Recitation on Bias-Variance and Regularization

ESE546

October 2020

Table of Contents

Table of Contents

Standard Supervised Learning

- Task is known. Dataset $\mathcal{D} = \{x_i, y_i\}_{i=1}^n$ is given.
- Choose a hypothesis space $\mathcal{F}$ (eg. a specific architecture of a neural network)

Standard Supervised Learning

- Task is known. Dataset $\mathcal{D} = \{x_i, y_i\}_{i=1}^{n}$ is given.
- Choose a hypothesis space $\mathcal{F}$ (eg. a specific architecture of a neural network)
  - You need to be able to parametrize $f_w \in \mathcal{F}$

Standard Supervised Learning

- Task is known. Dataset $\mathcal{D} = \{x_i, y_i\}_{i=1}^{n}$ is given.
- Choose a hypothesis space $\mathcal{F}$ (eg. a specific architecture of a neural network)
  - You need to be able to parametrize $f_w \in \mathcal{F}$
- Choose a training rule (and a loss):
  - (ERM): $\min_w \sum_{i=1}^{n} l(f_w(x_i), y_i)$
  - (MLE): $\max_w \sum_{i=1}^{n} \ln p(y_i|x_i; w)$
- Those two above are (part of) your "inductive bias".
- Tom Mitchel: *"an inductive bias of a learner is the set of additional assumptions sufficient to justify its inductive inferences as deductive inferences"*.
- Inductive bias if $\mathcal{F}$ is any CNN ?

Standard Supervised Learning

- Can $\mathcal{F}_\theta$ or $Loss_\theta$?

Standard Supervised Learning

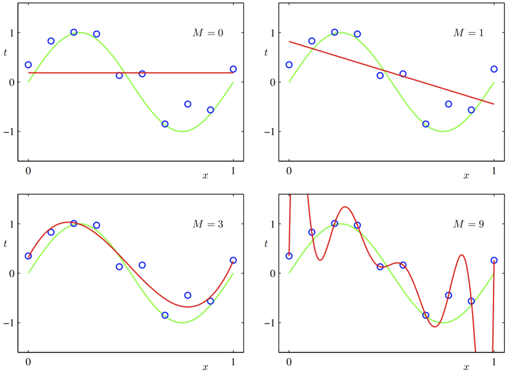
- Can $\mathcal{F}_\theta$ or $Loss_\theta$?
- Polynomials of order M or regularization with parameter λ.
- How can you find those parameters in our supervised setting?

Standard Supervised Learning

- Can $\mathcal{F}_\theta$ or $Loss_\theta$?
- Polynomials of order M or regularization with parameter λ.
- How can you find those parameters in our supervised setting?
- Minimum-cross validation error is actually another inductive bias...

Overfitting

- What is the issue with the rules above?



- Model adapts to noise.
- Complexity of the model should describe the complexity of the task not the amount of data gathered.

Regularization

- Idea! Penalize some models inside the hypothesis space more than others according to complexity.
- How to measure complexity? No. of weights, description length, BIC, AIC ?
- Double descent may indicate that our methods on measuring model complexity is inadequate.
- Note! We should not penalize, equivalent solutions, differently!
- How is training error affected with regularization ?
- Is there such thing as "overfitting to the validation set"?

Weight Shrinking

- In this class a regularizer is an explicit function of the parameters.
- $L'(w) = L(w) + \Omega(w)$.
- We have already seen some regularizers:

Weight Shrinking

- In this class a regularizer is an explicit function of the parameters.
- $L'(w) = L(w) + \Omega(w)$.
- We have already seen some regularizers:
  - Weight Decay: $\Omega(w) = \frac{\lambda}{2}\|w\|^2$

Weight Shrinking

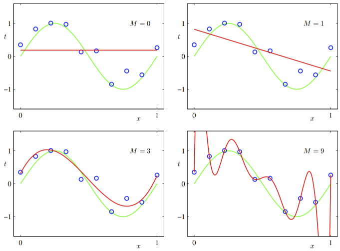
- In this class a regularizer is an explicit function of the parameters.
- $L'(w) = L(w) + \Omega(w)$.
- We have already seen some regularizers:
  - Weight Decay: $\Omega(w) = \frac{\lambda}{2}\|w\|^2$
  - Dropout (linear): $\Omega = \frac{p}{1-p} w^\top diag(X^\top X) w$

Weight Shrinking

- In this class a regularizer is an explicit function of the parameters.
- $L'(w) = L(w) + \Omega(w)$.
- We have already seen some regularizers:
  - Weight Decay: $\Omega(w) = \frac{\lambda}{2}\|w\|^2$
  - Dropout (linear): $\Omega = \frac{p}{1-p} w^\top diag(X^\top X) w$
- What is the reasoning behind penalizing large weights?

Weight Shrinking

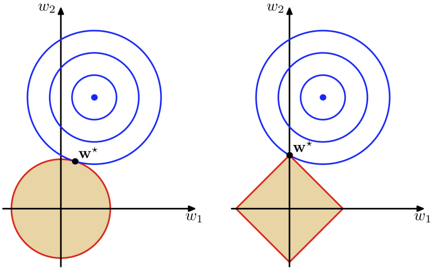
- In this class a regularizer is an explicit function of the parameters.
- $L'(w) = L(w) + \Omega(w)$.
- We have already seen some regularizers:
  - Weight Decay: $\Omega(w) = \frac{\lambda}{2}\|w\|^2$
  - Dropout (linear): $\Omega = \frac{p}{1-p} w^\top diag(X^\top X) w$
- What is the reasoning behind penalizing large weights?



