Generalization Properties of Deep Neural Networks Through The Prism of Interpolation

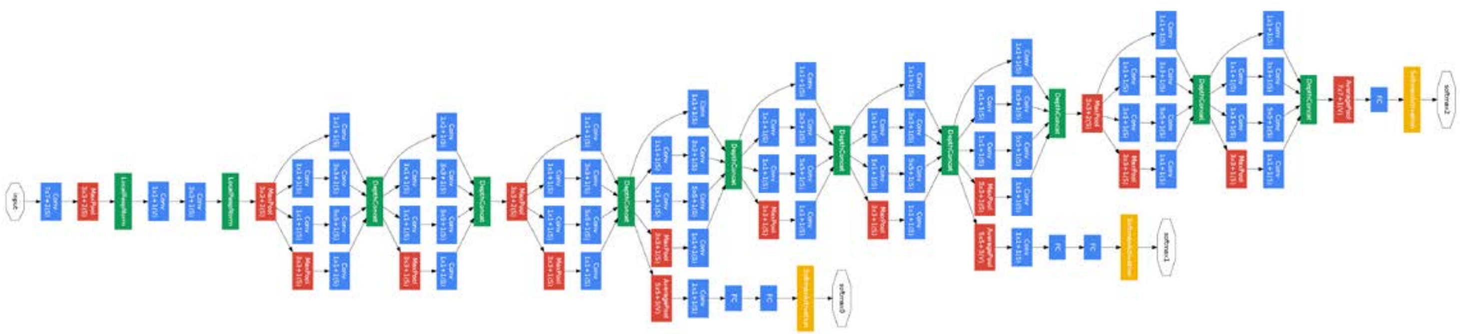
Mikhail Belkin
University of California San Diego,
Halıcıoğlu Data Science Institute

Mode Workshop, Sept 2021

Based on

Fit without fear: remarkable mathematical phenomena of deep learning through the prism of interpolation

(Acta Numerica 2021, arxiv: 2105.14368)

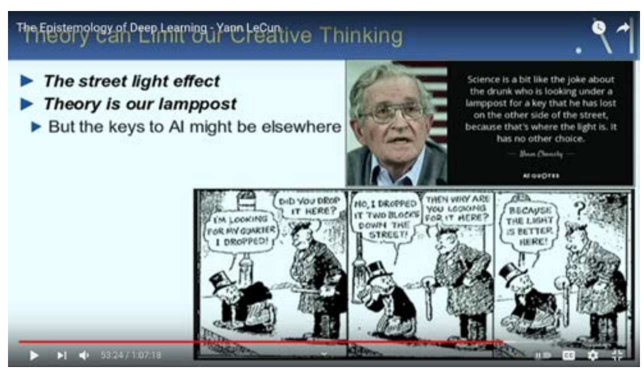


GoogLeNet, Szegedy, et al 2014.

Crisis of ML theory

“Machine learning has become alchemy” (A. Rahimi, B. Recht, NIPS 2017). https://youtu.be/x7psGHgatGM?t=722

ML theory “looking for lost keys under a lamp post, because that's where the light is" (Y. Lecun, 2018).
https://youtu.be/gG5NCkMerHU?t=3189



Yann Lecun:

IPAM talk, 2018

Deep learning breaks some basic rules of statistics.

Written in 1995

Leo Breiman
Statistics Department, University of California, Berkeley, CA 94305;
e-mail: leo@stat.berkeley.edu

Reflections After Refereeing Papers for NIPS

For instance, there are many important questions regarding neural networks which are largely unanswered. There seem to be conflicting stories regarding the following issues:

- Why don't heavily parameterized neural networks overfit the data?
- What is the effective number of parameters?
- Why doesn't backpropagation head for a poor local minima?
- When should one stop the backpropagation and use the current parameters?

Two key questions:

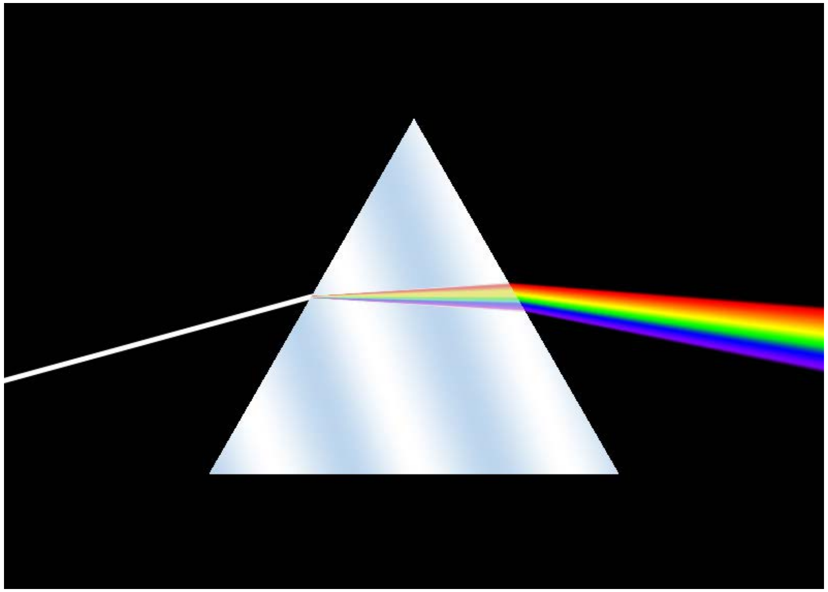
1. Generalization.

Why do neural networks generalize to unseen data?

2. Optimization.

Why can non-convex objective functions be optimized?

The Prism



"*destroyed all the poetry of the rainbow, by reducing it to the prismatic colours.*"' J. Keats

A prism allows analysis by separating a complex mixture of colors into simpler individual components.

The problem of generalization

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 algorithms (including neural networks) and theoretical analyses for ML are based on ERM:

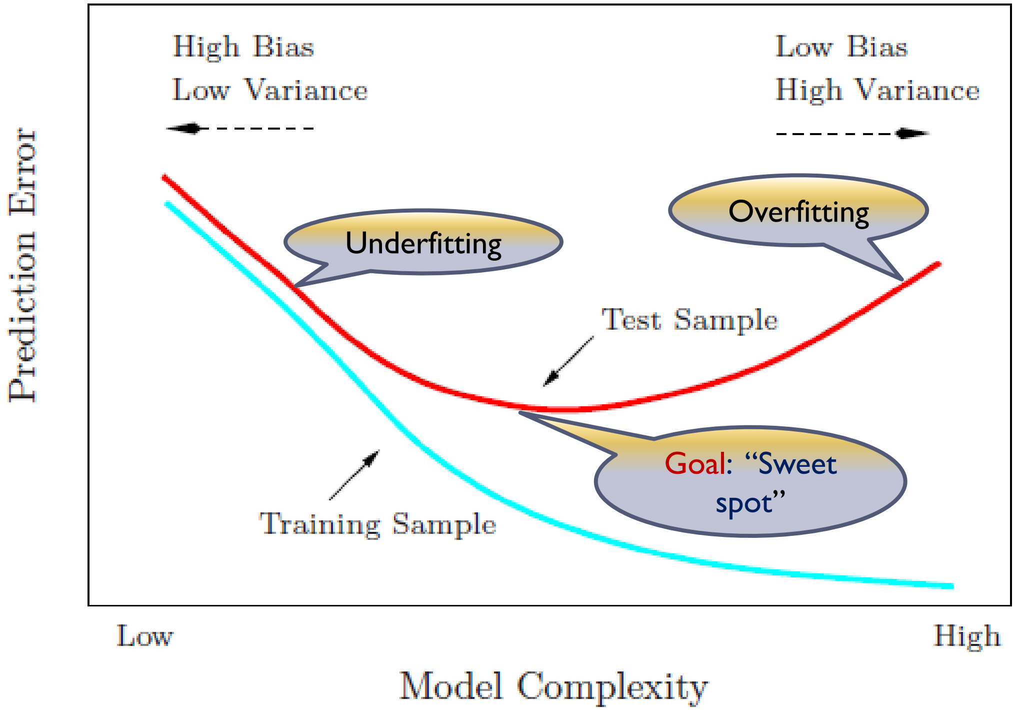
Empirical risk

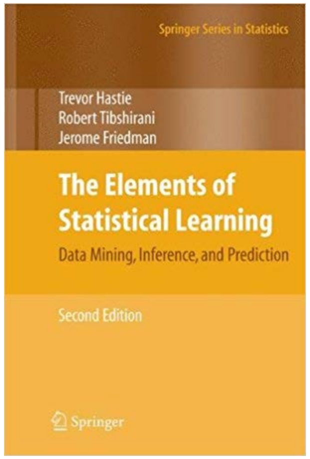
$$f^*_{ERM} = \underset{f \in \mathcal{H}}{argmin} \; \frac{1}{n} \sum_{training\ data} L(f(x_i), y_i)$$

Minimize empirical risk over a class of functions $\mathcal{H}$.

Key question – choice of $\mathcal{H}$.

