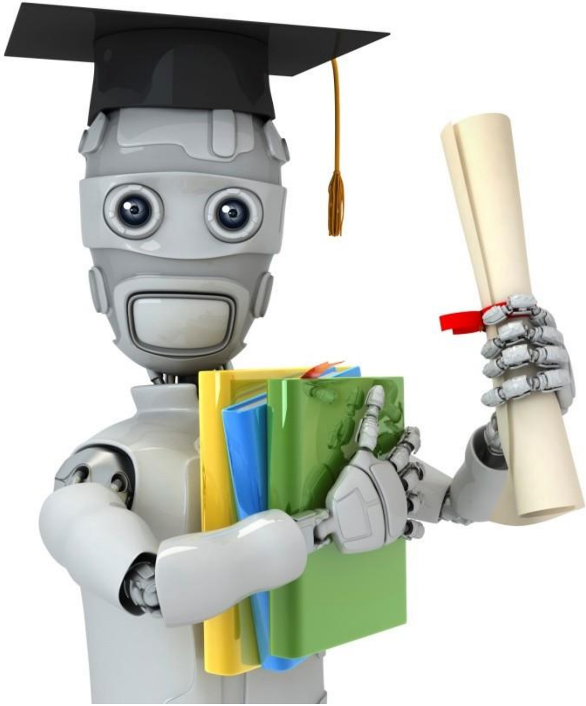


Machine Learning

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 $(x_1^2, x_2^2, x_1x_2, \text{etc.})$
- Try decreasing λ
- Try increasing λ

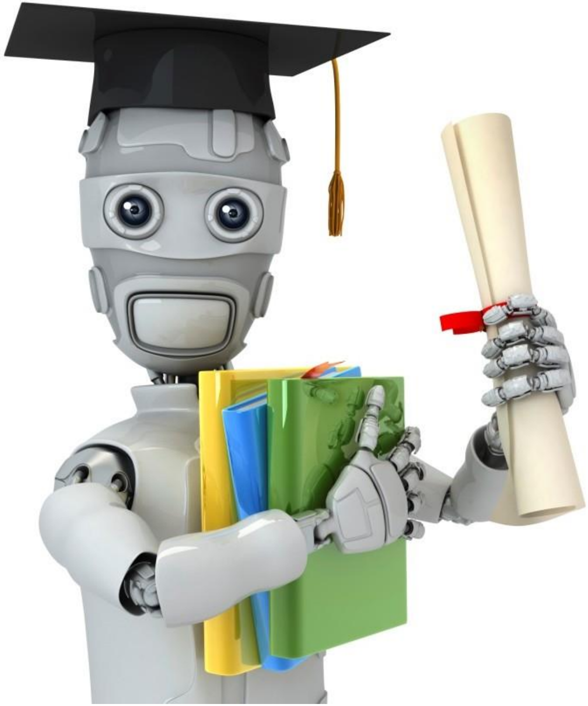
Machine learning diagnostic:

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.

Which of the following statements about diagnostics are true? Check all that apply.

- [ ] It’s hard to tell what will work to improve a learning algorithm, so the best approach is to go with gut feeling and just see what works.
- [ ] Diagnostics can give guidance as to what might be more fruitful things to try to improve a learning algorithm.
- [ ] Diagnostics can be time-consuming to implement and try, but they can still be a very good use of your time.
- [ ] A diagnostic can sometimes rule out certain courses of action (changes to your learning algorithm) as being unlikely to improve its performance significantly.

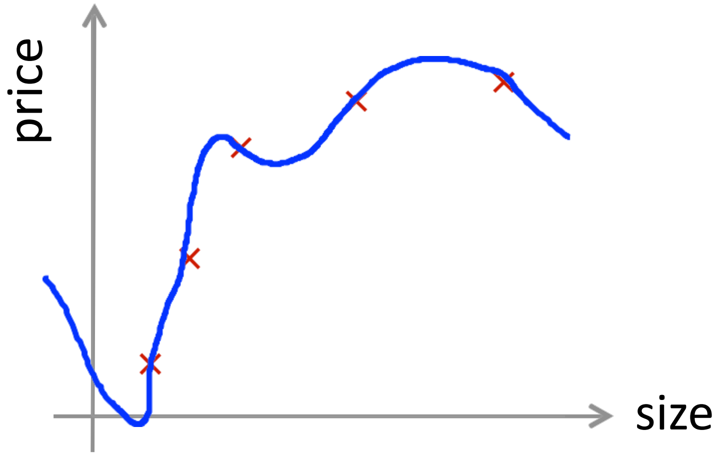


Machine Learning

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

30%

Test set

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

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

m_{test} = no. of test example

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

Andrew Ng

Suppose an implementation of linear regression (without regularization) is badly overfitting the training set. In this case, we would expect:

- The training error $J(\theta)$ to be **low** and the test error $J_{\text{test}}(\theta)$ to be **high**
- The training error $J(\theta)$ to be **low** and the test error $J_{\text{test}}(\theta)$ to be **low**
- The training error $J(\theta)$ to be **high** and the test error $J_{\text{test}}(\theta)$ to be **low**
- The training error $J(\theta)$ to be **high** and the test error $J_{\text{test}}(\theta)$ to be **high**

Training/testing procedure for linear regression

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

- Compute test set error:

