Notes : Week 6 part 1

Advice for Applying machine learning: Deciding what to try next

debugging a learning algorithm:

Suppose you have implemented regularized linear regression to predict housing prices

$$\rightarrow J(\theta) = \frac{1}{2m}\left[\sum_{i=1}^{m}\left(h_\theta(x^{(i)}) - y^{(i)}\right)^2 + \lambda \sum_{j=1}^{m} \theta_j^2\right]$$

However you test your hypothesis on a new set of houses, you find that makes unacceptably large errors in predictions - What should you try next?

- get more training examples.
- try smaller set of features.
- try getting additional features.
- try adding polynomial features (x_1^2, x_2^2, x_1x_2, etc)
- try decreasing λ
- try increasing λ

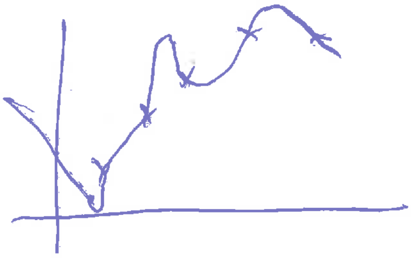
Machine Learning Diagnostic:

Diagnostic: A test you can use to gain insight what is/isn't working with a learning algorithm, and gain guidance as to how best improve its performance

→ Diagnostics can take time to implement, but doing so can be a very good use of time.

Notes: Week 6

Advice for applying machine learning: evaluating a hypothesis



$h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$

fails to generalize to new examples not in training set:

x_1 = size of house
x_2 = no of bedrooms
x_3 = no of floors
⋮
x_{1000}

→ hard to imagine what this function even looks like.

evaluating your hypothesis:

	size	price	
70%	~	~	Training set
30%	~	~	Test set

Training set → $x^{(1)}, y^{(1)}$
$x^{(2)}, y^{(2)}$
⋮
$x^{(m)}, y^{(m)}$

Test set → $x^{(1)}_{test}, y^{(1)}_{test}$
⋮
$x^{(m)}_{test}, y^{(m)}_{test}$

m_{test} = no. of test examples in $(x^{(i)}_{test}, y^{(i)}_{test})$

Training / testing procedure for $\boxed{\text{linear regression}}$

- Learn parameter θ from training data (minimizing training error $J(\theta)$) → 70%. take θ from traing set and plug it in here
- Compute test set error.

$$J_{test}(\theta) = \frac{1}{2m_{test}} \sum_{i=1}^{m_{test}} \left(h_\theta(x^{(i)}_{test}) - y^{(i)}_{test}\right)^2.$$

Training / testing procedure for logistic regression

- Learn parameter θ from training data.
- Compute test set error:

$$J_{test}(\theta) = -\frac{1}{m_{test}} \sum_{i=1}^{m_{test}} y_{test}^{(i)} \log h_\theta(x_{test}^{(i)}) + (1-y_{test}^{(i)}) \log h_\theta(x_{test}^{(i)})$$

- Misclassification error (0/1 misclassification error)

$$err(h_\theta(x), y) = \begin{cases} 1 & \text{if } h_\theta(x) \geq 0.5,\ y=0 \text{ or if } h_\theta(x) < 0.5,\ y=1 \ \} \text{ error} \\ 0 & \rightarrow \text{otherwise} \end{cases}$$

$\rightarrow$ opposite to what we expect

$$\text{Test error} = \underbrace{\frac{1}{m_{test}} \sum_{i=1}^{m_{test}} err(h_\theta(x_{test}^{(i)}), y_{test}^{(i)})}$$

This is exactly the fraction of examples on the test set that the hypothesis has mislabelled.

Notes: Week 6

Advice for applying machine learning: Model selection and training / Validation / test sets

Model selection problem

d=1 1. $h_\theta(x) = \theta_0 + \theta_1 x \rightarrow \theta^{(1)}$ — $J_{test}(\theta^{(1)})$

d = degree of polynomial

d=2 2. $h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 \rightarrow \theta^{(2)} \rightarrow J_{test}(\theta^{(2)})$

d=3 3. $h_\theta(x) = \theta_0 + \theta_1 x + \dots + \theta_3 x^3 \rightarrow \theta^{(3)} \rightarrow J_{test}(\theta^{(3)})$

test set error.

