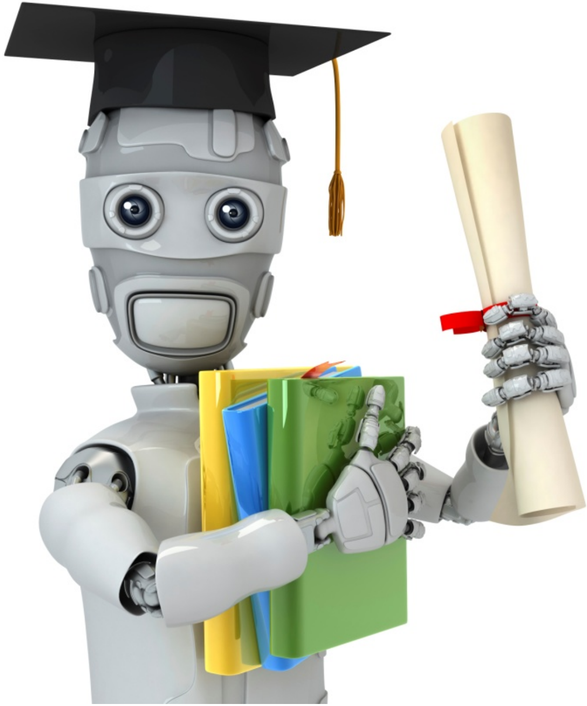


Machine Learning

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.

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

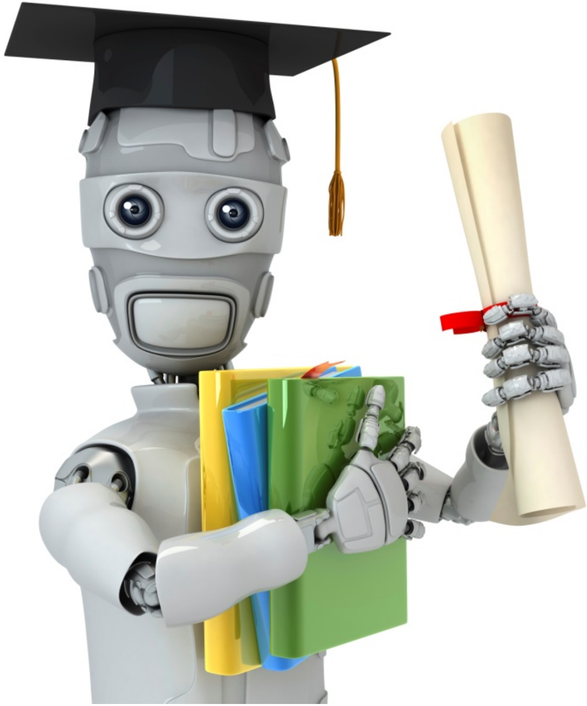
However, when you test your hypothesis on a new set of houses, you find that it makes unacceptably large errors in its predictions. What should you try next?

- Get more training examples
- Try smaller sets of features $x_1, x_2, x_3, \ldots, x_{100}$
- Try getting additional features
- Try adding polynomial features $(\underline{x_1^2}, \underline{x_2^2}, \underline{x_1 x_2}, \text{etc.})$
- Try decreasing λ
- Try increasing λ

Machine learning diagnostic:

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

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

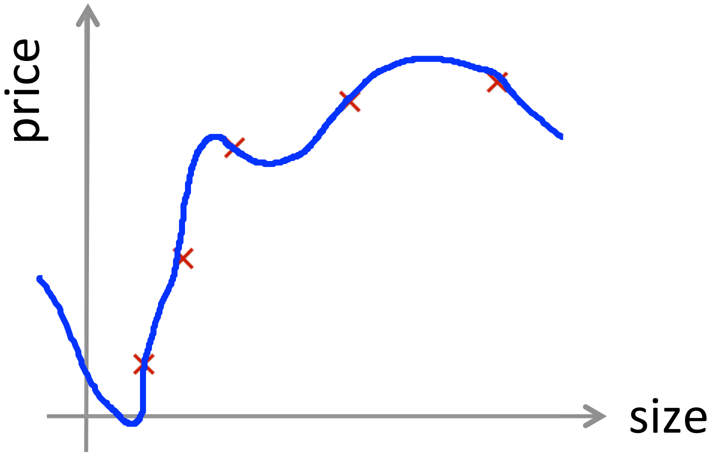


Machine Learning

Advice for applying machine learning

Evaluating a hypothesis

Evaluating your 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

$x_4 =$ age of house

$x_5 =$ average income in neighborhood

$x_6 =$ kitchen size

$\vdots$

x_{100}

Evaluating your hypothesis

Dataset:

Size	Price
2104	400
1600	330
2400	369
1416	232
3000	540
1985	300
1534	315
1427	199
1380	212
1494	243

70%

Training set

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

30%

Test set

$$(x^{(1)}_{test}, y^{(1)}_{test})$$
$$(x^{(2)}_{test}, y^{(2)}_{test})$$
$$\vdots$$
$$(x^{(m_{test})}_{test}, y^{(m_{test})}_{test})$$

m_{test} = no. of test example

$(x^{(1)}_{test}, y^{(1)}_{test})$

Andrew Ng

Training/testing procedure for linear regression

→ - Learn parameter $\underline{\theta}$ from training data (minimizing training error $\underbrace{J(\theta)}_{70\%}$)