Classical U-shaped generalization curve





The ERM/SRM theory of learning

Goal of ML: $f^* = \arg\min_{f} E_{unseen\ data}\, L(f(x), y)$

Goal of ERM: $f^*_{ERM} = \arg\min_{f_w \in \mathcal{H}} \frac{1}{n} \sum_{training\ data} L(f_w(x_i), y_i)$

1. *The theory of induction is based on the* *uniform law of large numbers.*
2. *Effective methods of inference must include* *capacity control.*

V. Vapnik, Statistical Learning Theory, 1998

…

Uniform law of large numbers

Empirical loss of any $f \in \mathcal{H}$ approximates expected loss of f.

$$\mathcal{L}_{emp}(f) = \frac{1}{n}\sum\nolimits_{training\ data} L\big(f_w(x_i), y_i\big) \approx E_{unseen\ data}\, L\big(f(x), y\big)$$

Hence

$$\mathcal{L}_{emp}(f^*_{ERM}) \approx E_{unseen\ data}\, L(f^*_{ERM}(x), y)$$

WYSIWYG Generalization bounds

WYSIWYG bounds VC-dim, fat shattering, Rademacher, covering numbers, margin…

Classically VC-dimension

Expected risk: what you get

Empirical risk: what you see

$$E\left(L\left(f^*_{ERM}, y\right)\right) \leq \frac{1}{n}\sum L\left(f^*_{ERM}(x_i), y_i\right) + O^*\left(\sqrt{\frac{c}{n}}\right)$$

Capacity control

...

6.1 THE SCHEME OF THE STRUCTURAL RISK MINIMIZATION INDUCTION PRINCIPLE 223

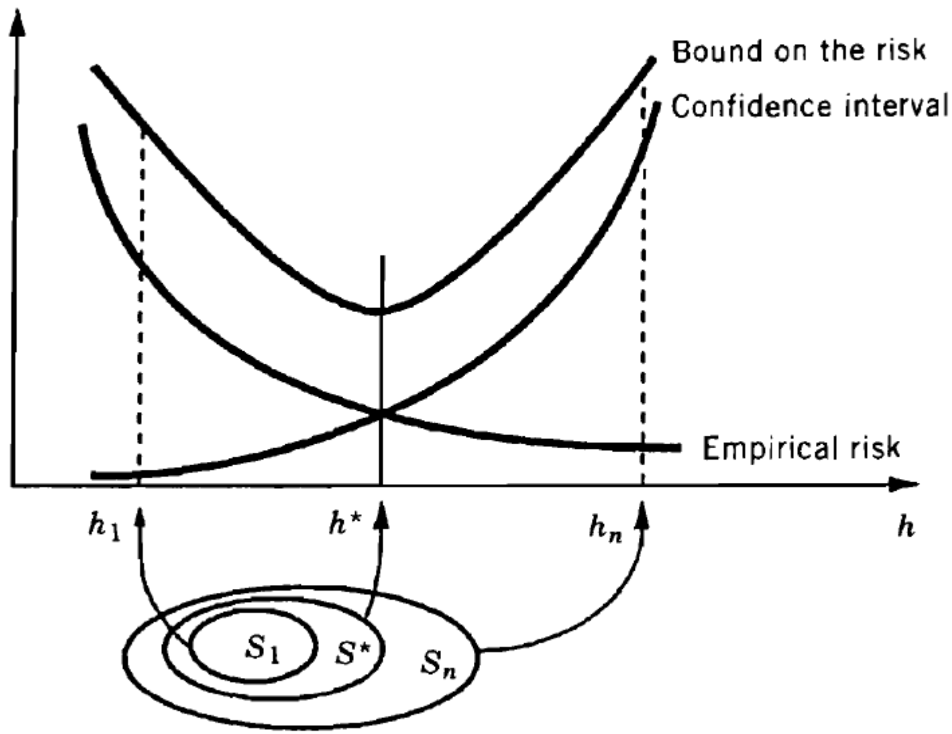


FIGURE 6.2. The bound on the risk is the sum of the empirical risk and of the confidence interval. The empirical risk is decreased with the index of element of the structure, while the confidence interval is increased. The smallest bound of the risk is achieved on some appropriate element of the structure.

V. Vapnik, **Statistical Learning Theory**, 1998

Data-dependent WYSIWYG bounds

Why do we need uniform laws of large numbers, when most $f \in \mathcal{H}$ are useless for prediction?

$$E(L(f^*_{ERM}, y)) \leq \frac{1}{n} \sum L\left(f^*_{ERM}(x_i), y_i\right) + O^*\left(\sqrt{\frac{c(X)}{n}}\right)$$

Margin and other "a posteriori" bounds allow $\mathcal{H}$ and c to be data-dependent.

Interpolation

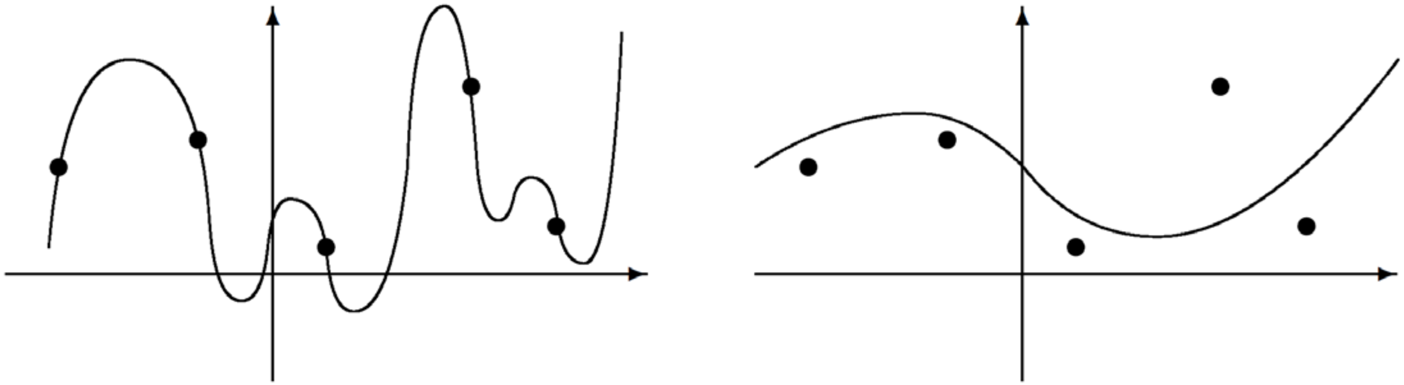
f interpolates if $\forall_i \; f(x_i) = y_i$

Test loss $\qquad$ Training loss

$$\underbrace{E\Big(L\big(f(x),y\big)\Big)}_{\neq 0} \leq \cancel{\underbrace{\frac{1}{n}\sum L\left(f(x_i),y_i\right)}_{= 0}} + O^*\left(\sqrt{\frac{c}{n}}\right)$$

WYSIWIG bounds imply interpolation should not generalize.

Does interpolation overfit?



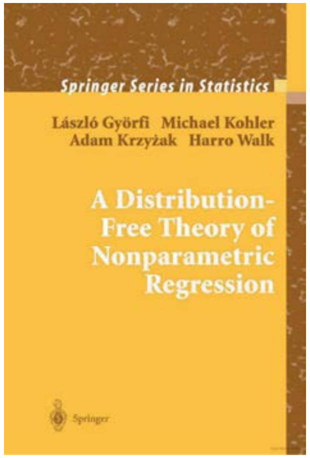
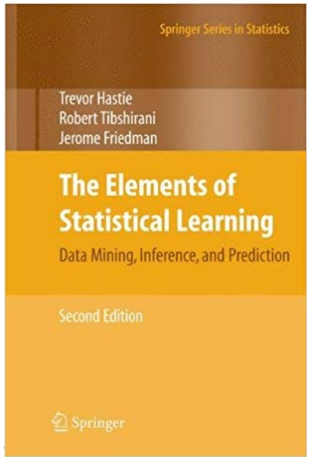


Figure 2.3. The estimate on the right seems to be more reasonable than the estimate on the left, which interpolates the data.



However, a model with zero training error is overfit to the training data and will typically generalize poorly.

