Overparameterization, Implicit Bias, and Double-Descent

Machine Learning Fundamentals for Economists

Jesse Perla

jesse.perla@ubc.ca

University of British Columbia

Table of contents

Overview

Summary

- Given the previous lectures on deep-learning, we can now come back to optimization methods and regularization
- At this point you should be convinced that:
  1. It is possible to fit very flexible approximations with high-dimensional parameters, far more than the number of data points (i.e., **overparameterized**)
  2. Having flexible functional forms in our $\mathcal{H}$ lets us use representations which may help us in many ways

Remain Skeptical!

1. Given our understanding of the classic bias-variance tradeoff
    - What about overfitting?
2. The ERM process for complicated problems is not globally convex
    - First-order methods only help me go to (convex) local optima?
    - With overparameterization, are local optima even convex?

Overview

Warning: active, incomplete literature in math, CS, and statistics.

But we can still give some intuition:

1. First explore why the bias-variance tradeoff breaks down and give a mental model for how we can solve problems in high dimensions
2. Then explore the optimization process in high dimensions and explain why with enough dimensions (and randomness) it becomes easier
3. Finally, explain different sources of regularization (implicit and explicit) that occur during optimization, which helps explain why double descent can occur

Double-Descent and the Bias-Variance Tradeoff

Recall the Decomposition of Errors from ERM

- Without repeating the entire notation, we established our goal as minimizing the difference between the ideal

$$\mathbb{E}_{\mathcal{D}\sim\mu^*}\left[\min_{f\in\mathcal{H}} R(f,\mathcal{D}) - \min_{f\in\mathcal{F}} R(f,\mu^*)\right] = \varepsilon_{\text{app}}(\mathcal{H}) + \varepsilon_{\text{gen}}(\mathcal{H})$$

- That is, a classic tradeoff between approximation error and generalization error given some true distribution μ^* and some samples $\mathcal{D}$

The Bias-Variance Tradeoff

- Richer, deeper models enable better representations
  - → Can we find them given the massive number of parameters?
  - → Or do they overfit, find spurious patterns which fail to generalize?
- Classic Bias-Variance tradeoff in statistics suggests a sweet spot
  - → Too few parameters and you get too much "bias", high $\varepsilon_{\text{app}}(\mathcal{H})$
  - → As you approach the number of parameters to the number of data points, you get too much "variance", high $\varepsilon_{\text{gen}}(\mathcal{H})$
  - → At the "interpolation threshold" it fits perfectly (i.e., the training error is zero) but the generalization is terrible
- So classic statistics suggests looking for the sweet spot below the interpolation threshold

Classic Statistics is (Often) Wrong!

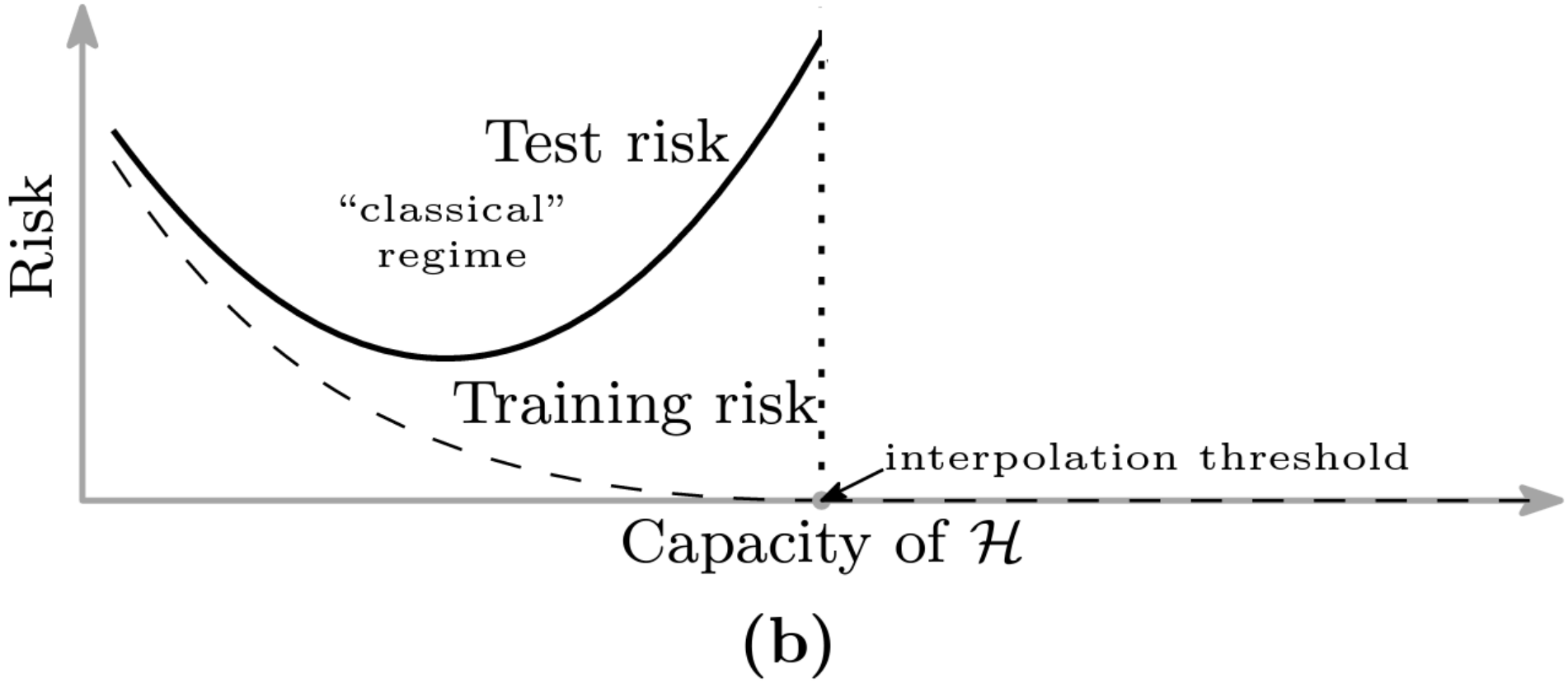
The solution to overfitting is to keep adding more parameters

