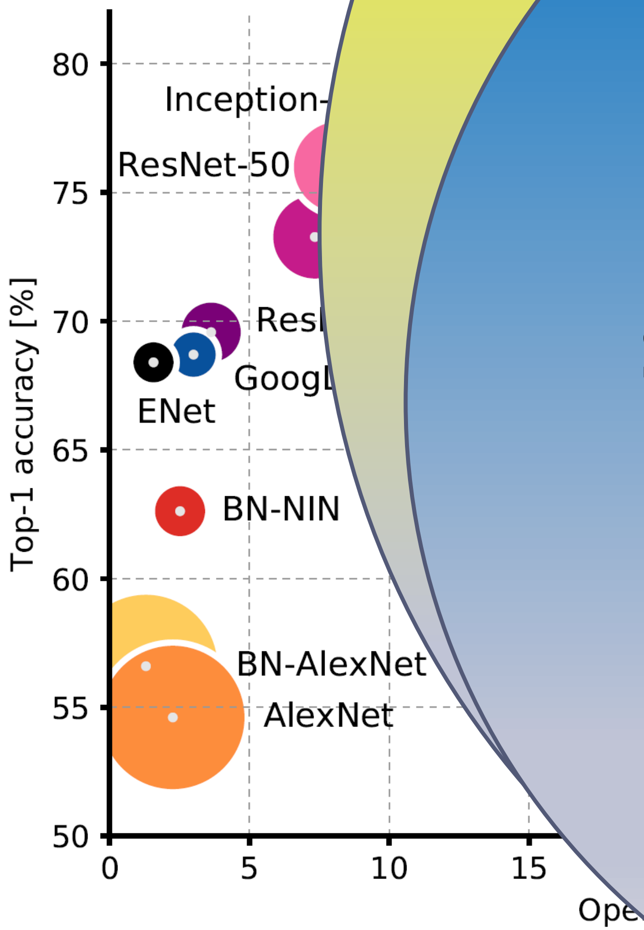
Neural networks, wide and deep, singular kernels and Bayes optimality

Mikhail Belkin
University of California San Diego,
Halıcıoğlu Data Science Institute

Collaborators: Adit Radhakrishnan, Caroline Uhler, Chaoyue Liu, Libin Zhu, Like Hui, Alexander Rakhlin, Alexander Tsybakov

Non-Linear and High Dimensional Inference,
IHP, Paris, Oct 2022

La



From Canziani, et al., 2017.

Switch Transformer, 2021:
1.6 trillion parameters

This talk

- A taxonomy and of infinitely wide and deep interpolating networks.
- Equivalence with singular kernel machines for classification.
- Bayes optimality for classification but not regression.

The problem of machine learning

Input: data $(x_i, y_i),\ i = 1..n\ , x_i \in \mathbb{R}^d, y_i \in \{-1,1\}$ (classification)

Goal: construct $f^*: \mathbb{R}^d \to \mathbb{R}$, that best "generalizes" to new data.

Under the standard statistical assumptions:

$$f^* = \underset{f}{argmin}\, E_{unseen\ data}\, L(f(x), y)$$

Empirical Risk Minimization

Most ML algorithms are based on minimizing empirical risk.

Empirical risk

$$f^*_{ERM} = \underset{f \in \mathcal{H}}{argmin} \frac{1}{n} \sum_{training\ data} L(f(x_i), y_i)$$

Tension between empirical risk and expected risk.

Interpolation and benign over-fitting

Recent understanding: interpolation does not contradict generalization.

…

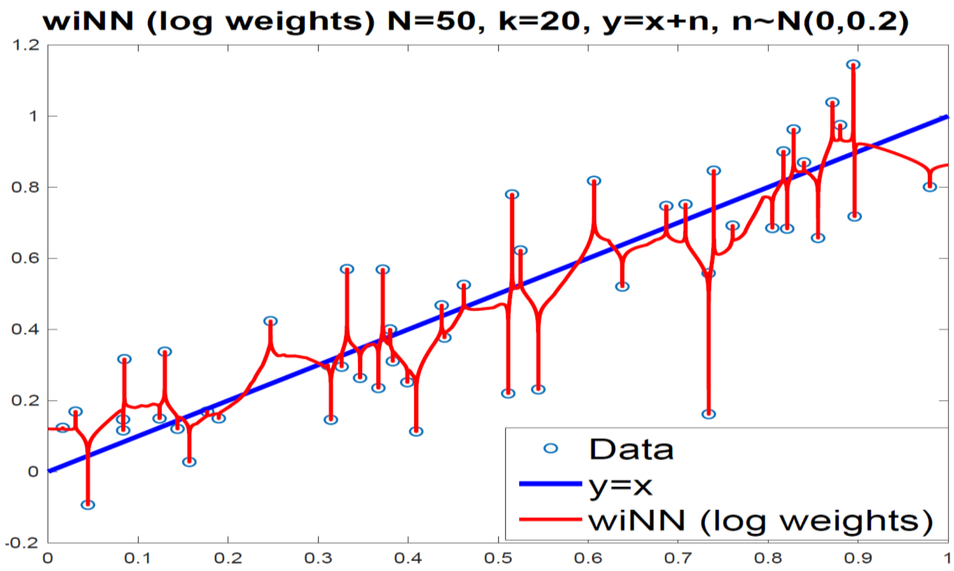


Table 18: ASR results on training and test set, error rate

Model	Task	train with square loss (%) Train	train with square loss (%) Test	train with cross-entropy (%) Train	train with cross-entropy (%) Test	square loss w/ same epochs as CE (%) Train	square loss w/ same epochs as CE (%) Test
Attention+CTC	TIMIT (PER)	0.9	20.8	4.8	20.8	0.9	20.8
(Kim et al., 2017)	TIMIT (CER)	4.5	32.5	11.6	33.4	4.5	32.5
VGG+BLSTMP	WSJ (WER)*	0.7	5.1	0.3	5.3	0.7	5.1
(Moritz et al., 2019)	WSJ (CER)*	0.3	2.4	0.1	2.5	0.3	2.4
VGG+BLSTM	Librispeech (WER)*	0.8	9.8	0.4	10.6	0.8	10.3
(Moritz et al., 2019)	Librispeech (CER)*	0.6	9.7	0.3	10.7	0.6	10.2
Transformer	WSJ (WER)*	0.7	5.7	0.5	5.8	0.7	5.7
(Watanabe et al., 2018)	Librispeech (WER)*	0.9	9.4	1.2	9.2	0.9	9.4

* For WSJ and Librispeech, we take 10% of the training set for the evaluation of the training error rate.