- Compute test set error:

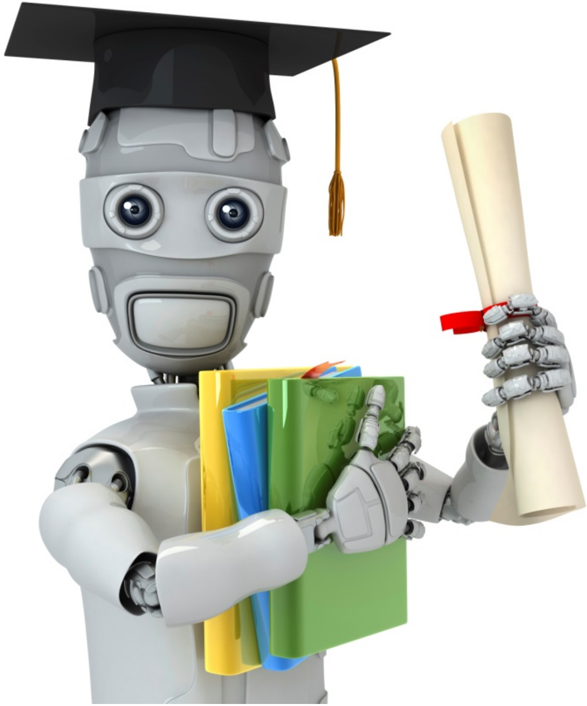
$$\underset{\uparrow}{J_{test}(\theta)} = \frac{1}{2m_{test}} \sum_{i=1}^{m_{test}} \underset{\uparrow}{\underline{\left(h_\theta(x^{(i)}_{test}) - y^{(i)}_{test}\right)^2}}$$

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):

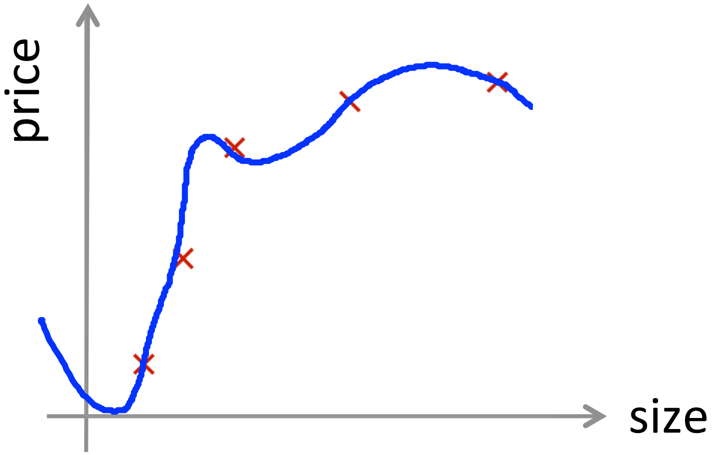


Machine Learning

Advice for applying machine learning

Model selection and training/validation/test sets

Overfitting example



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

Once parameters $\theta_0, \theta_1, \ldots, \theta_4$ were fit to some set of data (training set), the error of the parameters as measured on that data (the training error $J(\theta)$) is likely to be lower than the actual generalization error.

$\rightarrow \boxed{d = \text{degree of polynomial}}$

Model selection

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

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

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

$\vdots \qquad \vdots \qquad \vdots$

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

Choose $\boxed{\theta_0 + \ldots \theta_5 x^5} \leftarrow$

You should not use the test set to choose the regularization parameter, as you will then have an artificially low value for test error and it will not give a good estimate of generalization error. The cross validation lets us find the “just right” setting of the regularization parameter given the fixed model parameters learned from the training set. We can then use this to find the test error without risking an optimistic estimate of generalization error.

$\boxed{\Theta_0, \Theta_1, \ldots}$

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

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

We should not use test set data to choose any of the parameters (regularization and model parameters).If we develop new features by examining the test set, then we may end up choosing features that work well specifically for the test set, so Jtest(θ) is no longer a good estimate of how well we generalize to new examples.

Evaluating your hypothesis

Dataset:

Size	Price
2104	400
1600	330
2400	369
1416	232
3000	540
1985	300
1534	315
1427	199
1380	212
1494	243

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:

$$\rightarrow \quad J_{train}(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2 \qquad J(\theta)$$

Cross Validation error:

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

Test error:

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

Model selection

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

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

$d=3$ 3. $h_\theta(x) = \theta_0 + \theta_1 x + \cdots + \theta_3 x^3 \longrightarrow \theta^{(3)}$