Does interpolation overfit?

model	# params	random crop	weight decay	train accuracy	test accuracy
Inception	1,649,402	yes	yes	100.0	89.05
		yes	no	100.0	89.31
		no	yes	100.0	86.03
		no	no	100.0	85.75

[CIFAR 10, from *Understanding deep learning requires rethinking generalization*, Zhang, et al, 2017]

Boosting the margin:
A new explanation for the effectiveness of voting methods

1998

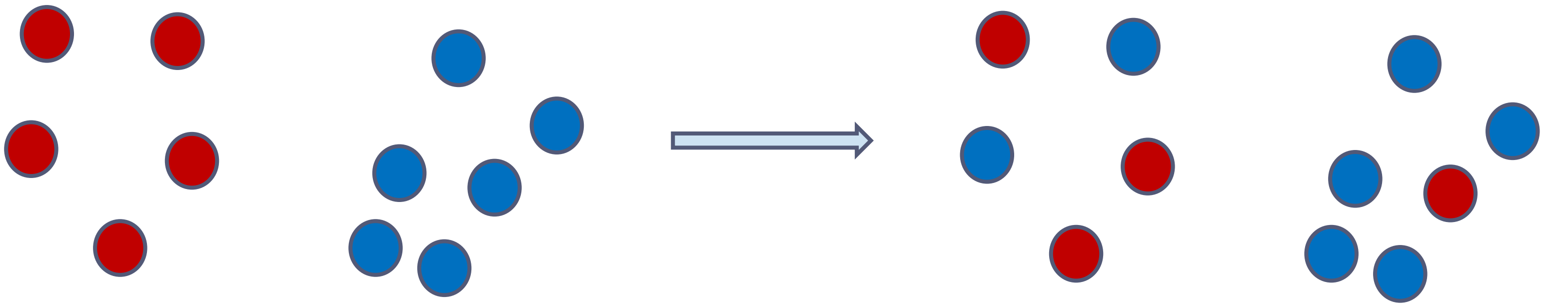
Robert E. Schapire **Yoav Freund** **Peter Bartlett** **Wee Sun Lee**

Abstract. One of the surprising recurring phenomena observed in experiments with boosting is that the test error of the generated hypothesis usually does not increase as its size becomes very large, and often is observed to decrease even after the training error reaches zero. In this paper, we

Suggestive, yet does not directly invalidate WYSIWYG bounds.

How to test model complexity?

Add label noise.

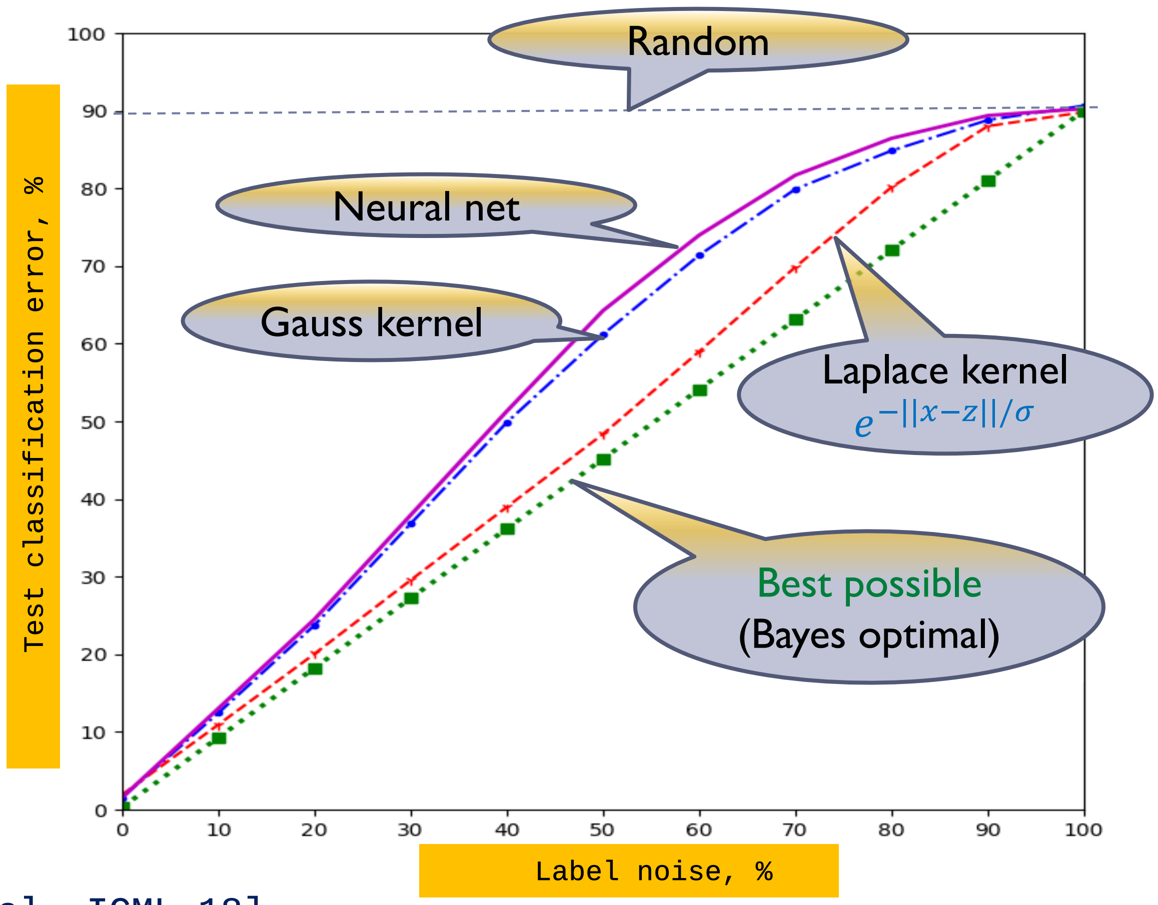


Model complexity grows necessary to fit data grows, but Bayes opt. does not change!

Expect overfitting to become severe as model complexity grows.

Interpolation does not overfit even for very noisy data

All methods (except Bayes optimal) have zero training square loss.

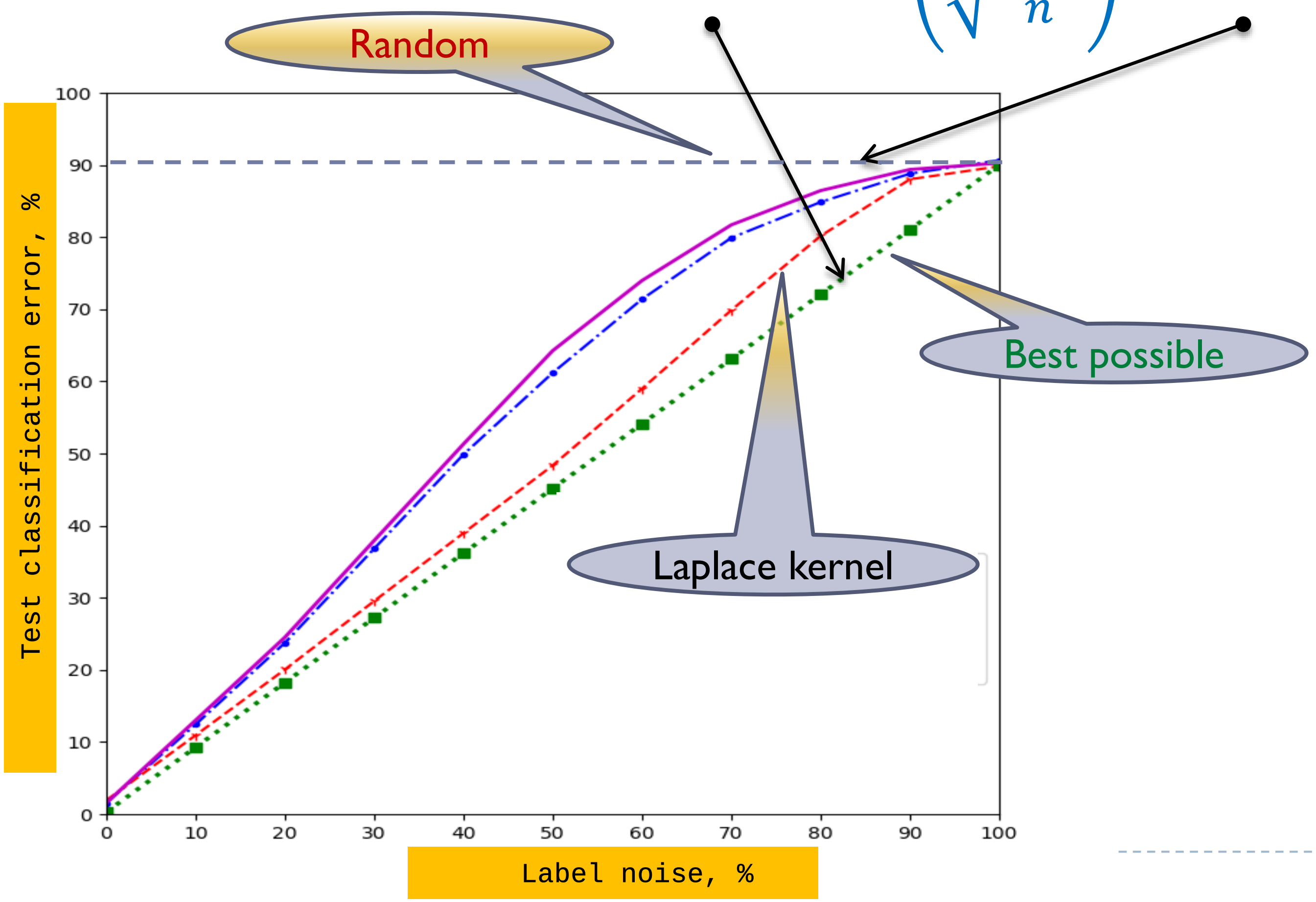


[B., Ma, Mandal, ICML 18]

Bounds?