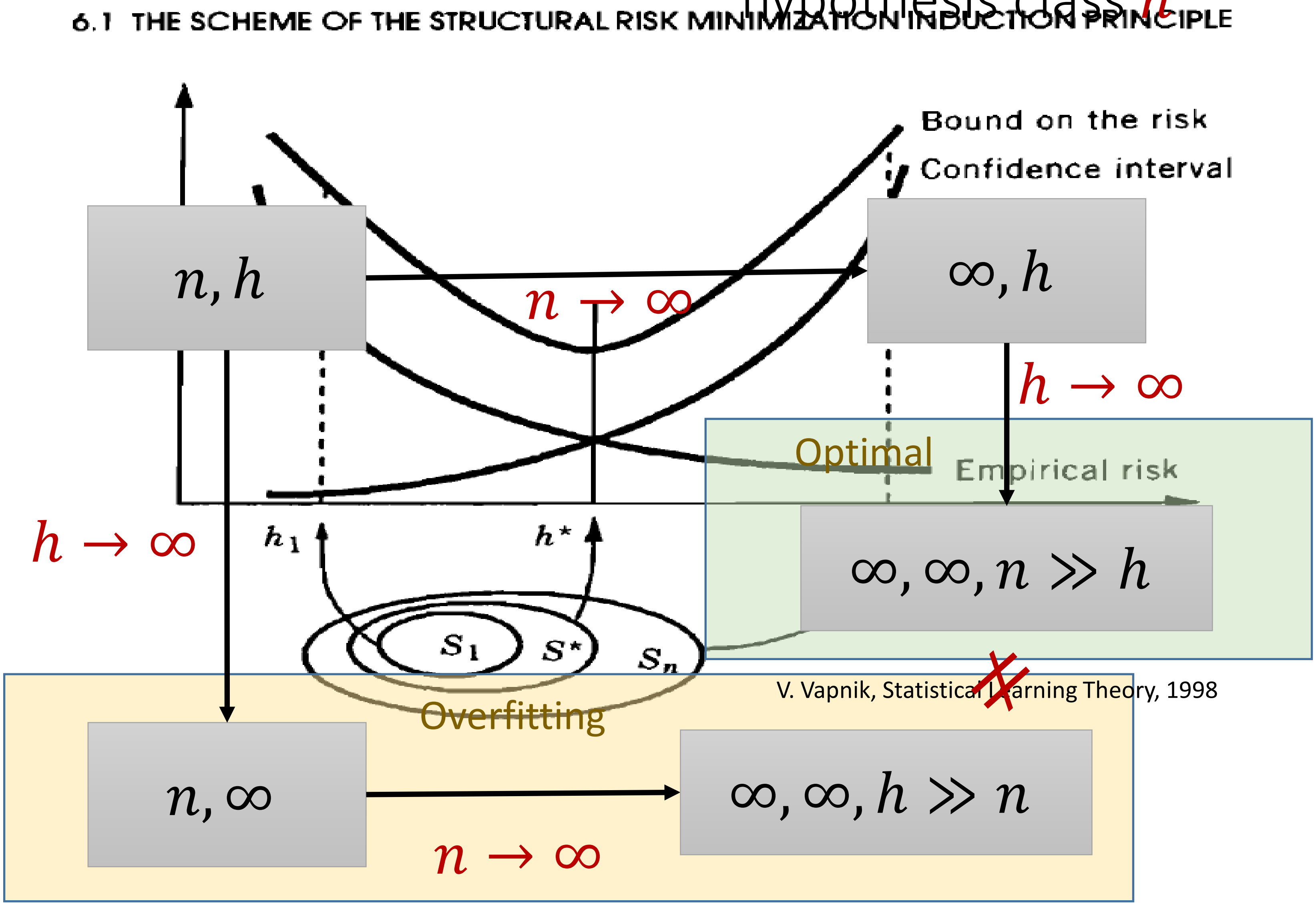
Train loss at optimal early stopping is far below test loss.

[Hui, B., ICLR 2021]

Classical: non-commutative

number of points n

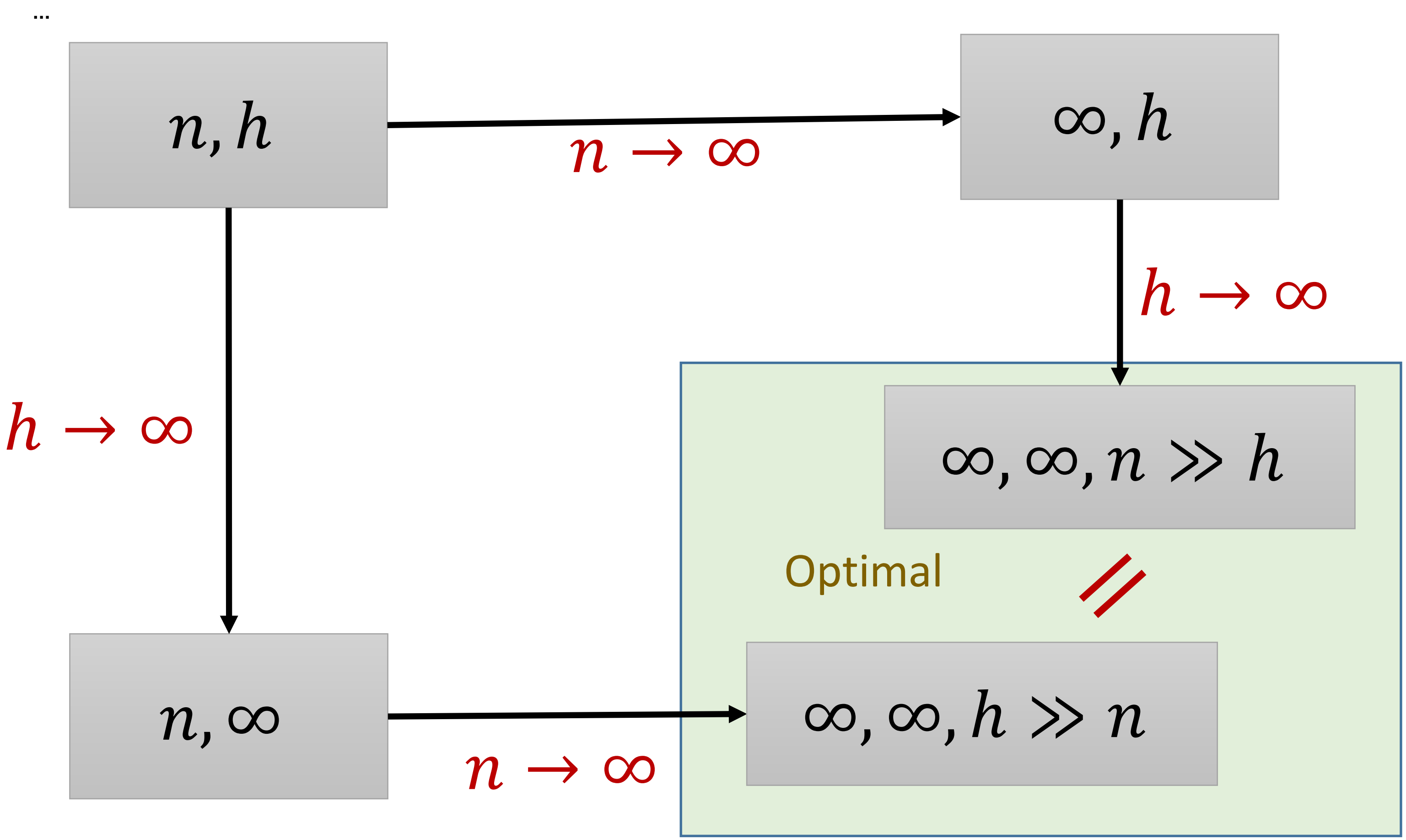
hypothesis class h



“Modern” commutative

number of points n

hypothesis class h



Kernel machines

Beautiful classical statistical/mathematical theory based on Reproducing Kernel Hilbert Spaces (RKHS) -- Hilbert Space of functions with bounded evaluation functionals.

