



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.

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

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?

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

## **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

30% Test Set

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

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

$m_{test} = \text{no. of test example}$   
 $(x_{test}^{(i)}, y_{test}^{(i)})$

# Training/testing procedure for linear regression

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

- Compute test set error:

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

## Training/testing procedure for logistic regression

- Learn parameter  $\theta$  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):





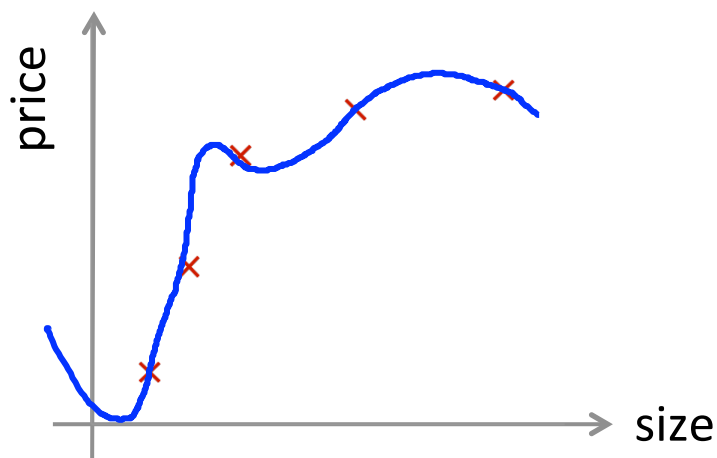
Machine Learning

# 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  $\theta_0, \theta_1, \dots, \theta_4$  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.

→  $d = \text{degree of polynomial}$  ↓

## Model selection

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

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

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

⋮

⋮

⋮

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

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

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

$\Theta^{(5)}$

$\Theta_0, \Theta_1, \dots$  ↑

Problem:  $J_{\text{test}}(\theta^{(5)})$  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.

# Evaluating your hypothesis

Dataset:

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



# Train/validation/test error

Training error:

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

$J(\theta)$

Cross Validation error:

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

Test error:

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

## Model selection

$$\begin{array}{llll}
 \text{d:1} & 1. & h_{\theta}(x) = \theta_0 + \theta_1 x & \xrightarrow{\min J(\theta)} \Theta^{(1)} \rightarrow J_{cv}(\Theta^{(1)}) \\
 \text{d:2} & 2. & h_{\theta}(x) = \theta_0 + \theta_1 x + \theta_2 x^2 & \xrightarrow{\quad} \Theta^{(2)} \rightarrow J_{cv}(\Theta^{(2)}) \\
 \text{d:3} & 3. & h_{\theta}(x) = \theta_0 + \theta_1 x + \cdots + \theta_3 x^3 & \xrightarrow{\quad} \Theta^{(3)} \rightarrow J_{cv}(\Theta^{(3)}) \\
 & \vdots & & \\
 \text{d:10} & 10. & h_{\theta}(x) = \theta_0 + \theta_1 x + \cdots + \theta_{10} x^{10} & \xrightarrow{\quad} \Theta^{(10)} \rightarrow J_{cv}(\Theta^{(10)})
 \end{array}$$

$\underline{d=4} \xrightarrow{\quad} \uparrow$

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

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



Machine Learning

# Advice for applying machine learning

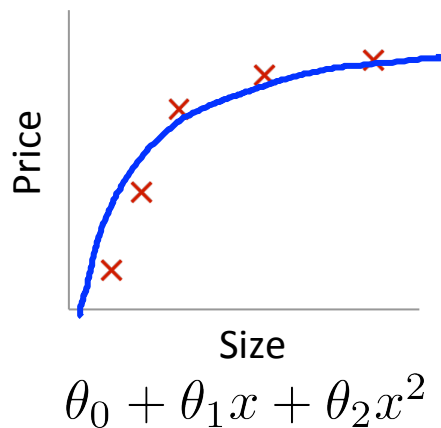
---

## Diagnosing bias vs. variance

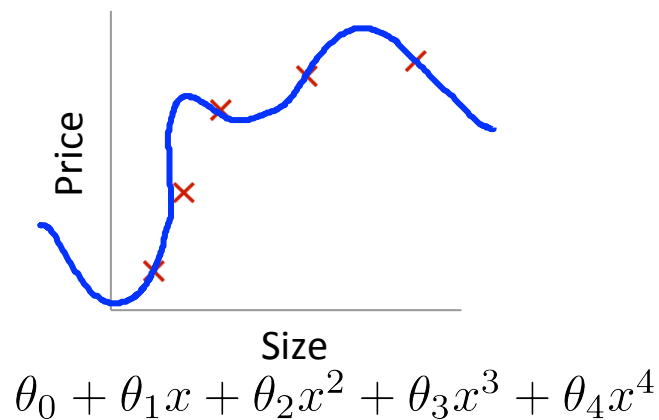
# Bias/variance



High bias  
(underfit)  
 $d=1$



“Just right”  
 $d=2$



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. ( $J_{cv}(\theta)$  or  $J_{test}(\theta)$  is high.) Is it a bias problem or a variance problem?