- The evaluation criteria here is the correct one: generalization performance
- Your intuition should be that this requires some sort of regularization
- In most of these methods you will “interpolate” the training data for zero training loss

Some References

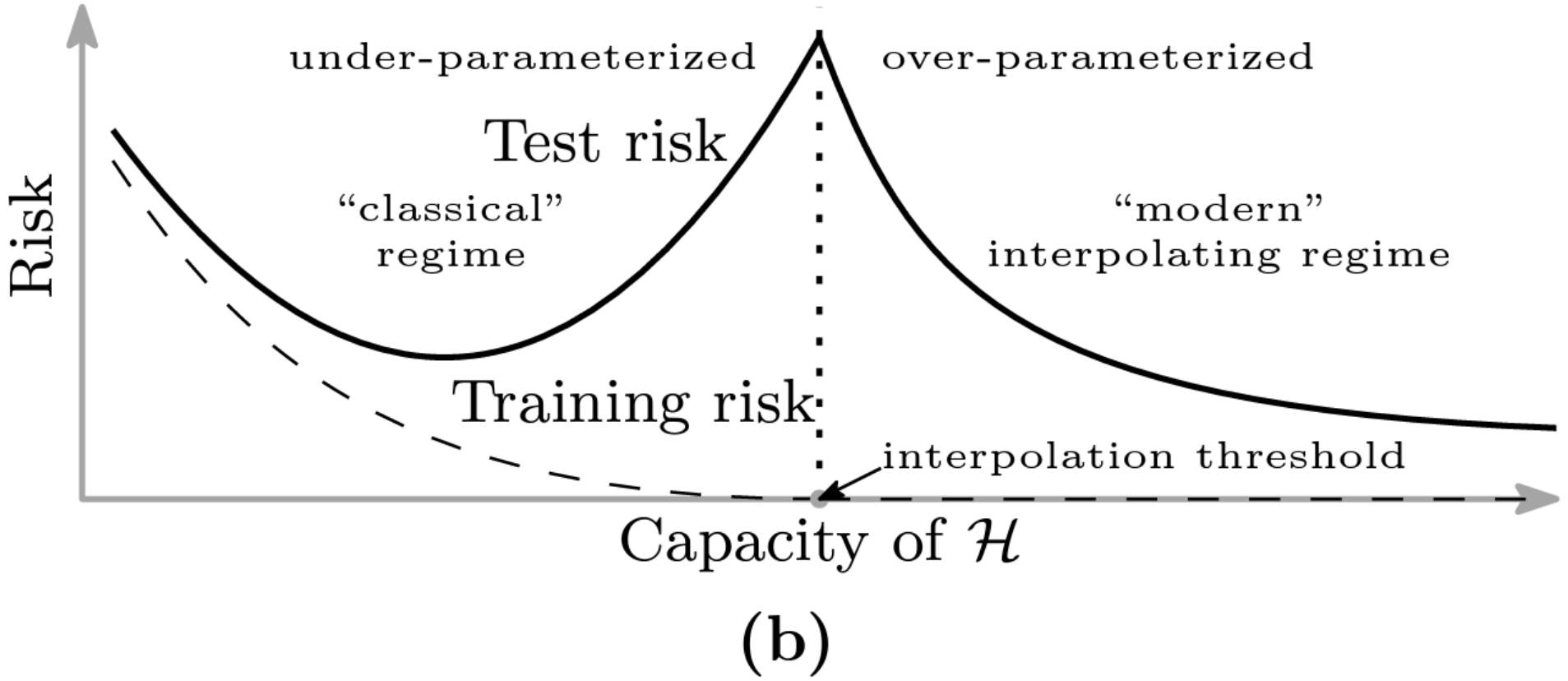
- Wilson (2025) for an accessible broad overview
- Reconciling modern machine-learning practice and the classical bias–variance trade-off
- Deep Double Descent: Where Bigger Models and More Data Hurt
- Loss landscapes and optimization in over-parameterized non-linear systems and neural networks
- Fit without fear: remarkable mathematical phenomena of deep learning through the prism of interpolation
- Also includes Linear models:
  - → Benign Overfitting in Linear Regression
  - → Surprises in High-Dimensional Ridgeless Least Squares Interpolation