⋮

etc d=10 10. $h_\theta(x) = \theta_0 + \theta_1 x + \dots + \theta_{10} x^{10} \rightarrow \underbrace{\theta^{(10)}}$ — $J_{test}(\theta^{(10)})$

thetas from different hypotheses.

Compute all $J_{test}(\theta)$'s

See which model has the lowest test set error

so for this example choose:

$\theta_0 + \dots \theta_5 x^5$

How well does the model generalize?
Report test set error $\boxed{J_{test}(\theta^{(5)})}$

Problem: $J_{test}(\theta^{(5)})$ is likely to be an optimistic estimate of generalization error i.e. our extra parameter (d = degree of polynomial) is fit to the test set.

↳ just like in the ~~past~~ extra practise exercise in homework exercise 4. The NN overfit our training data (the handwritten characters), however this highly optimized NN would not be all that useful for say a new dataset of handwritten characters

Notes: Week 6

Advice for applying machine learning: Model selection and training / Validation / test sets

evaluating your hypothesis

Dataset:

	Size	Price	
60%	~	~	Training set.
20%	~	~	Cross Validation Set (CV)
20%	~	~	test set.

$(x^{(1)}, y^{(1)})$
$(x^{(2)}, y^{(2)})$
$\vdots$
(x^{m}, y^{m})

$(x_{cv}^{(1)}, y_{cv}^{(1)})$
$(x_{cv}^{(2)}, y_{cv}^{(2)})$
$\vdots$
$(x_{cv}^{m_{cv}}, y_{cv}^{m_{cv}})$

?

m_{cv} = no. of CV example.

$(x_{cv}^{(i)}, y_{cv}^{(i)})$

$(x_{test}^{(1)}, y_{test}^{(1)})$
$(x_{test}^{(2)}, y_{test}^{(2)})$
$\vdots$
$x_{test}^{(m_{test})}, y_{test}^{m_{test}}$

m_{test}

Train / Validation / test error

Training error:

$$J_{train}(\theta) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta(x^{(i)}) - y^{(i)}\right)^2.$$

Cross - Validation error

$$J_{cv}(\theta) = \frac{1}{2m_{cv}}\sum_{i=1}^{m_{cv}}\left(h_\theta(x_{cv}^{(i)}) - y_{cv}^{(i)}\right)^2$$

Test error:

$$J_{test}(\theta) = \frac{1}{2m_{test}}\sum_{i=1}^{m_{test}}\left(h_\theta(x_{test}^{(i)}) - y_{test}^{(i)}\right)^2$$

Model Selection

$d=1$ 1. $h_\theta(x) = \theta_0 + \theta_1 x \rightarrow \min_\theta J(\theta) \rightarrow \theta^1 \rightarrow J_{cv}(\theta^{(1)})$

$d=2$ 2. $h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 \longrightarrow \theta^2 \longrightarrow J_{cv}(\theta^{(2)})$ lowest c.v.

$\boxed{e.g}$ Pick: $J_{cv}(\theta^{(4)})$ e.g. lets assume that θ^4 had the lowest error.

⋮

$d=10$ 10. $h_\theta(x) = \theta_0 + \theta_1 x + \dots + \theta_{10} x^{10} \longrightarrow \theta^{10} \longrightarrow J_{cv}(\theta^{10})$

pick $\theta_0 + \theta_1 x_1 + \dots + \theta_4 x^4$ $d=4$

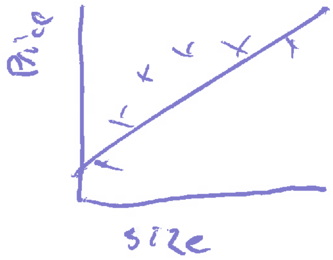
estimate generalization error for test set: $J_{test}(\theta^4)$

Notes : Week 6

Advice for applying machine learning : Diagnosing bias vs. Variance

bias = underfitting
variance = overfitting

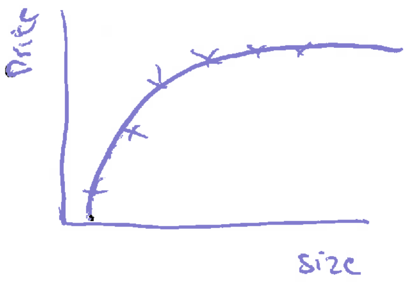
bias / variance



high bias
(underfit)

$d=1$

$\theta_0+\theta_1 x$



"Just right"

$d=2$.

