Advice for applying Machine Learning

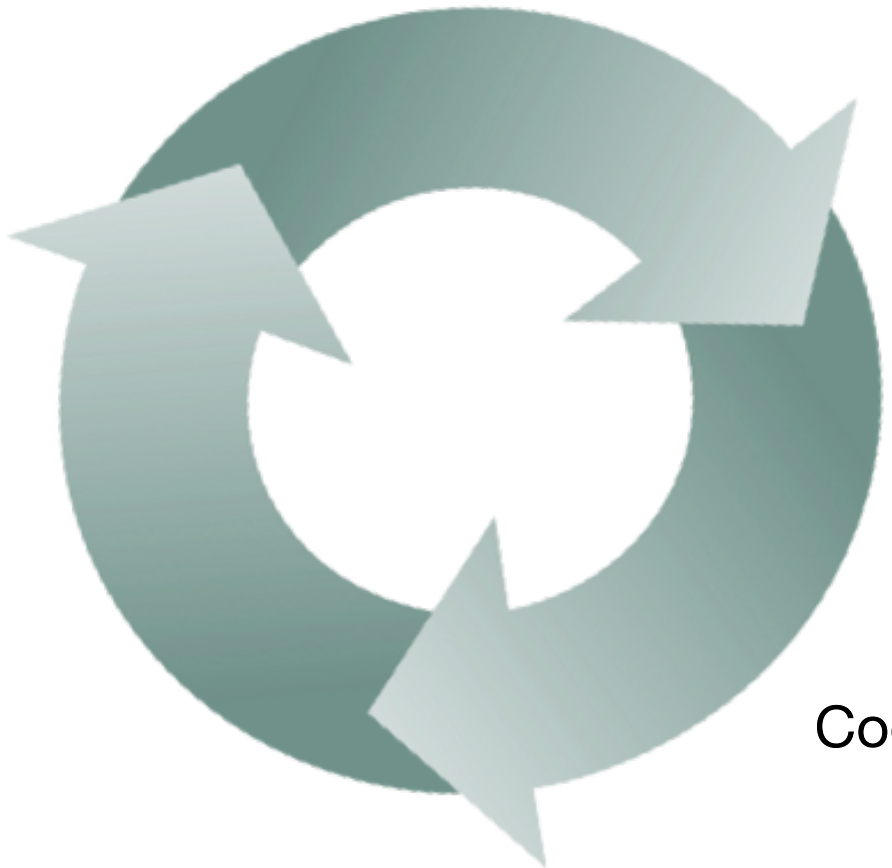
Andrew Ng

Advice for applying machine learning

Deciding what to try next

Machine learning cycle

Idea



Experiment

Code

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
- 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

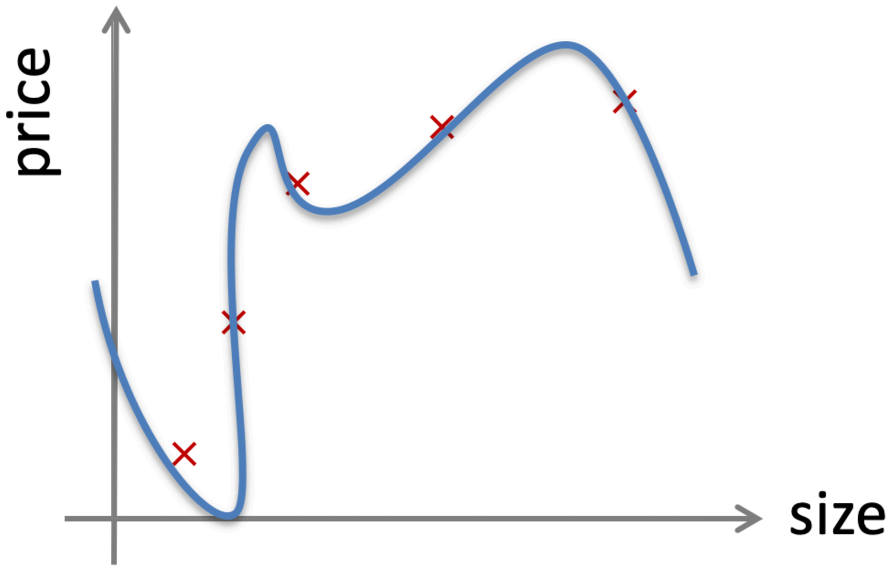
- Diagnostic: A test that you can run to gain insight what is/isn’t working with a learning algorithm, and gain guidance as to how best to improve its performance.

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

Advice for applying machine learning

Evaluating a hypothesis

Evaluating your hypothesis



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

Overfit: 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

How well does the model generalize on data not used for training?

dataset

	Size	Price		
70%	2104	400	$(x^{(1)}, y^{(1)})$	training set
	1600	330	$(x^{(2)}, y^{(2)})$	
	2400	369	$\vdots$	
	1416	232		
	3000	540		
	1985	300	$(x^{(m)}, y^{(m)})$	
	1534	315		
30%	1427	199	$(x_{test}^{(1)}, y_{test}^{(1)})$	test set
	1380	212	$(x_{test}^{(2)}, y_{test}^{(2)})$	
	1494	243	$\vdots$	
			$(x_{test}^{(m_{test})}, y_{test}^{(m_{test})})$	

Training/testing procedure for linear regression

- Learn parameter θ from training data (minimizing training error $J(\theta)$)

- Compute test set error:

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

Training/testing procedure for logistic regression

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

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

- Misclassification error (0/1 misclassification error):