Classic Statistics: Single Descent



from https://arxiv.org/abs/1812.11118

Double Descent



from https://arxiv.org/abs/1812.11118

Example Fitting Small Number of Data Points

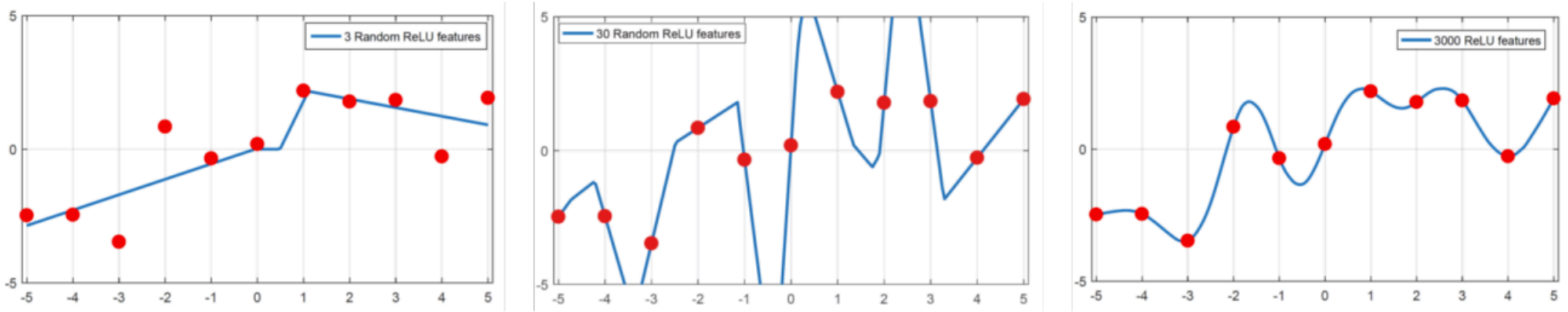


Figure 6: Illustration of double descent for Random ReLU networks in one dimension. Left: Classical under-parameterized regime (3 parameters). Middle: Standard over-fitting, slightly above the interpolation threshold (30 parameters). Right: "Modern" heavily over-parameterized regime (3000 parameters).