$\theta_0+\theta_1 x+\theta_2 x^2$

High variance
(overfit)

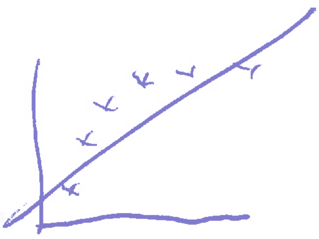
$d=4$

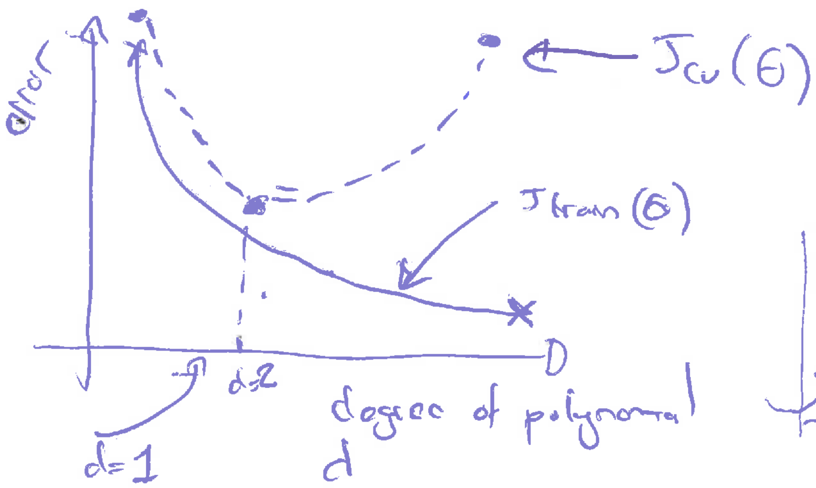
$\theta_0+\theta_1 x+\theta_2 x^2$
$+\theta_3 x^3+\theta_4 x^4$

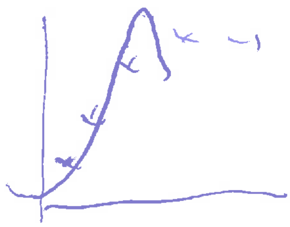
Bias / variance

Training error : $J_{train}(\theta) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$

Cross validation error : $J_{cv}(\theta) = \frac{1}{2m_{cv}}\sum_{i=1}^{m_{cv}}\left(h_\theta(x_{cv}^{(i)}) - y_{cv}^{(i)}\right)^2$

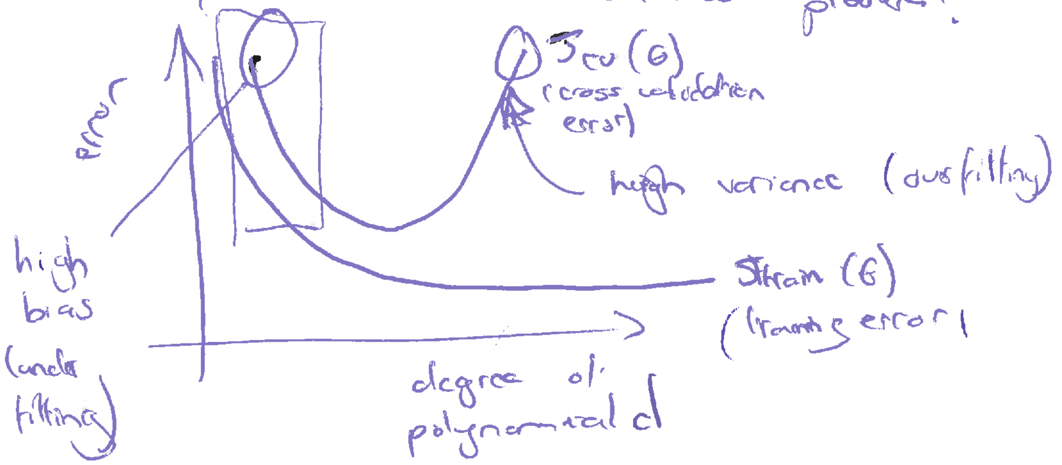






Diagnosing bias vs. variance

Suppose your learning algorithm is performing less well than I was hoping. ($J_{cv}(\theta)$ or $J_{test}(\theta)$ is high) Is it a bias problem or a variance problem?