RKHS Theory [Aronszajn,…, 1950s]

Splines [Parzen, Wahba,…, 1970-80s]

Kernel machines [Vapnik,…, 1990s]

Wide neural networks [Jacot, Gabriel, Hogler, …,2020s]

Kernels

Any RKHS corresponds to a PSD kernel.

$$k(x,y) = e^{-\frac{\|x-y\|^2}{\sigma^2}}$$ Gaussian kernel

$$k(x,y) = e^{-\frac{\|x-y\|}{\sigma}}$$ Laplace kernel

Many others…

Interpolating Kernel machines

$$f^* = argmin_{f \in \mathcal{H}, \forall_i f(x_i)=y\_i} \|f\|_{\mathcal{H}}$$

Representer theorem -- solution:

$$f^*(x) = \sum_i \alpha_i k(x_i, x), \ \boldsymbol{\alpha} = K^{-1}\boldsymbol{y}$$

$$K_{ij} = k(x_i, x_j)$$

Width -- Transition to linearity

Very wide neural networks (w. linear output layer)

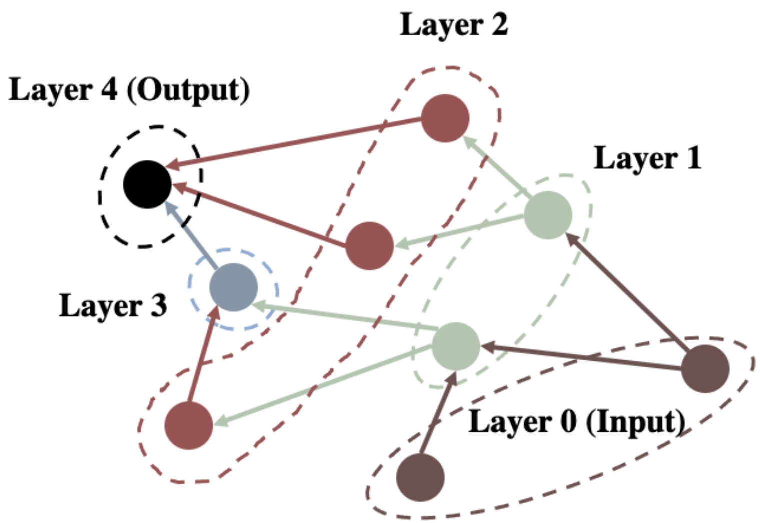
= linear functions of parameters

= kernel machines.

$$k_w(x, z) = \langle \nabla_w f_w(x), \nabla_w f_w(z) \rangle$$

First identified in [Jacot, Gabriel, Hogler, 18] as constant NTK along the training trajectory for the model $f_w(x)$.

Transition to linearity in DAG neural networks



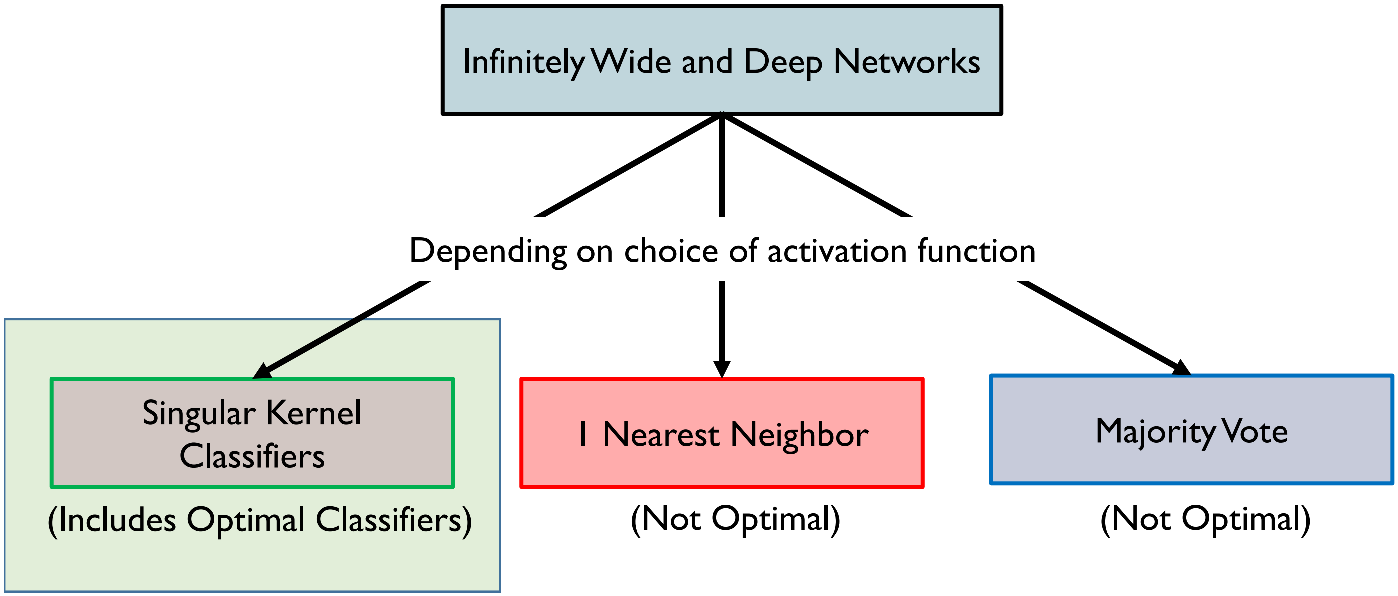
Theorem:

For a neural network (w. arbitrary activations) corresponding to a DAG with minimum in-degree m (width) and linear output layer

$$\|H(F)(w)\| = O\left(1/\sqrt{m}\right)$$

[Zhu, Liu, B., NeurIPS 2022]

Width + Depth



...