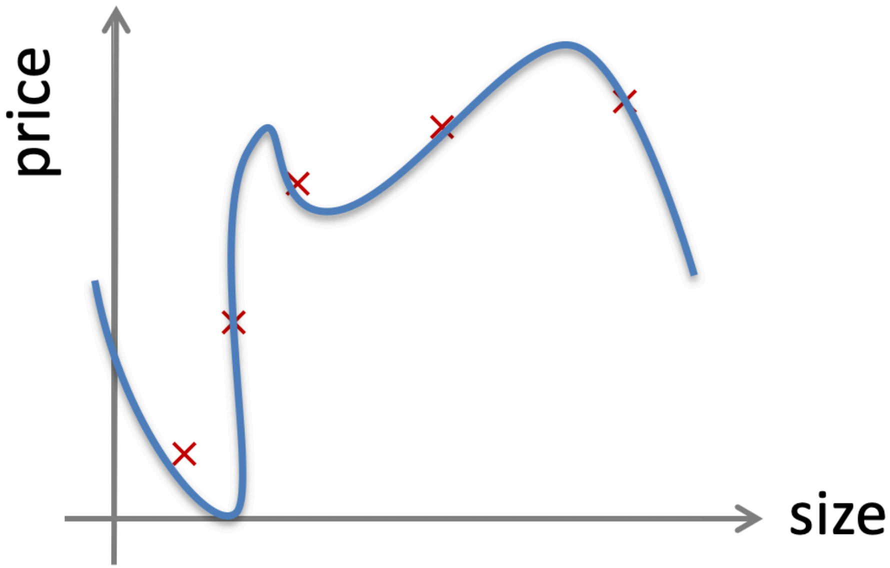
$$err(h_\theta(x), y) = \begin{cases} 1 & \text{if } h_\theta(x) \geq 0.5, y = 0 \\ & \text{or if } h_\theta(x) < 0.5, y = 1 \\ 0 & \text{otherwise} \end{cases}$$

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

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

Model selection

1. $h_\theta(x) = \theta_0 + \theta_1 x$ $\theta^{(1)} \rightarrow J_{test}(\theta^{(1)})$
2. $h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2$ $\theta^{(2)} \rightarrow J_{test}(\theta^{(2)})$
3. $h_\theta(x) = \theta_0 + \theta_1 x + \cdots + \theta_3 x^3$

$\vdots$ $\vdots$

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

Choose $\theta_0 + \ldots \theta_5 x^5$

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

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
	1600	330
60%	2400	369
	1416	232
	3000	540
	1985	300
20%	1534	315
	1427	199
20%	1380	212
	1494	243

training set

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

cross-validation set

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

test set

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

Train/validation/test error

Training error:

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

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

1. $h_\theta(x) = \theta_0 + \theta_1 x$ $\quad\quad \theta^{(1)} \rightarrow J_{cv}(\theta^{(1)})$
2. $h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2$ $\quad\quad \theta^{(2)} \rightarrow J_{cv}(\theta^{(2)})$
3. $h_\theta(x) = \theta_0 + \theta_1 x + \cdots + \theta_3 x^3$

$\vdots$

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

Pick the model with minimum cross validation error

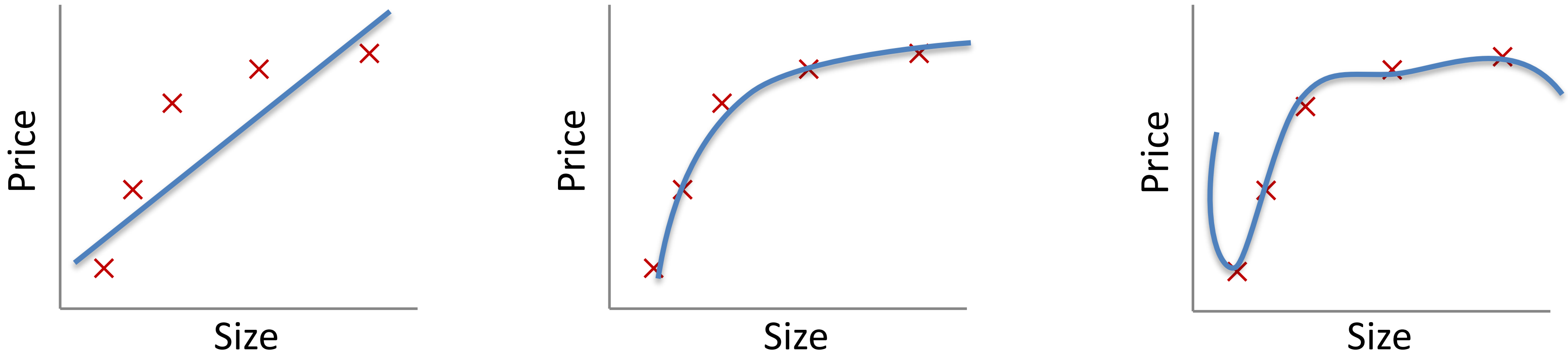
$$\theta_0 + \theta_1 x_1 + \cdots + \theta_4 x^4$$

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

Advice for applying machine learning

Diagnosing bias vs. variance

Bias/variance (easy in 2D)



$\theta_0 + \theta_1 x$

High bias
(underfit)

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

"Just right"