from https://arxiv.org/pdf/2105.14368.pdf

Example with Increasing “Width” of Neural Network

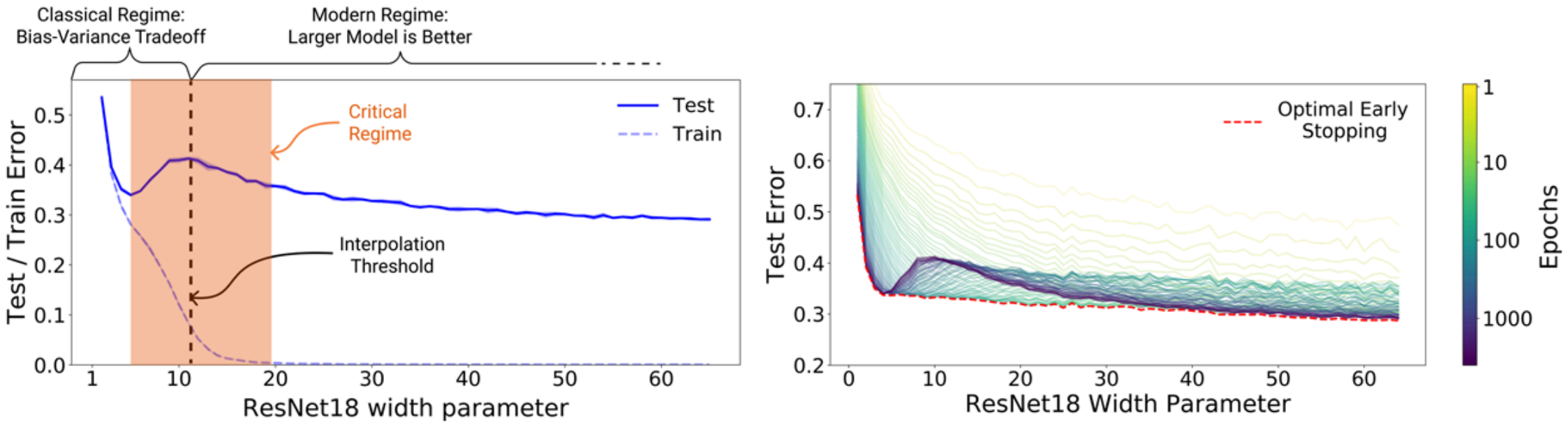


Figure 1: **Left:** Train and test error as a function of model size, for ResNet18s of varying width on CIFAR-10 with 15% label noise. **Right:** Test error, shown for varying train epochs. All models trained using Adam for 4K epochs. The largest model (width 64) corresponds to standard ResNet18.

from https://arxiv.org/pdf/1912.02292.pdf

Overparameterized Models Interpolate

- One interpretation is that ML optimization methods with sufficient parameters find a min-norm interpolating solution. Minimizing ERM:

$$\min_{f \in \mathcal{H}} \frac{1}{N} \sum_{n=1}^{N} \ell(f, x_n, y_n)$$

- With large enough $\mathcal{H}$ this interpolates, i.e. $\ell(f, x_n, y_n) = 0$ for all n
- Remember overdetermined LLS and the ridgeless regression
  - → This did not require enormous amounts of data

Min-Norm Interpretation

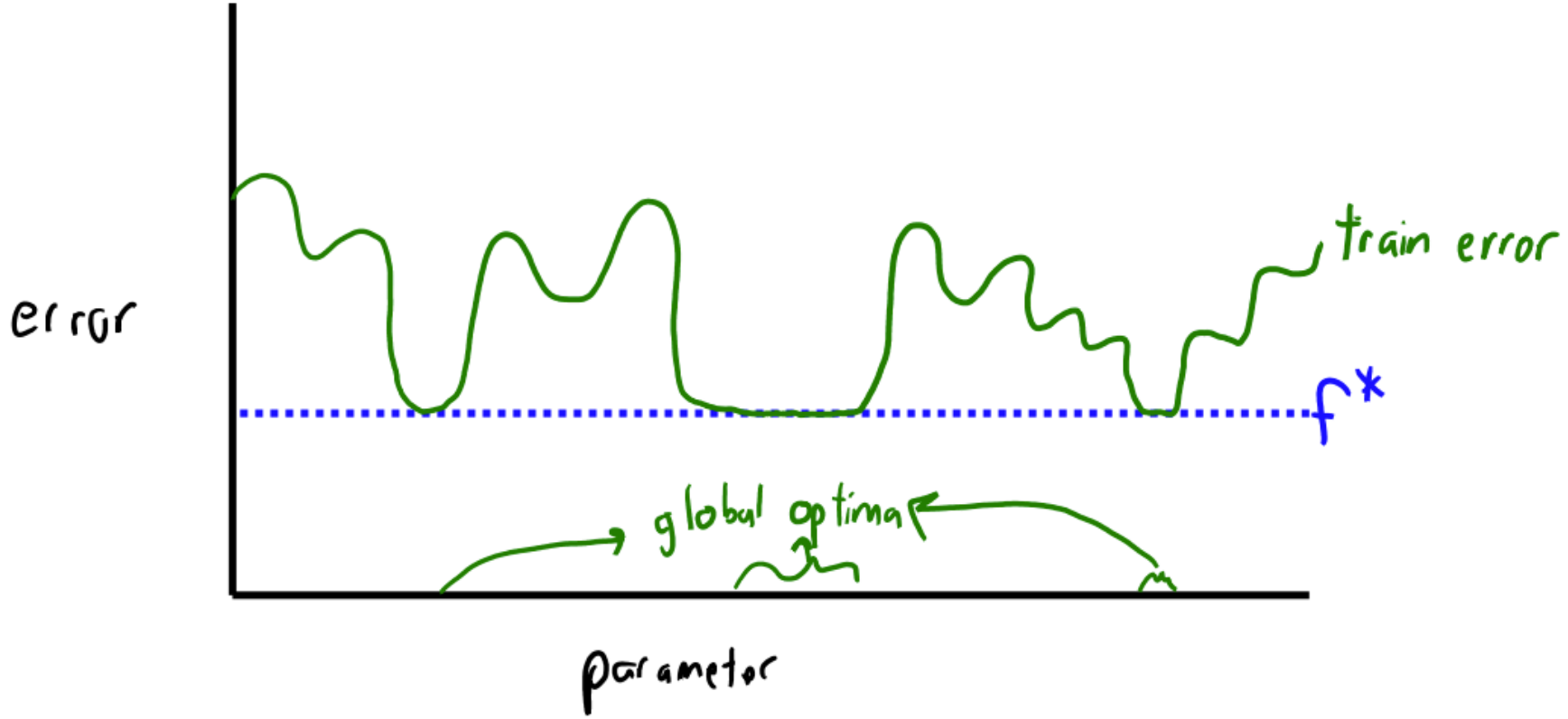
- With first-order optimizers can often interpret as solving