[Radhakrishnan, B., Uhler, 2022]

Inverse and Direct methods

Kernel machine (inverse):

$$y(x) = \text{sign}\left(Y_n K_n^{-1} K(\mathrm{X},\mathrm{x})\right)$$

$$(K_n)_{ij} = k\left(x_i, x_j\right) \quad Y_n = (y_1, \dots, y_n),$$

$$K(\mathrm{X},\mathrm{x}) = \left(k(x_1,x), \dots, k(x_n,x)\right)^T$$

Kernel smoother (Nadaraya-Watson)(direct):

$$y(x) = \text{sign}\left(\frac{\sum y_i k(x_i,x)}{\sum k(x_i,x)}\right) = \text{sign}\left(Y_n K(X,x)\right)$$

Singular kernels and Interpolating Nadaraya-Watson schemes

$$f(x) = \frac{\sum y_i k(x_i, x)}{\sum k(x_i, x)}$$

$$k(x_i, x) = \frac{1}{||x - x_i||^\alpha}$$

Claim: NW predictors with singular kernels are interpolating schemes: $\forall_i \, f(x) = y_i$

(Shepard's interpolation, 1968)

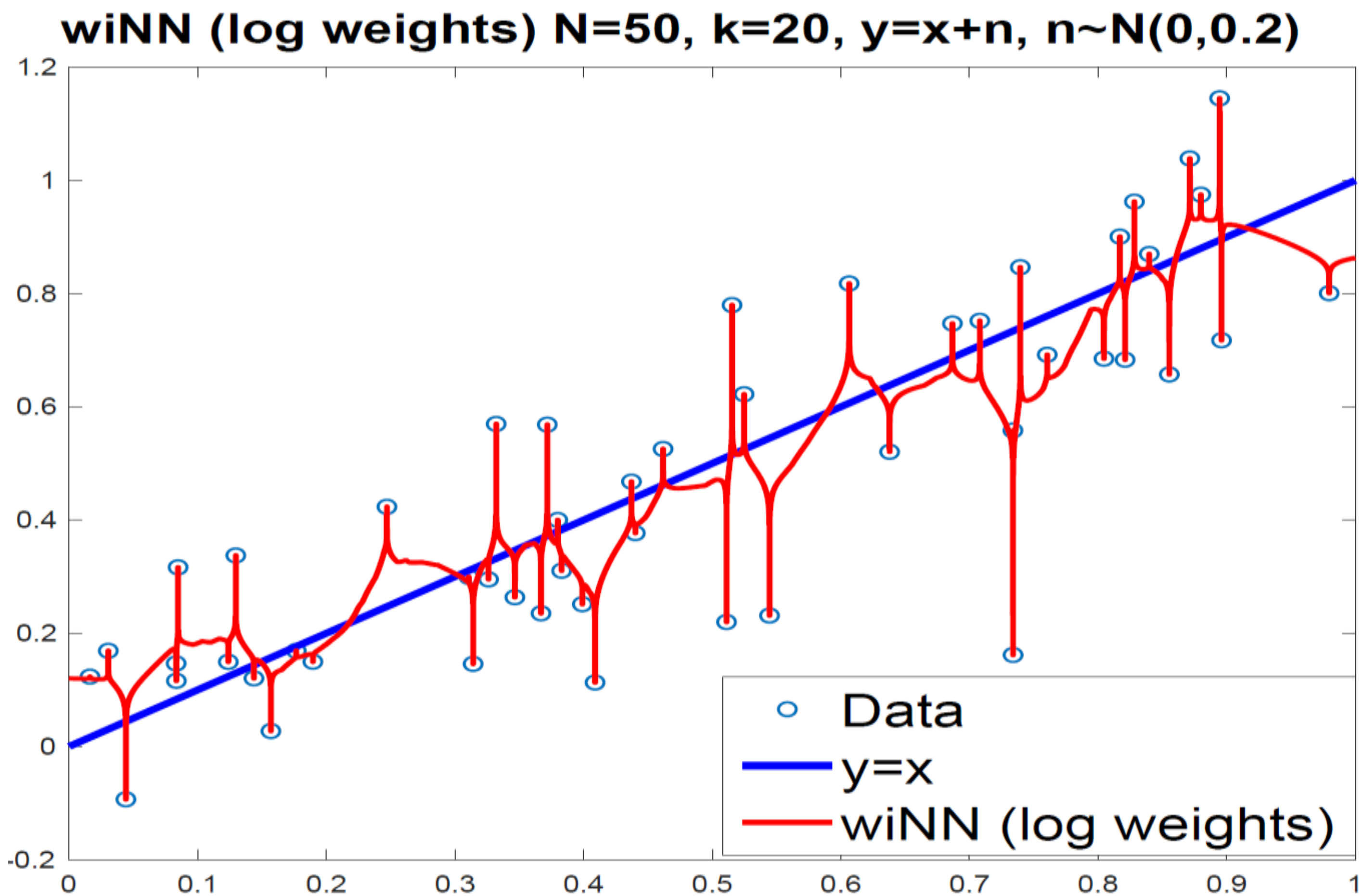
Optimality of Interpolating Nadaraya-Watson schemes

Consistency for regression. [Devroye, Györfi, Krzyzak, 98]
(the Hilbert scheme) $\alpha = d$.

Optimality for regression/classification:

Weighted interpolated schemes with certain singular kernels are consistent (converge to Bayes optimal) for classification in any dimension. Moreover, statistically (minimax) optimal for regression in any dimension.

[B., Hsu, Mitra, NeuriPS 18], followup [B., Rakhlin, Tsybakov, AIStats 19]


wiNN (log weights) N=50, k=20, y=x+n, n~N(0,0.2)
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Data
y=x
wiNN (log weights)

Equivalence of direct and inverse methods

Claim. Direct and inverse methods with singular kernels are equivalent for classification:

$$y(x) = \text{sign}\left(\frac{\sum y_i k(x_i, x)}{\sum k(x_i, x)}\right) = \text{sign}\left(Y_n K_n^{-1} K(X, x)\right)$$

Proof.