What kind of generalization bound could work here?

$$0.7 < O^*\left(\sqrt{\frac{c(n)}{n}}\right) < 0.9$$



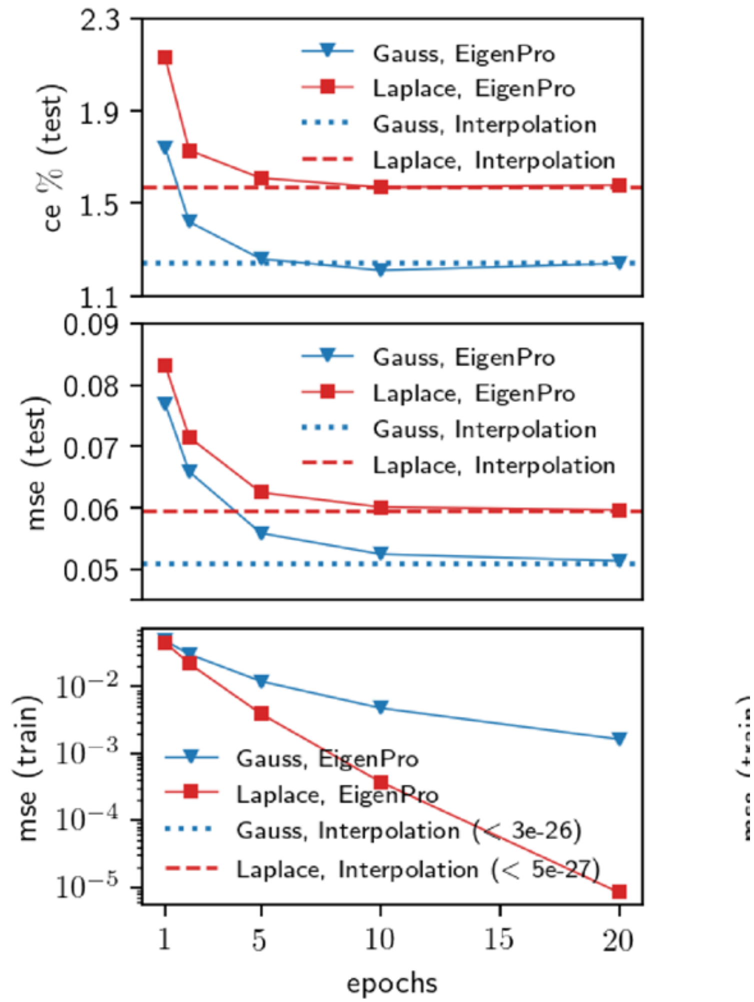
Why bounds fail

$$\overset{\texttt{correct}}{0.7} < O^*\left(\sqrt{\frac{c(n)}{n}}\right) < \overset{\texttt{useful}}{0.9} \qquad n \to \infty$$

1. The constant in O^* needs to be exact. There are no bounds like that.

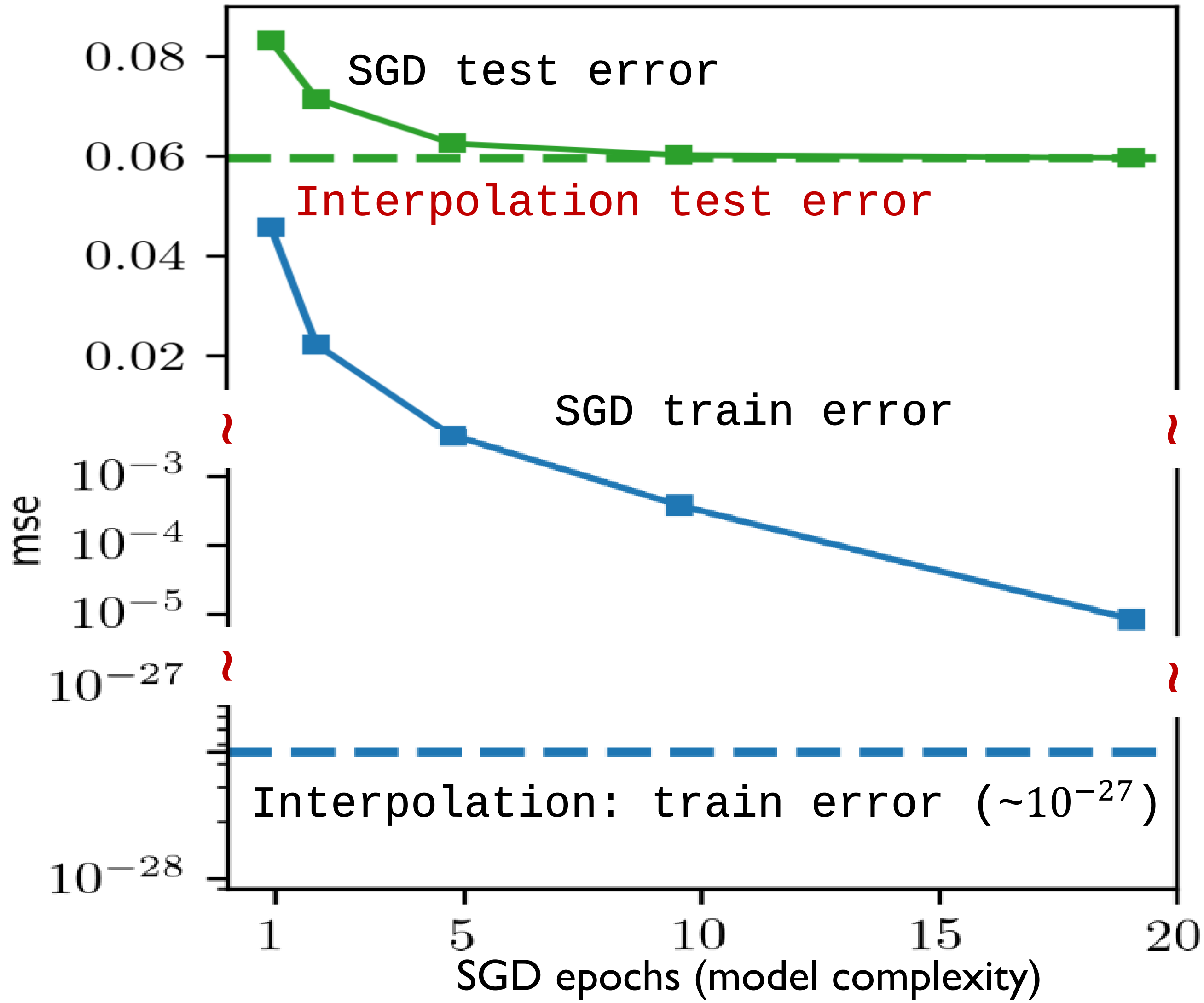
2. Conceptually, how would the quantity $c(n)$ "know" about the Bayes risk?

Recent work: [Nagarajan, Kolter, 19; Bartlett, Long 20]



2.3
1.9
1.5
1.1
ce % (test)
Gauss, EigenPro
Laplace, EigenPro
Gauss, Interpolation
Laplace, Interpolation
0.09
0.08
0.07
0.06
0.05
mse (test)
Gauss, EigenPro
Laplace, EigenPro
Gauss, Interpolation
Laplace, Interpolation
10^{-2}
10^{-3}
10^{-4}
10^{-5}
mse (train)
Gauss, EigenPro
Laplace, EigenPro
Gauss, Interpolation (< 3e-26)
Laplace, Interpolation (< 5e-27)
1
5
10
15
20
epochs

(a) MNIST



SGD test error
Interpolation test error
SGD train error
Interpolation: train error (~10^{-27})
mse
0.08
0.06
0.04
0.02
10^{-3}
10^{-4}
10^{-5}
10^{-27}
10^{-28}
1
5
10
15
20
SGD epochs (model complexity)

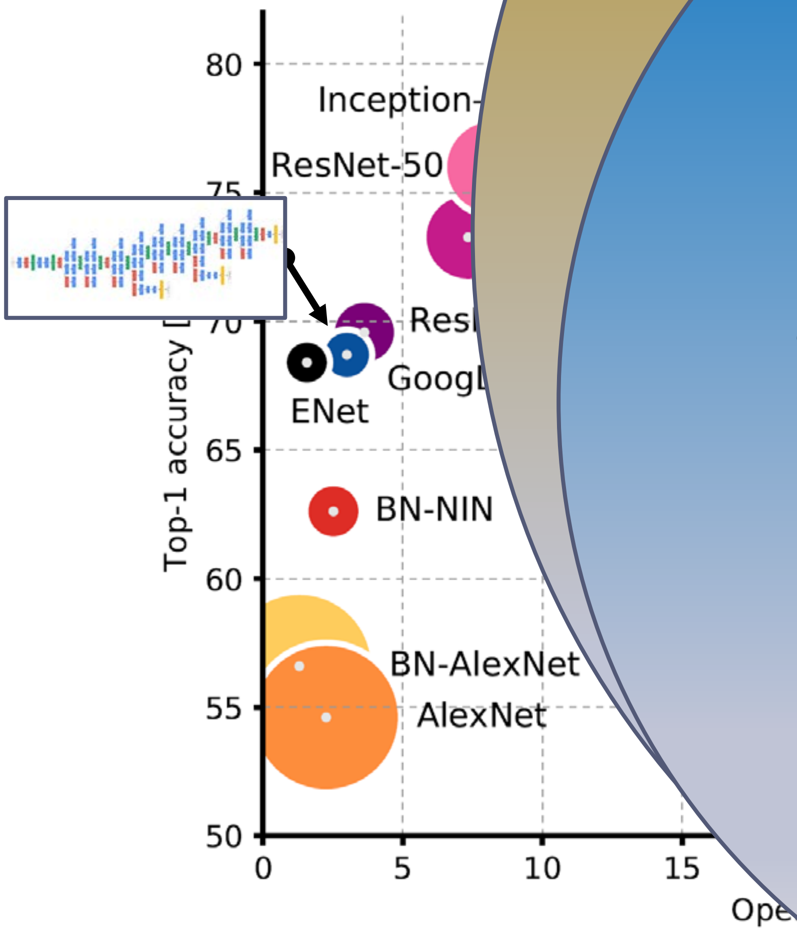
Interpolation is best practice for deep learning

From Ruslan Salakhutdinov's tutorial (Simons Institute, 2017):

The best way to solve the problem from practical standpoint is you build a very big system … basically you want to make sure you hit the zero training error.

