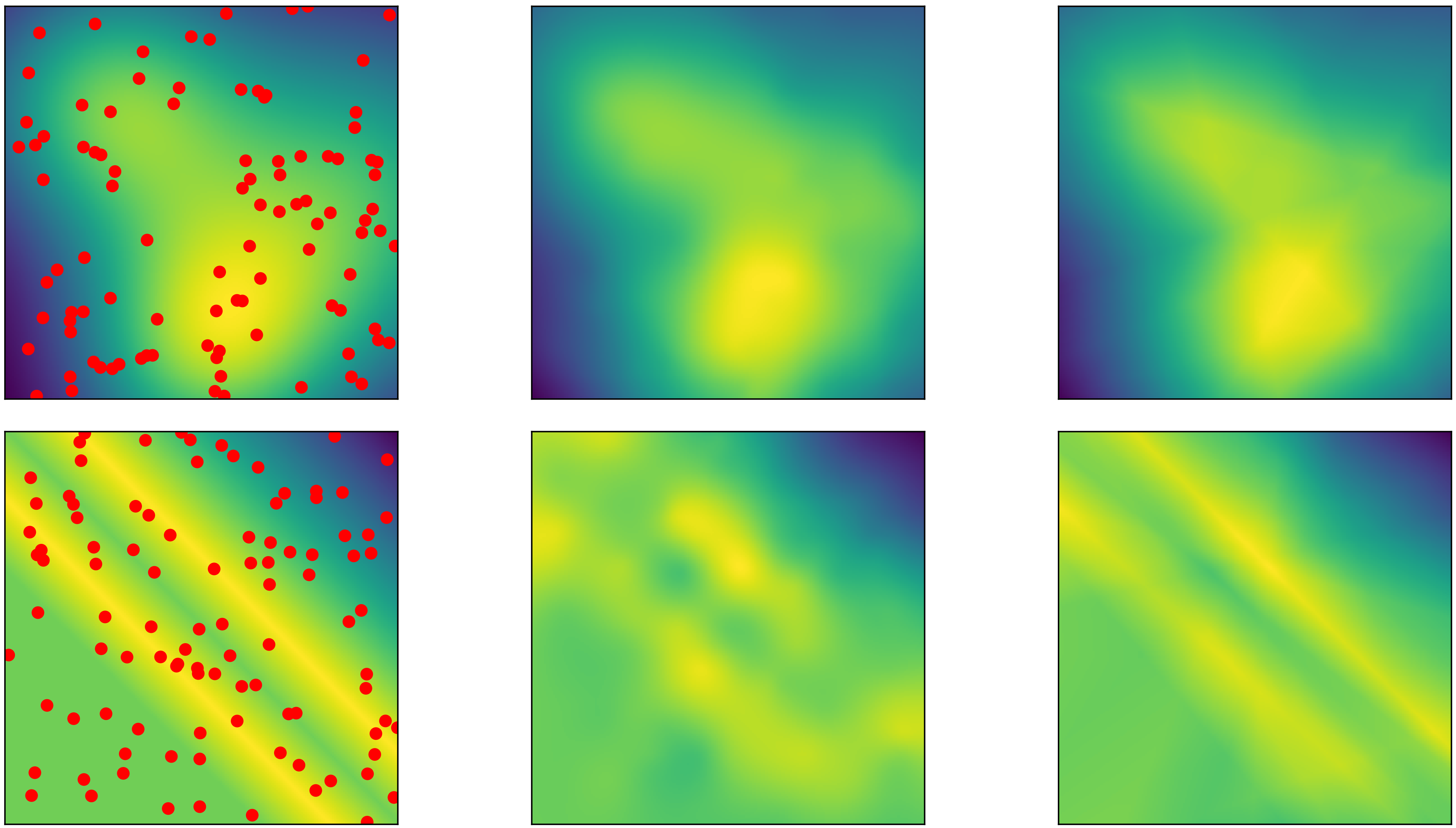
Neural networks learn $\mathscr{R}\,\mathrm{BV}^2$-functions.