Bias (underfit):

$$\rightarrow J_{train}(\theta) \text{ will be high} \\ J_{cv}(\theta) \approx J_{train}(\theta)$$

Variance (overfit):

$$\rightarrow J_{train}(\theta) \text{ will be low} \\ J_{cv}(\theta) \gg J_{train}(\theta)$$

$\Rightarrow$



Machine Learning

# Advice for applying machine learning

---

## Regularization and bias/variance

# Linear regression with regularization

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

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



Large  $\lambda$

→ High bias (underfit)

→  $\lambda = 10000$ .  $\theta_1 \approx 0, \theta_2 \approx 0, \dots$   
 $h_{\theta}(x) \approx \theta_0$



Intermediate  $\lambda$

“Just right”



→ Small  $\lambda$

High variance (overfit)

→  $\lambda = 0$

## Choosing the regularization parameter $\lambda$

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

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

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

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

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

$J(\theta)$

$J_{train}$   
 $J_{cv}$   
 $J_{test}$

## Choosing the regularization parameter $\lambda$

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

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

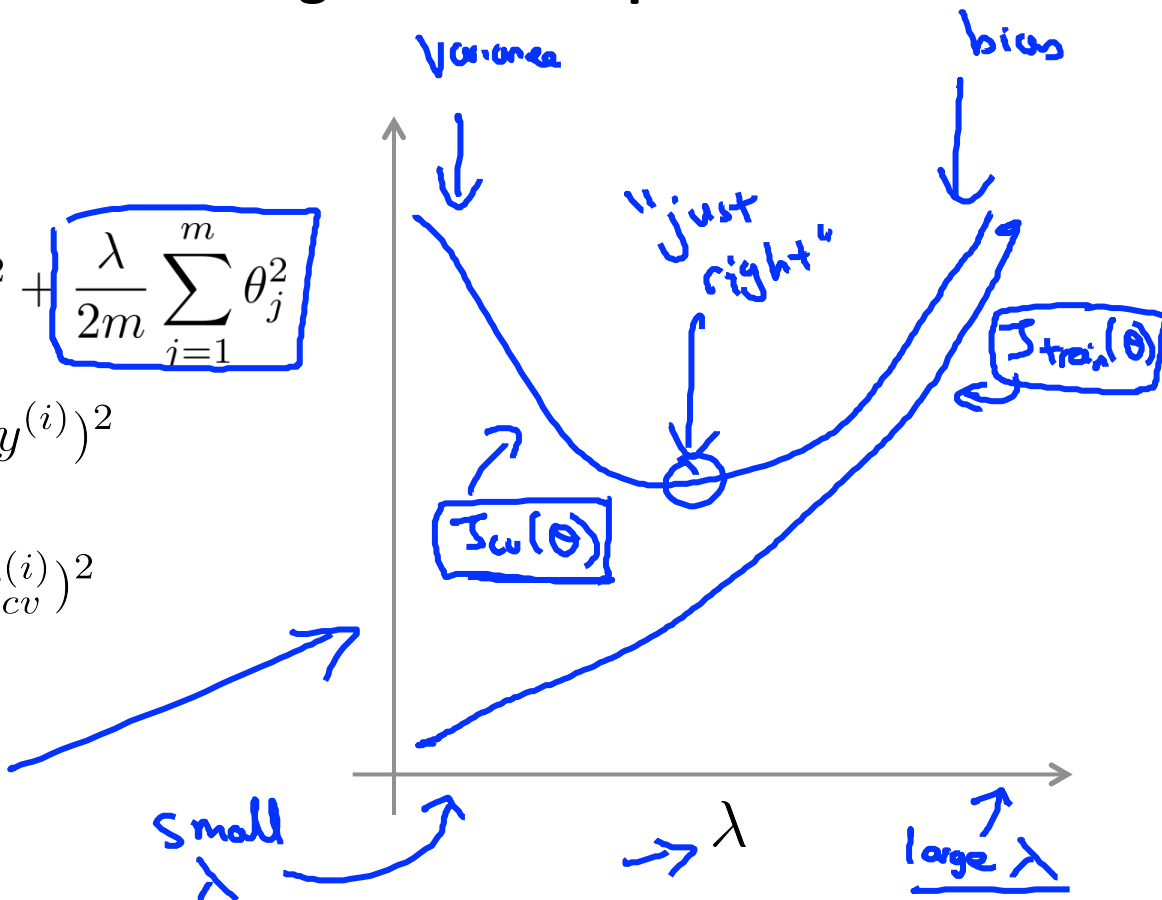
1. Try  $\lambda = 0 \leftarrow \uparrow \rightarrow \min_{\theta} J(\theta) \rightarrow \theta^{(1)} \rightarrow J_w(\theta^{(1)})$
  2. Try  $\lambda = 0.01$   $\rightarrow \min_{\theta} J(\theta) \rightarrow \theta^{(2)} \rightarrow J_w(\theta^{(2)})$
  3. Try  $\lambda = 0.02$   $\rightarrow \theta^{(3)} \rightarrow J_w(\theta^{(3)})$
  4. Try  $\lambda = 0.04$   $\vdots$
  5. Try  $\lambda = 0.08$   $\rightarrow \theta^{(5)} \rightarrow J_w(\theta^{(5)})$
  - $\vdots$
  12. Try  $\lambda = 10$   $\rightarrow \theta^{(12)} \rightarrow J_w(\theta^{(12)})$   
 $\uparrow \quad \underline{10.24}$
- Pick (say)  $\theta^{(5)}$ . Test error:  $J_{\text{test}}(\theta^{(5)})$

