Machine Learning

Lecture 16: Generalization 3

Hao Tang

December 11, 2022

Reusing test sets

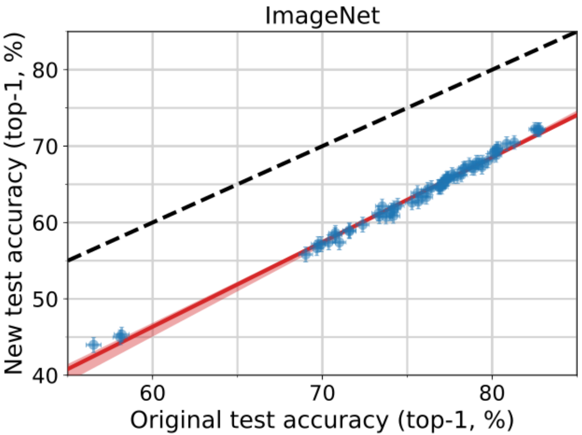


Image credit: (Recht *et al.*, 2019)

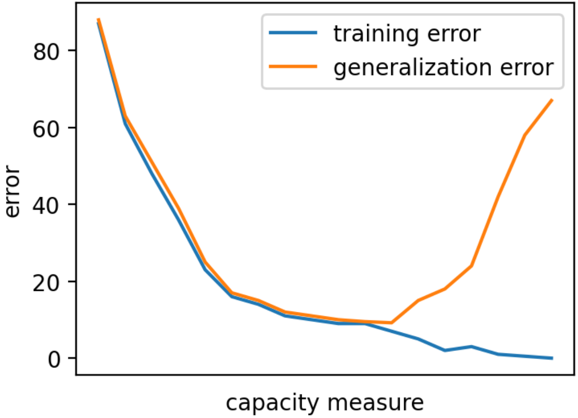
Capacity-generalization tradeoff

- With probability $1 - \delta$, for all $h \in \mathcal{H}$,

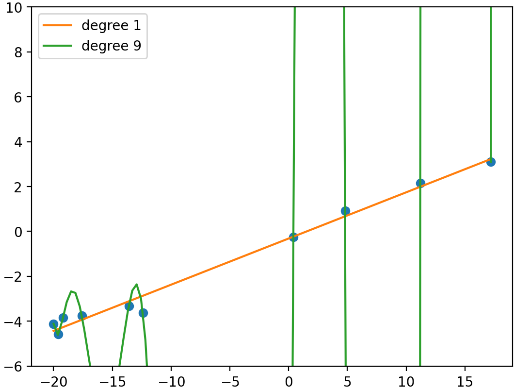
$$L_{\mathcal{D}}(h) \leq L_S(h) + \sqrt{\frac{8d \log(en/d) + \log(1/\delta)}{n}} \tag{1}$$

- As the capacity of $\mathcal{H}$ increases, $\min_{h \in \mathcal{H}} L_S(h)$ drops but the second term goes up.

Capacity-generalization tradeoff



Failure case 2



Large hypothesis classes

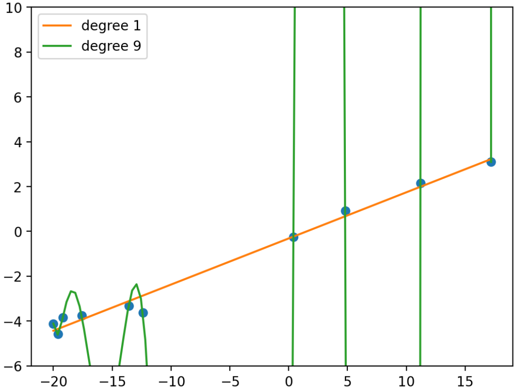
- Compare

$$\mathcal{H}_1 = \text{the set of two-layer neural networks with 512 hidden units} \tag{2}$$

$$\mathcal{H}_2 = \text{the set of all two-layer neural networks} \tag{3}$$

- $\mathcal{H}_1$ has a finite VC dimension, while the VC dimension of $\mathcal{H}_2$ is infinite!
- It is much easier (and tempting) to reduce the training error by increasing the hypothesis class.