P. and Nowak (2021, Journal of Machine Learning Research)

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

Neural Networks Adapt to Directional Smoothness

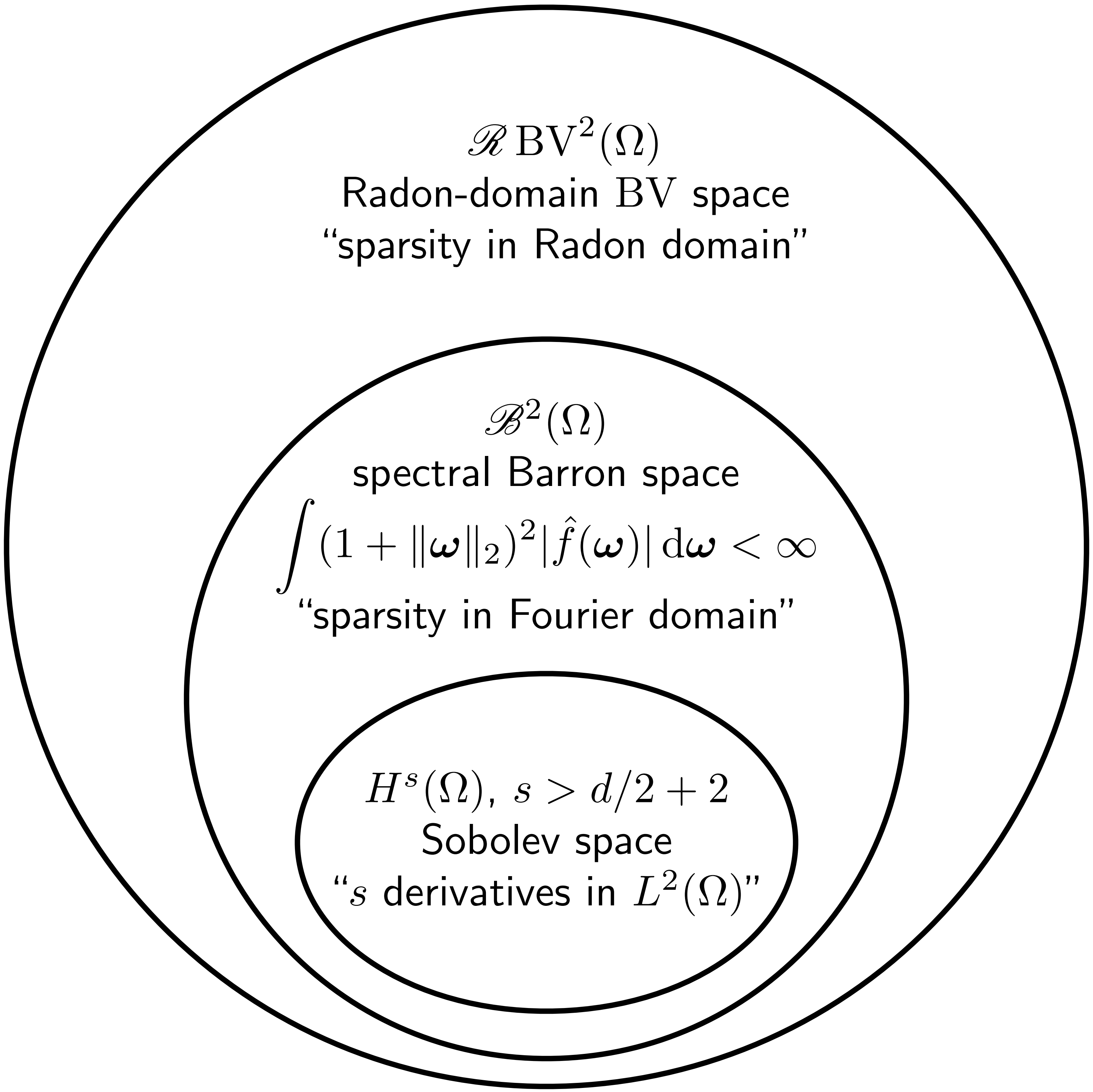
True function and noisy data	Thin-plate spline (kernel method)	Neural network (nonlinear method)