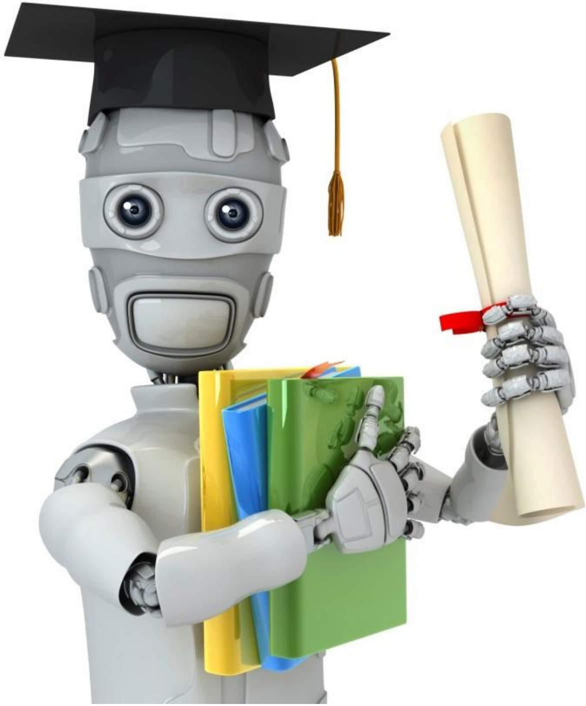
$$J_{test}(\theta) = \frac{1}{2m_{test}} \sum_{i=1}^{m_{test}} \left(h_\theta(x^{(i)}_{test}) - y^{(i)}_{test} \right)^2$$

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):

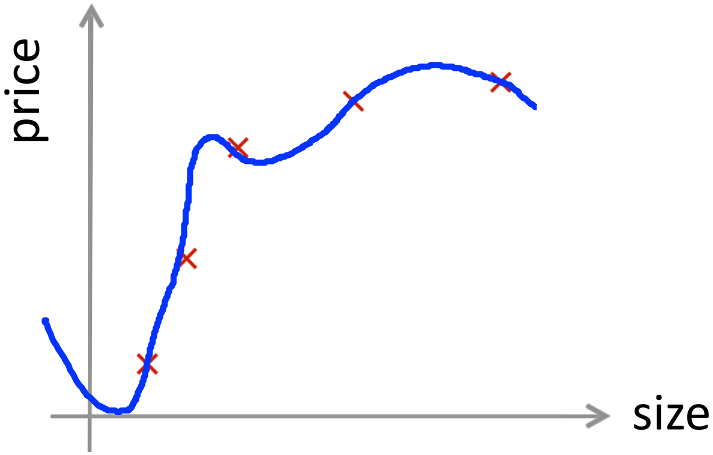


Machine Learning

Advice for applying machine learning

Model selection and training/validation/test sets

Overfitting example



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

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

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

$d=3$ 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 $\theta_0 + \ldots \theta_5 x^5$

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

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 test set.

Evaluating your hypothesis

Dataset:

	Size	Price	
	2104	400	Training set
	1600	330	
60%	2400	369	
	1416	232	
	3000	540	
	1985	300	
20%	1534	315	Cross validation set (CV)
	1427	199	
20%	1380	212	test set
	1494	243	

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

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

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:

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

$J(\theta)$

Cross Validation error:

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

Test error:

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

Model selection

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

$d=2$ 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. $h_\theta(x) = \theta_0 + \theta_1 x + \cdots + \theta_{10} x^{10}$ $\longrightarrow \Theta^{(10)} \longrightarrow J_{cv}(\Theta^{(10)})$

$d=4$

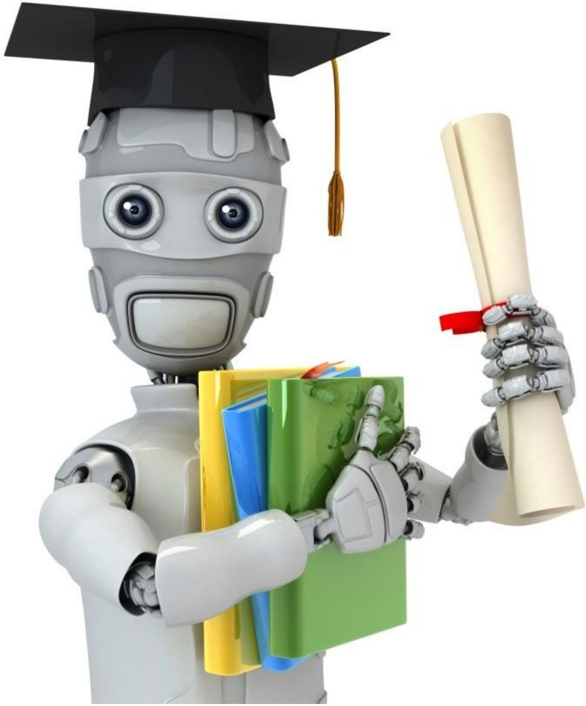
Pick $\theta_0 + \theta_1 x_1 + \cdots + \theta_4 x^4$

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

Andrew Ng

Consider the model selection procedure where we choose the degree of polynomial using a cross validation set. For the final model (with parameters θ), we might generally expect $J_{\mathrm{CV}}(\theta)$ To be lower than $J_{\mathrm{test}}(\theta)$ because:

- An extra parameter (d, the degree of the polynomial) has been fit to the cross validation set.
- An extra parameter (d, the degree of the polynomial) has been fit to the test set.
- The cross validation set is usually smaller than the test set.
- The cross validation set is usually larger than the test set.