Failure case 2



Failure case 2

- Compare

  $w_2 = [0.206, -0.317]$

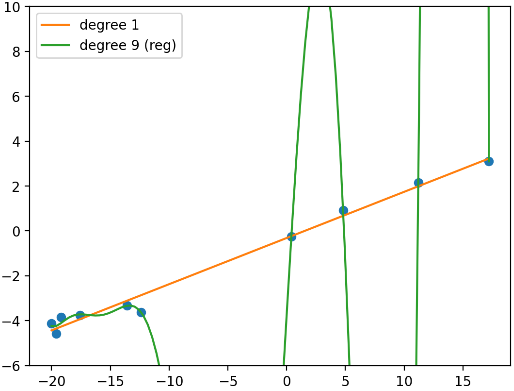
  $w_9 = [-30.69, 93.27, -2.65, -3.29, -0.124, 0.0248, 0.0017, 0.0000245, -0.00000423, -0.0000000857]$

- The learned weights are either too large or too small for degree 9.

- What if instead we optimize

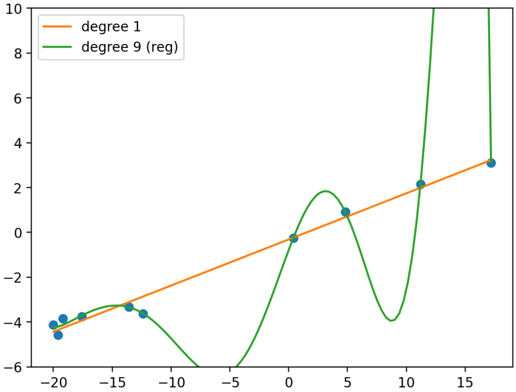
$$\min_{w \in \mathcal{H}} L_S(w) + \frac{\lambda}{2} \|w\|^2 \tag{4}$$

Failure case 2



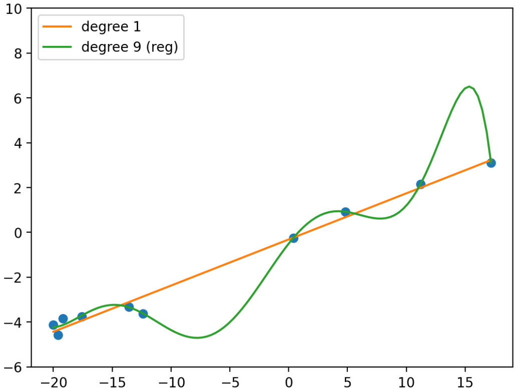
$$\lambda = 0.002$$

Failure case 2



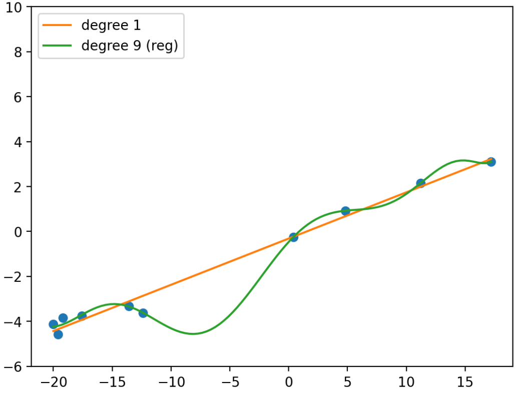
$\lambda = 0.02$

Failure case 2



$$\lambda = 0.1$$

Failure case 2



$$\lambda = 0.2$$

L_2 Regularization

- The term $\frac{\lambda}{2}\|w\|^2$ is called an L_2 regularizer.
- It is also known as weight decay.
- The expression

$$L_S(w) + \frac{\lambda}{2}\|w\|^2 \tag{5}$$

is the Lagrangian of

$$\begin{aligned} \min_w \quad & L_S(w) && (6) \\ \text{s.t.} \quad & \|w\| \leq B && (7) \end{aligned}$$

L_2 Regularization

- The L_2 regularizer has an effect of controlling the capacity of the hypothesis class.

- Compare

$$\mathcal{H} = \{x \mapsto w^\top \phi(x) : w \in \mathbb{R}^d\} \tag{8}$$

$$\mathcal{H} = \{x \mapsto w^\top \phi(x) : \|w\| \leq B\} \tag{9}$$

Shattering

- Given n data points, there are 2^n ways of label them $\{+1, -1\}$.
- A set of n points is shattered by $\mathcal{H}$ if there is an arrangement of n points such that classifiers in $\mathcal{H}$ can produce all 2^n ways of labeling.
- VC dimension is the largest number of points that $\mathcal{H}$ can shatter.

Rademacher complexity

- Rademacher complexity (in binary classification) on a data set S is defined as

$$\mathfrak{R}_S(\mathcal{H}) = \mathbb{E}_\sigma \left[\max_{h \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} \sigma_i h(x_i) \right], \tag{10}$$

  where $\sigma \in \{+1, -1\}^n$ is uniformly chosen.

- In words, Rademacher complexity measures how well a class of classifiers correlate with random noise.