	$M=0$	$M=1$	$M=6$	$M=9$
$w_0^\star$	0.19	0.82	0.31	0.35
$w_1^\star$		-1.27	7.99	232.37
$w_2^\star$			-25.43	-5321.83
$w_3^\star$			17.37	48568.31
$w_4^\star$				-231639.30
$w_5^\star$				640042.26
$w_6^\star$				-1061800.52
$w_7^\star$				1042400.18
$w_8^\star$				-557682.99
$w_9^\star$				125201.43

Weight Decay

- $l(w) + \frac{\lambda}{2}\|w\|^2 + \frac{\lambda_0}{2}|w_0|^2$ $(\lambda_0 = 0)$
- How is regularization fitted into our probabilistic framework?
  - (MAP) $max_w \ln p(w|\mathcal{D}) \propto max_w\{\ln p(Y|X;w) + \ln p(w)\}$
  - Weight decay: $\implies p(w) \sim \mathcal{N}(0, \lambda^{-1}I)$
- During training: $w^{t+1} = (1 - \eta\lambda)w^t - \nabla_w l(w^t)$
  - Can you see the weight shrinking?
- Other regularizers penalize solutions differently:



- Regularization in deep learning is complicated.

Weight Decay in Deep Learning

- Single hidden layer network:

$$z_j = h(w_j^\top x + w_{j0})$$
$$y_k = w_k^\top z + w_{k0}$$

- What if $\tilde{x}_i = ax_i + b$? We expect the network to adjust its weights so that the classification is consistent.
- Indeed,

$$\tilde{w}_{ji} = \frac{1}{a} w_{ji}$$
$$\tilde{w}_{j0} = w_{j0} - \frac{b}{a} \sum_i w_{ji}$$

- Observe how the biases are adjusted.

Weight Decay in Deep Learning

- Single hidden layer network:

$$z_j = h(w_j^\top x + w_{j0})$$
$$y_k = w_k^\top z + w_{k0}$$

- What if $\tilde{y}_k = cy_k + d$? We can again rescale the weights:

$$\tilde{w}_{kj} = cw_{kj}$$
$$\tilde{w}_{k0} = cw_{k0} + d$$

- We should expect consistency out of any reasonable regularizer! Meaning, if we train using the original dataset and then a rescaled version, we should expect same predictions and only rescaled weights.

Weight Decay in Deep Learning

- Single hidden layer network:

$$z_j = h(w_j^\top x + w_{j0})$$
$$y_k = w_k^\top z + w_{k0}$$

- Is weight-decay "linear-transformation-invariant" ?
- As written above, regularizers should not favor one equivalent solution over another.
- Easy fix?

$$\Omega(w) = \frac{\lambda_1}{2} \sum_{w \in \mathcal{W}_1, no \;\; bias} w^2 + \frac{\lambda_2}{2} \sum_{w \in \mathcal{W}_2, no \;\; bias} w^2$$

- $\lambda_1 \to a^{1/2}\lambda_1, \;\; \lambda_2 \to c^{-1/2}\lambda_2$ and we get the scaled down weights!
- One more time: What about the bias?

Bias- Variance Tradeoff

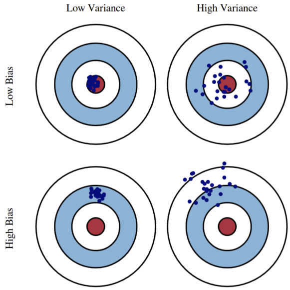
- Optimal Predictor may depend on the distribution $P(x, y)$ but not on the hypothesis class. *(Bayes Optimal Predictor)*.
  - If $P(x|y)$ do not overlap what is its error? Is this always the case?
  - Is $P(x, y)$ known?
- $\mathcal{D} \sim P^n(x, y)$. Also, $\hat{f}(x; \mathcal{D}) = \mathcal{L}(\mathcal{D})$. Learner can use any rule (or loss) to output $\hat{f}$. But we do not care about the learner's loss.
- $(x, y) \sim P(x, y)$: Test data
- Population Risk:

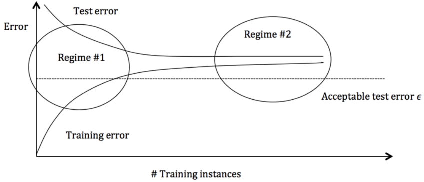
$$
\begin{aligned}
R(\hat{f}) = \mathbb{E}_{(x,y)\sim P, \mathcal{D}\sim P^n}[(\hat{f}(x;\mathcal{D}) - y)^2] = \\
= \mathbb{E}_{(x,y)\sim P}[(f^*(x) - y)^2] + \\
+ \mathbb{E}_{x,\mathcal{D}}[(\hat{f}(x;\mathcal{D}) - \mathbb{E}_{\mathcal{D}}[\hat{f}(x;\mathcal{D})])^2] + \\
+ \mathbb{E}_{x}[(f^*(x) - \mathbb{E}_{\mathcal{D}}[\hat{f}(x;\mathcal{D})])^2]
\end{aligned}
$$

where $f^*(x) = \mathbb{E}_y[y|x]$

Bias- Variance Tradeoff

- Why use a quadratic loss in population risk?
- $R(\hat{f}) = \textit{Bayes error} + \textit{Variance} + \textit{Bias}^2$
- High Bias: Underfitting, Erroneous Assumptions.
- High Variance: Overfitting, Sensitivity to Dataset fluctuations.
- Is the No. of weights a good complexity measure?
  - $f(x) = a\sin(bx)$





Quiz

- Adding More features
- Increasing the Number of Hidden Units
- Bagging.
- Gather More data (same data distribution)
- Data Augmentation.
- Adversarial Training.
- Increase Hypothesis Space.
- Increase p in Dropout (linear).
- Do some weight sharing (eg CNN)
- Early Stopping.