10.8 Interpolation and Double Descent

Double Descent

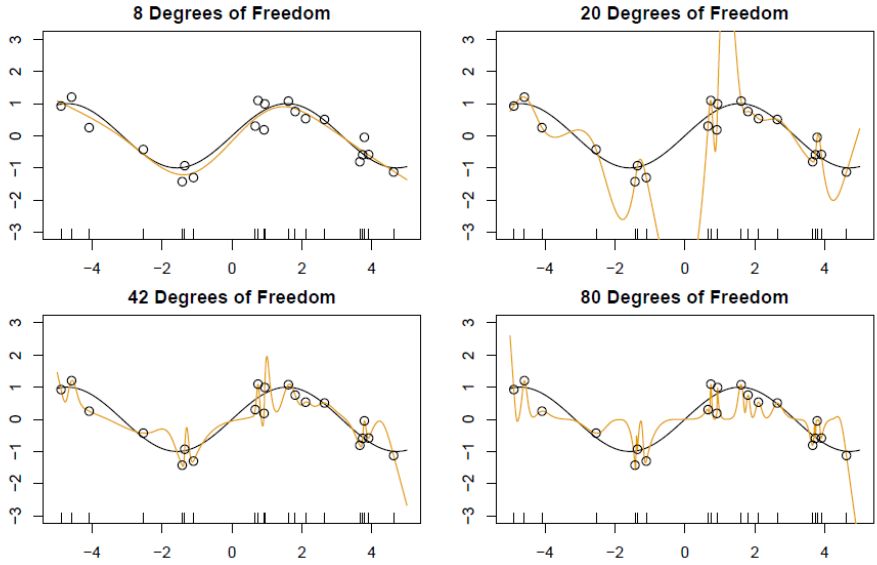
- With neural networks, it seems better to have too many hidden units than too few.
- Likewise more hidden layers better than few.
- Running stochastic gradient descent till zero training error often gives good out-of-sample error.
- Increasing the number of units or layers and again training till zero error sometimes gives even better out-of-sample error.

. . . what happened to overfitting and the usual bias-variance trade-off?

Simulation

- $y = \sin(x) + \epsilon$ with $x \sim U[-5, 5]$ and $\epsilon \sim N(0, 0.3)$
- Training set $n = 20$, test set very large $(10k)$
- Fit a natural spline to the data, with d degrees of freedom.
  - With $d = 20$, we fit the training data perfectly (all residuals $= 0$)
- With $d > 20$, we still fit the data exactly, but our solution isn't unique.
  - Among all zero-residual solutions, we pick the one with *minimum norm*, ie with the smallest $\sum_{j=1}^{d} \hat{\beta}_j^2$

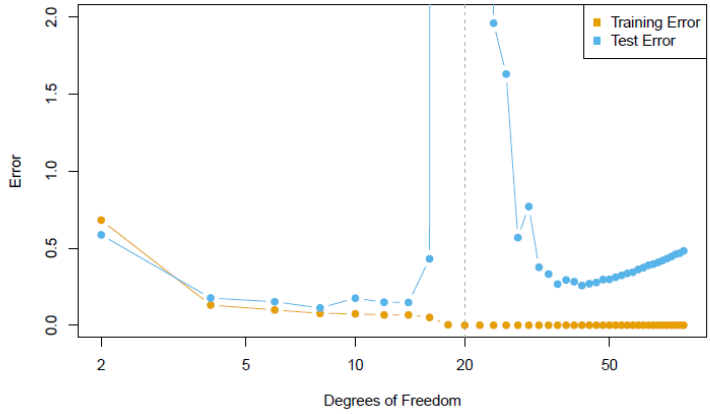
Simulation


8 Degrees of Freedom
20 Degrees of Freedom
42 Degrees of Freedom
80 Degrees of Freedom

Simulation

- Achieving a zero-residual solution with $d = 20$ is a stretch!
  - Only one solution, may need to be very wiggly.
- At d increases, it's easier to achieve this (and there are multiple ways to do it).

Simulation



- When $d \leq 20$, model is OLS and we see the usual bias-variance trade-off.
- When $d > 20$, we revert to minimum-norm.
  - As d increases above 20, $\sum_{j=1}^{d} \hat{\beta}_j^2$ decreases since it gets easier to achieve zero error.