$$\min_{f \in \mathcal{H}} \|f\|_S$$
$$\text{s.t. } \ell(f, x, y) = 0, \quad \text{for all } (x, y) \in \mathcal{D}$$

- Where S is some (semi-)norm we generally cannot characterize
  - → In some cases we can prove this is Sobolev 1,2 (e.g.,with LLS)
  - → In practice, chooses smooth functions (Occam's Razor)
- Whether these are the correct (i.e., generalizable) solutions depends on whether the S aligns with the properties of the true μ^*
  - → Having lots of data is neither necessary nor sufficient

Loss Landscapes in Overparameterized Models

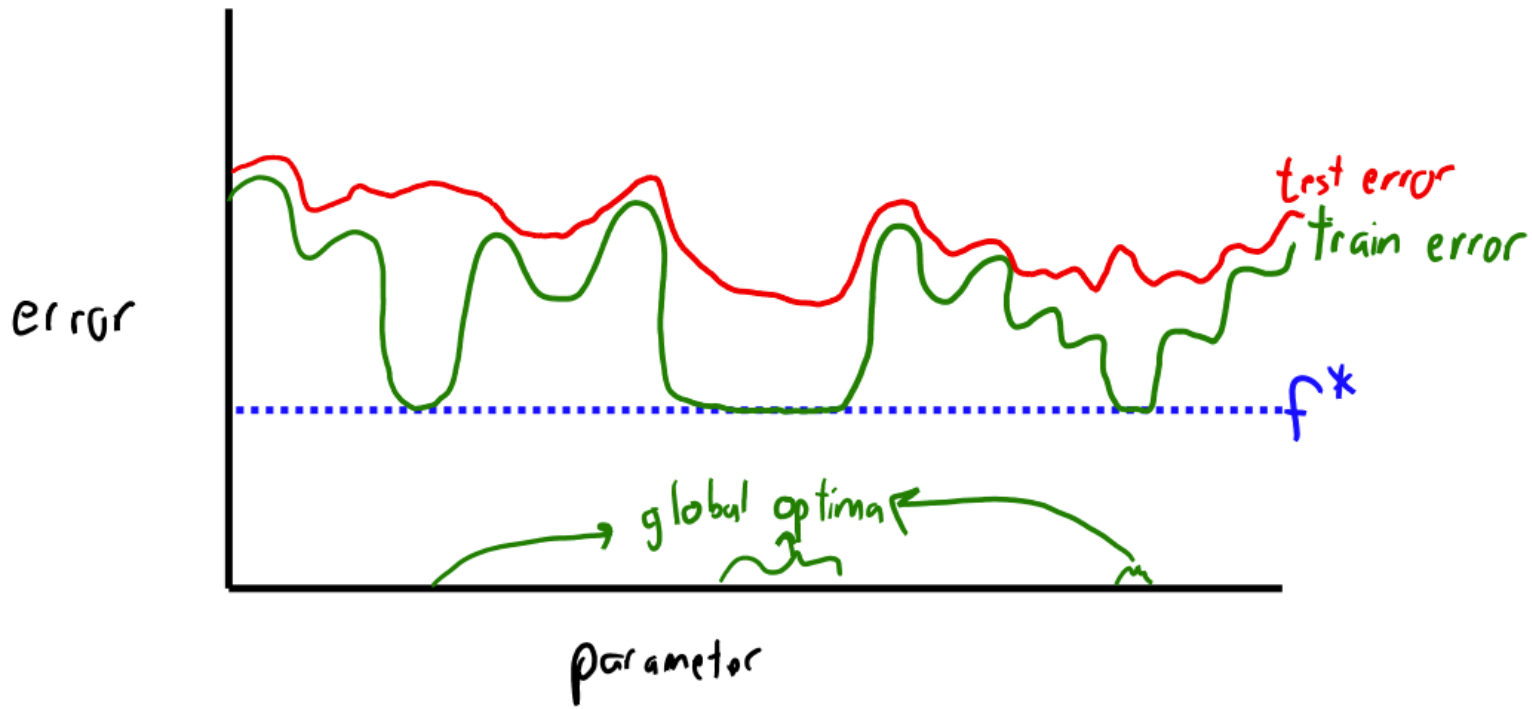
Overparameterized ERM Has Many Global Minima



