bias-variance tradeoff

Saeed Saremi

Assigned reading: 4.2, 4.3, 9.3.2

October 29, 2024

outline

- decomposition of the generalization error into bias^2, variance, and noise
  - definition of the generalization error
  - ✎ derivation of the decomposition
  - underfitting and overfitting in light of the bias-variance tradeoff
- the case of (highly) overparametrized neural networks
  - rethinking generalization in deep learning
  - double descent
  - grokking phenomenon
  - the regime of $p > d \cdot n$

motivating example

Regression setting:

- We are given a dataset:

$$\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{n}$$

- Fit the data with the polynomials

$$f(x;\theta) = \sum_{k=0}^{M} \theta_k x^k$$

- Regularized loss:

$$\mathcal{L}(\theta) = \frac{1}{2}\sum (f(x_i;\theta) - y_i)^2 + \frac{\lambda}{2}\|\theta\|^2$$

- We repeat this for many datasets!

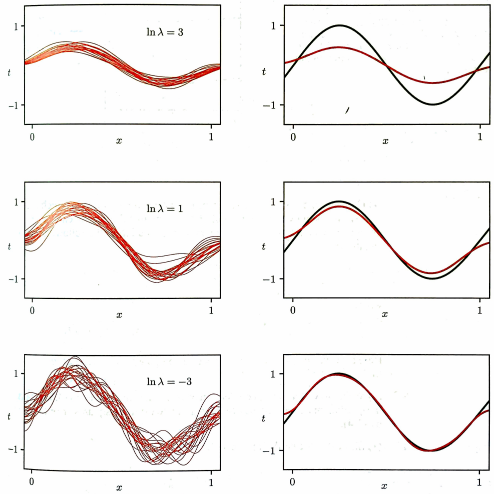


Figure 4.7 Illustration of the dependence of bias and variance on model complexity governed by a regularization parameter λ, using the sinusoidal data from Chapter 1. There are $L = 100$ data sets, each having $N = 25$ data points, and there are 24 Gaussian basis functions in the model so that the total number of parameters is $M = 25$ including the bias parameter. The left column shows the result of fitting the model to the data sets for various values of $\ln\lambda$ (for clarity, only 20 of the 100 fits are shown). The right column shows the corresponding average of the 100 fits (red) along with the sinusoidal function from which the data sets were generated (green).

expected output

- We are given a dataset
$$\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{n}$$
drawn i.i.d. from some (unknown) distribution $p(x, y)$.
- Throughout this lecture we consider the regression problem, where $y \in \mathbb{R}$.
- In general, given the the input x, the output y is not unique, whose uncertainty is captured by the conditional distribution $p(y|x)$.
- In this setup $\hat{f}(x) = \mathbb{E}[Y|x]$ (the expected output given x) is (Bayes) optimal for the squared loss
$$\mathcal{L}[f] = \int (f(x) - y)^2 p(x, y) dx\, dy.$$
- We use the following notation to denote the expected output (given x):
$$\bar{y}(x) = \mathbb{E}[Y|x] = \int y\, p(y|x) dy$$

Bayes optimal regression

$$\mathcal{L}[f] = \int (f(x) - y)^2 \, p(x,y) \, dx \, dy$$

$$\Rightarrow \frac{\delta \mathcal{L}}{\delta f}\Big|_{f=\hat{f}} = 0$$

$$\Rightarrow \int (\hat{f}(x) - y) \underbrace{p(x,y)}_{p(y|x)\,p(x)} \, dx \, dy = 0$$

$$\int p(y|x)\,dy = 1$$

$$\Rightarrow \int \Big(\hat{f}(x) - \underbrace{\bar{y}(x)}_{\mathbb{E}[Y/x]}\Big) \, p(x) \, dx = 0$$

$$\Rightarrow \boxed{\hat{f}(x) = \bar{y}(x)}$$

Question asked during the lecture

$$\overset{\text{posterior}}{p(y|x)} = \frac{\overset{\text{likelihood}}{p(x|y)} \; \overset{\text{prior}}{p(y)}}{p(x)}$$

$$p(x) \rightarrow \int p(x|y)\,p(y)\,dy$$

expected test error

- Given the dataset $\mathcal{D}$ we use an algorithm to fit the data, arriving at