Bias (underfit)

$J_{train}(\theta)$ will be high

$J_{cv}(\theta) \approx J_{train}(\theta)$

Variance (overfit)

$J_{train}(\theta)$ will be low

$J_{cv}(\theta) \gg J_{train}$

much greater than.

Note: Week 6

Advice for applying machine learning: Regularization and bias/variance

Linear regression with regularization

Model: $h_\theta(x) = \theta_0 + \theta_1 x \ldots \theta_4 x^4$

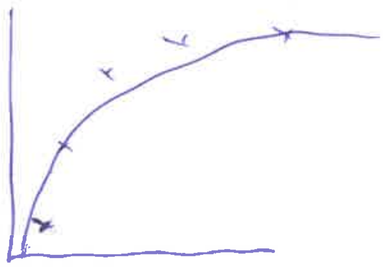
$$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta(x^{(i)}) - y^{(i)}\right)^2 + \boxed{\frac{\lambda}{2m}\sum_{j=1}^{m}\theta_j^2}$$

Large λ

High bias (underfit)

$\lambda = 10000$ $\theta_1 = 0, \theta_2 = 0$

$h_\theta(x) \cancel{\approx} \theta_0$



Intermediate λ

Just right

Small λ

High variance (overfit)

$\lambda = 0$

Choosing the regularization parameter λ

Model: $h_\theta(x) = \theta_0 + \theta_1 x \ldots \theta_4 x^4$

$$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta(x^{(i)}) - y^{(i)}\right)^2 + \frac{\lambda}{2m}\sum_{j=1}^{m}\theta_j^2$$

1) try $\lambda = 0$ $\rightarrow \min_\theta J(\theta) \rightarrow \theta^{(1)} \rightarrow J_{cv}(\theta^{(1)})$

2) $\lambda = 0.01$ $\rightarrow \min_\theta J(\theta) \rightarrow \theta^{(2)} \rightarrow J_{cv}(\theta^{(2)})$

3) $\lambda = 0.02$ $\quad\theta^{(3)}$

4) $\lambda = 0.04$

5) $\lambda = 0.08$ $\quad \theta^{(5)} \rightarrow J_{cv}(\theta^{(5)})$

12) $\lambda = 10$ $\rightarrow \theta^{(12)} \rightarrow J_{cv}(\theta^{(12)})$

multiples of 2.

pick which ever λ gives the lowest error of the cross-validation set.

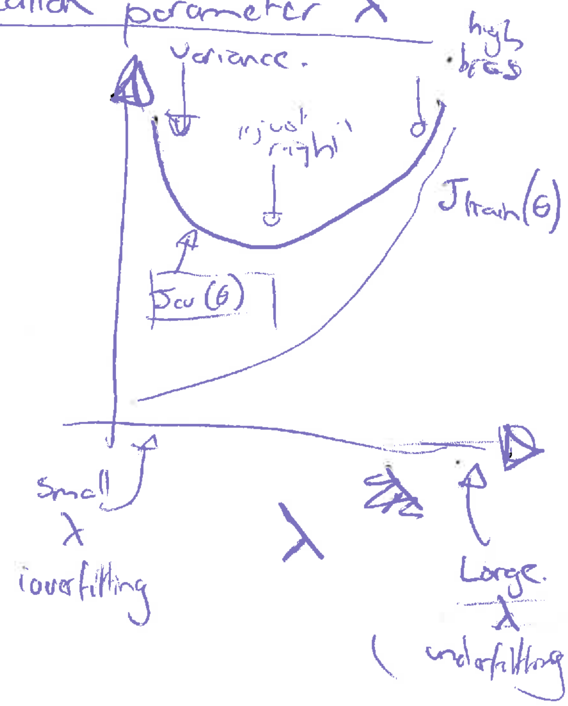
Pick (say) $\theta^{(5)}$. Test error: $J_{test}(\theta^{(5)})$