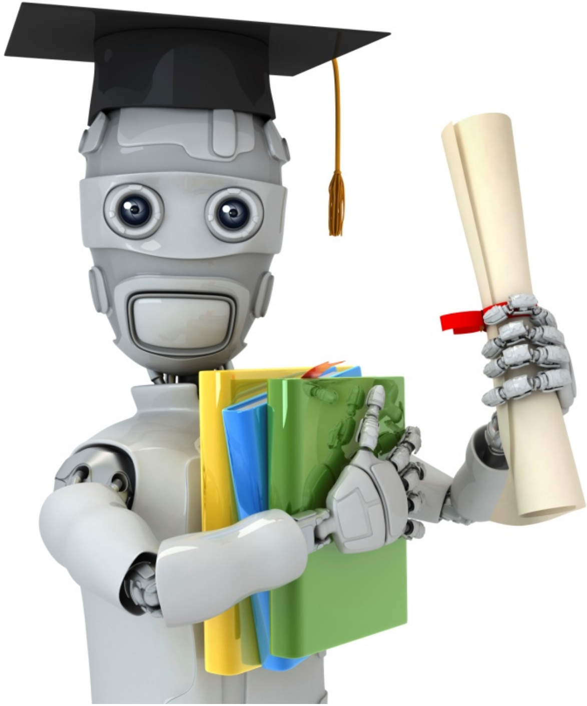
$\vdots \qquad \vdots \qquad \vdots \qquad J_{cv}(\theta^{(4)})$

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

$\underline{d=4} \longrightarrow$

Pick $\theta_0 + \theta_1 x_1 + \cdots + \theta_4 x^4$ ←

Estimate generalization error for test set $\underline{J_{test}(\theta^{(4)})}$ ←

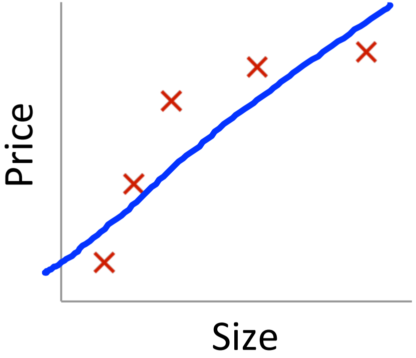


Machine Learning

Advice for applying machine learning

Diagnosing bias vs. variance

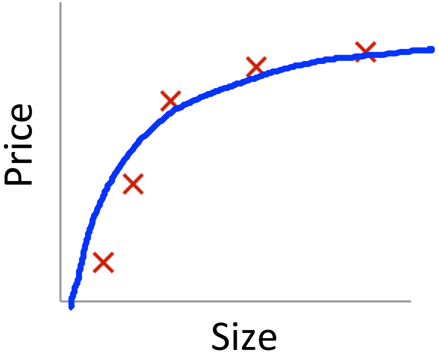
Bias/variance



$$\theta_0 + \theta_1 x$$

High bias
(underfit)

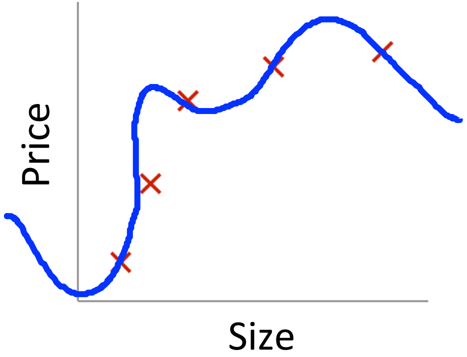
$d=1$



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

"Just right"

$d=2$



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

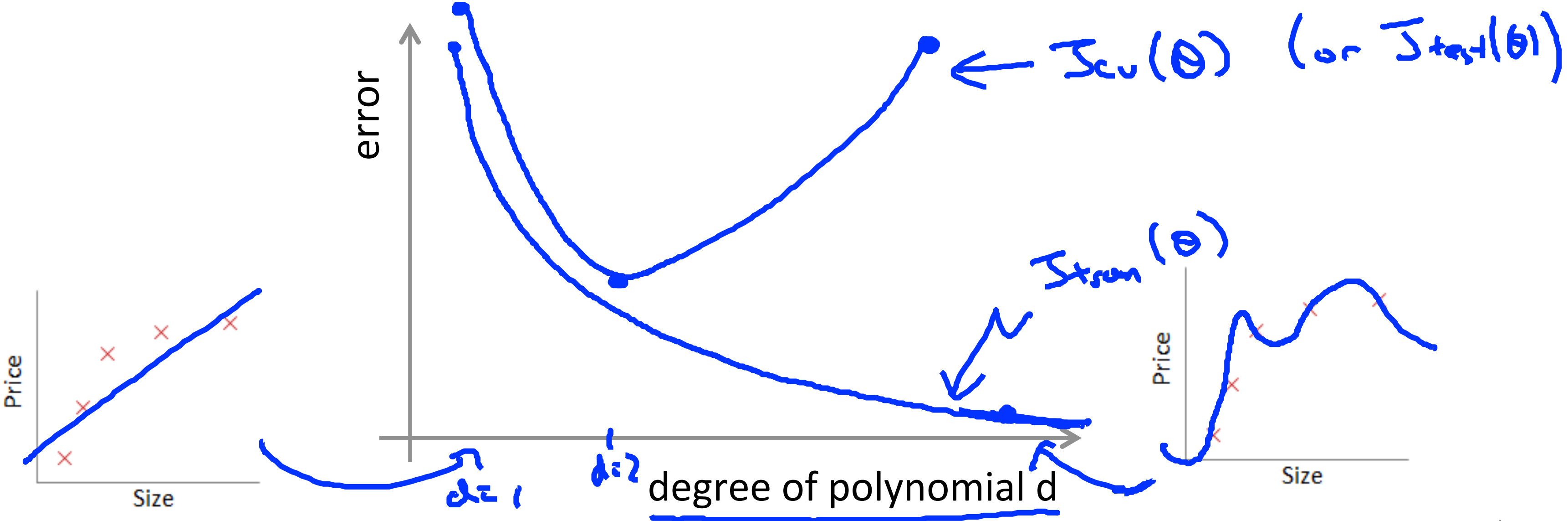
High variance
(overfit)

