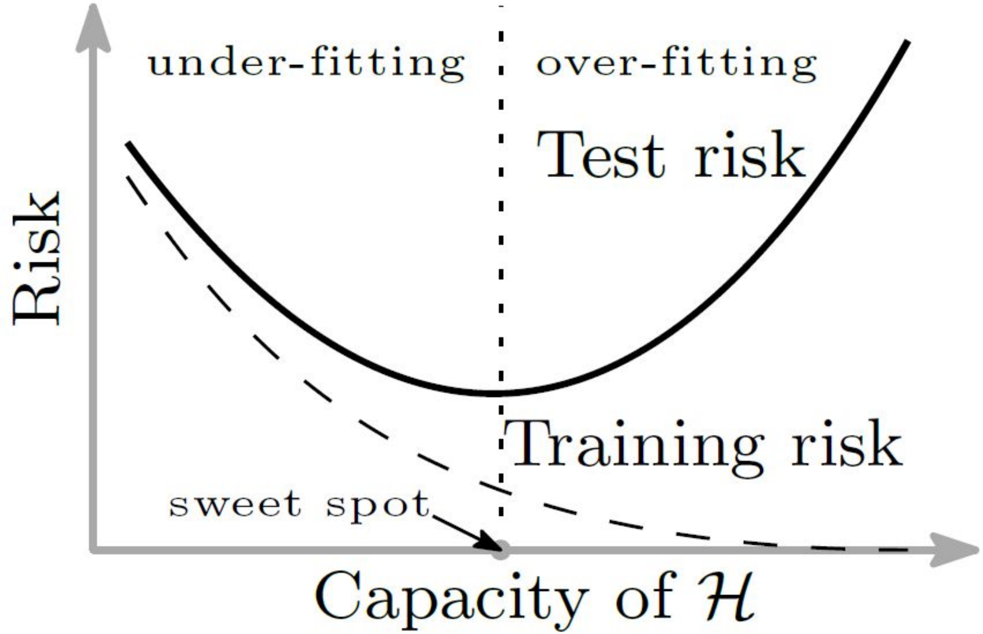
Reconciling modern machine learning practice and the bias-variance trade-off

Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal

Niklas Smedemark-Margulies and Nicholas Vann

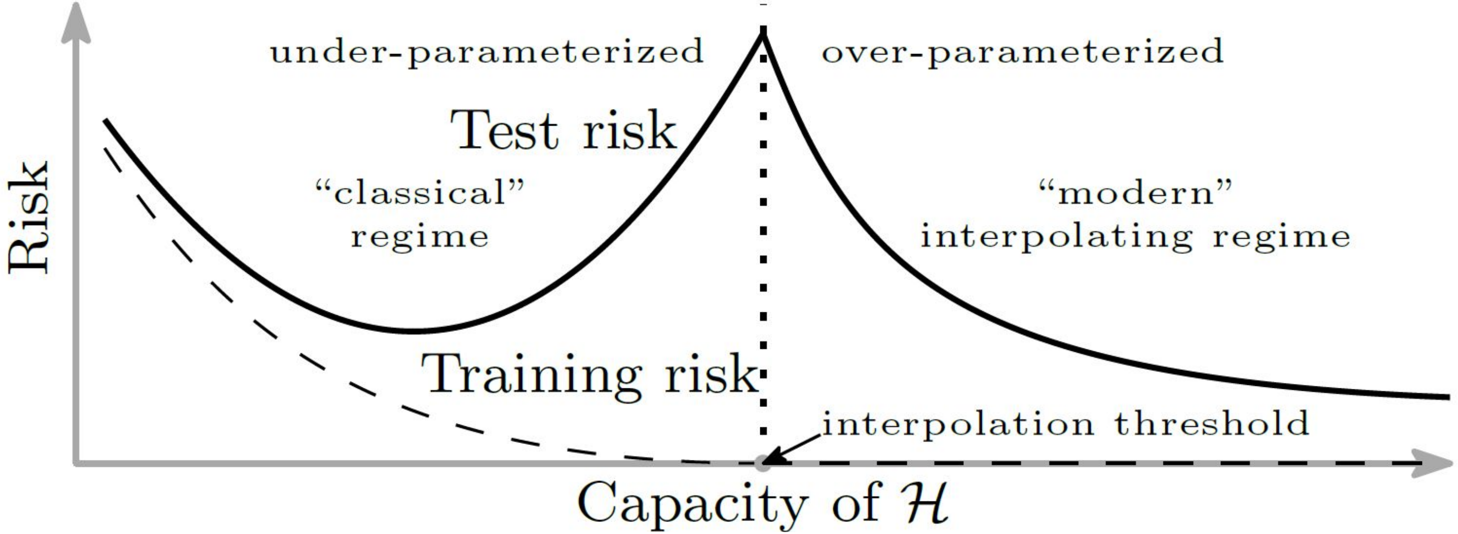
Introduction – Bias – Variance Trade Off