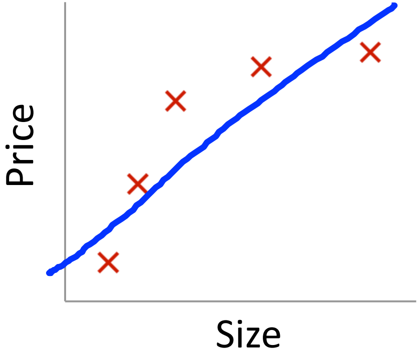


Machine Learning

Advice for applying machine learning

Diagnosing bias vs. variance

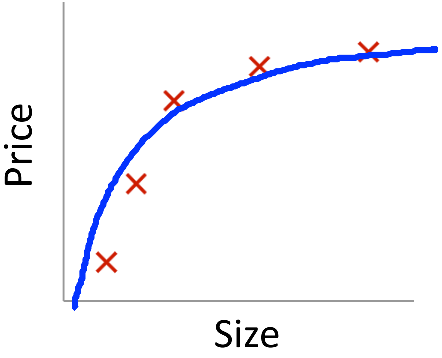
Bias/variance



$$\theta_0 + \theta_1 x$$

High bias
(underfit)

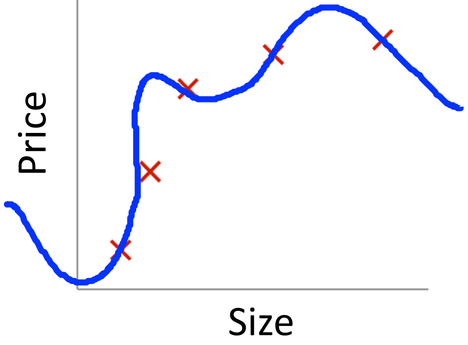
$d=1$



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

"Just right"

$d=2$



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

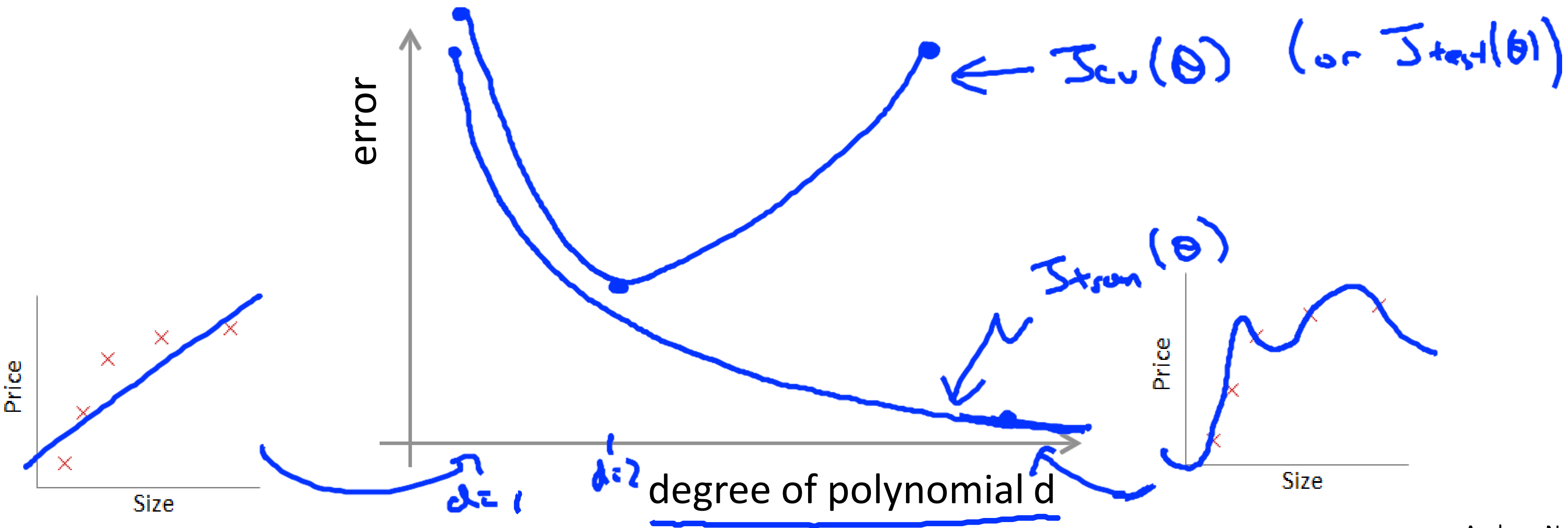
High variance
(overfit)

$d=4$

Bias/variance

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

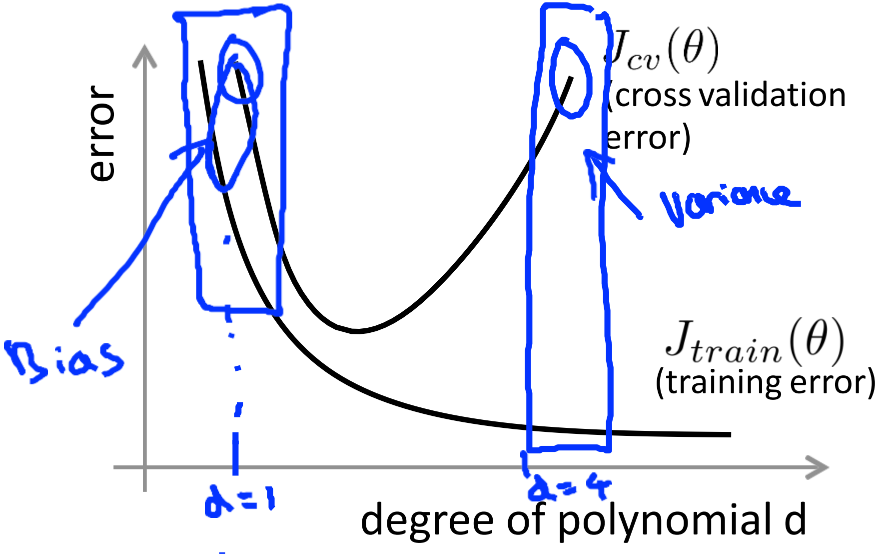
Cross validation error: $\underline{J_{cv}(\theta)} = \frac{1}{2m_{cv}}\sum_{i=1}^{m_{cv}}(h_\theta(x_{cv}^{(i)}) - y_{cv}^{(i)})^2$ (or $J_{test}(\theta)$)