$d=4$

Bias/variance

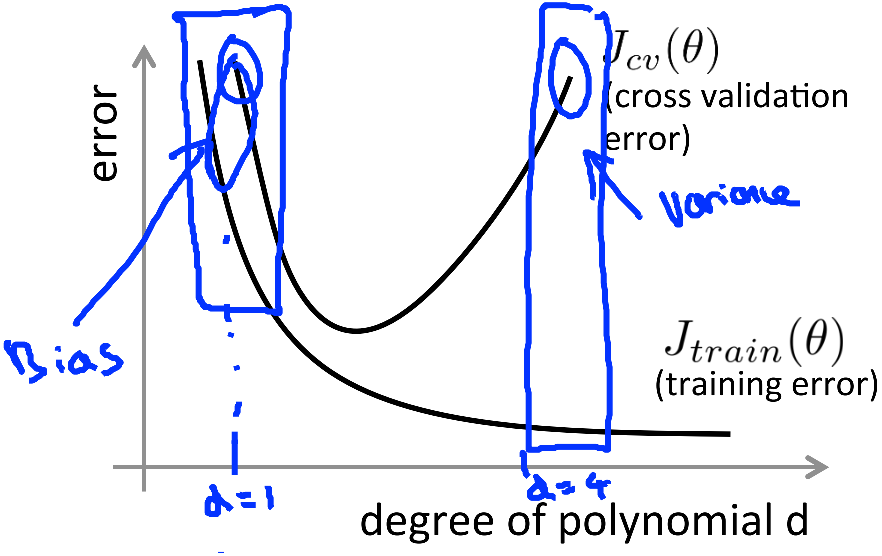
Training error: $J_{train}(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2$

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



Diagnosing bias vs. variance

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



Bias (underfit):

→ $J_{train}(\theta)$ will be high

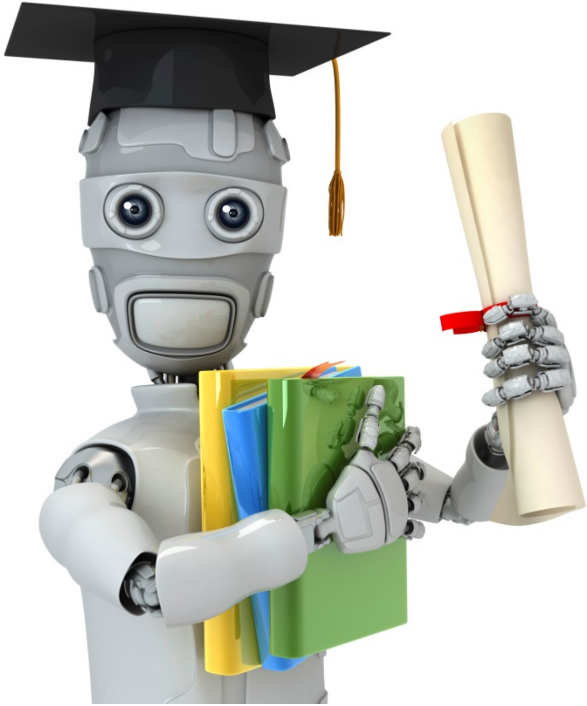
$J_{cv}(\theta) \approx J_{train}(\theta)$

Variance (overfit):

→ $J_{train}(\theta)$ will be low

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

$\gg$



Machine Learning

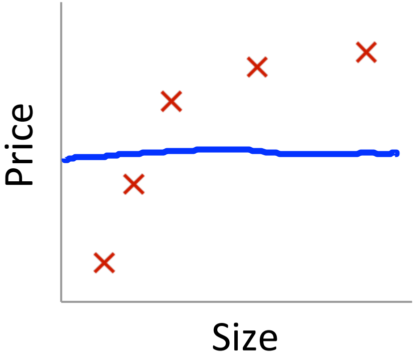
Advice for applying machine learning

Regularization and bias/variance

Linear regression with regularization

Model: $\boxed{h_\theta(x) = \theta_0} + \underline{\theta_1} x + \underline{\theta_2} x^2 + \underline{\theta_3} x^3 + \theta_4 x^4$ ←

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

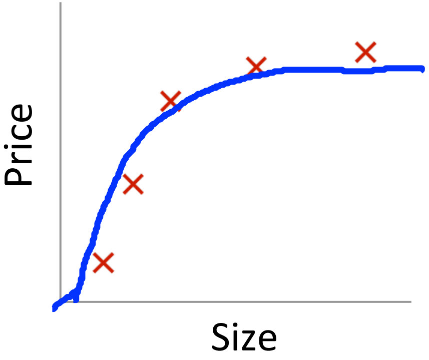


Large λ ←

→ High bias (underfit)