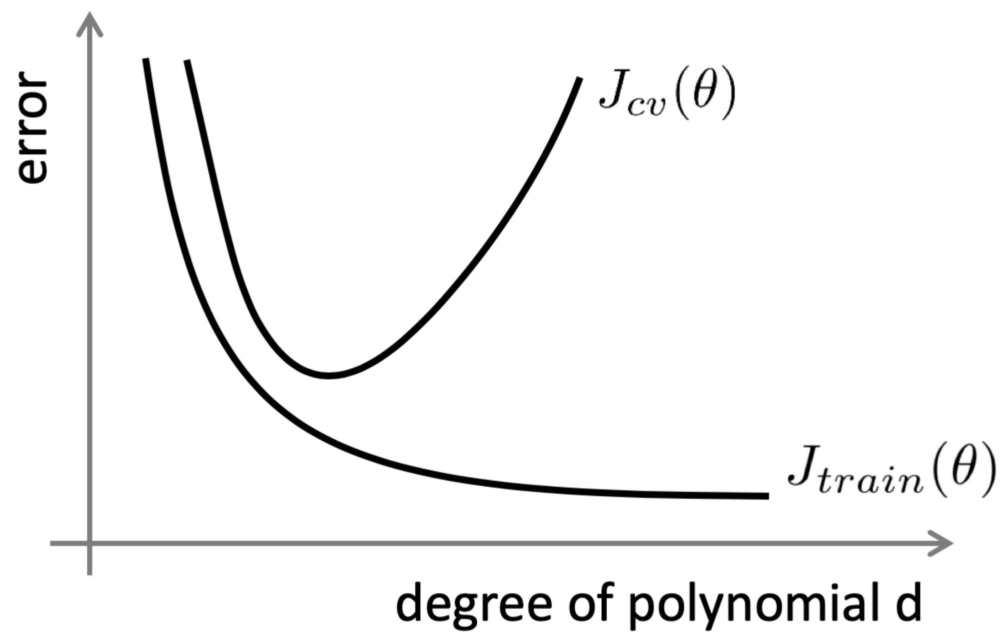
$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$

High variance
(overfit)

Bias/Variance

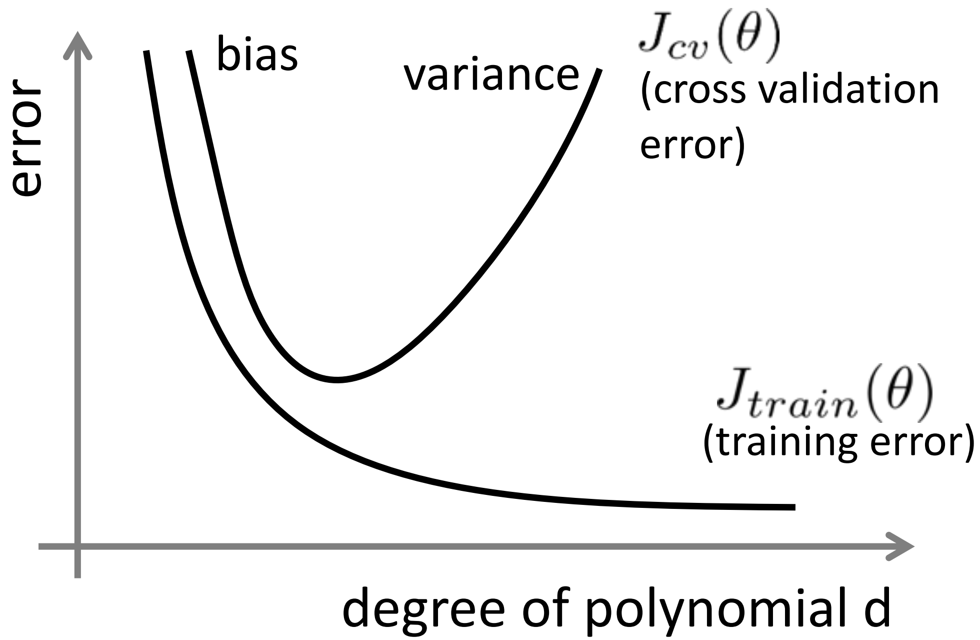
Training error: $J_{train}(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2$

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



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

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

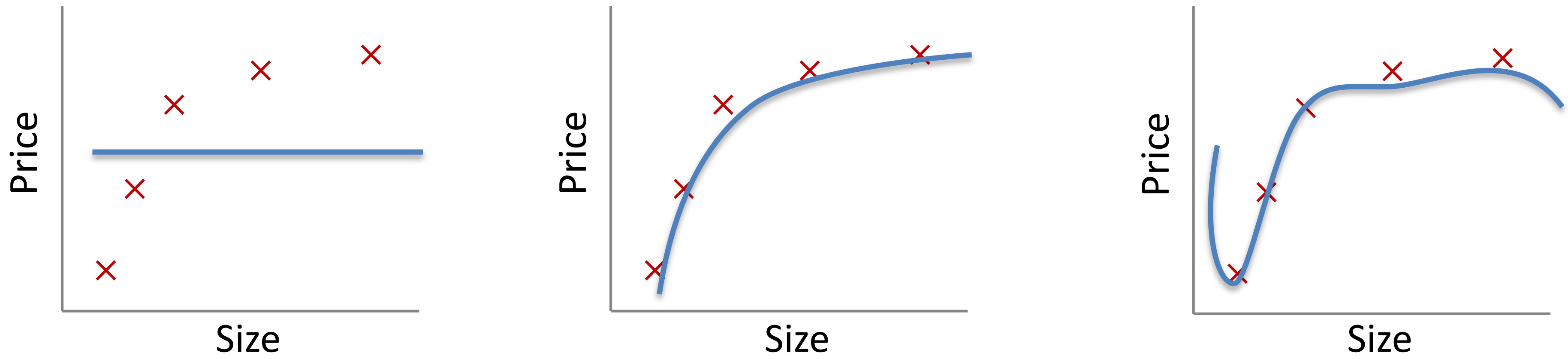
Advice for applying machine learning

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)

Intermediate λ
"Just right"

Small λ
High variance (overfit)