1. $\text{sign}\left(\frac{\sum y_i k(x_i,x)}{\sum k(x_i,x)}\right) = \text{sign}\left(Y_n K(X, x)\right)$

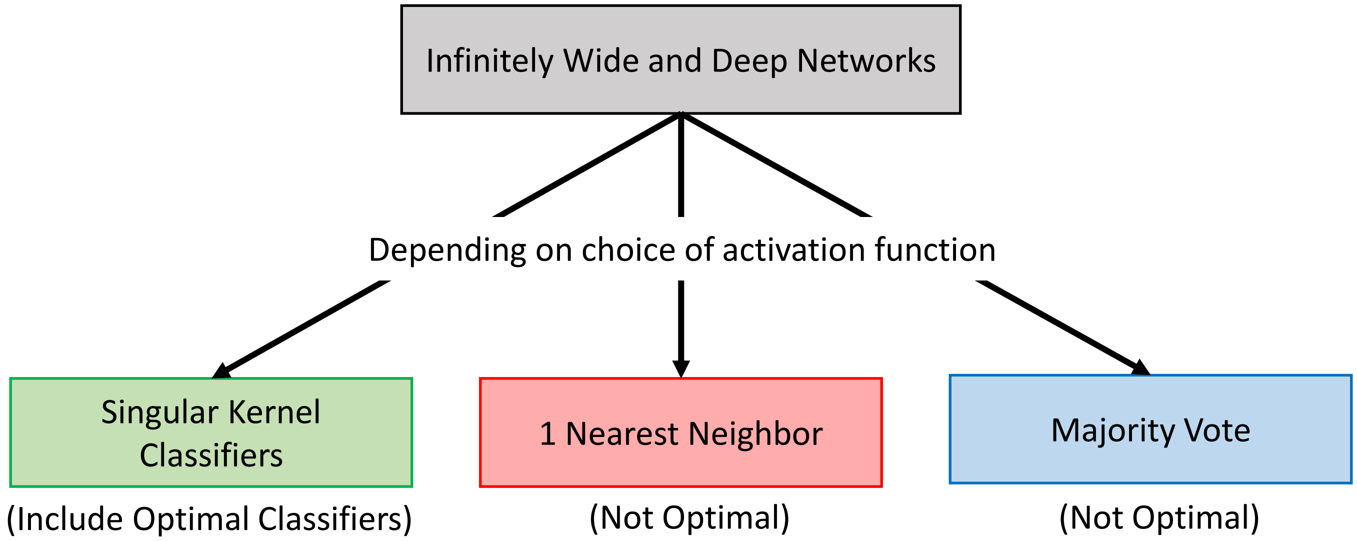
Off-diagonal terms

2. Singular kernels: $K_n = \infty I + G_n$

$$K_n^{-1} = (\infty I + G_n)^{-1} = \frac{1}{\infty}\left(I + \frac{G_n}{\infty}\right)^{-1} = \frac{1}{\infty}\left(I - \frac{1}{\infty} G_n\right) = \frac{1}{\infty} I$$

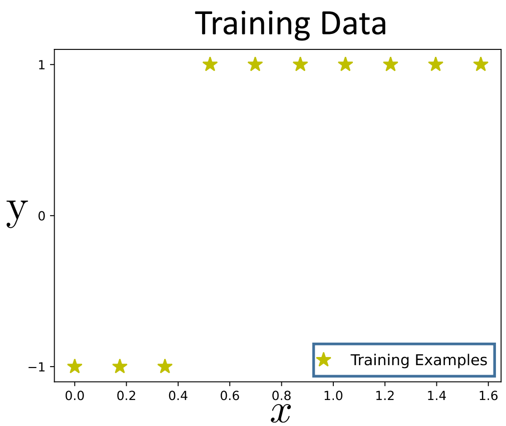
Wide and deep networks

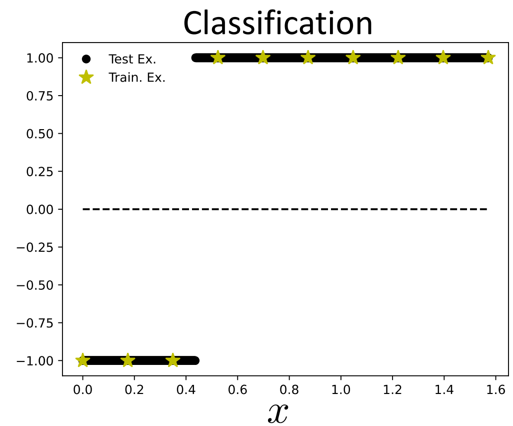
- A taxonomy of infinitely wide and deep classifiers (on a sphere) based on activation function.
- Construct infinitely wide and deep FC networks that, when trained using standard methods, achieve optimality for classification.

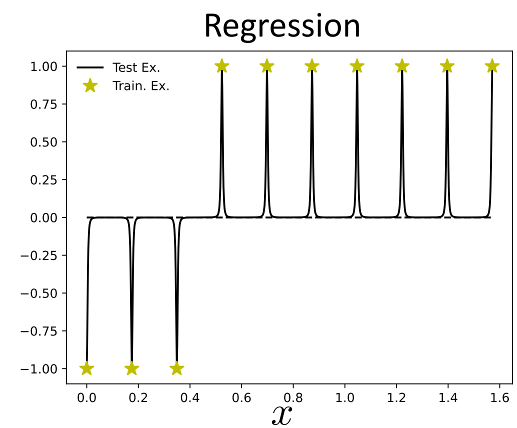


[A. Radhakrishnan, B., C. Uhler, 2022]

Regression vs Classification







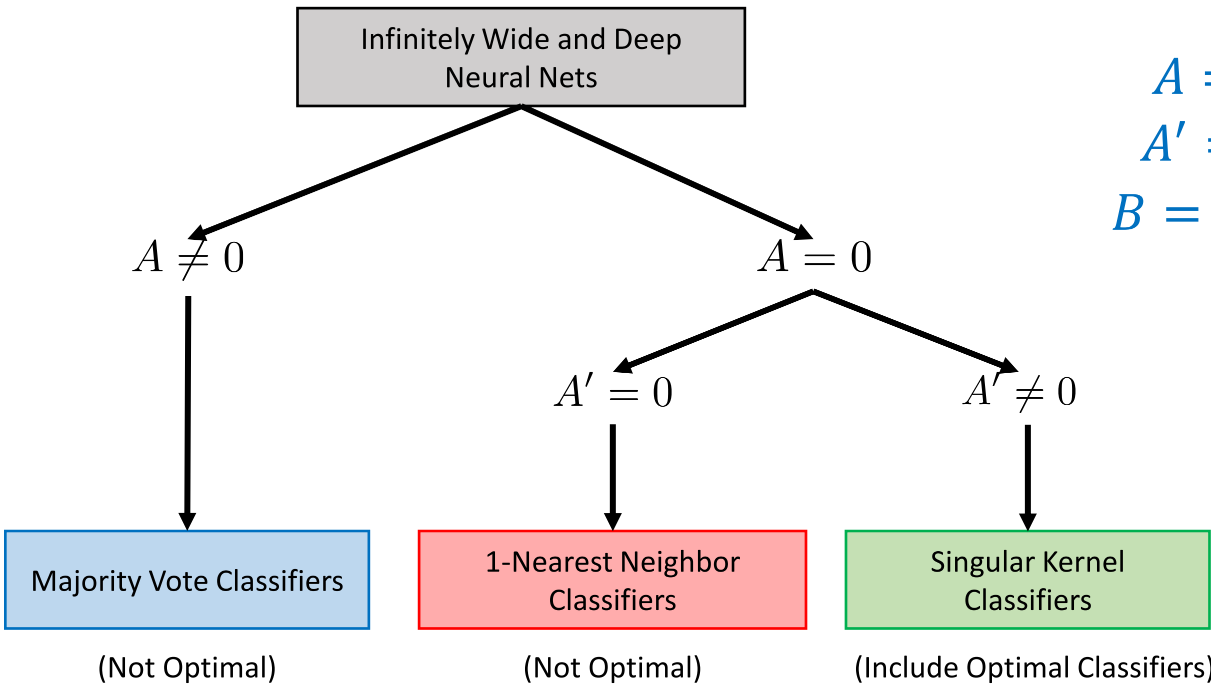
- Converges to delta function for regression but not for classification!