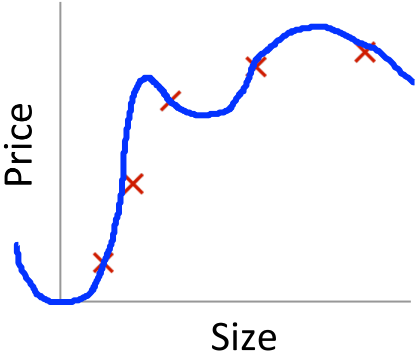
→ $\lambda = 10000.\ \underline{\theta_1 \approx 0, \theta_2 \approx 0,} \ldots$

$h_\theta(x) \approx \theta_0$



Intermediate λ ←

“Just right”



→ Small λ

High variance (overfit)

→ $\lambda = 0$

Choosing the regularization parameter λ

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

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

$J(\theta)$

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

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

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

J_{train}

J_{cv}

J_{test}

Choosing the regularization parameter λ

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

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

1. Try $\lambda = 0$ $\rightarrow \min_\theta J(\theta) \rightarrow \theta^{(1)} \rightarrow J_{cv}(\theta^{(1)})$
2. Try $\lambda = 0.01$ $\rightarrow \min_\theta J(\theta) \rightarrow \theta^{(2)} \rightarrow J_{cv}(\theta^{(2)})$
3. Try $\lambda = 0.02$ $\rightarrow \theta^{(3)} \rightarrow J_{cv}(\theta^{(3)})$
4. Try $\lambda = 0.04$

   $\vdots$
5. Try $\lambda = 0.08$ $\rightarrow \theta^{(5)}$ $J_{cv}(\theta^{(5)})$