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

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

Choosing the regularization parameter λ

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

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

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

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

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

Choosing the regularization parameter λ

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

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

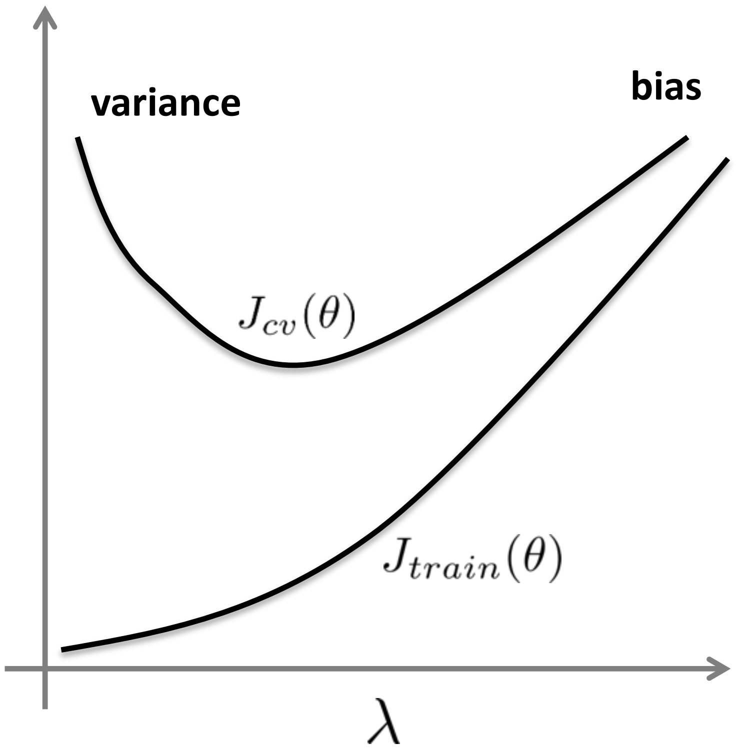
1. Try $\lambda = 0$ $\quad \theta^{(1)} \rightarrow J_{cv}(\theta^{(1)})$
2. Try $\lambda = 0.01$ $\quad \theta^{(2)} \rightarrow J_{cv}(\theta^{(2)})$
3. Try $\lambda = 0.02$
4. Try $\lambda = 0.04$ $\quad \vdots$
5. Try $\lambda = 0.08$

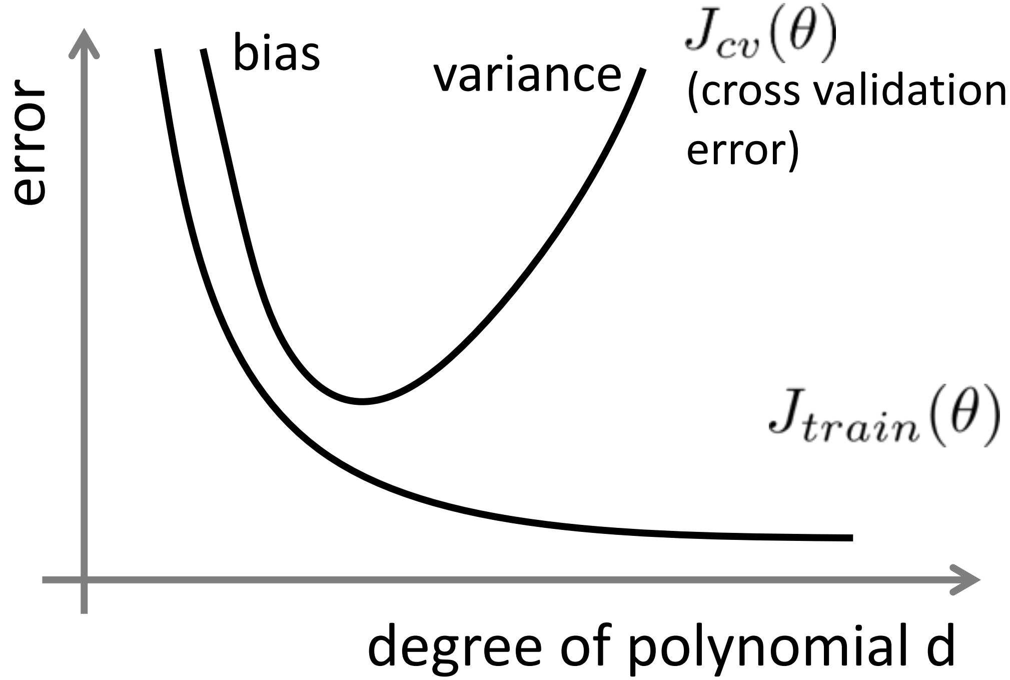
$\vdots$

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

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

Bias/variance as a function of the regularization parameter λ





One of the challenges with building machine learning systems is that there's so many things you could try, so many things you could change