$$f_{\mathcal{D}} : X \to Y.$$

- By definition, the expected test error for f_D is given by

$$\mathcal{L}[f_{\mathcal{D}}] = \int (f_{\mathcal{D}}(x) - y)^2 p(x, y) dx\, dy$$

model complexity

- We assume we fix the set of functions used to fit the data in our prediction model, which we denoted by $\mathcal{F}$. Therefore, $f_{\mathcal{D}} \in \mathcal{F}$.
- Some examples of $\mathcal{F}$ include the space of linear functions, the space of polynomials with a fixed degree, and finally (our favorite) the space of neural networks with a fixed architecture.
- In abuse of notation, we sometimes think of $\mathcal{F}$ (which perhaps comes with some hyperparameters) as the procedure/algorithm used to fit the dataset $\mathcal{D}$ to arrive at the function $f_{\mathcal{D}}$, captured by
$$f_{\mathcal{D}} = \mathcal{F}(\mathcal{D}).$$
- Intuitively, $\mathcal{F}$ encodes the model complexity.[1]

[1] For example if $\mathcal{F}$ is parametric class of functions $f_\theta \in \mathcal{F}$, the dimensions p of the parameters ($\theta \in \mathbb{R}^p$) is traditionally considered a surrogate for model complexity (although we will see a breakdown of this traditional paradigm).

expected regression function

- Consider a thought experiment by drawing many i.i.d. datasets

$$\mathcal{D}_j \overset{\text{iid}}{\sim} p^n, \; j \in [m].$$

- Now imagine fitting functions $f_{\mathcal{D}_j} : X \to Y$ to the dataset $\mathcal{D}_j$.
- All these m functions can be combined by taking their empirical mean:

$$\bar{f}_m = \frac{1}{m} \sum_j f_{\mathcal{D}_j},$$

which for large m converges to

$$\bar{f} = \int f_{\mathcal{D}} p(\mathcal{D}) d\mathcal{D}$$

generalization error

What we are interested in is the expected test error:

$$R = \mathbb{E}_{(x,y),\mathcal{D}} \left(f_{\mathcal{D}}(x) - y \right)^2$$

The quantity R, called the generalization error, tells us how our model performs when we evaluate it on many test samples (x, y) and over many datasets $\mathcal{D}$.

bias + variance + noise decomposition of R

$$R = \mathbb{E}_x\left[\left(\bar{f}(x) - \bar{y}(x)\right)^2\right] + \mathbb{E}_{x,\mathcal{D}}\left[\left(f_{\mathcal{D}}(x) - \bar{f}(x)\right)^2\right] + \mathbb{E}_{x,y}\left[\left(\bar{y}(x) - y\right)^2\right]$$

bias-variance-noise

the generalization error

Two "players":

$\bar{y}(x) = \mathbb{E}[Y|X]$

$\bar{f}(x) = \mathbb{E}_D f_D(x)$

$$R = \mathbb{E}_D \mathbb{E}_{(x,y)} (f_D(x) - y)^2$$

$$= \mathbb{E}_D \mathbb{E}_{(x,y)} (\underbrace{f_D(x) - \bar{f}(x)}_{A} + \underbrace{\bar{f}(x) - y}_{B})^2$$

$$A^2 + B^2 + 2AB$$

$$\text{"}AB\text{"} = \mathbb{E}_D \mathbb{E}_{(x,y)} (f_D - \bar{f}(x))(\bar{f}(x) - y)$$

$$= \mathbb{E}_{(x,y)} \underbrace{\left(\mathbb{E}_D(\cancel{f_D - \bar{f}(x)})\right)}_{0} (\underbrace{\bar{f}(x) - y}_{\text{does not depend on } D})$$

$$= 0$$

$$R = \mathbb{E}_D \mathbb{E}_{(x,y)} (f_D(x) - \bar{f}(x))^2 + \underbrace{(\bar{f}(x) - y)^2}$$

$$(\underbrace{\bar{f}(x) - \bar{y}(x)}_{a} + \underbrace{\bar{y}(x) - y}_{b})^2$$