See https://www.cs.ubc.ca/~schmidtm/Courses/440-W22/L7.pdf

- With overparameterization, there are many global minima
- In high dimensions the global minima become increasingly connected

Not All Global Minima are Alike



See https://www.cs.ubc.ca/~schmidtm/Courses/440-W22/L7.pdf

- Some parameters generalize (estimated with test error) better than others
- Intuition: better with more “volume” of parameters at low test error

Flat vs. Sharp Geometry

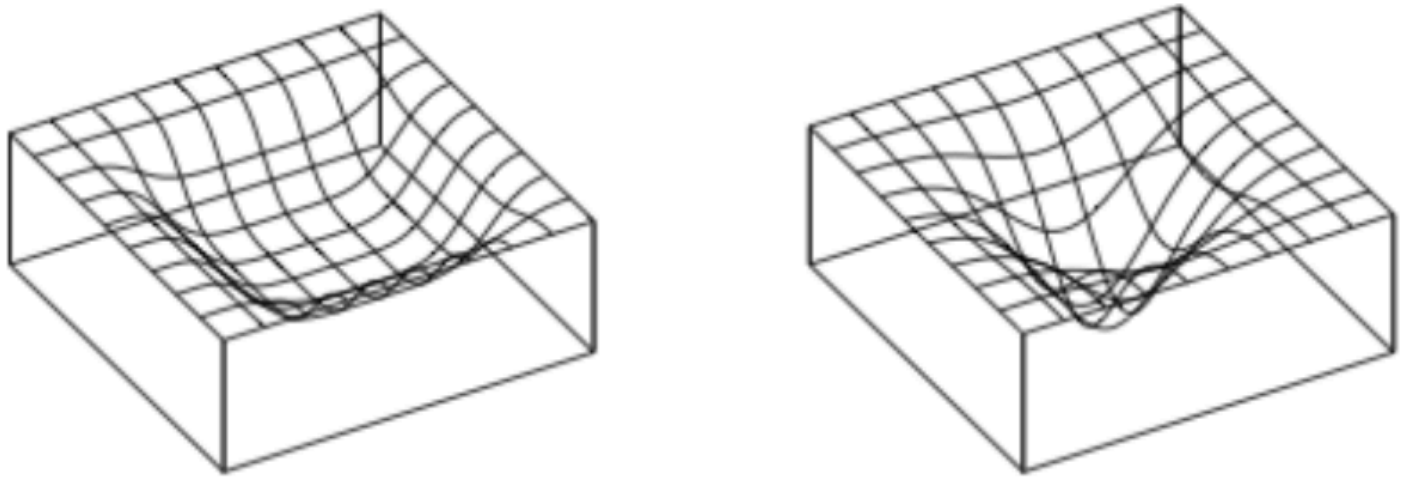


Figure 13.19: Flat vs sharp minima. From Figures 1 and 2 of [HS97a]. Used with kind permission of Jürgen Schmidhuber.

from ProbML Book 1 Section 13.5.6

- We often think of flatter geometry near minima as worse (e.g., not identified)
- With overparameterization this is the wrong intuition.