Bias / variance as a function of the regularization parameter λ

$$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta(x^{(i)})-y^{(i)}\right)^2 + \frac{\lambda}{2m}\sum_{j=1}^{m}\theta_j^2$$

$$J_{train}(\theta) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta(x^{(i)})-y^{(i)}\right)^2$$

$$\boxed{J_{cv}(\theta)} = \frac{1}{2m_{cv}}\sum_{i=1}^{m_{cv}}\left(h_\theta(x_{cv}^{(i)})-y_{cv}^{(i)}\right)^2$$



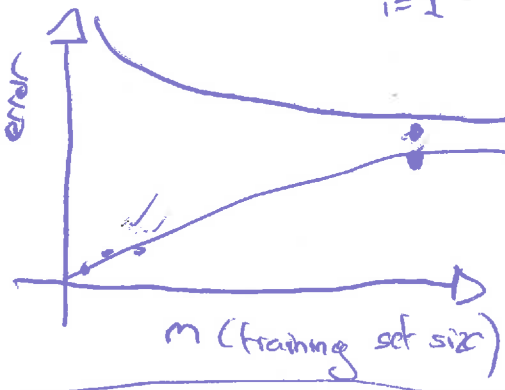
Week: 6

Advice for applying machine learning: learning curves

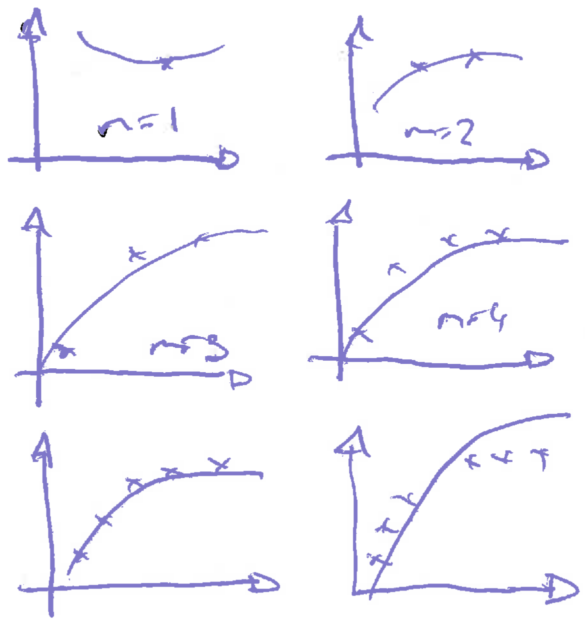
learning curves

$$J_{train}(\theta) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$

$$J_{cv}(\theta) = \frac{1}{2m_{cv}}\sum_{i=1}^{m_{cv}}\left(h_\theta(x^{(i)}_{cv}) - y^{(i)}_{cv}\right)^2$$

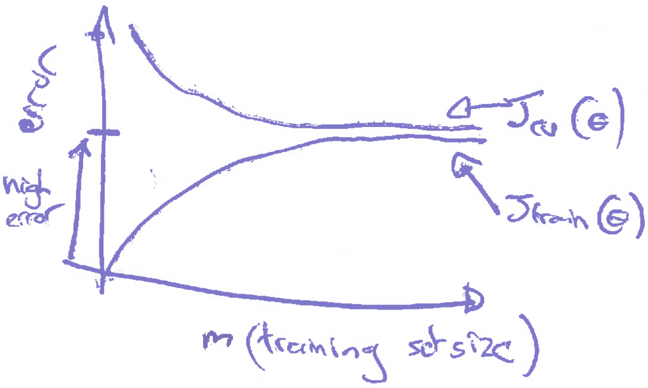


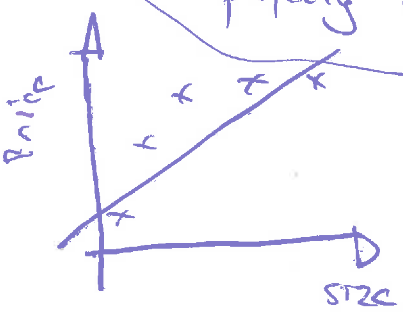
$h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2$



harder and harder to find a quadratic function that fits the data perfectly as $d \to 0 \ldots \infty +$