$$a^2 + b^2 + 2ab$$

$$\text{"}ab\text{"} = \overset{p(x,y)}{\mathbb{E}_{(x,y)}} (\bar{f}(x) - \bar{y}(x))(\bar{y}(x) - y)$$

$$= \mathbb{E}_x \mathbb{E}_{y|x} (\underbrace{\bar{f}(x) - \bar{y}(x)}_{\text{does not depend on } y}) (\underbrace{\bar{y}(x) - y}_{\mathbb{E}[Y|X]})$$

$$= (\quad)(\underbrace{\cancel{\bar{y}(x) - \bar{y}(x)}}_{0})$$

$$\Rightarrow \boxed{R = \underbrace{A^2}_{\text{variance}} + \underbrace{a^2}_{\text{bias}} + \underbrace{b^2}_{\text{noise}}}$$

(see lecture for the detailed expression

Question asked during the lecture

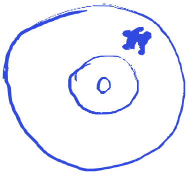
$$\mathbb{E}_{(x,y)} \square = \int \square \, p(x,y)\, dx\, dy$$

$$= \int \square \, p(y|x)\, p(x)\, dx\, dy$$

$$= \int p(x) \underbrace{\left(\int \square \, p(y|x)\, dy\right)}_{\mathbb{E}_{Y|x} \square} dx$$

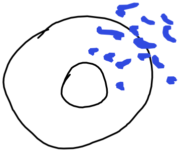
$$= \mathbb{E}_X \mathbb{E}_{Y|X} \square$$

bias + variance + noise decomposition of R



low variance

high bias



high bias

high variance

"ground truth" $\mathbb{E}[Y|X] = \int y\, p(y|x)\, dy$

mean

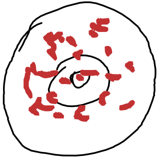
"true value"

$$R = \mathbb{E}_x\left[\left(\bar{f}(x) - \bar{y}(x)\right)^2\right] + \mathbb{E}_{x,\mathcal{D}}\left[\left(f_{\mathcal{D}}(x) - \bar{f}(x)\right)^2\right] + \mathbb{E}_{x,y}\left[\left(\bar{y}(x) - y\right)^2\right]$$

individual fit

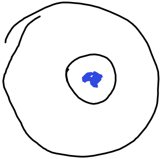
"mean response"

mean of my fits to many training sets



high variance

low bias



low bias

low variance

underfitting/overfitting in light of the bias-variance decomposition

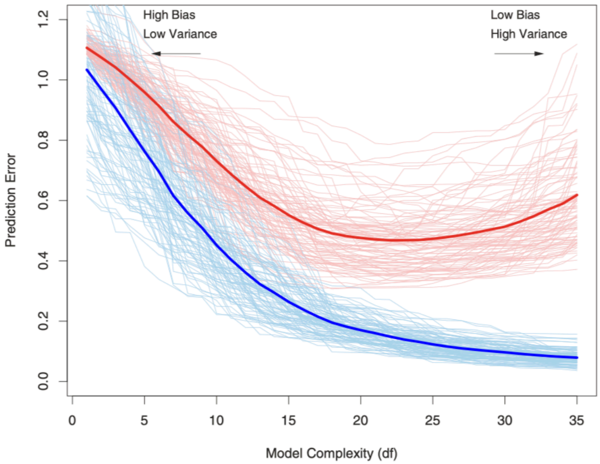


FIGURE 7.1. *Behavior of test sample and training sample error as the model complexity is varied. The light blue curves show the training error* $\overline{\mathrm{err}}$*, while the light red curves show the conditional test error* $\mathrm{Err}_{\mathcal{T}}$ *for* 100 *training sets of size* 50 *each, as the model complexity is increased. The solid curves show the expected test error* Err *and the expected training error* $\mathrm{E}[\overline{\mathrm{err}}]$.