Taxonomy of Infinitely Wide and Deep FC Nets

Depends on activation function ϕ.

$z \sim \mathcal{N}(0,1)$

$$A = \mathbb{E}(\phi(z))$$
$$A' = \mathbb{E}(\phi'(z))$$
$$B = \mathbb{E}([\phi'(z)]^2)$$



Examples:

ReLU	2nd Hermite Polynomial	Cubic Polynomial
$\phi(x) = \max(0, x)$	$\phi(x) = \frac{x^2 - 1}{\sqrt{2}}$	$\phi(x) = \frac{x^3 + (\sqrt{6} - 3)x}{\sqrt{12}}$

Singular Kernel Classifiers

Theorem A. ($A = 0, A' \neq 0$)

$$k^{\infty}(x,z) \sim \left(\frac{R(||x-z||)}{||x-z||^{\alpha}}\right)$$

Infinite depth NTK

$$\alpha = -\frac{4 \ln A'}{\ln B}$$

Consequence (using Devroye, Györfi, and Krzyzak (1998)):

If $\alpha = d$, infinite depth neural net predictor h^{∞} is Bayes optimal.

1-NN and Majority vote

Theorem B. ($A = 0 = A' = 0$)

$$sign(h^{\infty}) = 1 - NN(x)$$

Theorem C*. ($A \neq 0$)

$$sign(h^{\infty}) = \text{majority vote}\ (y_1, \dots, y_n\ )$$

What is going on?

Kernels for deep FC networks (on the unit sphere). Put $v = \langle x, z \rangle$

$$k^0(v) = v$$

$$k^l(v) = \underbrace{\psi(\ldots(\psi(v)\ldots)}_{l} + k^{l-1}(v)\psi'\underbrace{(\psi(\ldots(\psi(v)\ldots)}_{l-1}$$

$$\psi(v) = \mathbb{E}_{(u,t)\sim\mathcal{N}(0,\Lambda(v))}\,\phi(u)\phi(t), \qquad \Lambda = \begin{pmatrix} 1 & v \\ v & 1 \end{pmatrix}$$

Dual activation function.

[Daniely, Frostig, Singer, 16], [Jacot, Gabriel, Hogler, 18]

First key observation

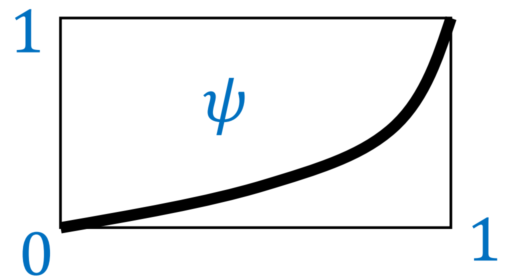
$$k^l(v) = \underbrace{\psi(\dots(\psi(v)\dots)}_{l} + k^{l-1}(v)\psi'(\psi(\dots(\psi(v)\dots)$$

$$\psi^l(v) = \psi(\dots(\psi(v)\dots)$$

Kernels $k^l(\langle x, z\rangle)$ and $\psi^l(\langle x, z\rangle)$ (after normalization) have poles of the same order at $v = 1$ as $l \to \infty$.

Iteration

ψ: $[0,1] \to [0,1]$ is convex and monotone.

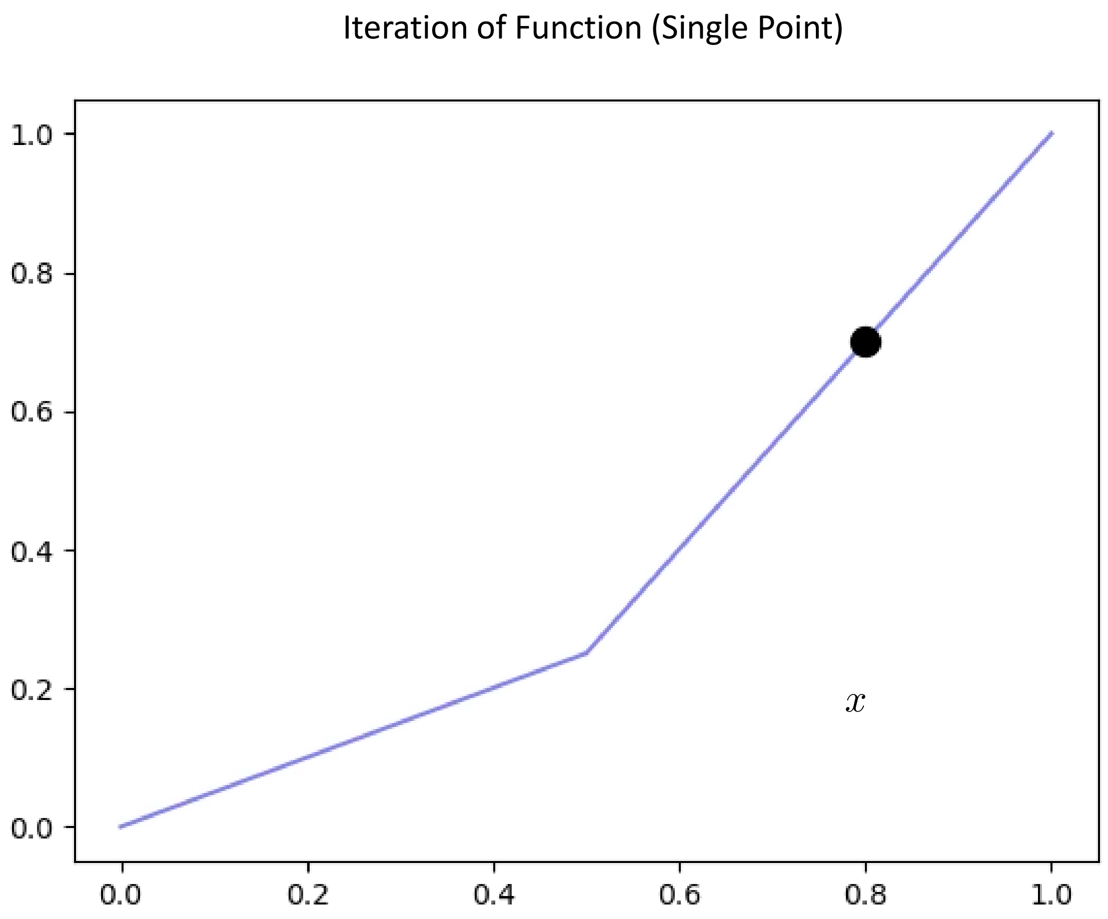


Two fixed points (case 1, sing. kernel):