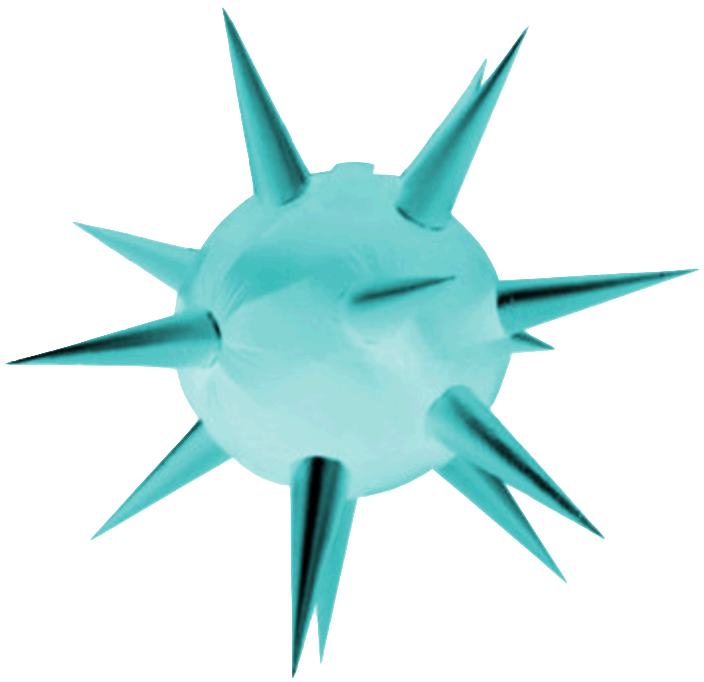


Variation in only a **few directions** is a defining characteristic of $\mathscr{R}\,\mathrm{BV}^2$.

P. and Nowak (2023, IEEE Transactions on Information Theory)

Neural Banach Spaces





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

P. and Nowak (2023, IEEE Transactions on Information Theory)

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

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

$-\alpha$

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

By the inequality of Carl (1981), this implies