# Bias/variance as a function of the regularization parameter $\lambda$

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

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

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





Machine Learning

# Advice for applying machine learning

---

## Learning curves



# Learning curves

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

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



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



## High bias



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



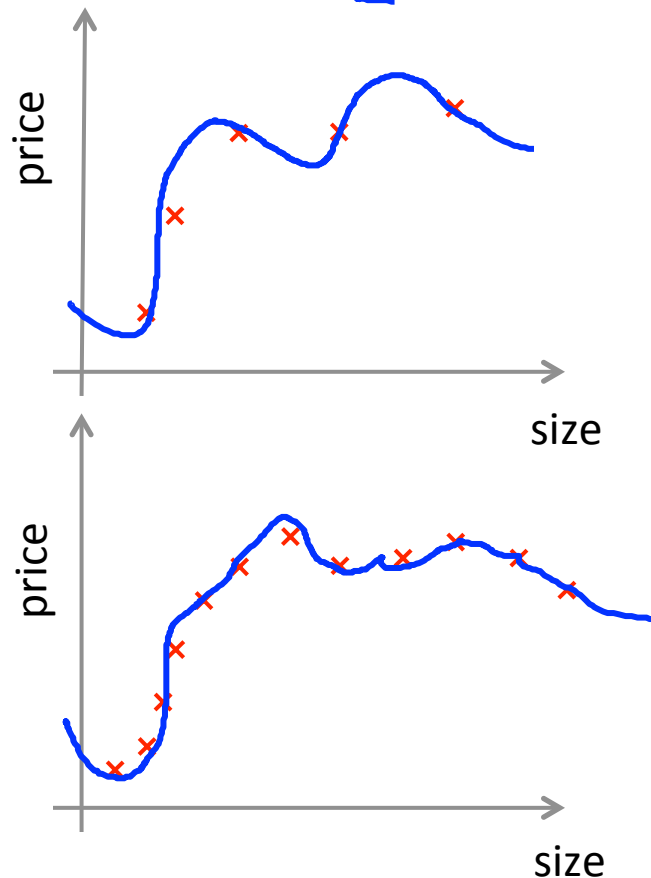
## High variance



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

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

(and small  $\lambda$ )  $\nearrow$





Machine Learning

# Advice for applying machine learning

---

## Deciding what to try next (revisited)

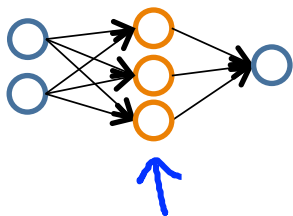
## Debugging a learning algorithm:

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 ( $x_1^2, x_2^2, x_1x_2$ , etc) → fixes high bias.
- Try decreasing  $\lambda$  → fixes high bias
- Try increasing  $\lambda$  → fixes high variance

# 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 ( $\lambda$ ) to address overfitting.

$J_{co}(\theta)$  ↑