Andrew Ng

Diagnosing bias vs. variance

Suppose your learning algorithm is performing less well than you were 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

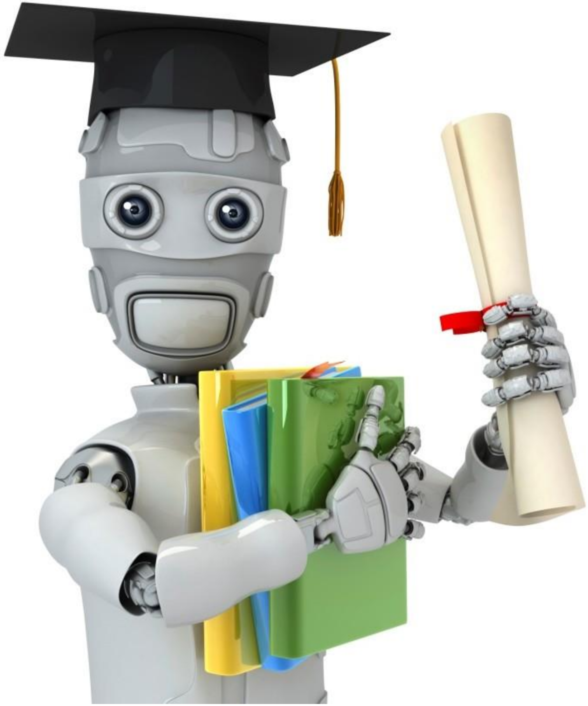
$J_{cv}(\theta) \underline{\gg} J_{train}(\theta)$

$\gg$

Andrew Ng

Suppose you have a classification problem. The (misclassification) error is defined as $\frac{1}{m}\sum_{i=1}^{m} \mathrm{err}(h_\theta(x^{(i)}), y^{(i)})$, and the cross validation (misclassification) error is similarly defined, using the cross validation examples $(x_{\mathrm{cv}}^{(1)}, y_{\mathrm{cv}}^{(1)}), \ldots, (x_{\mathrm{cv}}^{(m_{\mathrm{cv}})}, y_{\mathrm{cv}}^{(m_{\mathrm{cv}})})$. Suppose your training error is 0.10, and your cross validation error is 0.30. What problem is the algorithm most likely to be suffering from?

- High bias (overfitting)
- High bias (underfitting)
- High variance (overfitting)
- High variance (underfitting)



Machine Learning

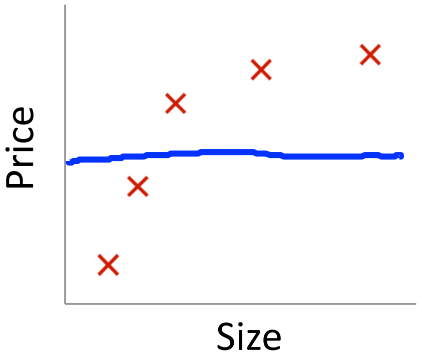
Advice for applying machine learning

Regularization and bias/variance

Linear regression with regularization

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

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

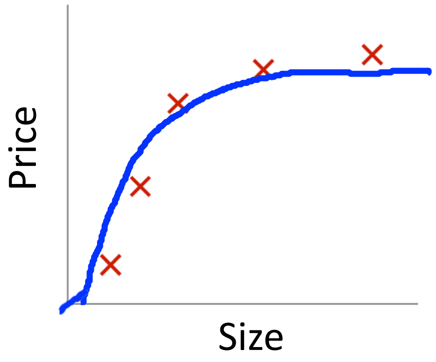


Large λ

High bias (underfit)

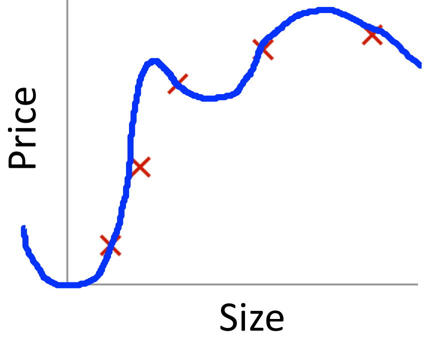
$\lambda = 10000.\ \theta_1 \approx 0, \theta_2 \approx 0, \ldots$

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



Intermediate λ

"Just right"



Small λ

High variance (overfit)

$\lambda = 0$

Choosing the regularization parameter λ

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

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

$J(\theta)$

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

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

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

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 $\lambda = \underline{0.01}$ $\rightarrow \min_\theta J(\theta) \rightarrow \Theta^{(2)} \rightarrow J_{cv}(\Theta^{(2)})$
3. Try $\lambda = \underline{0.02}$ $\rightarrow \Theta^{(3)} \rightarrow J_{cv}(\Theta^{(3)})$
4. Try $\lambda = \underline{0.04}$
5. Try $\lambda = 0.08$ $\rightarrow \Theta^{(5)}$ $J_{cv}(\Theta^{(5)})$