$$\log \mathcal{N}(\delta, U(\mathscr{R}\,\mathrm{BV}^2), \|\cdot\|_{L^\infty(\Omega)}) = \widetilde{O}(\delta^{-\frac{2d}{d+3}}) = \widetilde{O}(\delta^{-2}).$$

$-\frac{1}{\alpha}$

unit ball

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

P. and Nowak (2023, IEEE Transactions on Information Theory)

Minimax Optimality of Neural Networks

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

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

weight decay objective

satisfies

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

no curse

minimax rate

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

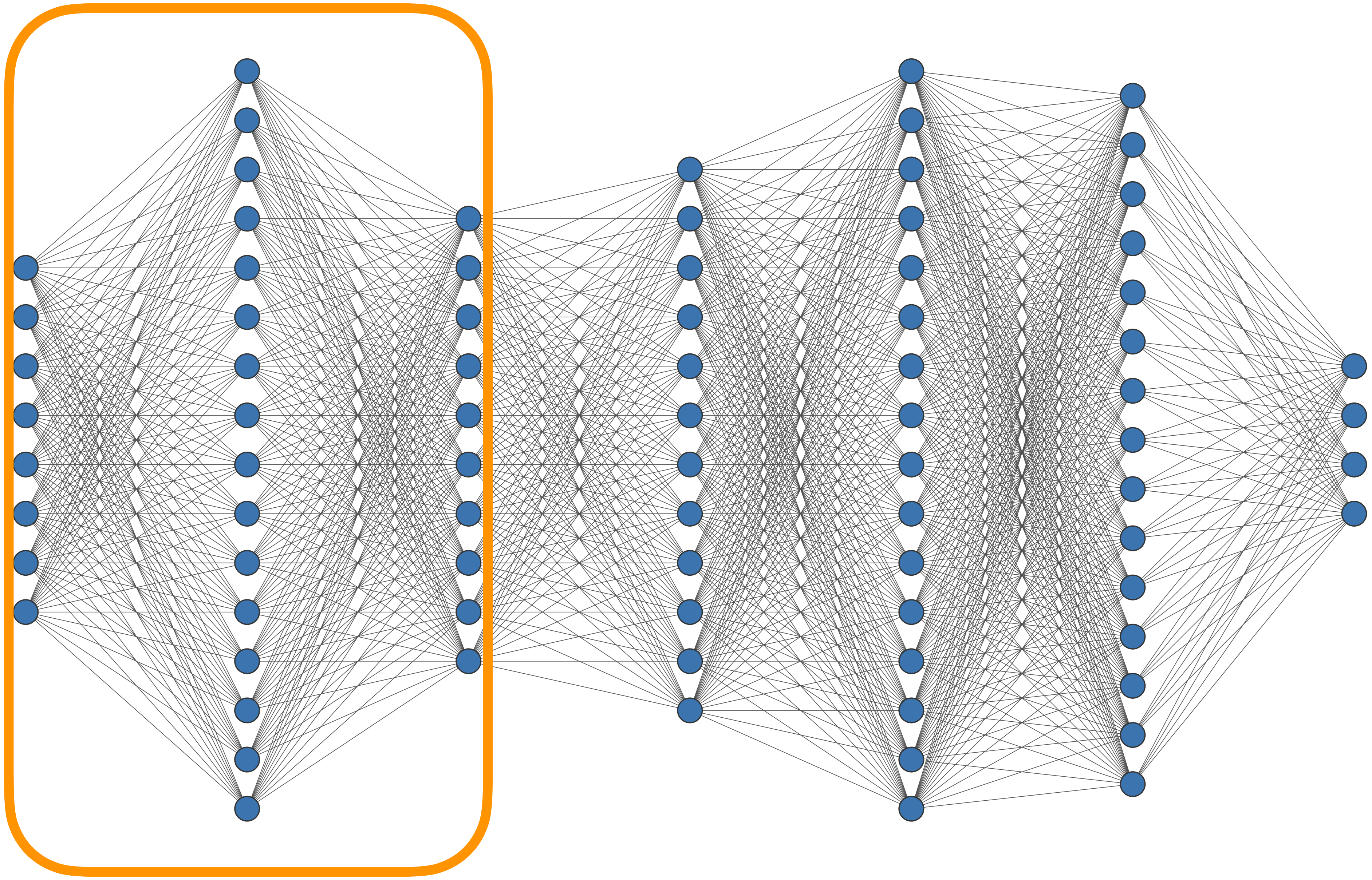
Linear minimax lower bound: $N^{-\frac{3}{d+3}}$