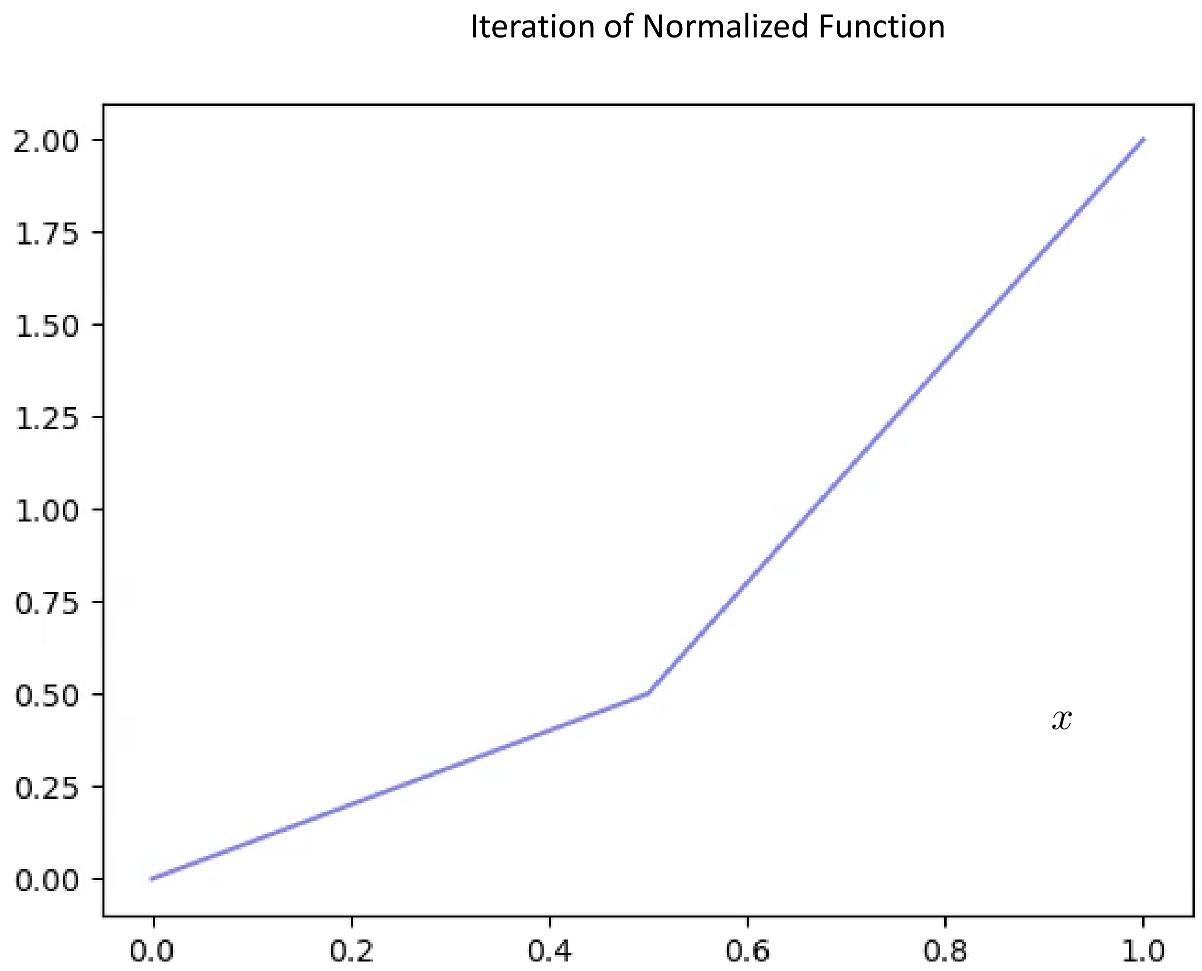
$\mathrm{A} = \psi(0) = 0,\ \mathrm{A}' = \psi(1) = 1$

What does $\psi^l(v) = \psi(\ldots(\psi(v)\ldots)$ look like?