suggested (classical) recipe

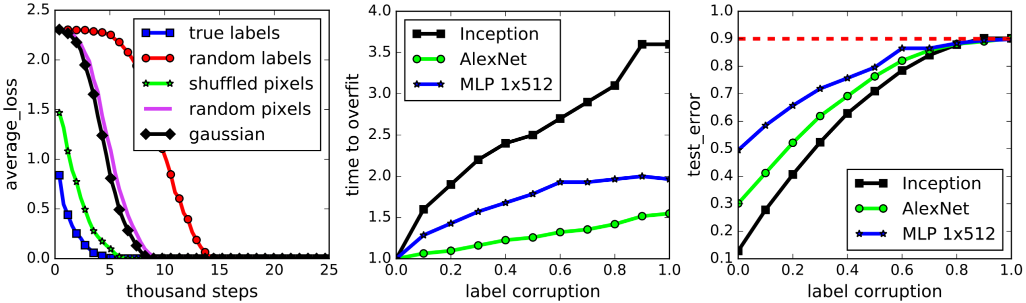
- The "capacity" of the model is a hyperparameter
- We choose this based on the error on a "validation set" (subset of training set put aside for this purpose)
- We expect that as we increase capacity we hit a "sweet spot" that we discover using performance on the validation set
- We we are using capacity smaller than optimal, we are "underfitting", when we use capacity larger than optimal, we are "overfitting".

suggested (modern) recipe

- Train on the largest overparametrized model that one can
- Totally happy with the training set error becoming small
- Do not overfitting!
- Below we discuss some recent literature on this topic.

fear

"rethinking generalization"



(a) learning curves (b) convergence slowdown (c) generalization error growth

Figure: Understanding deep learning requires rethinking generalization (Zhang et al, 2017)

double descent

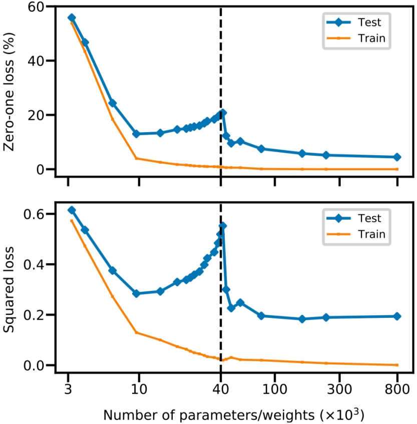


Fig. 3. Double-descent risk curve for a fully connected neural network on MNIST. Shown are training and test risks of a network with a single layer of H hidden units, learned on a subset of MNIST ($n = 4 \cdot 10^3$, $d = 784$, $K = 10$ classes). The number of parameters is $(d+1) \cdot H + (H+1) \cdot K$. The interpolation threshold (black dashed line) is observed at $n \cdot K$.

Figure: *Reconciling modern machine-learning practice and the classical bias-variance trade-off*

grokking phenomenon[2]

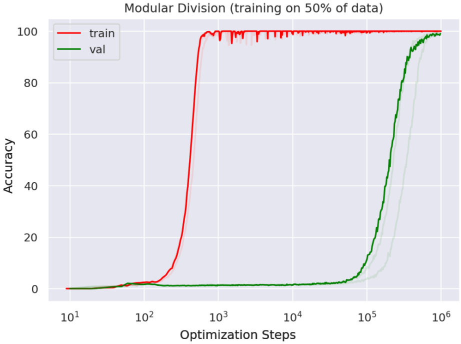


Figure: Grokking: Generalization Beyond Overfitting on Small Algorithmic Datasets (Power et al. 2022)

[2] The term "grok" originates from the 1961 science fiction novel Stranger in a Strange Land by Robert Heinlein. In the book, it is a word from the Martian language that means to deeply and intuitively understand something.

$p > d \cdot n$ regime

> *Classically, data interpolation with a parametrized model class is possible as long as the number of parameters is larger than the number of equations to be satisfied. A puzzling phenomenon in deep learning is that models are trained with many more parameters than what this classical theory would suggest. We propose a partial theoretical explanation for this phenomenon. We prove that for a broad class of data distributions and model classes,* ***overparametrization*** *is* ***necessary*** *if one wants to interpolate the data* ***smoothly****. Namely we show that* smooth *interpolation requires d times more parameters than mere interpolation, where d is the ambient data dimension. [...] We also give an interpretation of our result as an improved generalization bound for model classes consisting of smooth functions.*
>
> *A Universal Law of Robustness via Isoperimetry, Bubeck and Sellke, 2021.*