the curse

P. and Nowak (2023, IEEE Transactions on Information Theory)

What Does All of This Mean for Learning With Deep Neural Networks?

Layers of Vector-Valued Shallow Networks

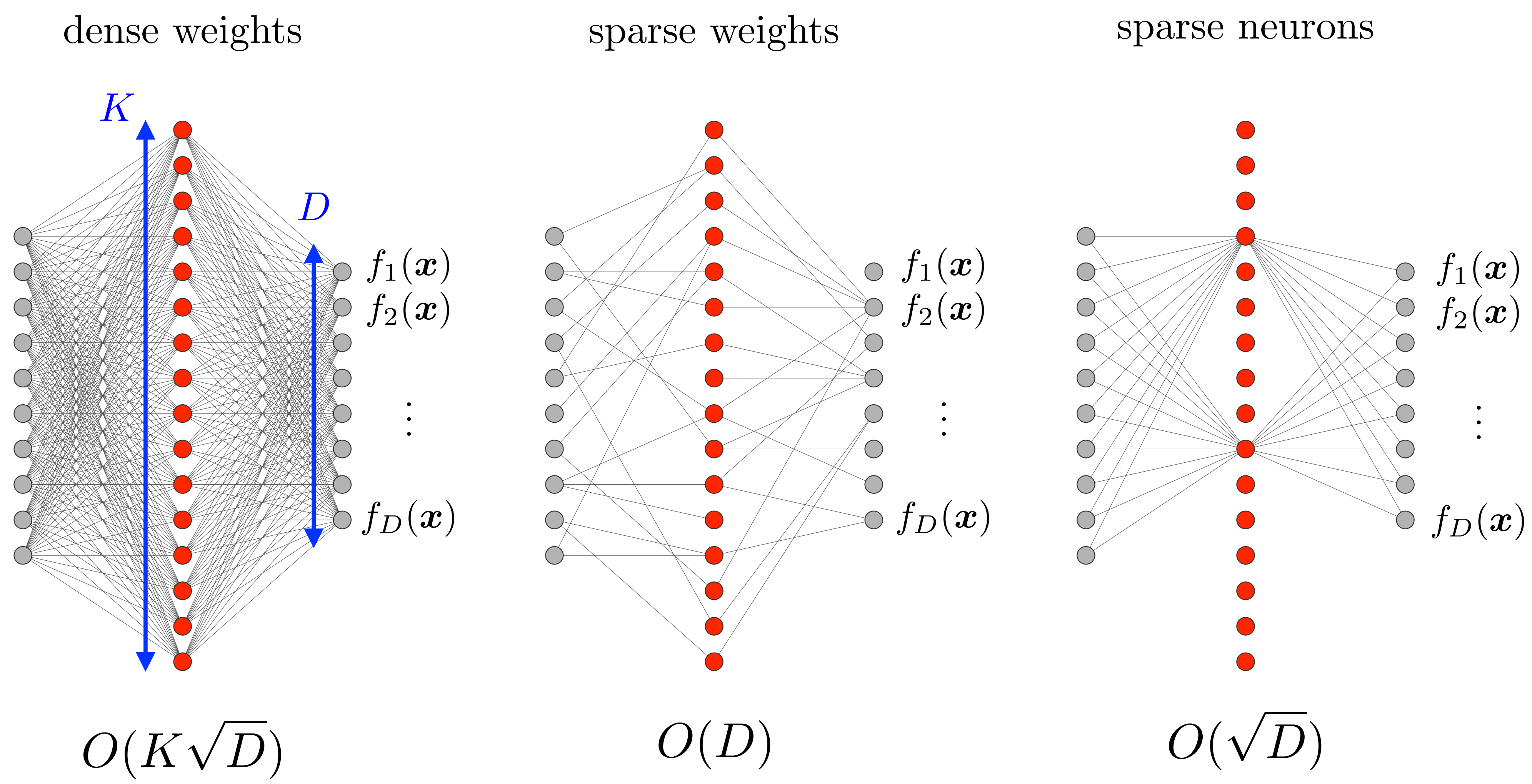


Deep Neural Networks are **Layers** of Shallow Vector-Valued Networks

The Structured Sparsity of Weight Decay

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

weight decay
$\Longleftrightarrow$
non-convex multitask lasso



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

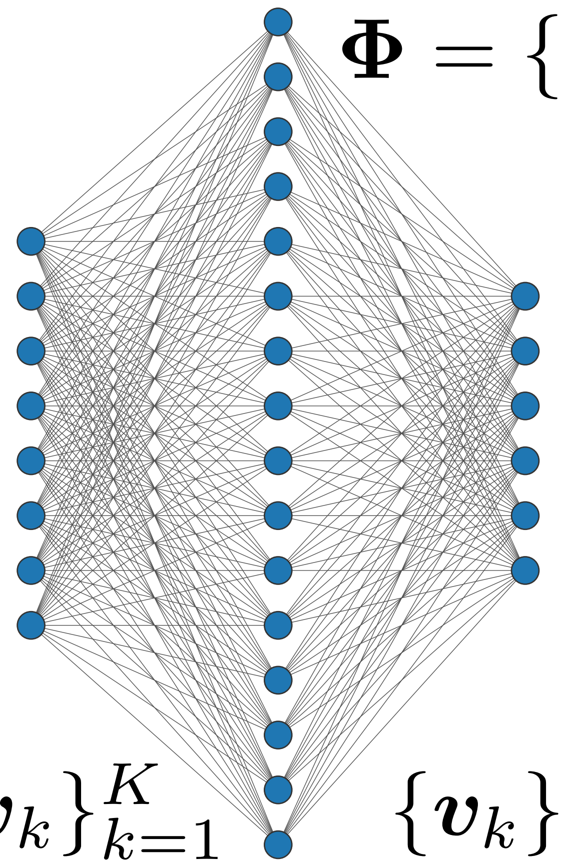
Weight decay favors outputs that "share" neurons (sparse neurons)

Tight Bounds on Widths

Consider one ReLU layer within a **trained** deep neural network
with weight decay
to a global minimizer