$\vdots$

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

$\underline{10.24}$

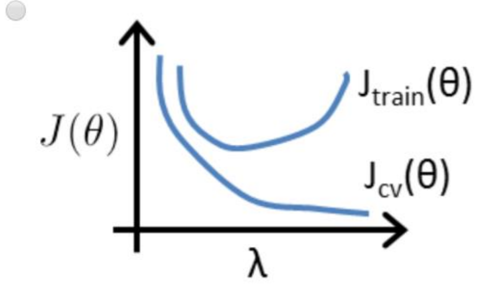
Pick (say) $\theta^{(5)}$. Test error: $\underline{J_{test}}(\Theta^{(5)})$

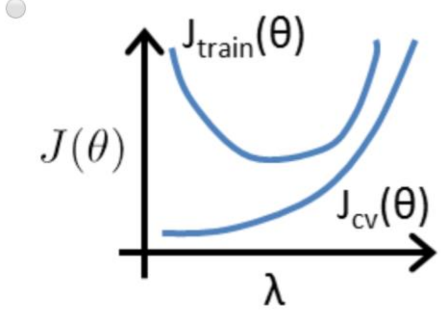
Andrew Ng

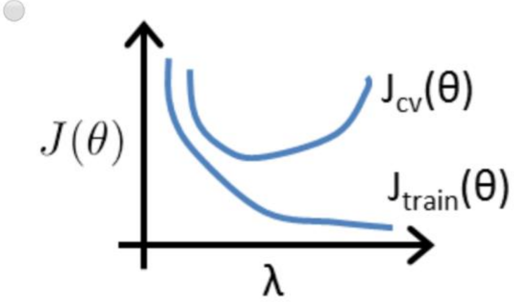
Consider regularized logistic regression. Let

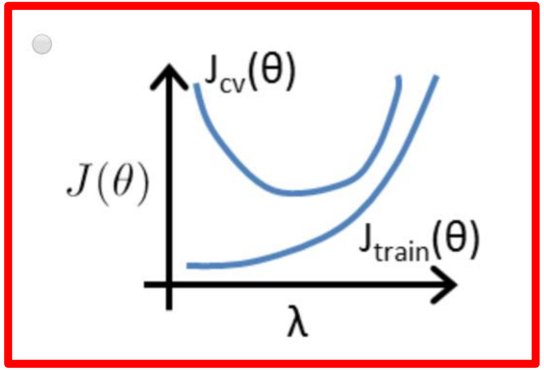
- $J(\theta) = \frac{1}{2m}\left[\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2 + \lambda \sum_{j=2}^{n} \theta_j^2\right]$
- $J_{\text{train}}(\theta) = \frac{1}{2m_{\text{train}}}\left[\sum_{i=1}^{m_{\text{train}}} (h_\theta(x_{\text{train}}^{(i)}) - y_{\text{train}}^{(i)})^2\right]$
- $J_{\text{CV}}(\theta) = \frac{1}{2m_{\text{CV}}}\left[\sum_{i=1}^{m_{\text{CV}}} (h_\theta(x_{\text{CV}}^{(i)}) - y_{\text{CV}}^{(i)})^2\right]$

Suppose you plot J_{train} and J_{CV} as a function of the regularization parameter λ. which of the following plots do you expect to get?