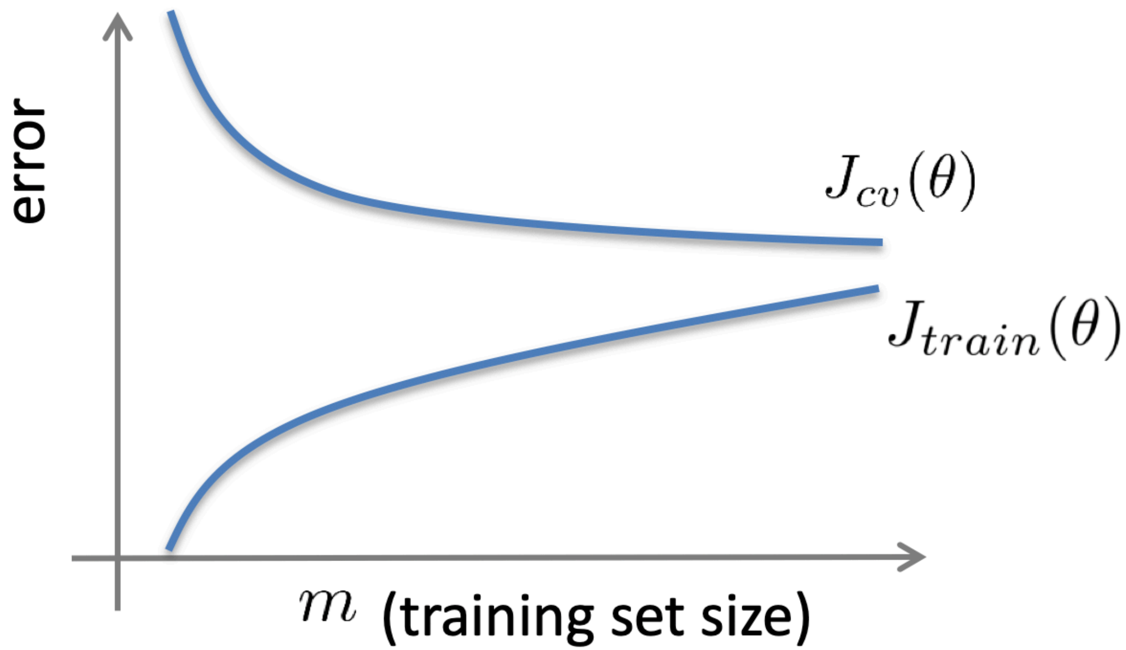
Advice for applying machine learning

Learning curves

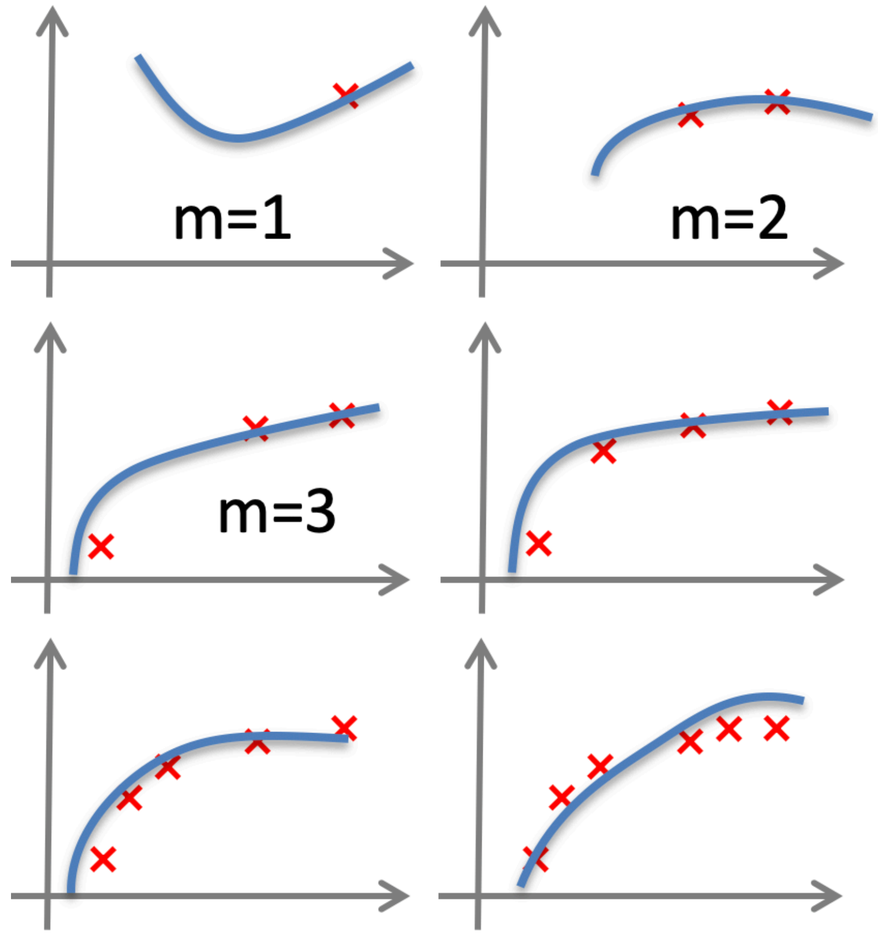
Learning curves

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

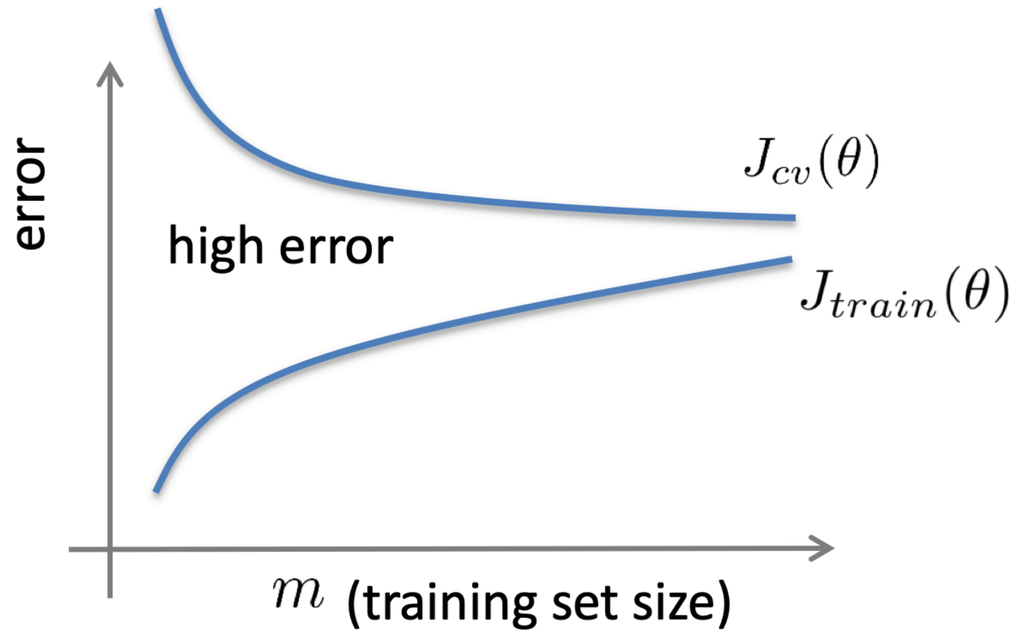
$$J_{cv}(\theta) = \frac{1}{2m_{cv}}\sum_{i=1}^{m_{cv}}(h_\theta(x_{cv}^{(i)}) - y_{cv}^{(i)})^2$$