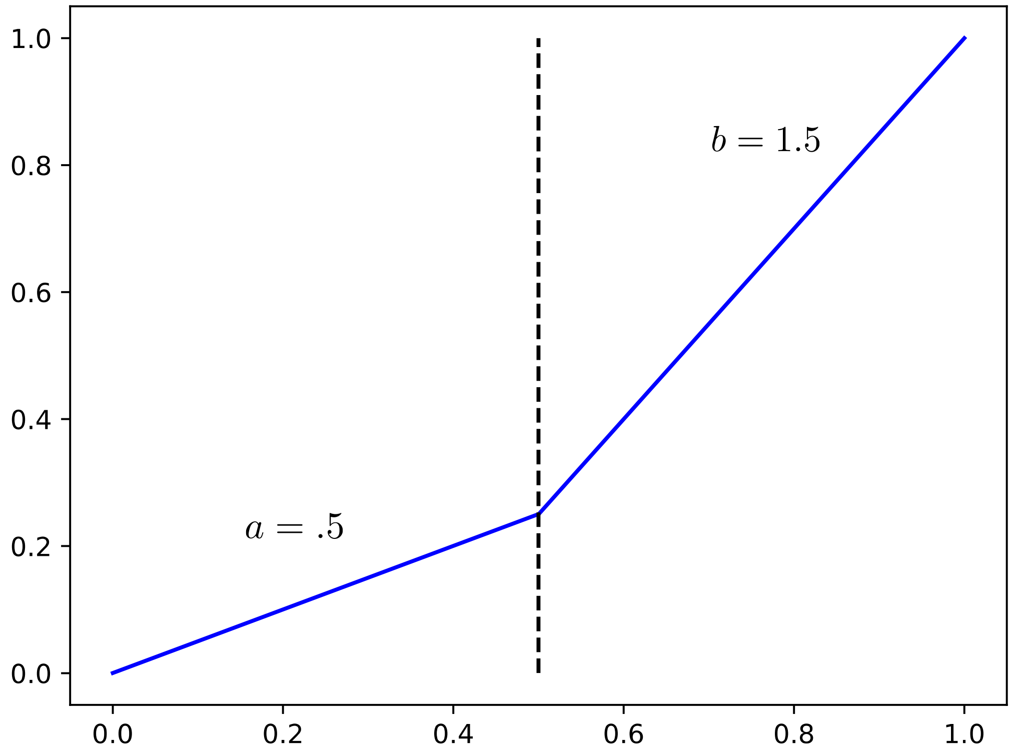
Piecewise linear function



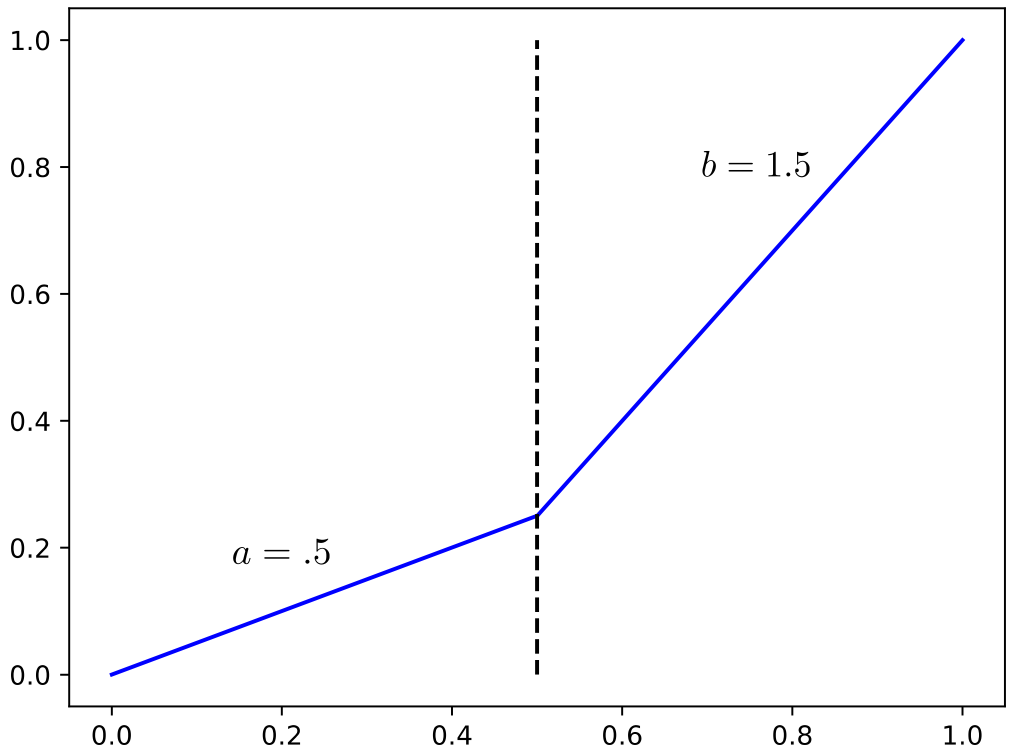
Iteration on the unit interval



Iteration



$\psi^{\infty}(v) = \lim_{l \to \infty} \frac{\psi^{l}(v)}{a^{l}}$ is well-defined and has a pole at 1.

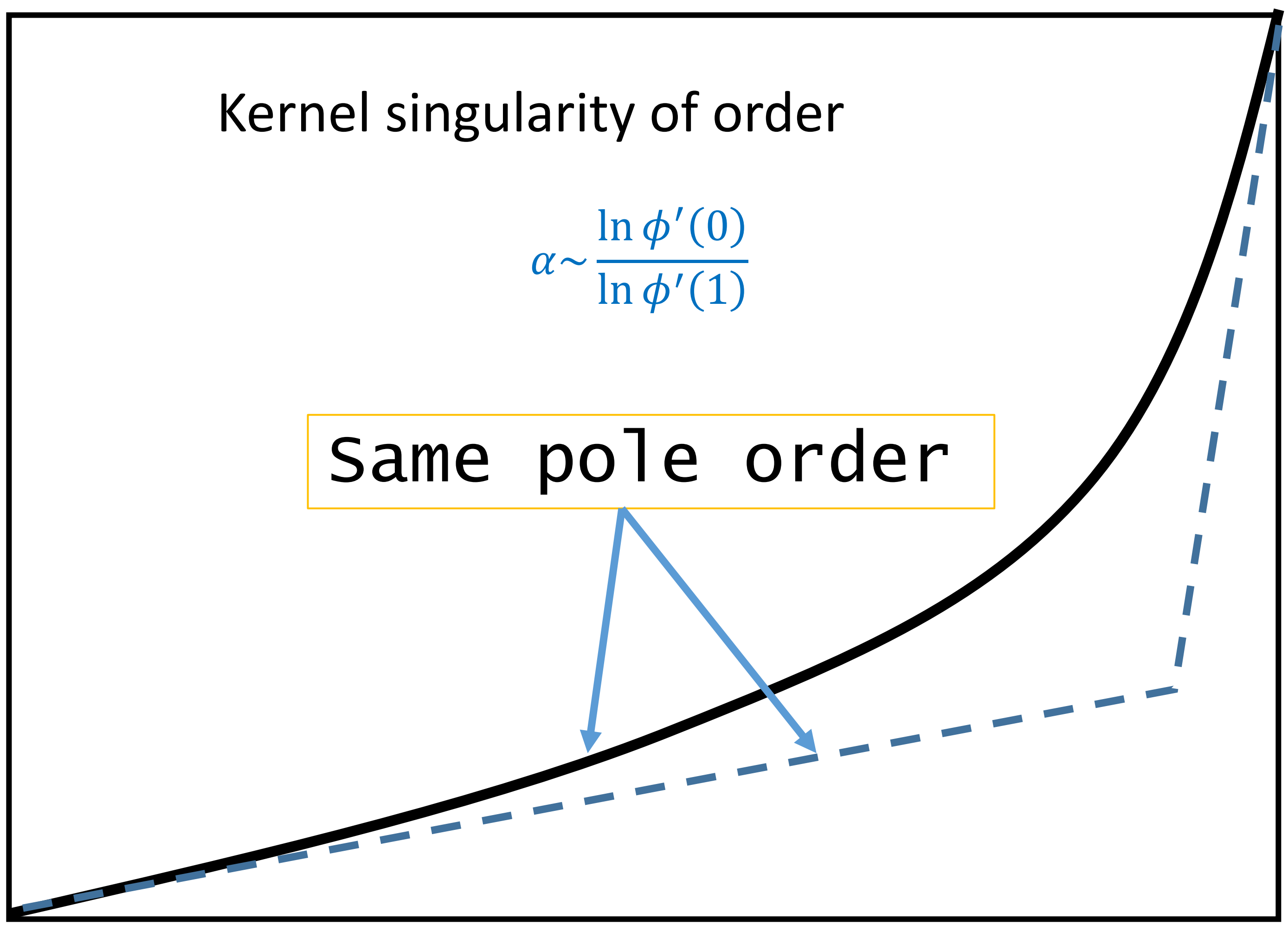


What matters are the slopes.

Kernel singularity of order

$$\alpha \sim -\frac{\ln b}{\ln a}$$

Key observation



Summary

Remarkable properties of neural networks:

- Wide neural networks = kernel machines.

- Deep neural networks = kernel machines with singular kernels = NW schemes for classification.

- Clear separation between regression and classification.