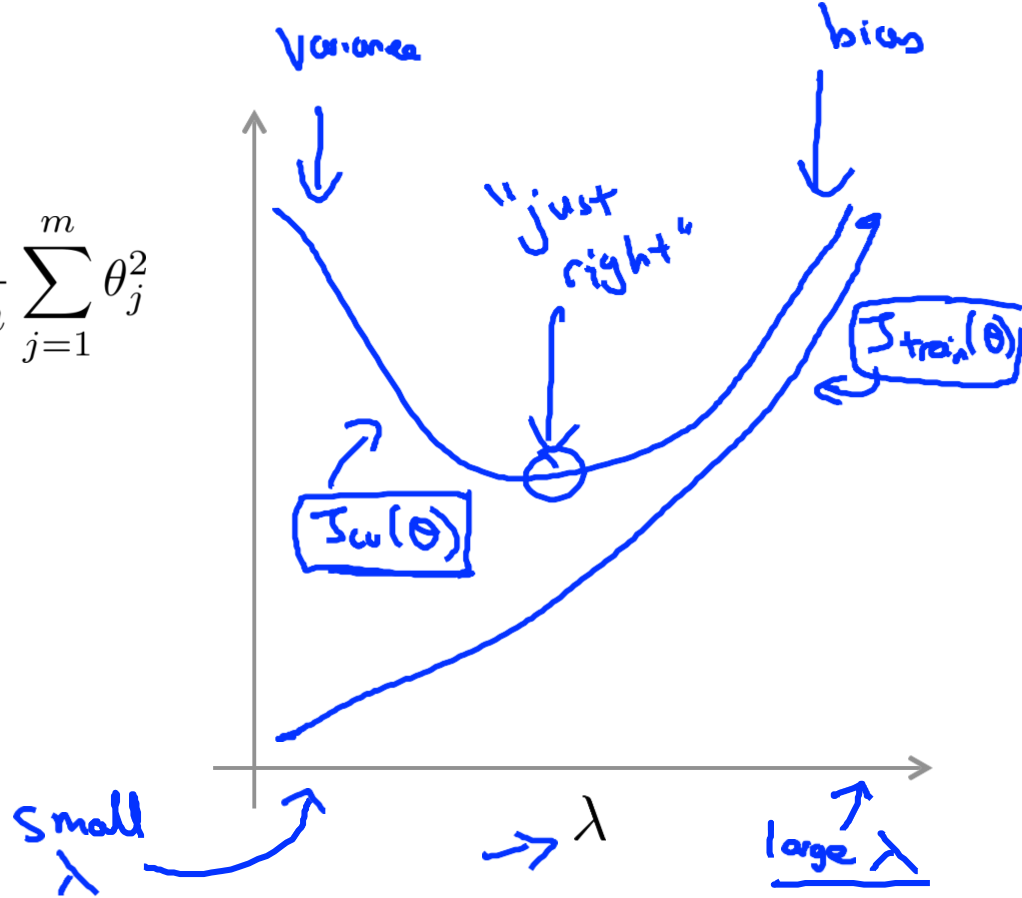


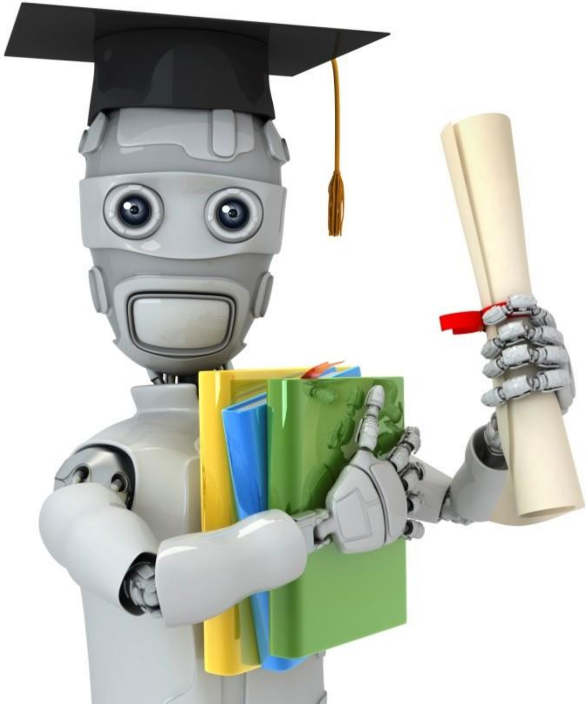
Bias/variance as a function of the regularization parameter λ