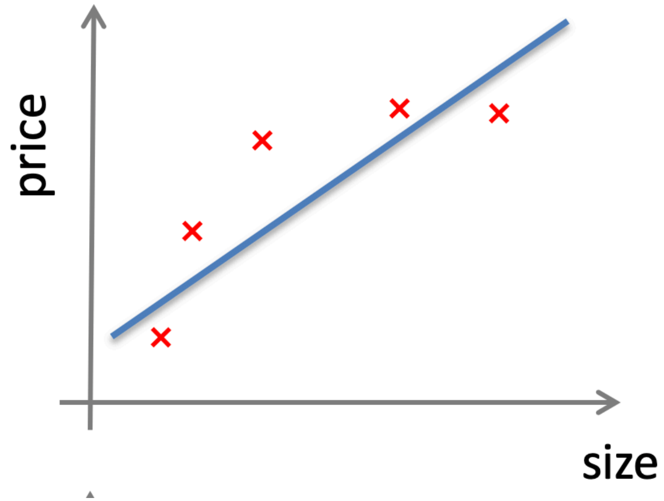
$$h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2$$

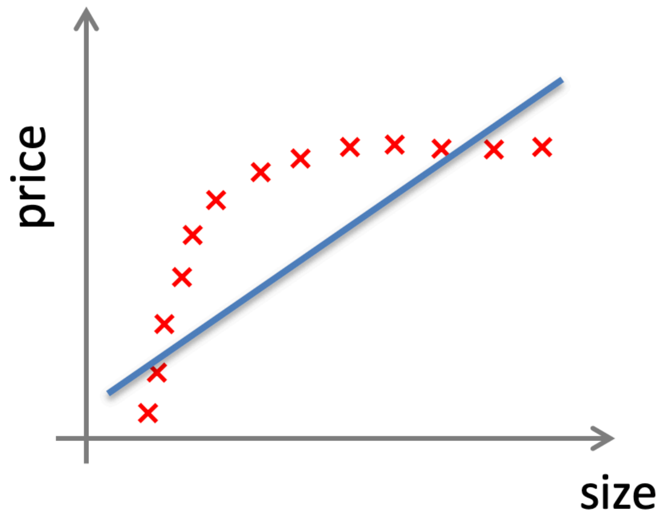


High bias



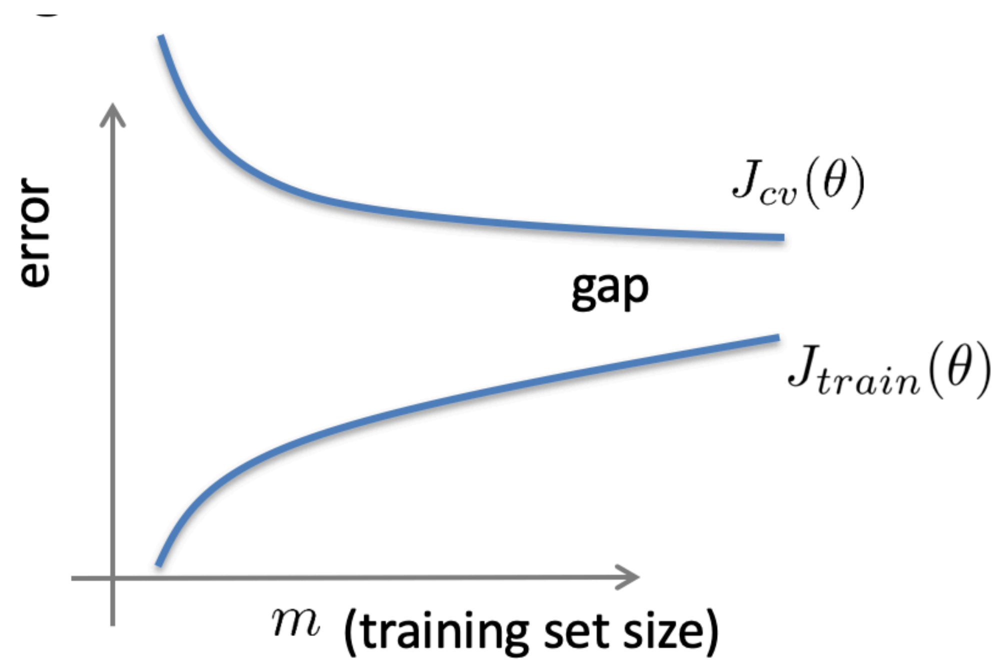
$$h_\theta(x) = \theta_0 + \theta_1 x$$





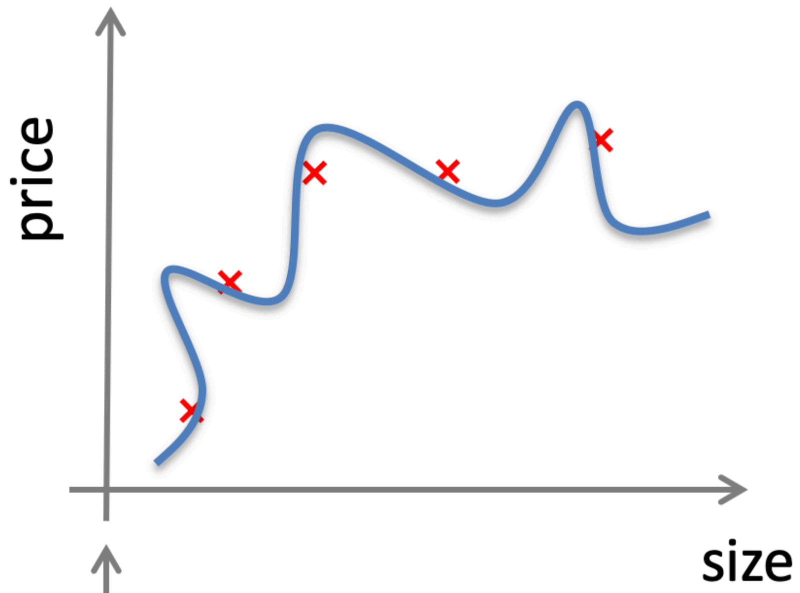
If a learning algorithm is suffering from high bias, getting more training data will not (by itself) help much.