$\mathbf{\Phi} = \{\phi_n\}_{n=1}^N$

$\boldsymbol{x}_n$



$\mathbf{\Psi} = \{\psi_n\}_{n=1}^N$

$\{\boldsymbol{w}_k\}_{k=1}^K$ $\{\boldsymbol{v}_k\}_{k=1}^K$

push the magnitude of $\boldsymbol{w}_k$ into $\boldsymbol{v}_k$

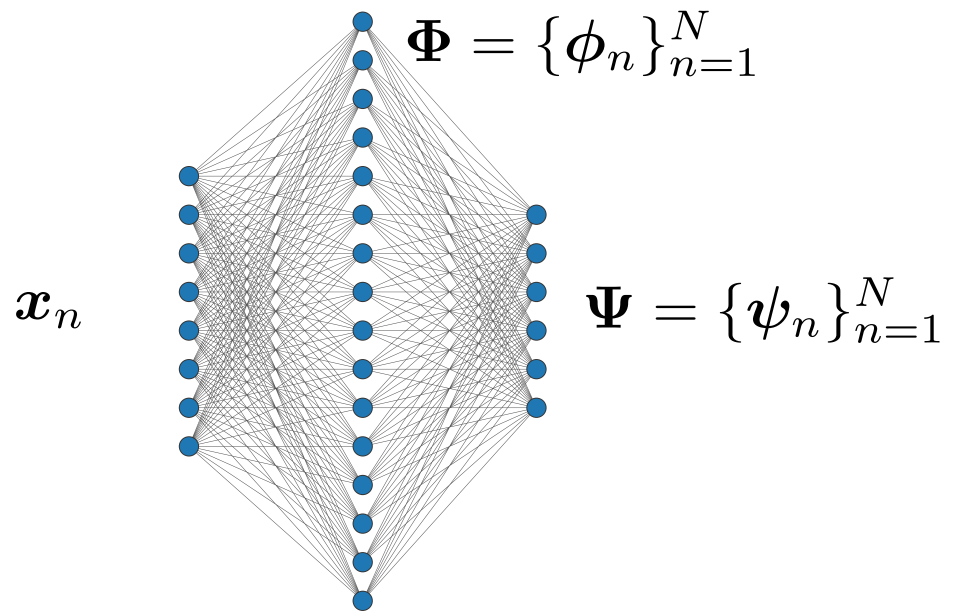
At each layer, the weight decay solution minimizes

multitask lasso

$$\min_{\{\boldsymbol{v}_k\}_{k=1}^K} \sum_{k=1}^K \|\boldsymbol{v}_k\|_2 \quad \text{s.t.} \quad \mathbf{\Psi} = \mathbf{V}\mathbf{\Phi}.$$

Shenouda, **P.**, Lee, and Nowak (2023+)

Tight Bounds on Widths



$$\min_{\{v_k\}_{k=1}^K} \sum_{k=1}^{K} \|v_k\|_2 \quad \text{s.t.} \quad \Psi = \mathbf{V}\Phi.$$

Low-rank data embeddings have been observed empirically by Huh et al. (2022).

Layer Width Theorem (Shenouda, P., Lee and Nowak 2023+)

Let Φ denote the post-activation features and Ψ denote the neuron outputs of any ReLU layer in a **trained** DNN (minimizes the weight decay objective). Then, there exists a representation with

$$K \leq \operatorname{rank}(\Phi)\operatorname{rank}(\Psi) \leq N^2$$

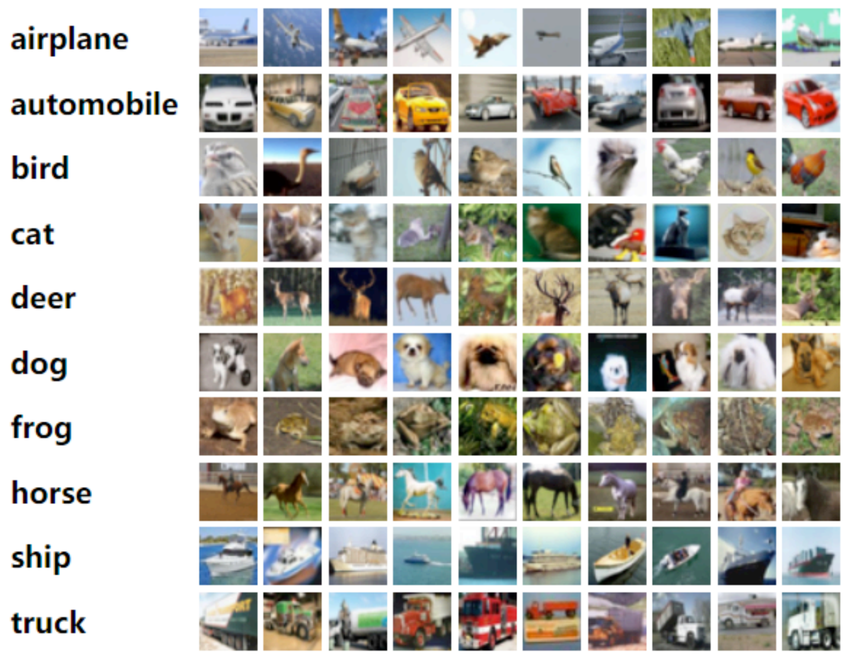
Bound of Jacot (2023): $N(N+1)$.

neurons. The representation can be found by solving a **convex multitask lasso** problem.