   $\vdots$

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

0.24

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

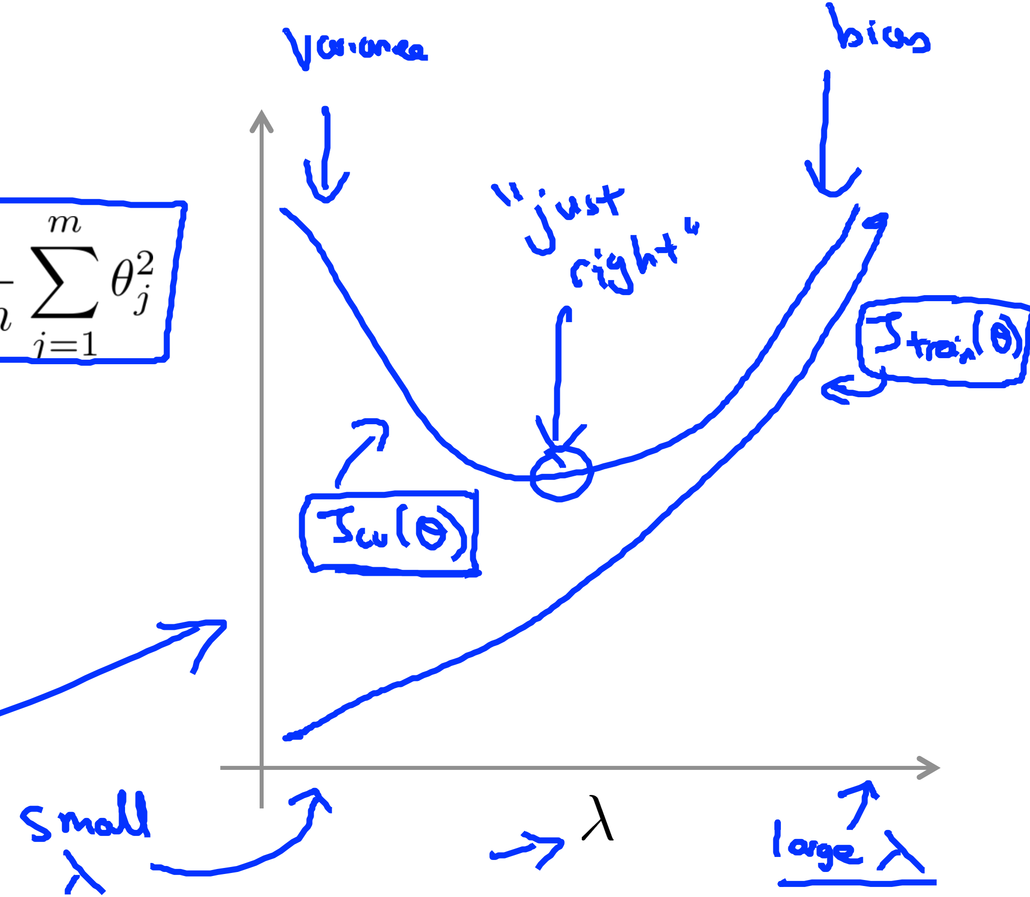
Andrew Ng

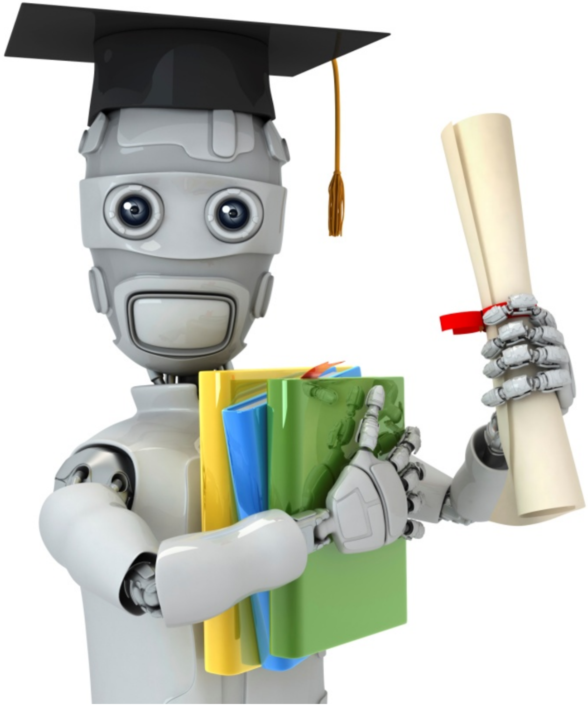
Bias/variance as a function of the regularization parameter λ

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

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

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





Machine Learning

Advice for applying machine learning

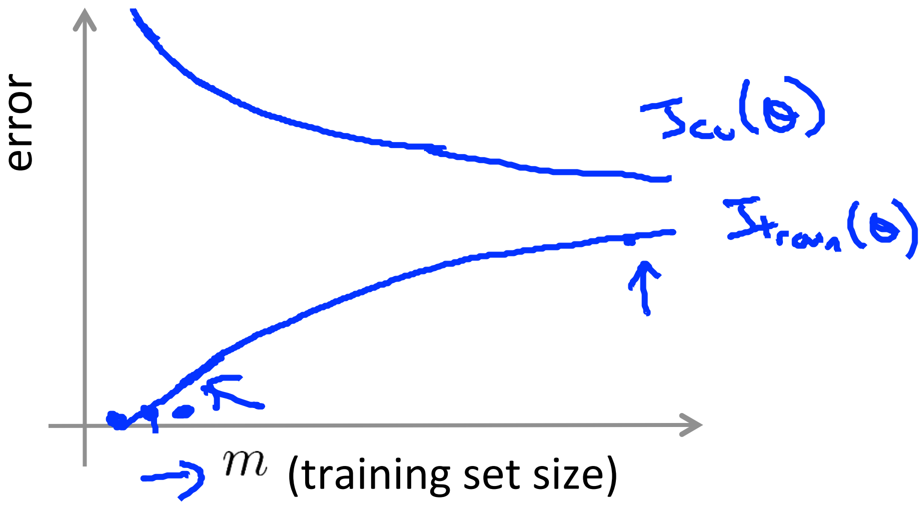
Learning curves

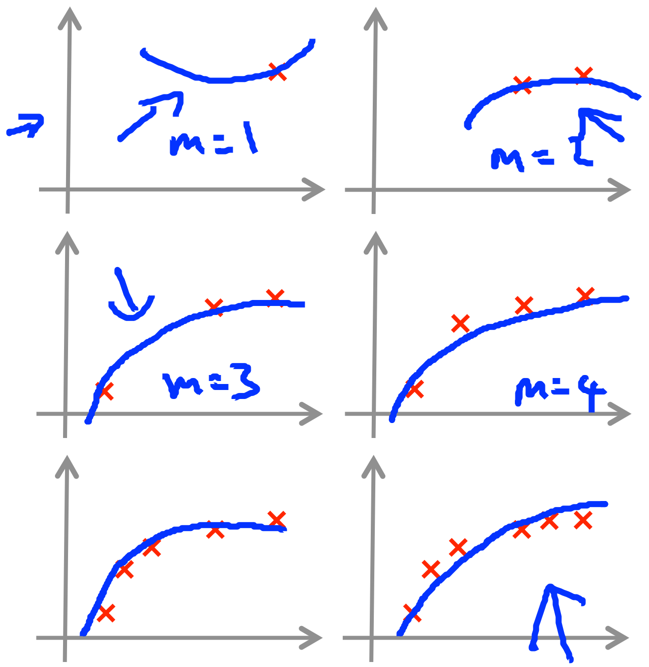
Learning curves

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

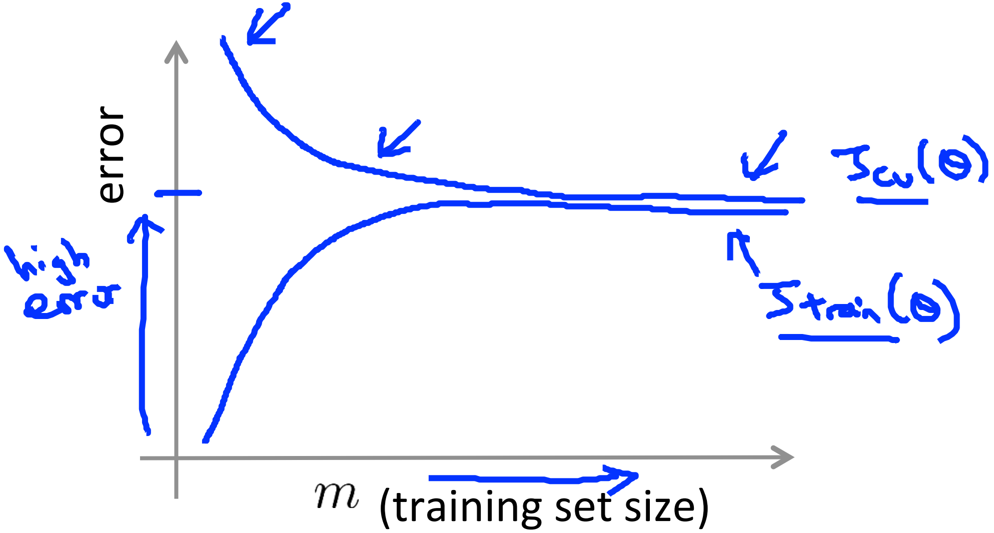
$$\underline{J_{train}(\theta)} = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$