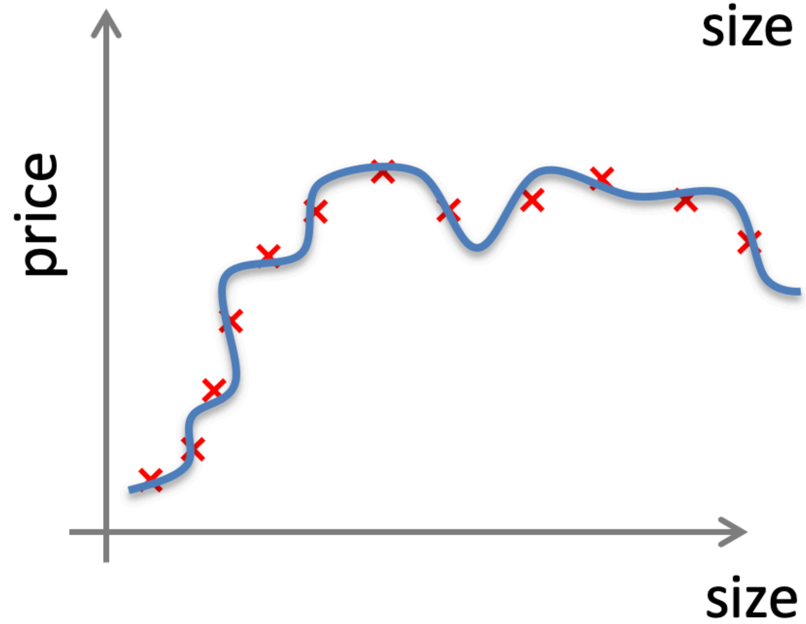
High variance



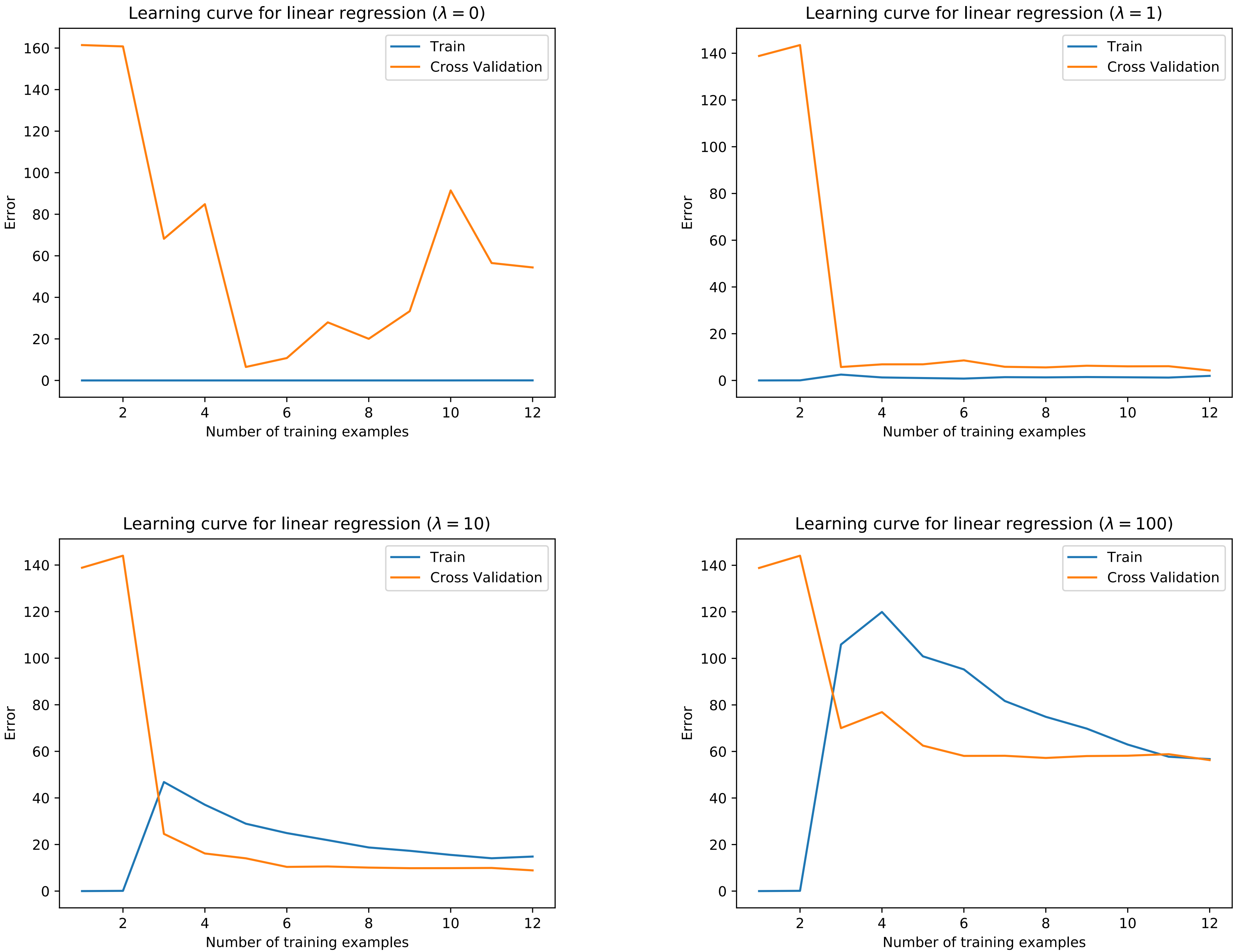
$$h_\theta(x) = \theta_0 + \theta_1 x + \cdots + \theta_{100} x^{100}$$

(and small λ)





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



Learning curve for linear regression ($\lambda = 0$)
Learning curve for linear regression ($\lambda = 1$)
Learning curve for linear regression ($\lambda = 10$)
Learning curve for linear regression ($\lambda = 100$)
Train
Cross Validation
Error
Number of training examples

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?