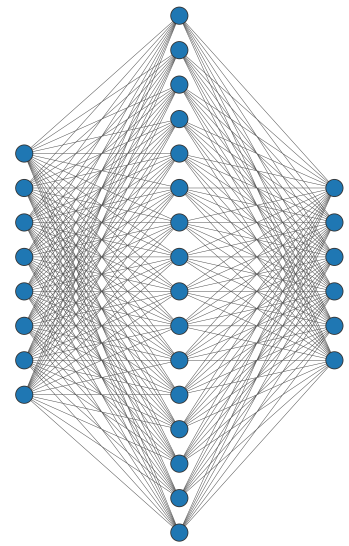
Shenouda, **P.**, Lee, and Nowak (2023+)

Application: Principled DNN Compression

VGG-19 trained with weight decay on CIFAR-10.



final ReLU layer
$K = 512$ neurons



output dimension
$D = 10$

Theory: There exists a representation with

$$\leq \mathrm{rank}(\mathbf{\Phi})\,\mathrm{rank}(\mathbf{\Psi}) \approx 10 \cdot 10 = 100 \text{ neurons.}$$

	original network	compressed network
active neurons	512	47
test accuracy	93.92%	93.88%
train loss	0.0104	0.0112

$10\times$ compression!
no change in performance!

Shenouda, **P.**, Lee, and Nowak (2023+)

Summary

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

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

Weight decay is secretly a **sparsity-promoting** regularization scheme.

- promotes **neuron sharing** (structured sparsity)
- motivates the design of **principled** DNN compression schemes

This is Just the Beginning!

Going Forward: Theory

What kinds of functions do **structured neural architectures** learn?

- Attention mechanisms and **transformers**
- Orthogonal weight normalization: $\mathbf{W}^\mathsf{T}\mathbf{W} = \mathbf{I}$ **P.** and Unser (2023+)

What are the fundamental limits of **shallow** networks?

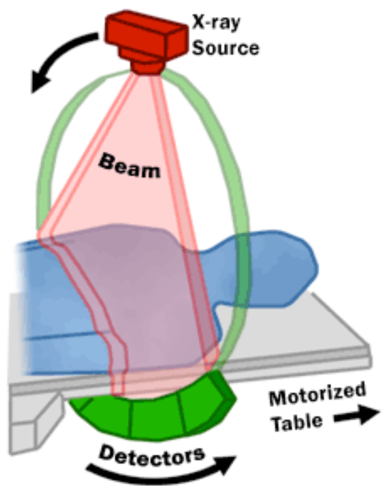
- $\mathscr{R}\mathrm{BV}^2$ does not capture everything DeVore, Nowak, **P.** and Siegel (2023+)
- Characterization of the **approximation spaces** of shallow networks
- **Quantitative** depth separation results

Going Forward: Applications

Function-space view on **implicit neural representations**

- Implicitly defined, continuous, differentiable signal representations parameterized by neural networks

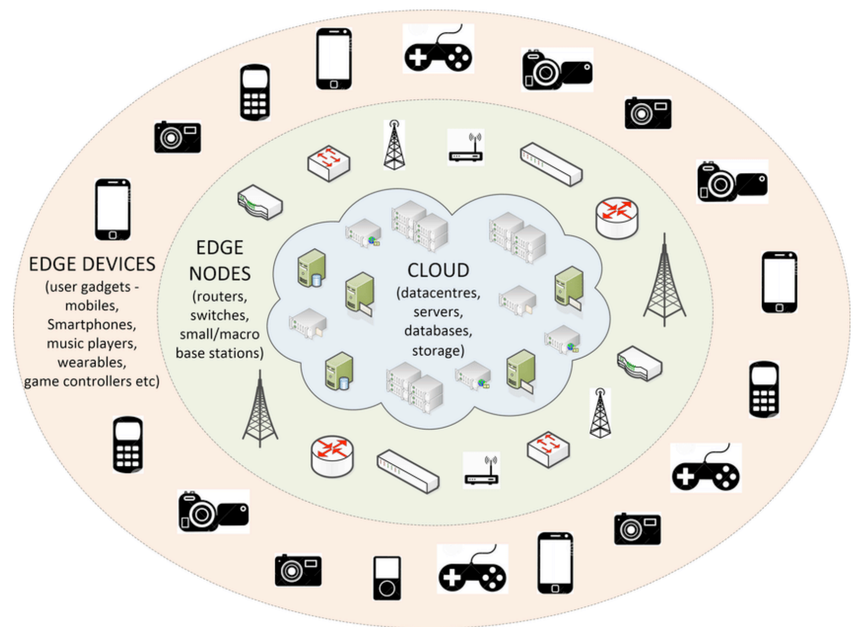
Stanley (2007)