  → Flat minima generalize better because they have similar loss over a larger volumes of parameter space. Less dependent on individual data

SGD Finds Flat Minima

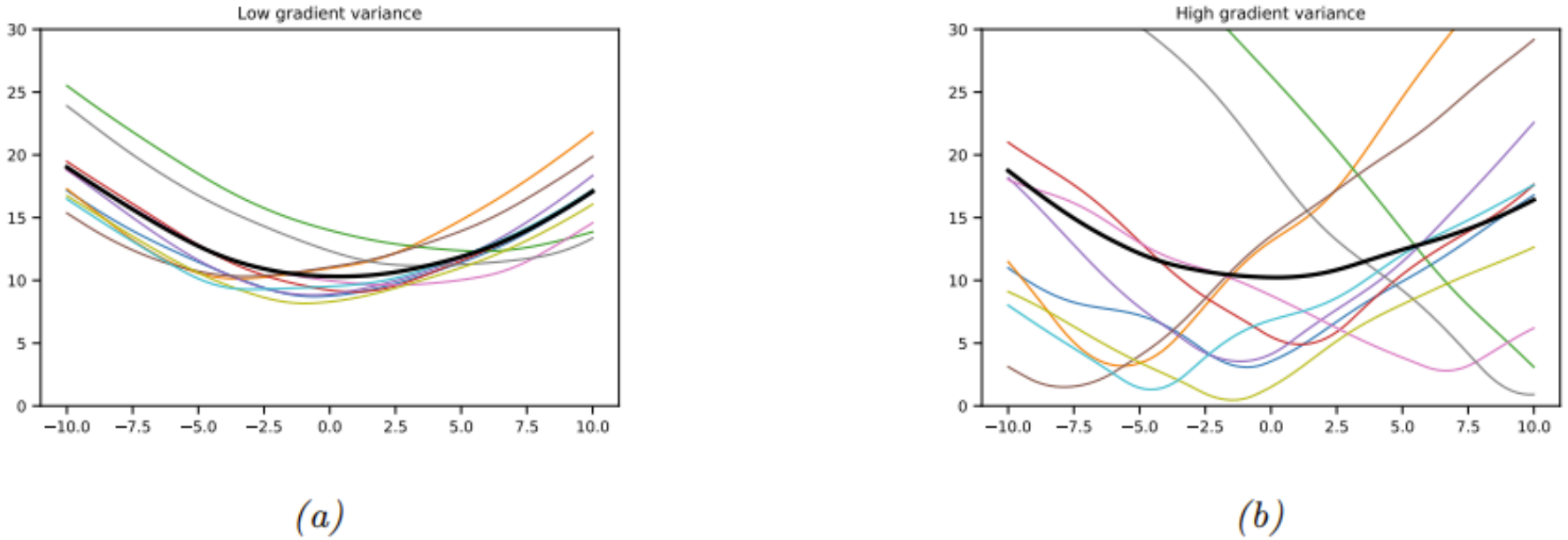
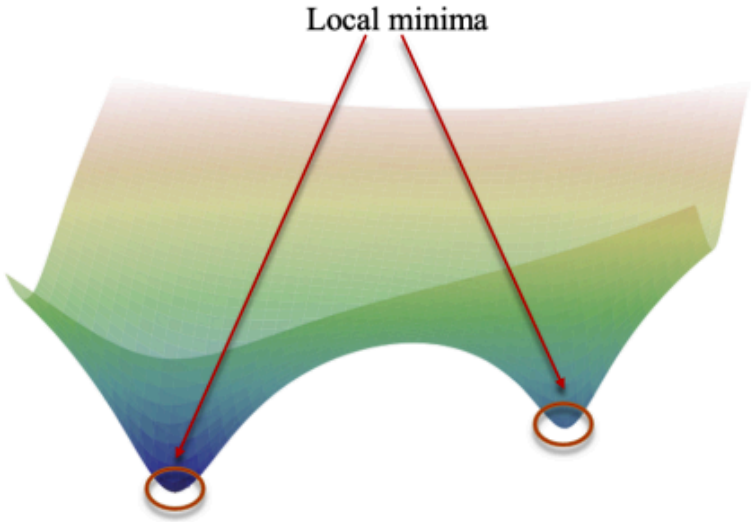


Figure 13.20: Each curve shows how the loss varies across parameter values for a given minibatch. (a) A stable local minimum. (b) An unstable local minimum. Generated by sgd_minima_variance.ipynb. Adapted from https://bit.ly/3wTc1L6.