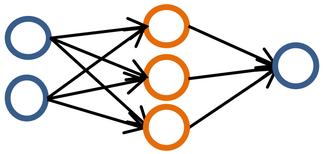
- Get more training examples: fixes high variance
- Try smaller sets of features: fixes high variance
- Try getting additional features: fixes high bias
- Try adding polynomial features: fixes high bias
- Try decreasing λ: fixes high bias
- Try increasing λ: fixes high variance

Advice for applying machine learning

Neural networks

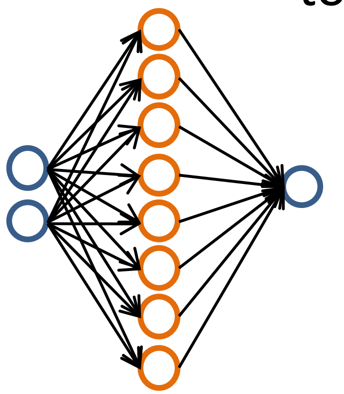
Neural networks and overfitting

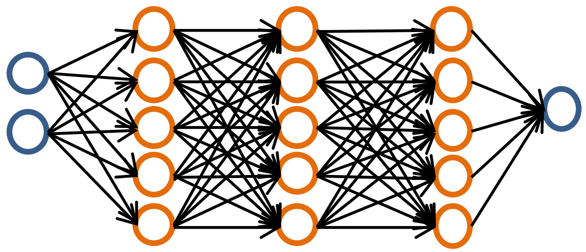
"Small" neural network
(fewer parameters; more
prone to underfitting)



Computationally cheaper

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

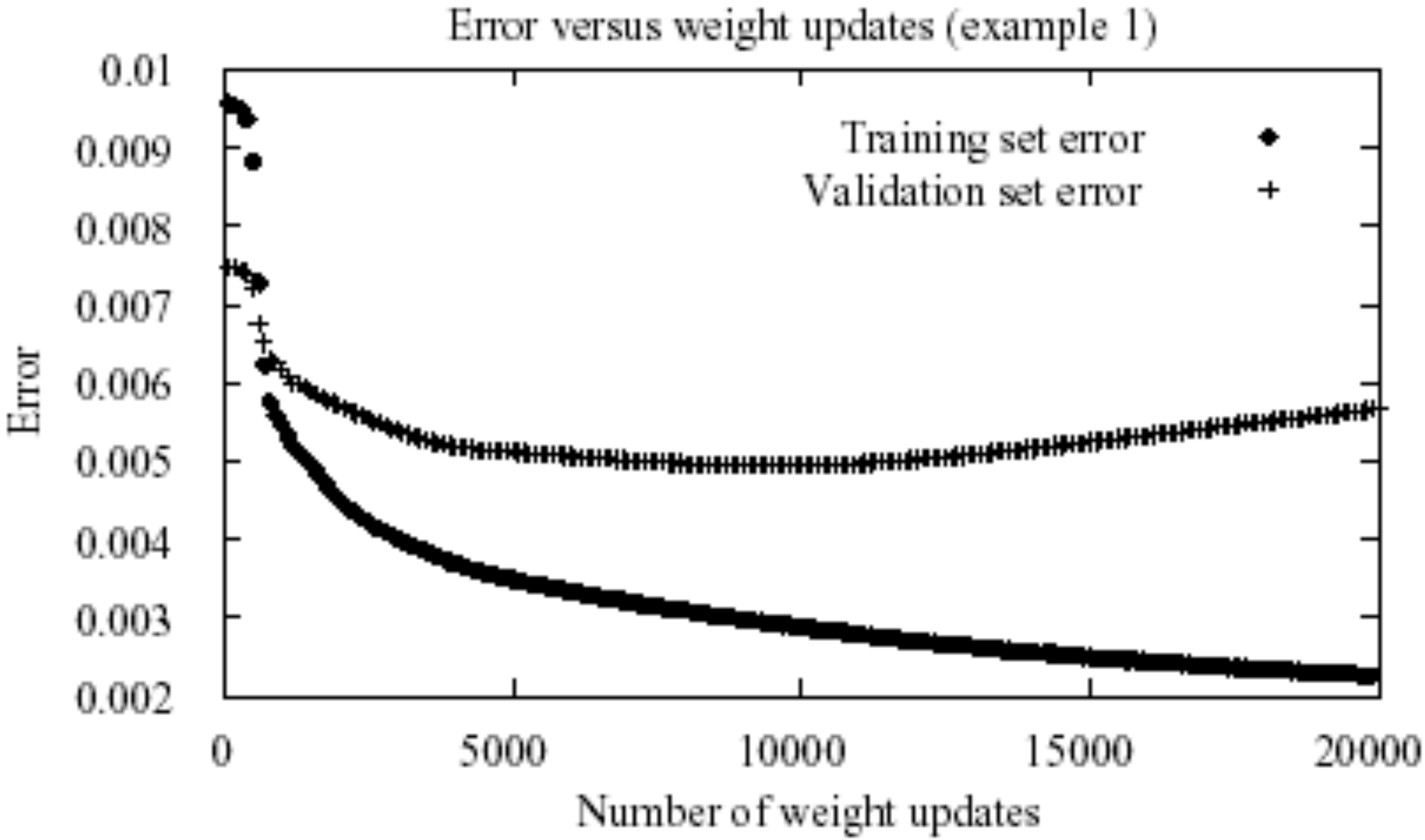




Computationally more expensive

Use regularization (λ) to address overfitting

Neural networks: early stopping to avoid overfitting



Neural networks: early stopping to avoid overfitting