Further tuning is needed for state-of-the-art results, but already works well at this point.

La



From Canziani, et al., 2017.

Switch Transformer, 2021:
1.6 trillion parameters

The “puzzle” of generalization

Interpolation does not appear to overfit contrary to ML/statistical beliefs.

Yet the practice of deep learning is arguably closer to interpolation than to classical settings.

New “theory of induction” cannot be based on uniform laws of large numbers with capacity control.

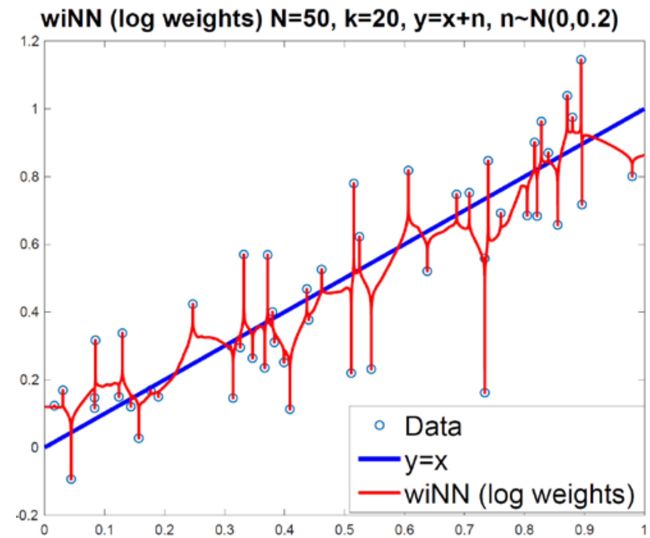
Can interpolation generalize?

Interpolated k-NN schemes

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

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

(cf. Shepard's interpolation)



Theorem:

Weighted (interpolated) k-nn 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]

A curious corollary

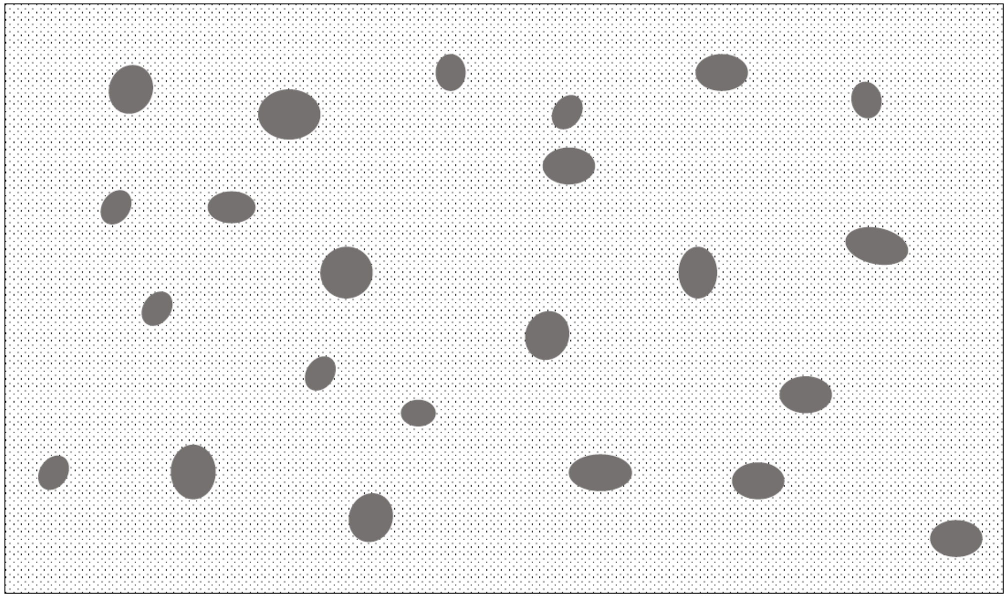
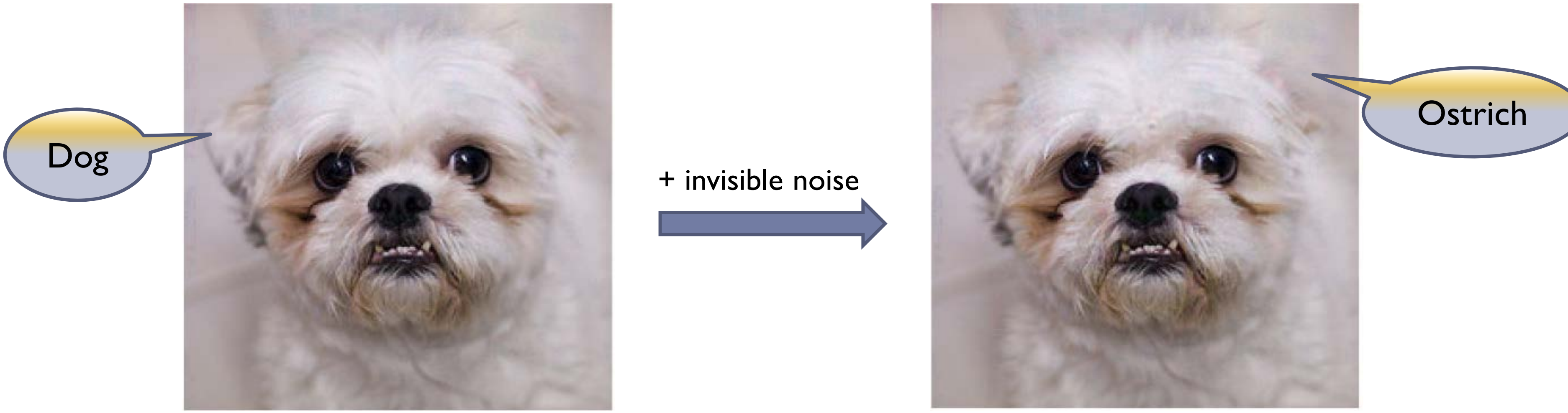


Figure 11: Raisin bread: The "raisins" are basins where the interpolating predictor f_{int} disagrees with the optimal predictor f^*, surrounding "noisy" data points. The union of basins is an everywhere dense set of zero measure (as $n \to \infty$).

Interpolation and adversarial examples



From Szegedy, at al, *Intriguing properties of neural networks*, ICLR 2014

Theorem: adversarial examples for interpolated classifiers are asymptotically dense (assuming the labels are not deterministic).

caveat emptor: possibly only one of the mechanisms.

This talk so far:

A. Interpolation empirically aligns with generalization.
B. Theory of interpolation cannot be based on uniform bounds.
C. Statistical validity of interpolating nearest neighbor methods.

There is a mismatch between A and C.

Methods we analyze have no complexity control/optimization,
Yet practical methods choose the largest technologically feasible models.

Key questions for new theory: dependence of generalization on model complexity.

Parametric families

ReLU Networks $ReLU(x) = \max(x, 0)$,

Neural network with hidden layer of size d:

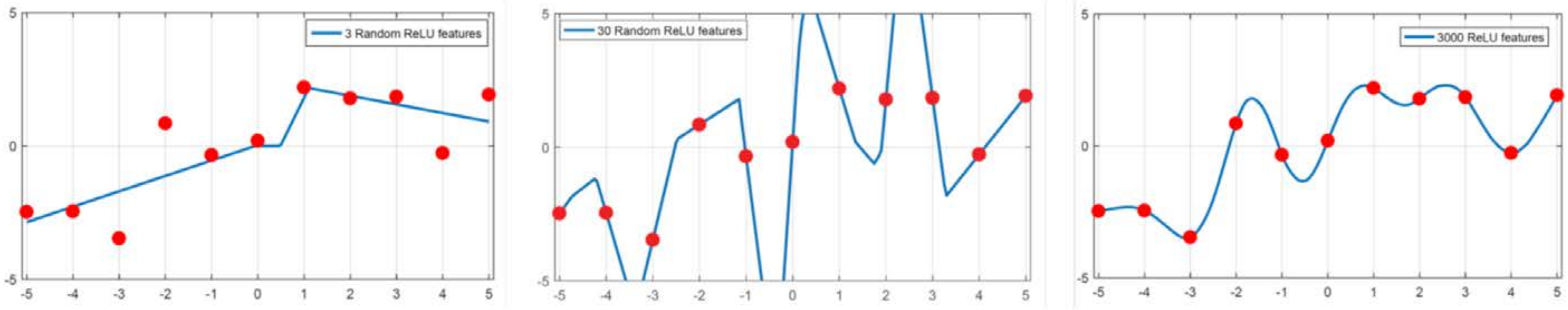
$$h_d(x) = \sum_{j=1}^{d} \alpha_j \, ReLU(b_j x + c_j)$$

Random ReLU features: b_j, c_j fixed chosen at random.