Introduction

- Classical machine learning looks for the “sweet spot” where training risk is low but not at the cost of test risk (bias-variance trade off)
- However, modern methods like neural networks are often designed to have little to no training risk and are still accurate on test data
- This is due to the fact that the function class capacity is increased well beyond the point of reaching zero training risk, functionally extending beyond the traditional U shaped curve

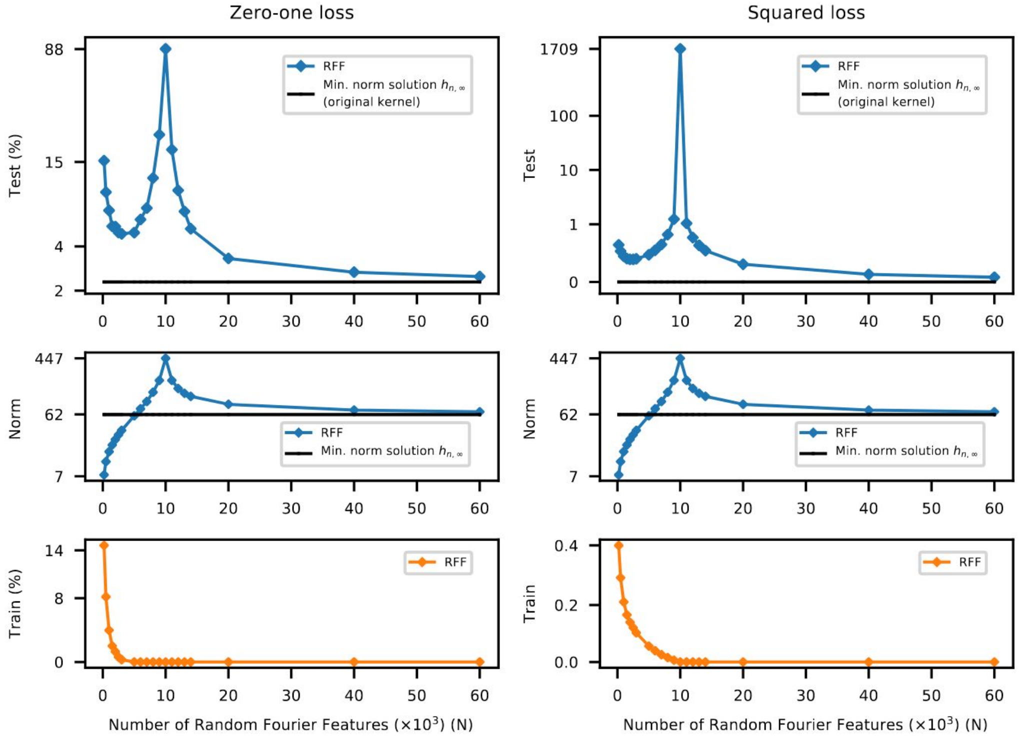
Introduction – Double Descent



Random Fourier Features

$$h(x) = \sum_{k=1}^{N} a_k \phi(x; v_k)$$

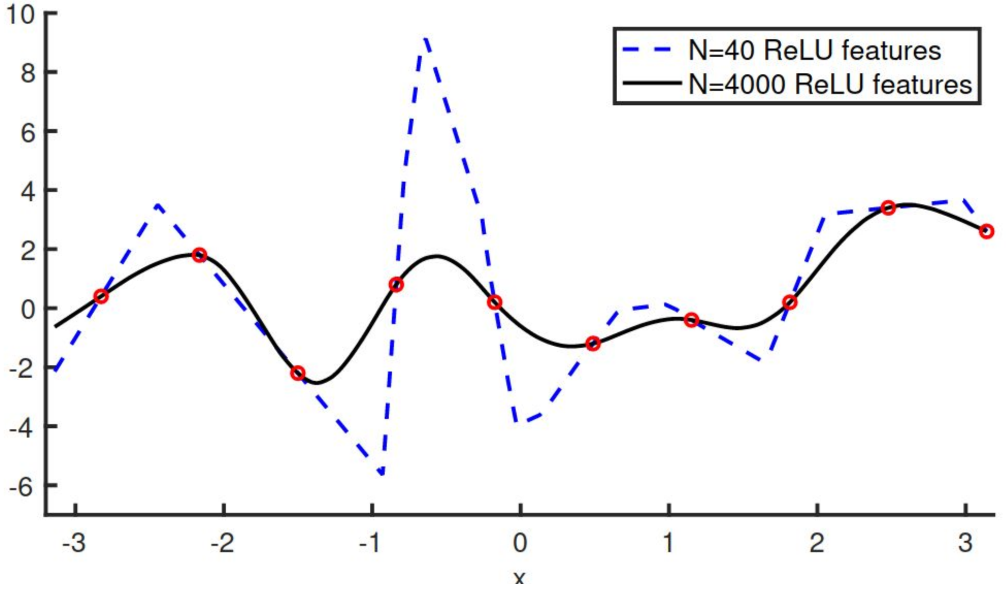
$$\text{where} \quad \phi(x; v) := e^{\sqrt{-1}\langle v, x\rangle}$$



Random ReLU Features

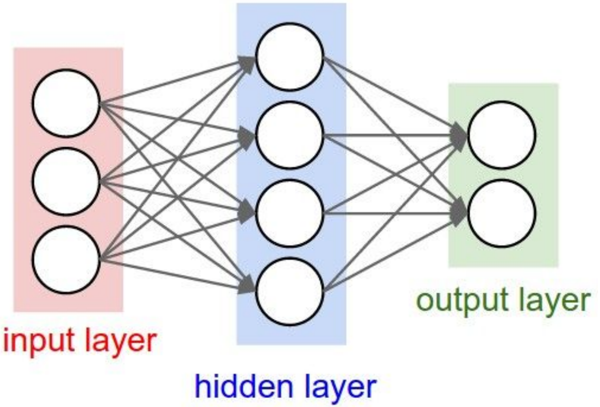
$$h(x) = \sum_{k=1}^{N} a_k \phi(x; v_k)$$