- Rademacher complexity (in binary classification) for n points is defined as

$$\mathfrak{R}_n(\mathcal{H}) = \mathbb{E}_{S \sim \mathcal{D}^n} [\mathfrak{R}_S(\mathcal{H})]. \tag{11}$$

Rademacher generalization bounds

- With probability $1-\delta$, for all $h \in \mathcal{H}$

$$L_{\mathcal{D}}(h) \leq L_S(h) + \mathfrak{R}_n(\mathcal{H}) + \sqrt{\frac{\log(1/\delta)}{2n}} \tag{12}$$

- With probability $1-\delta$, for all $h \in \mathcal{H}$

$$L_{\mathcal{D}}(h) \leq L_S(h) + \mathfrak{R}_S(\mathcal{H}) + 3\sqrt{\frac{\log(2/\delta)}{2n}} \tag{13}$$

Linear classifiers with bounded norm

- If $S = \{x : \|x\| \leq r\}$ and $\mathcal{H} = \{x \mapsto w^\top x : \|w\| \leq B\}$,

$$\mathfrak{R}_S(\mathcal{H}) \leq \sqrt{\frac{r^2 B^2}{n}} \tag{14}$$

Stability

- If we replace a data point in the data set, do you get a very different classifier?
- We say that the learning algorithm is stable is changing a data point does not change the the classifier by much.
- If S is the data set, then $S^{(i)}$ is the same data set with the i-th data point replaced with another random data point.

Stability

- Stable learning algorithms don't overfit.

$$\mathbb{E}_{S\sim\mathcal{D}^n}[L_{\mathcal{D}}(A(S)) - L_S(A(S))] = \mathbb{E}_{\substack{i\sim U(n)\\ S\sim\mathcal{D}^n\\ (x,y)\sim\mathcal{D}}}[\ell(A(S^{(i)})(x_i), y_i) - \ell(A(S)(x_i), y_i)] \tag{15}$$

- Proof

$$\mathbb{E}_S[L_{\mathcal{D}}(A(S))] = \mathbb{E}_S[\mathbb{E}_{(x,y)\sim\mathcal{D}}[\ell(A(S)(x), y)]] = \mathbb{E}_S[\mathbb{E}_{(x,y)\sim\mathcal{D}}[\ell(A(S^{(i)})(x_i), y_i)]] \tag{16}$$

$$\mathbb{E}_S[L_S(A(S))] = \mathbb{E}_S[\mathbb{E}_{i\sim U(n)}[\ell(A(S)(x_i), y_i)]] \tag{17}$$

Lipschitz loss

- If the loss is ρ-Lipschitz continuous,

$$\ell(A(S^{(i)})(x_i), y_i) - \ell(A(S)(x_i), y_i) \leq \rho \|A(S^{(i)}) - A(S)\|. \tag{18}$$

- We only need a bound on $\|A(S^{(i)}) - A(S)\|$.

Lipschitz and strongly convex

- If a function is λ-strongly convex,

$$\frac{\lambda}{2}\|x - x^*\|^2 \leq f(x) - f(x^*) \tag{19}$$

  where x^* is the minimizer.

- If we can bound $f(x) - f(x^*)$, then we can have bound on $\|x - x^*\|$.

- We will then let $x = A(S^{(i)})$ and $x^* = A(S)$.

L_2 regularizer

- $\frac{\lambda}{2}\|w\|^2$ is λ-strongly convex.
- $L_S(w) + \frac{\lambda}{2}\|w\|^2$ is λ-strongly convex if $L_S(w)$ is convex.
- Adding a L_2 regularizer makes learning stable.
- If we choose $A(S) = \operatorname{argmin}_{w \in \mathcal{H}} L_S(w) + \frac{\lambda}{2}\|w\|^2$, we get

$$\|A(S^{(i)}) - A(S)\| \leq \frac{2\rho}{\lambda n}. \tag{20}$$

- In the end, we have

$$\mathbb{E}_{S\sim\mathcal{D}^n}[L_{\mathcal{D}}(A(S)) - L_S(A(S))] \leq \frac{2\rho^2}{\lambda n} \tag{21}$$

Hypothesis class limited by the learning algorithm

- Compare

$$\mathcal{H}_1 = \text{the set of all two-layer neural networks} \tag{22}$$

$$\mathcal{H}_2 = \text{the set of all two-layer neural networks with bounded norm } B \tag{23}$$

$$\mathcal{H}_3 = \text{the set of all two-layer neural networks searched with } t \text{ gradient updates} \tag{24}$$

- $\mathcal{H}_1$ has infinite VC dimension, while the last two has bounded Rademacher complexity.