Trained by linear regression over α_j :

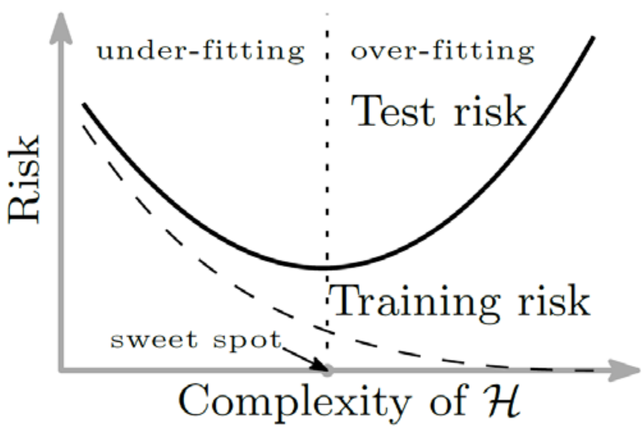
$$h_d^* = arg\min_{\alpha} \sum (h_d(x_i) - y_i)^2$$

Interpolation and over-parameterization

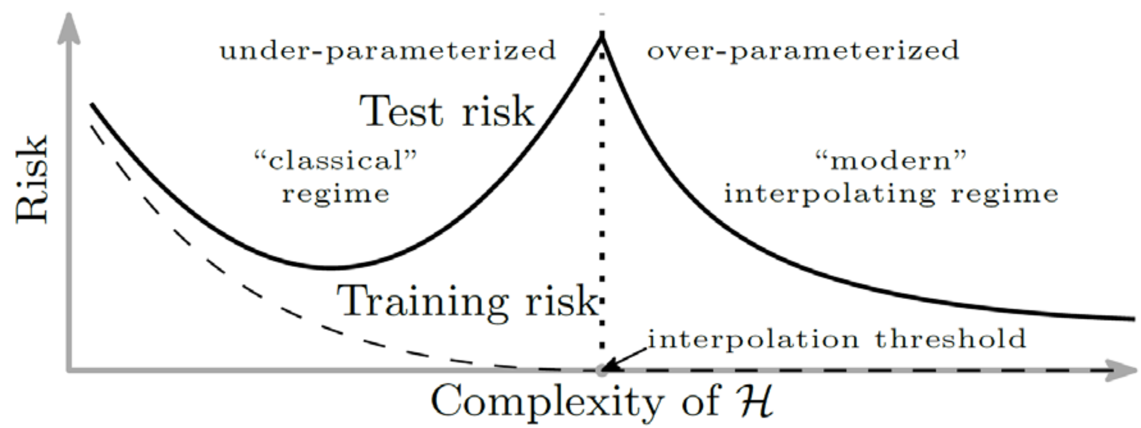


Double descent risk curve

Classical risk curve



New "double descent" risk curve

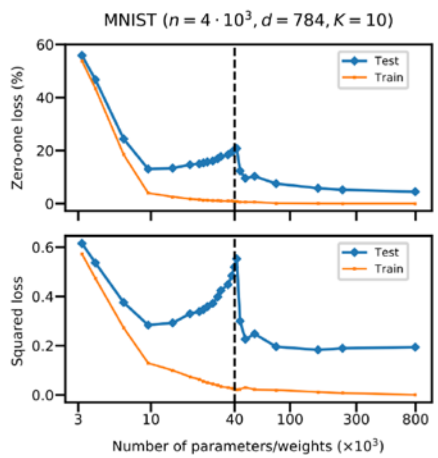


Two key points:

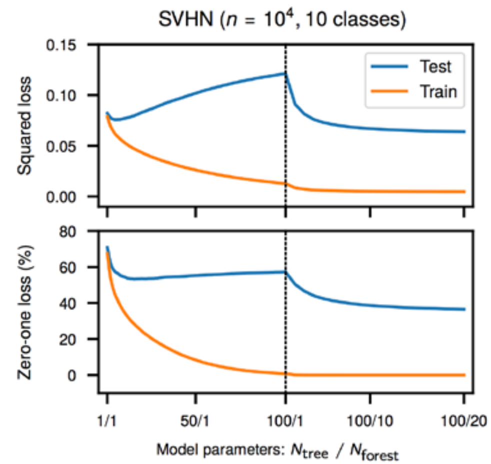
- The classical curve ends where modern ML starts.
- Very complex models can outperform "classical" models

[B., Hsu, Ma, Mandal, PNAS 2019]

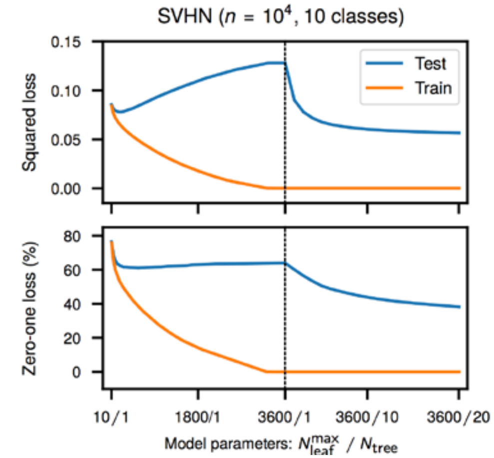
Fully connected network

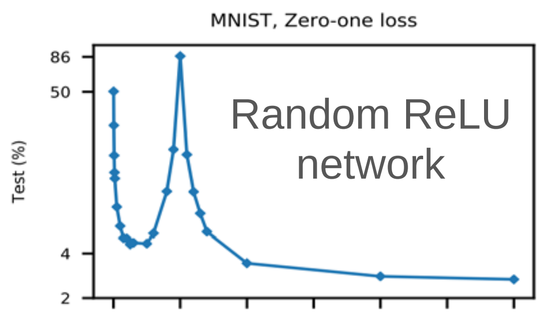


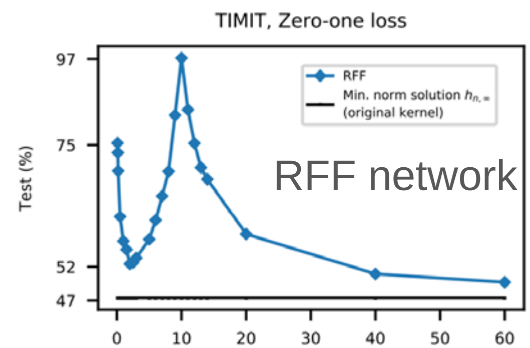
Random Forest

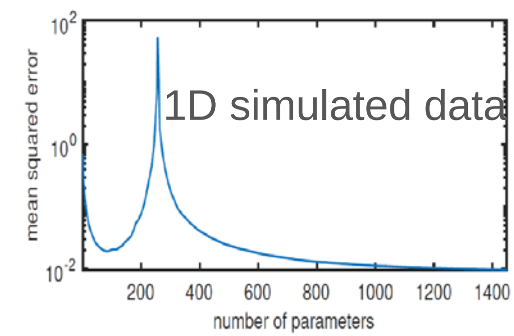


L2-boost

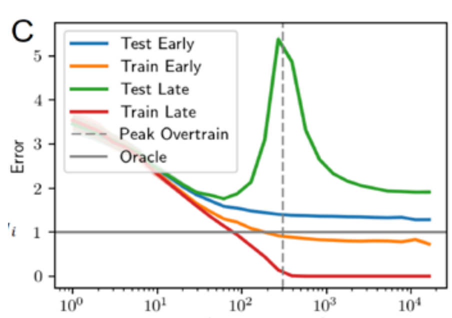




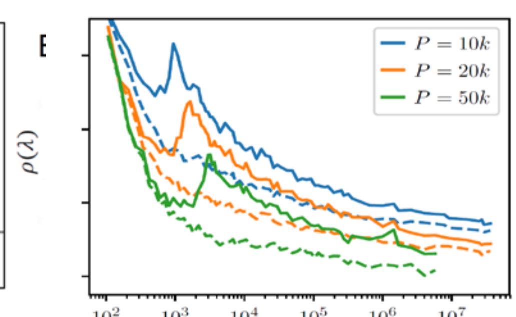




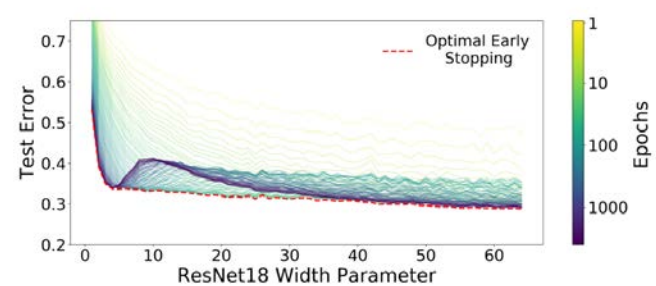
[B., Hsu, Ma, Mandal, 18]



Advani, Saxe, 2017



Spigler, et al, 2018



Nakkiran, et al, ICLR 2020

Double descent in linear/kernel models

Interpolated linear models provide insights for DNN.