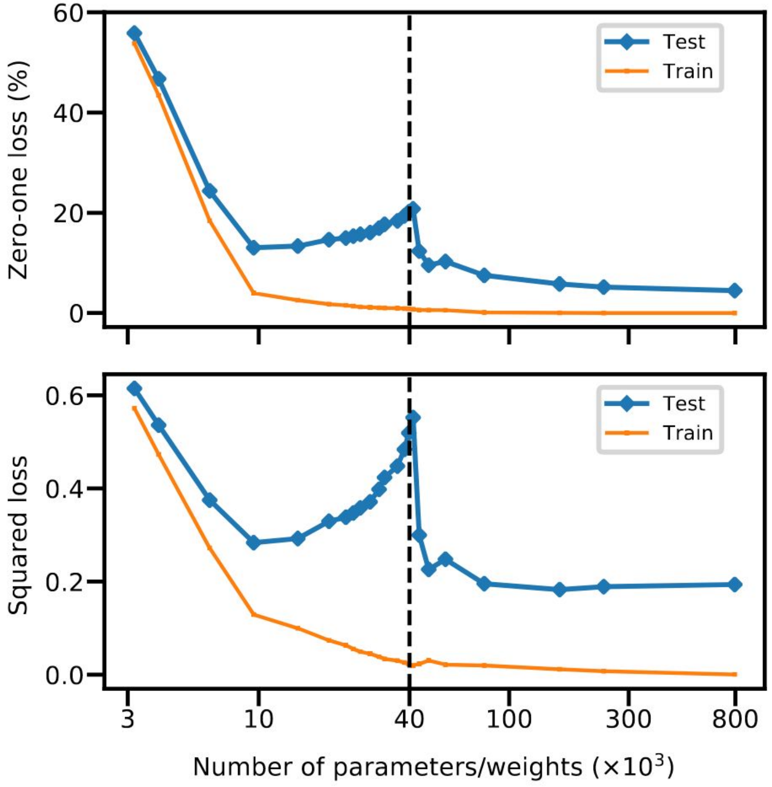
$$\text{where} \quad \phi(x; v) := \max(\langle v, x \rangle, 0)$$



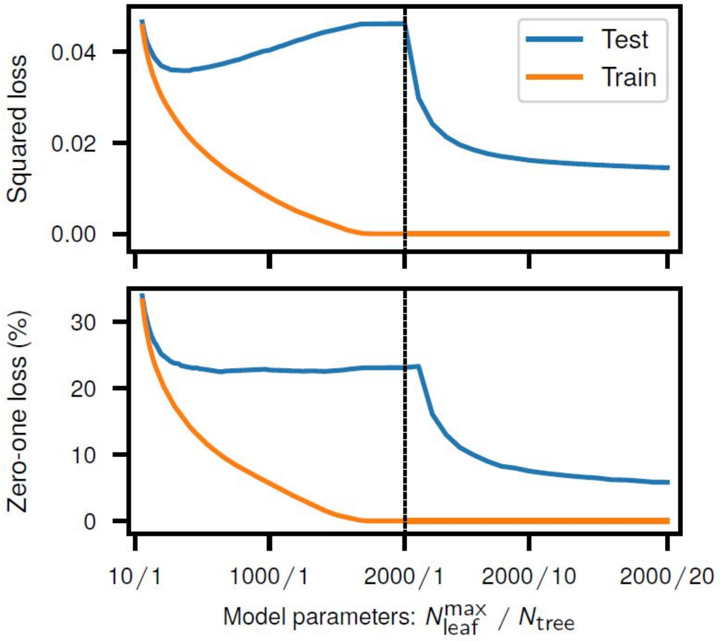
Neural Network

Single hidden layer of varying size, e.g.:



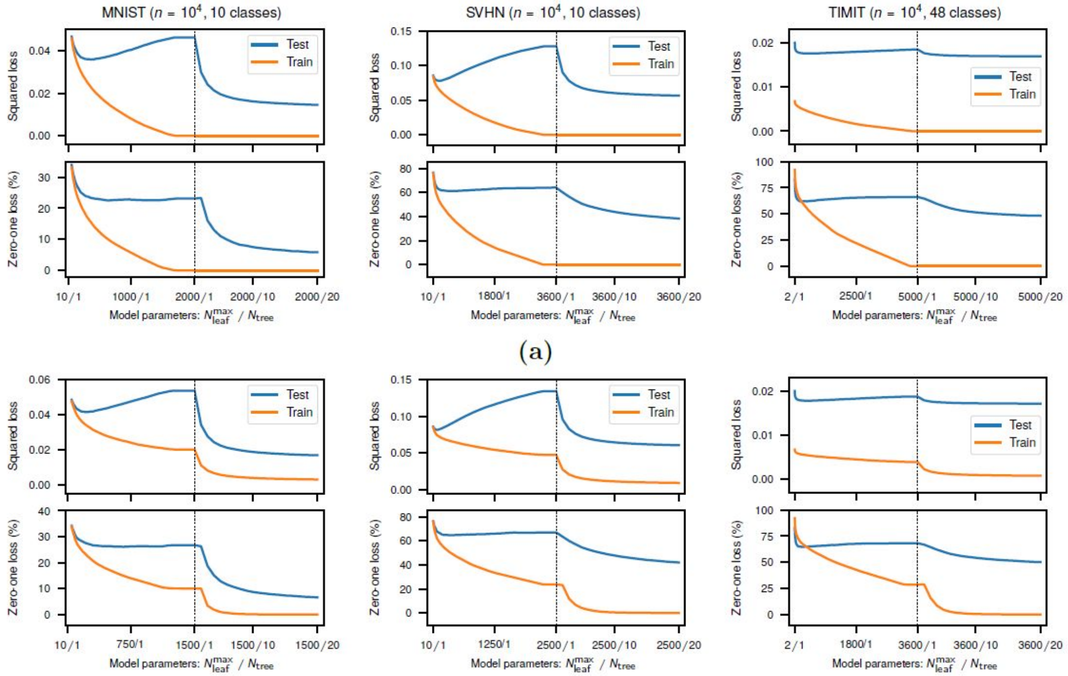


Decision Trees and Ensemble Methods



- The double descent curve can be seen in more classical machine learning methods as well
- By including multiple trees such as in random forests the method is effectively extended beyond the interpolation point

Decision Trees and Ensemble Methods



(a)

Activity 1: Decision Trees

Consider a problem with: 10^4 labeled training items.

Given the following 4 possible random forest models, which one do we expect contains the optimal model? (Optimal = lowest expected test error)?

How about the least optimal model?

A) 5 x 10^3 max leaves, one tree
B) 10^4 max leaves, one tree
C) 10^4 max leaves per tree, two trees
D) 5 x 10^3 max leaves per tree, 10 trees

Activity 2: Choose the ideal RFF model

Consider a regression problem with: ~10^8 labeled training items.

Which of the following Random Fourier Features models do you think will perform best (best = lowest expected test error)? How about worst?

(Recall that *N* is the number of fixed random vectors we use to produce features)

A) $N = \sim 7.2 \times 10^3$

B) $N = \sim 4.5 \times 10^3$

C) $N = \sim 1.8 \times 10^5$

D) $N = \sim 3.6 \times 10^8$

Conclusion

- Historical Absence
  - Regularization prevents interpolation
  - Interpolation happens in a narrow range of settings for NN
  - RFF models have traditionally been used with N $\ll$ n for better run time so models beyond the interpolation threshold were not considered
- Inductive bias
  - Occam's Razor, the smoothest model that fits the data is likely to generalize best
- Practical Considerations
  - Larger models may be easier to optimize with SGD as well
- Still need precise definitions of model complexity, esp. for NN
  - We can think about # parameters, # effective parameters, VC dimension

TL;DR

- We saw the **bias-variance** tradeoff, aka the **underfit-overfit** tradeoff
- Previously, ML theory told us:
  - There is a “**sweet spot**” of model complexity, where we will have the best possible performance on test data.
  - Achieving perfect accuracy on your training data is probably a <u>bad idea</u> because you are likely to be **overfitting**
- Nonetheless, experimenters discovered that very large models can achieve perfect training accuracy and still do very well on test data.
- This paper tells us how to reconcile this phenomenon by moving model complexity beyond the interpolation threshold