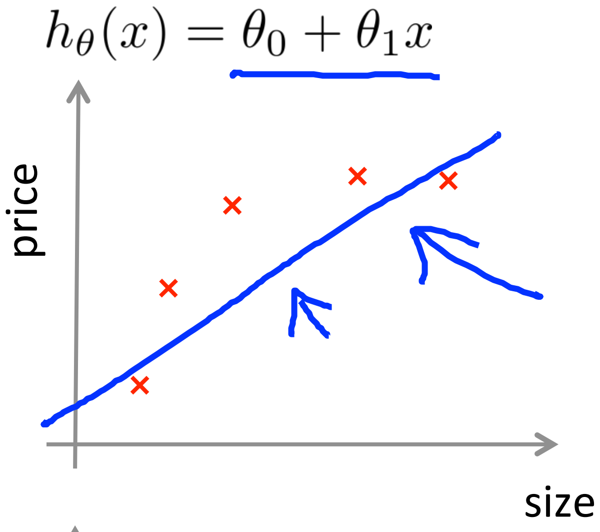
$$J_{cv}(\theta) = \frac{1}{2m_{cv}}\sum_{i=1}^{m_{cv}}\left(h_\theta(x^{(i)}_{cv}) - y^{(i)}_{cv}\right)^2$$



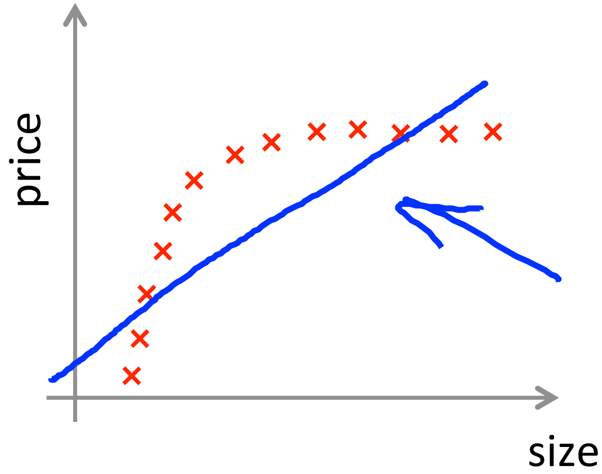


High bias





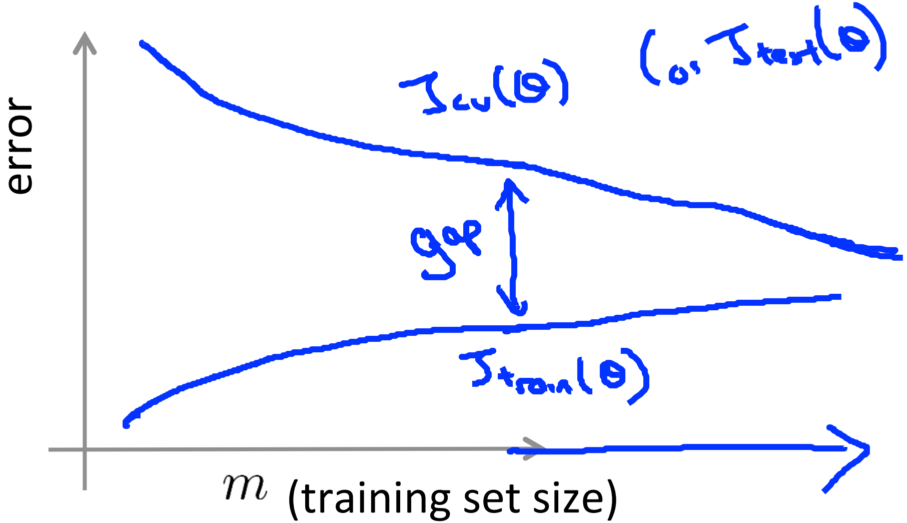
If a learning algorithm is suffering from high bias, getting more training data will not (by itself) help much.