Some recent work on generalization in linear/kernel models:

[Bartlett, Long, Lugosi, Tsigler 19],
[Hastie, Montanari, Rosset, Tibshirani 19] [Mitra, 19],
[Muthukumar, Vodrahalli, Sahai, 19] [Mei, Montanari, 19]
[Liang, Rakhlin, 19], [Liang, Rakhlin, Zhai, 19] [Xu, Hsu, 19]

Choosing maximum number of features is provably optimal under the "weak random feature" model. [B., Hsu, Xu, 19].

Deep Neural ReLU networks = Laplace RKHS
[Chen, Xu, 20],[Bietti, Bach 20]

ERM and Interpolation (linear)

Classical ERM:

$$f^*_{ERM} = arg\min_{f \in \mathcal{H}} \frac{1}{n} \sum\nolimits_{training\ data} L(f(x_i), y_i)$$

Modern ML/interpolation:

$$f^*_{int} = arg \min_{\substack{f \in \mathcal{H} \\ \forall_i\, f(x_i) = y_i}} ||f||$$

Norm minimization hidden within the dynamics of SGD.
Looks like ERM superficially.

Framework for modern ML

Occam's razor based on inductive bias:
Maximize smoothness subject to interpolating the data.

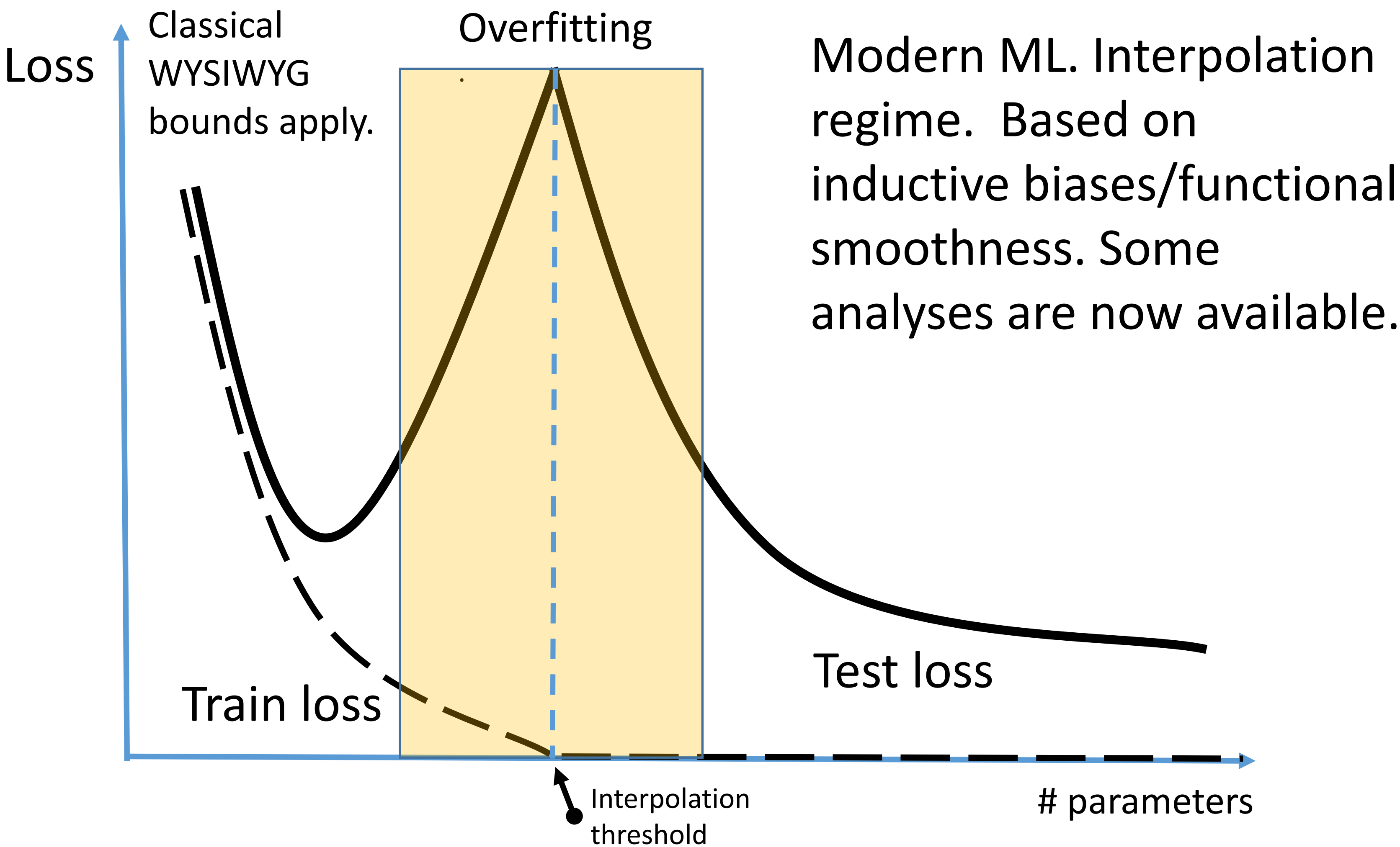
Three ways to increase smoothness:

- Explicit: minimum functional norm solutions
  - Exact: kernel machines.
  - Approximate: RFF, ReLU features.
- Implicit: SGD/optimization (Neural networks)
- Averaging (Bagging, L2-boost).

All coincide for kernel machines.

Interesting recent work: smoothness may require over-parameterization in parametric families [Bubeck, Selke, 21]

The landscape of generalization



Key question

Why is SGD so successful in optimizing highly non-linear neural networks?

Traditional view:

tractable optimization = (local) convexity

Learning as solving a system of equations

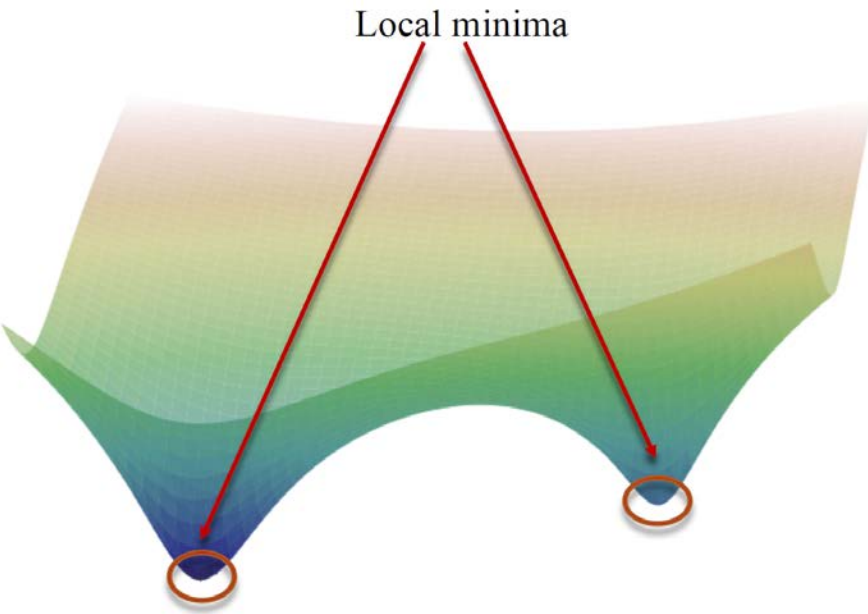
Fitting data = solving a system of non-linear equations $f_w(x_i) \approx y_i$:

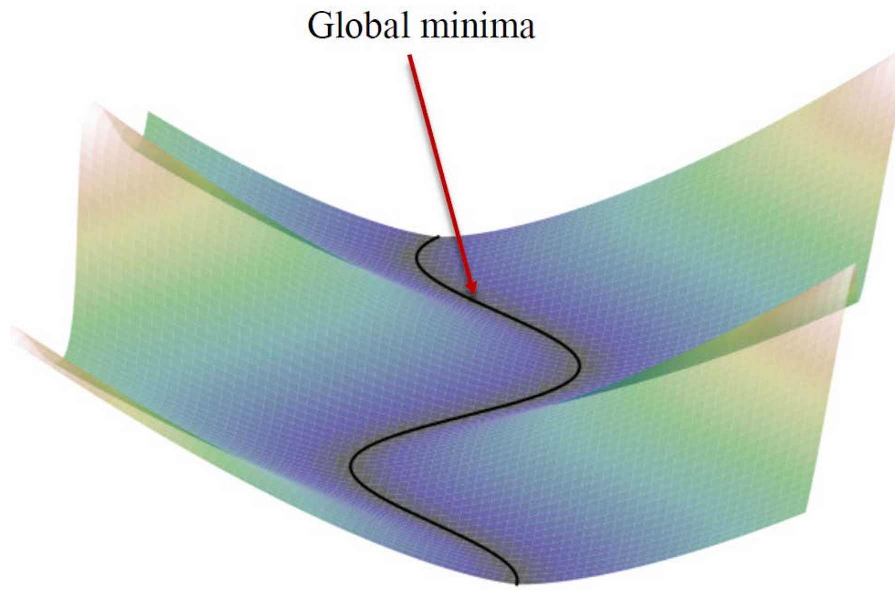
$$F(w) = y, \qquad F: \mathbb{R}^m \to \mathbb{R}^n$$

Equivalent to minimizing (square loss)

$$L(\mathrm{w}) = ||F(w) - y||^2$$

Under and over-parameterization





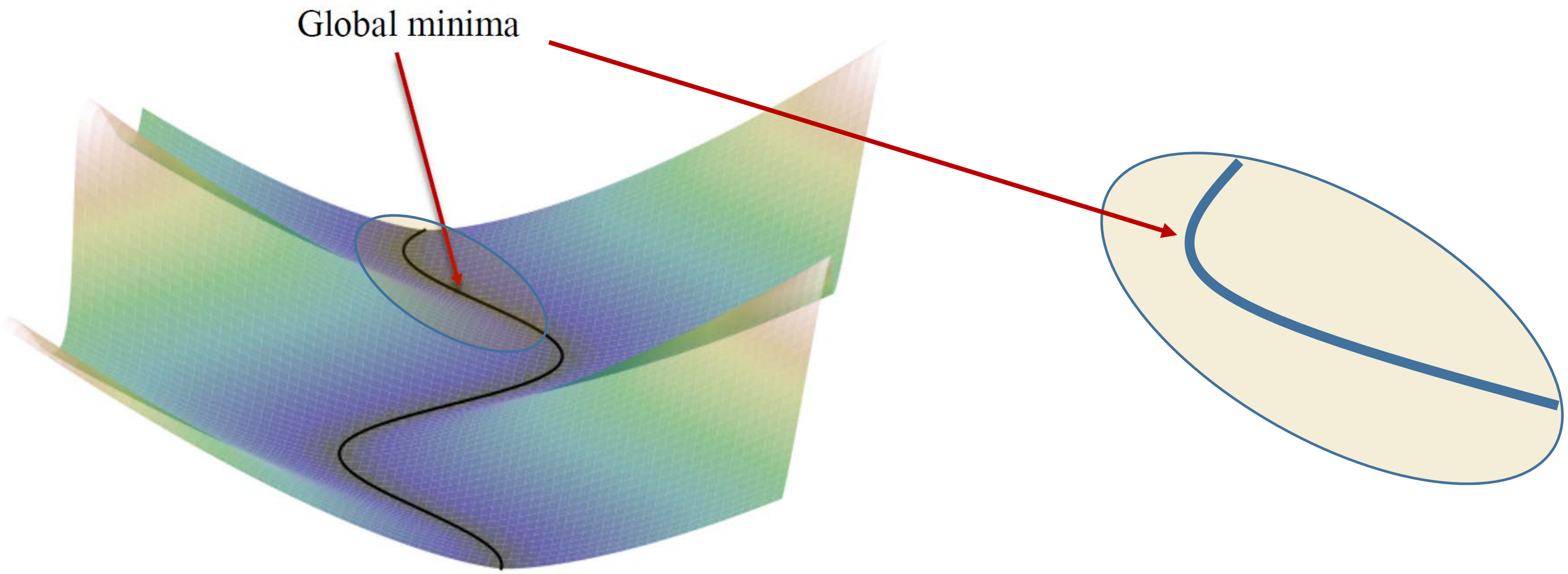
Classical underparameterized landscape $m \leq n$:

Isolated local minima

Overparameterized landscape $m > n$:

Manifolds of global minima

Essential non-convexity



“Theorem”: Landscapes of over-parameterized systems are never convex, even locally.

Proof: If $L(\mathrm{w})$ is locally convex, the manifold of minima cannot have curvature (must be a line segment).

Theory of optimization for over-parameterized systems cannot be based on (local) convexity.

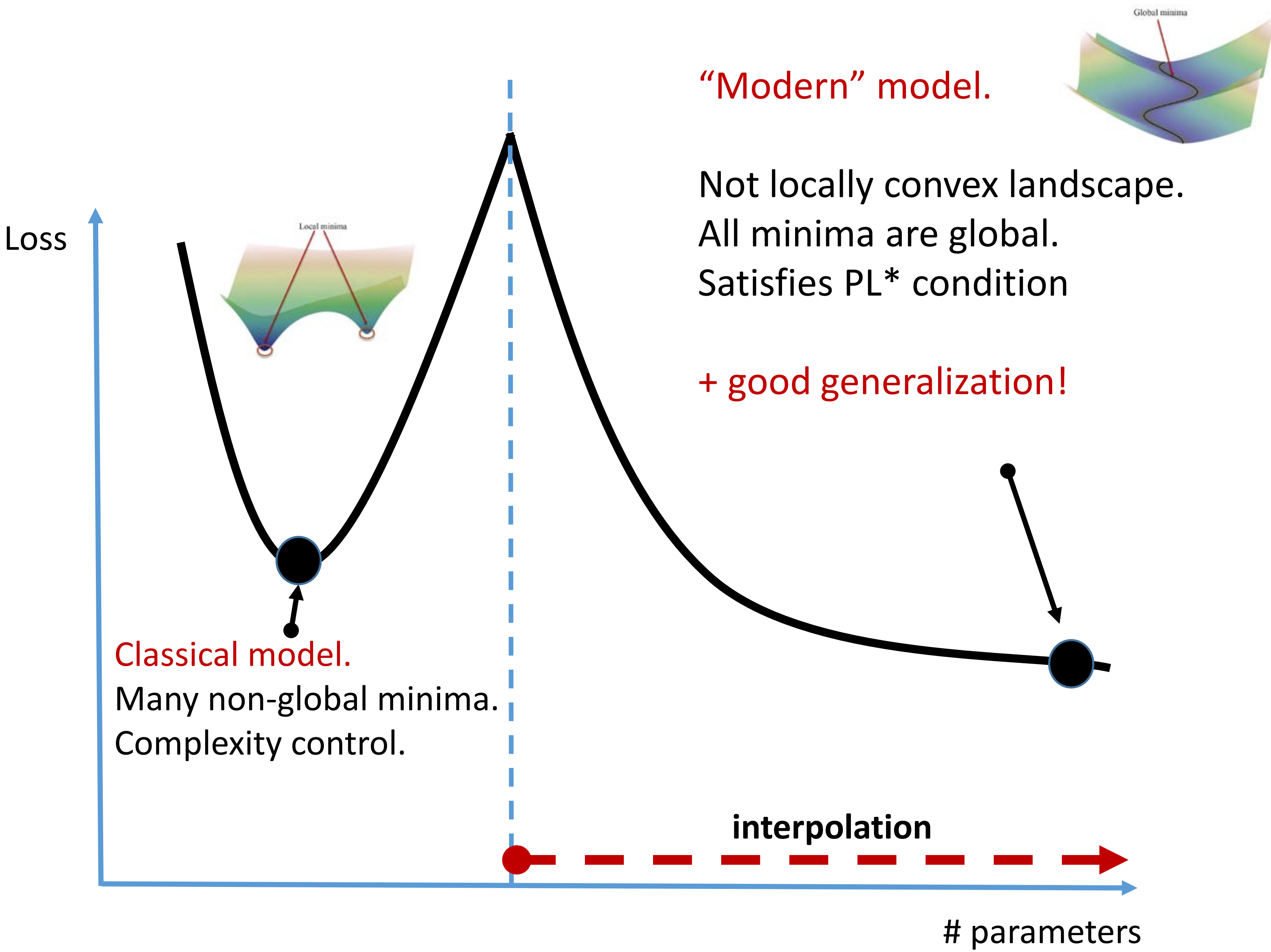
From convexity to PL

Polyak-Lojasiewicz (PL) condition (1963)

$$||\nabla L(w)||^2 \geq \mu\,(L(w) - L(w^*))$$

+ First order.
+ Guarantees convergence of GD.
+ Invariant under “nice” transformations of w.

Modern and classical models



Collaborators:

Chaoyue Liu, Ohio State University-> Facebook

Siyuan Ma, OSU -> Google

Soumik Mandal, Ohio State University

Libin Zhu, UCSD

Raef Bassily, Ohio State University

Daniel Hsu, Columbia University

Partha Mitra, Spring Harbor Labs

Sasha Rakhlin, MIT

Sasha Tsybakov, ENSAE

Thank you