- Gained popularity for denoising, compression, and inverse problems (e.g., cryo-EM, CT)

Fundamental limits of DNN compression

- Fast inference on **edge devices** and **embedded systems**



Research Vision

Towards **trustworthy** and **reliable** deep learning in practice.

P. and Nowak (2021, J. Mach. Learn. Res.)
P. and Nowak (2023, IEEE Trans. Inf. Theory)
P. and Nowak (2023, IEEE Signal Process. Mag.)
P. and Unser (2023, arXiv)
DeVore, Nowak, **P.**, and Siegel (2023, arXiv)

Shenouda, **P.**, Lee, and Nowak (2023, arXiv)

Theory

Numerical Investigations

e.g., compressed DNNs on the edge

ongoing research:
INRs for image denoising

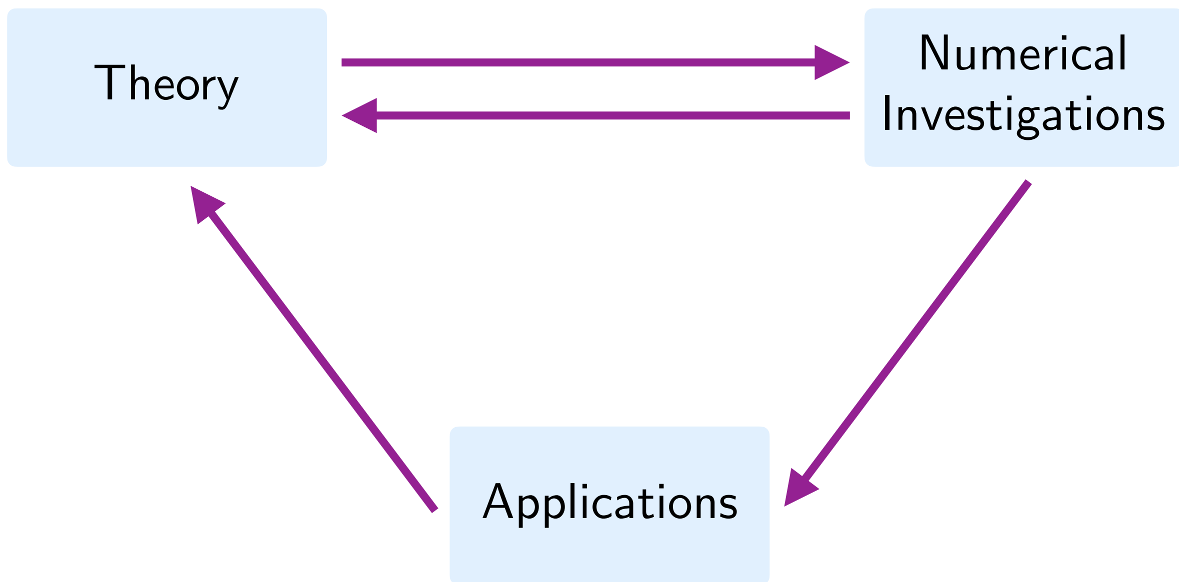
Applications

e.g., implicit neural representations

Collaborations with domain experts in

- computer engineering
- imaging science

Conclusion



Questions?

Collaborators:

Rob Nowak, UW–Madison, USA
Joe Shenouda, UW–Madison, USA
Kangwook Lee, UW–Madison, USA
Ron DeVore, Texas A&M University, USA
Jonathan Siegel, Texas A&M University, USA
Michael Unser, EPFL, Switzerland
Pakshal Bohra, EPFL, Switzerland
Mehrsa Pourya, EPFL, Switzerland
Stan Ducotterd, EPFL, Switzerland

Funding:

NSF
Office of Naval Research Science & Technology
erc
Swiss National Science Foundation