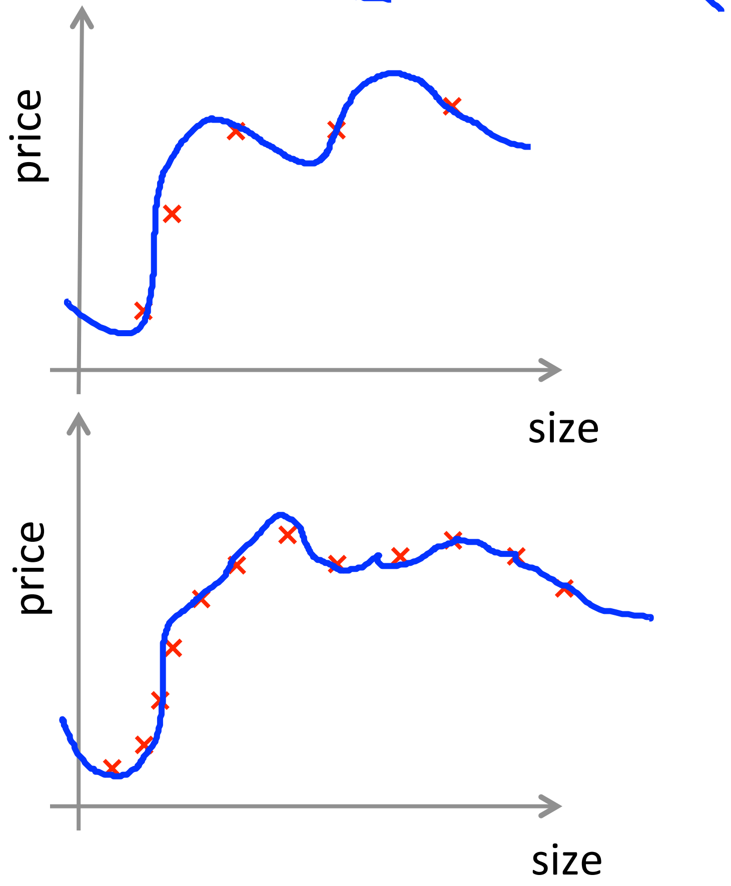


High variance

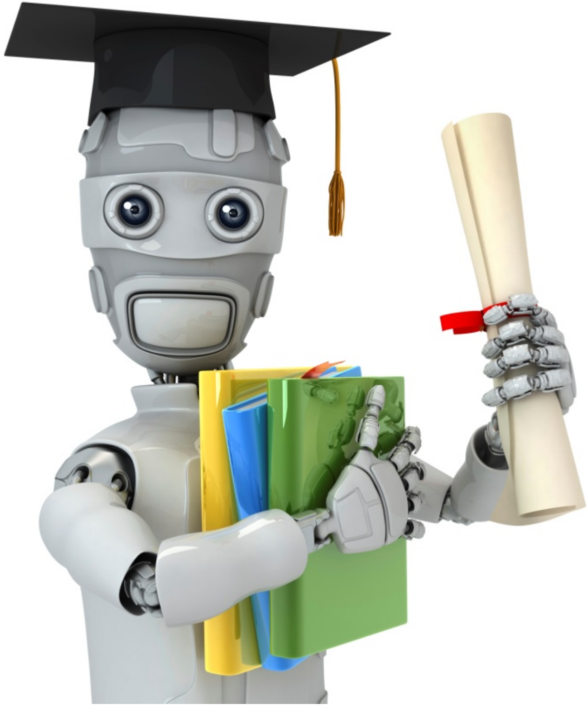
$$h_\theta(x) = \theta_0 + \theta_1 x + \cdots + \theta_{100} x^{100}$$

(and small λ)





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



Machine Learning

Advice for applying machine learning

Deciding what to try next (revisited)

Debugging a learning algorithm:

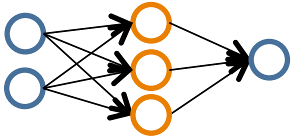
Suppose you have implemented regularized linear regression to predict housing prices. However, when you test your hypothesis in a new set of houses, you find that it makes unacceptably large errors in its prediction. What should you try next?

- Get more training examples → fixes high variance
- Try smaller sets of features → fixes high variance
- Try getting additional features → fixes high bias
- Try adding polynomial features $(x_1^2, x_2^2, x_1x_2, \text{etc})$ → fixes high bias.
- Try decreasing λ → fixes high bias
- Try increasing λ → fixes high variance

Adding more layers will increase model complexity, making the variance problem worse.

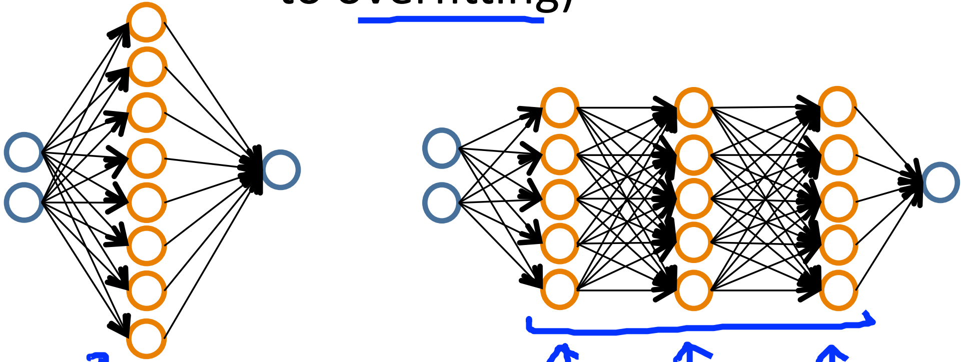
Neural networks and overfitting

"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.

$J_{cv}(\Theta)$

Andrew Ng