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

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

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





Machine Learning

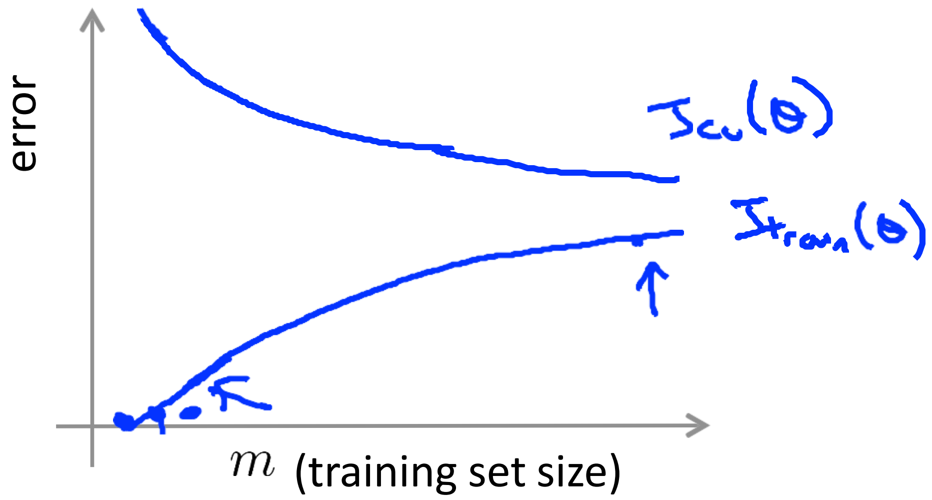
Advice for applying machine learning

Learning curves

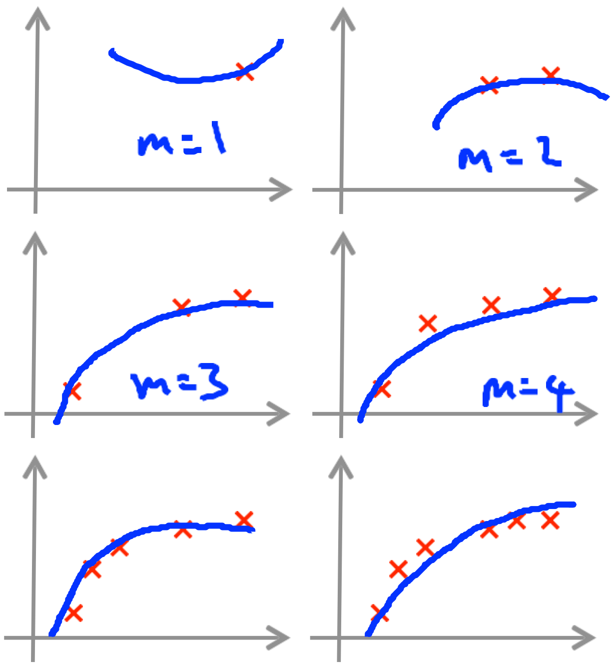
Learning curves

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

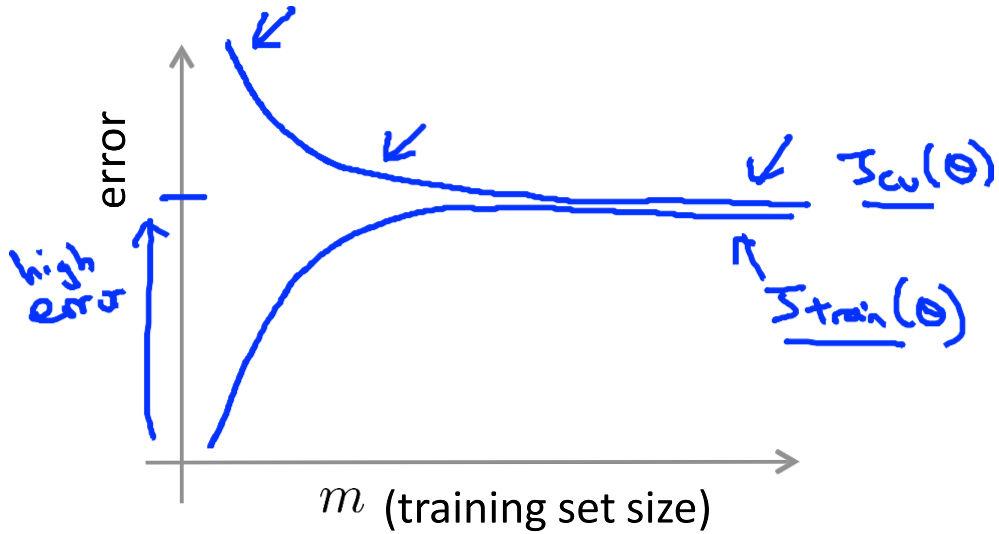
$$J_{cv}(\theta) = \frac{1}{2m_{cv}}\sum_{i=1}^{m_{cv}}(h_\theta(x_{cv}^{(i)}) - y_{cv}^{(i)})^2$$



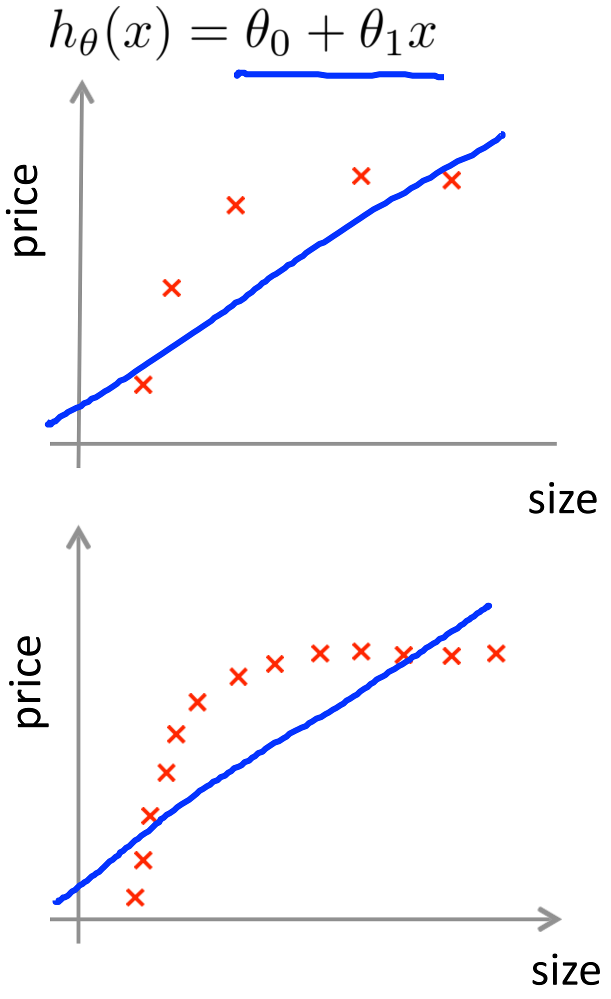
$$h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2$$