High bias (underfitting)

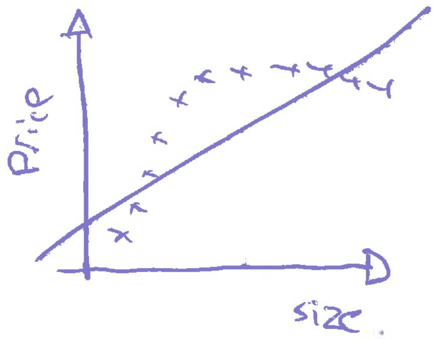




$\boxed{h_\theta(x) = \theta_0 + \theta_1 x}$

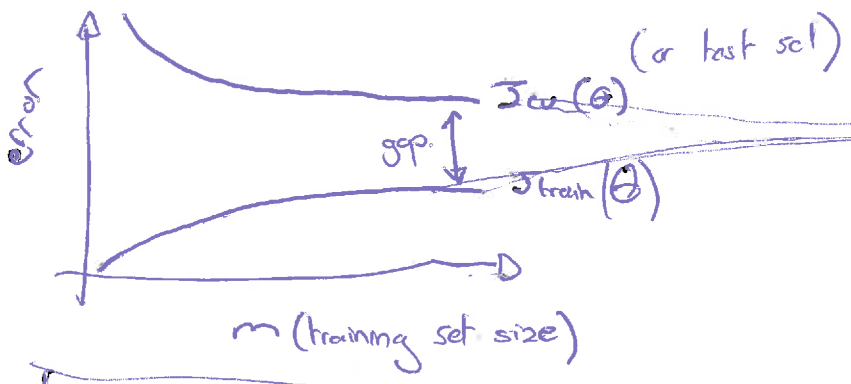
straight line

If a learning algorithm is suffering from high bias, getting more training data will not (by itself) help much.



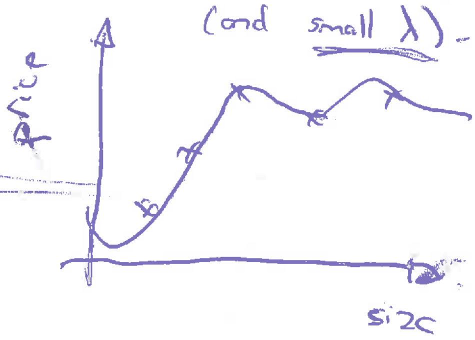
more data
↓
still underfitting (high bias)

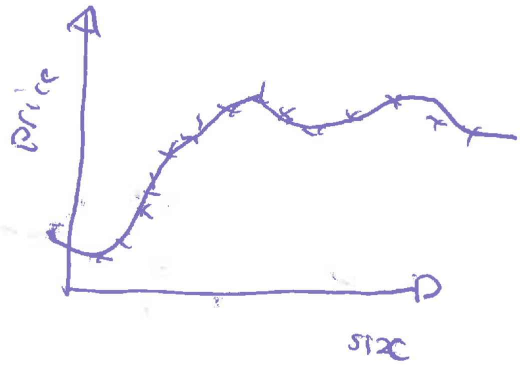
High variance



$h_\theta(x) = \theta_0 + \theta_1 x + \dots + \theta_{100} x^{100}$
(and small λ).

high order polynomial





If a learning algorithm is suffering from high variance, getting more training data is likely to help.

Week 6

Advice for applying machine learning: deciding what to try next (revisited)

debugging a learning algorithm

- get more examples → fixes high variance
- try smaller sets of features → fixes high variance
- try getting additional features → fixing high bias
- try adding polynomial features → ~~adding polynomial features~~ fixes high bias
- try decreasing λ → fixes high bias (underfitting)
- " increasing λ → fixes high variance (overfitting)

Neural networks and overfitting

"small" neural network:
(fewer parameters; more prone to underfitting)

→ computationally cheaper

"large" neural network
(more parameters; more prone to overfitting)

→ Computationally more expensive.

Use regularization (λ) to address overfitting