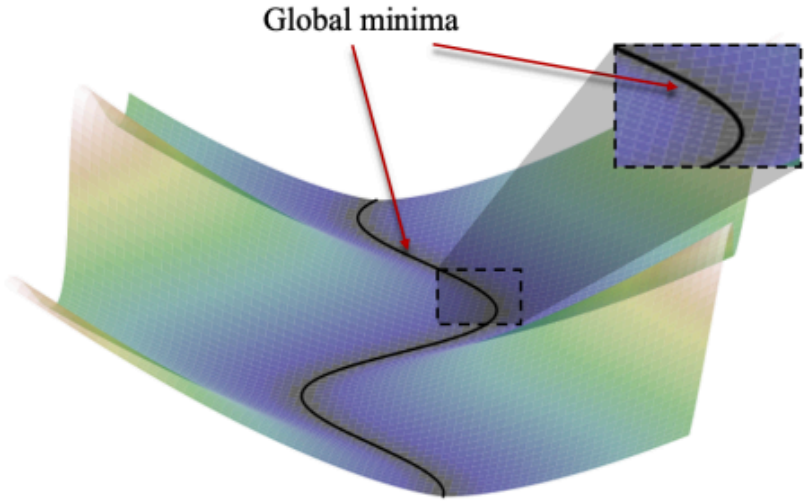
from ProbML Book 1 Section 13.5.6

- SGD finds more stable local minima, less sensitive to underlying data

Overparameterized Loss Landscapes



(a) Loss landscape of under-parameterized models



(b) Loss landscape of over-parameterized models

from https://arxiv.org/pdf/2003.00307.pdf

- Intuition of optimization in low dimensions often breaks down in high dimensions and with overparameterization
- Many classic examples where adding in more parameters makes optimization easier (e.g., more degrees of freedom to move)

Loose Intuition from this Literature

- **Warning:** this literature is a work in progress, and hard to prove things formally for general problems
- However some loose intuition from Loss landscapes and optimization in over-parameterized non-linear systems and neural networks and others
- As you move from low-dimensional, underparameterized to high-dimensional overparameterized problems:
  - → All local minima increasingly become global minima (i.e., interpolation)
  - → Global minima are increasingly connected
  - → Nothing is convex, but increasingly the minima seem to fulfill a related condition (PL^*) which is all we need

Interpretations on the Importance of Flatness

- Bayesian Interpretations:
  - → Flat corresponds to more posterior uncertainty (see ProbML Book 1 Section 13.5.6 and ProbML Book 2 Section 17.4.1)
  - → Larger region in parameter space with similar loss
  - → Less sensitive to small changes in the data (i.e., overfitting)

Regularization

Implicit/Inductive Regularization

- The theme of much of this literature is on the idea of “hidden/implicit/inductive” bias/regularization that occurs during optimization of ML models
- We saw this already in the LLS section where we showed how underdetermined LLS solved with most algorithms ends up at the ridgeless regression solution
- This does not mean that implicit regularization is always enough, but it helps us understand its role (and whether it fights against us).

Sources of Regularization

- We see this with just gradient descent in LLS, so not specific to nonlinear, having lots of parameters, or stochastic optimization
- SGD itself has regularizing effects
- Sometimes manual regularization is needed
  - → Early stopping, Weight decay (L2 regularization), etc.
- See **ProbML Book 1** Section 13.5.6
- See **Mark Schmidt's CPSC440 Notes on Double-Descent Curves** for more on regularization

Stochastic Gradients

- See ProbML Book 1 Section 13.5.6
- As we saw before, randomness in SGD and its variants can help us find flatter minima
- The key is that it is the right sort of noise, which helps us avoid sharp minima and concentrate on flatter minima
- This is true even if all local minima are global minima and connected
  - → The connected minima might even make it easier to traverse to the flatter regions

No Free Lunch!

Choose Algorithms for the Problem Class

- It is worth ending with a restatement of the **no free lunch theorems**

  → No optimization algorithm is universally superior; any algorithm's performance is problem-dependent, such that its average performance across all possible problems is the same as that of any other algorithm.

- In practice, this means that we will need to carefully choose the optimization methods, hypothesis class $\mathcal{H}$, and regularization with a particular problem in mind by understanding the problem class well
- Luckily: it seems many problems in economics are very similar, so we can use priors to choose methods, and HPO software lets us experiment

References

Wilson, Andrew Gordon. 2025. “Deep Learning Is Not so Mysterious or Different.” https://arxiv.org/abs/2503.02113.