High bias



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

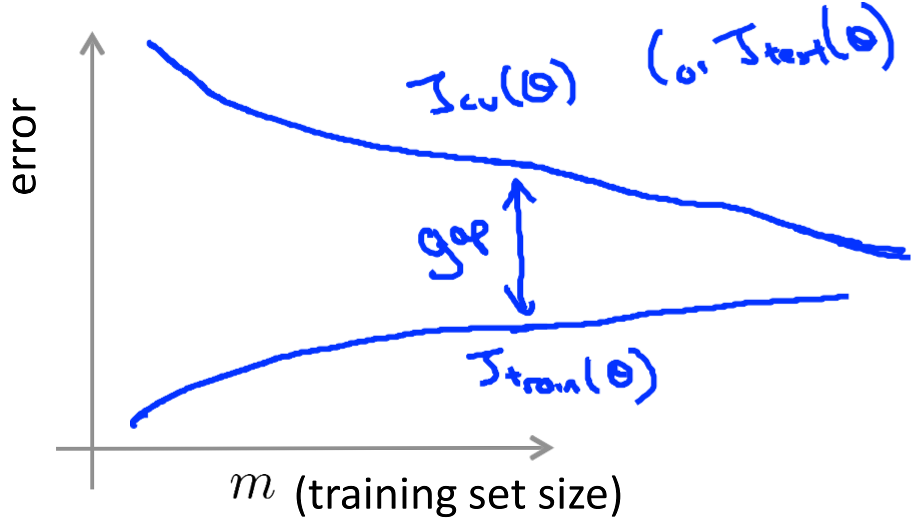


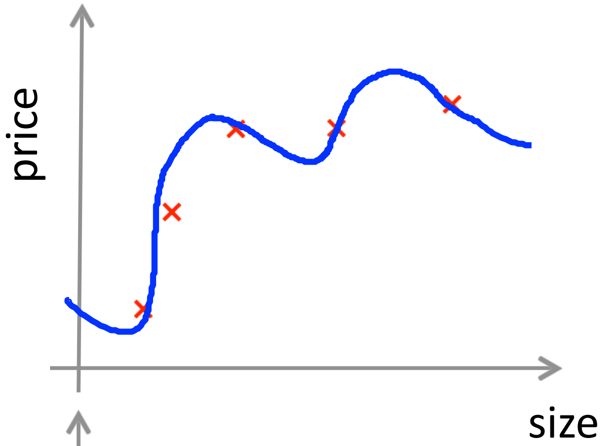
Andrew Ng

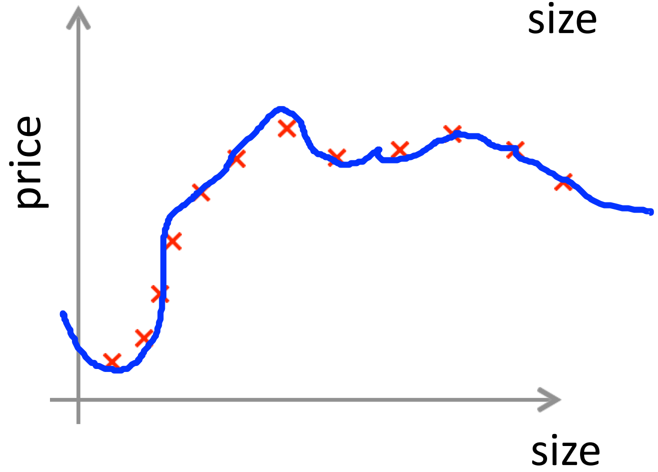
High variance

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

(and small λ)





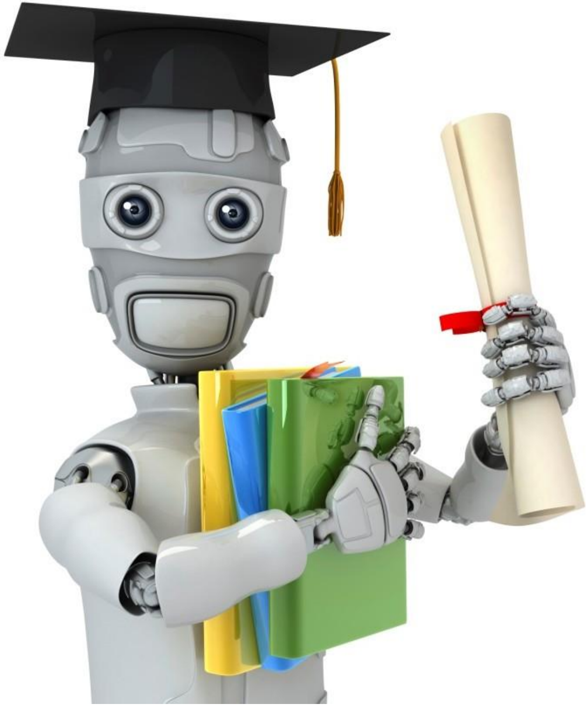


If a learning algorithm is suffering from high variance, getting more training data is likely to help.

Andrew Ng

In which of the following circumstances is getting more training data likely to significantly help a learning algorithm's performance?

- [ ] Algorithm is suffering from high bias.
- [ ] Algorithm is suffering from high variance.
- [ ] $J_{\mathrm{CV}}(\theta)$ (cross validation error) is much larger than $J_{\mathrm{train}}(\theta)$ (training error).
- [ ] $J_{\mathrm{CV}}(\theta)$ (cross validation error) is about the same as $J_{\mathrm{train}}(\theta)$ (training error).



Machine Learning

Advice for applying machine learning

Deciding what to try next (revisited)

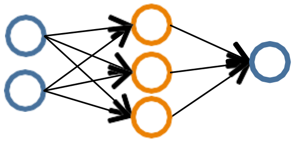
Debugging a learning algorithm:

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 $(x_1^2, x_2^2, x_1 x_2, \text{etc})$ → fixes high bias.
- Try decreasing λ → fixes high bias
- Try increasing λ → fixes high variance

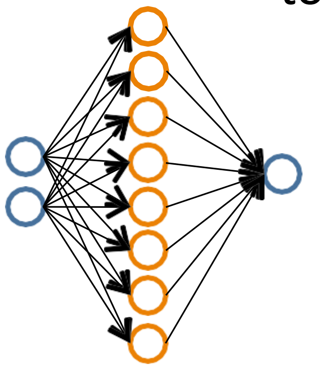
Neural networks and overfiBng

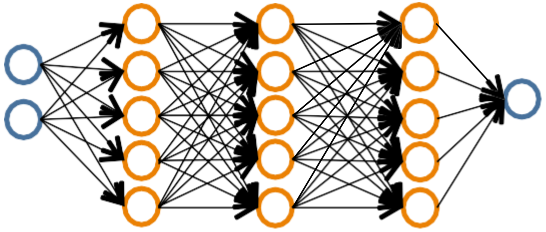
“Small” neural network
(fewer parameters; more
prone to underfitting)



Computationally cheaper

“Large” neural network
(more parameters; more prone
to overfitting)





Computationally more expensive.

Use regularization (λ) to address overfitting.

$J_{cv}(\Theta)$

Suppose you fit a neural network with one hidden layer to a training set. You find that the cross validation error $J_{\text{CV}}(\theta)$ is much larger than the training error $J_{\text{train}}(\theta)$. Is increasing the number of hidden units likely to help?

- ○ Yes, because this increases the number of parameters and lets the network represent more complex functions.
- ○ Yes, because it is currently suffering from high bias.
- ○ No, because it is currently suffering from high bias, so adding hidden units is unlikely to help.
- ○ No, because it is currently suffering from high variance, so adding hidden units is unlikely to help.