

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?

- 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 , etc.)
- Try decreasing λ
- Try increasing λ

$x_1, x_2, x_3, \dots, x_{100}$

Evaluating your hypothesis

Dataset:

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

20%

30%

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

→ $(x_{test}^{(1)}, y_{test}^{(1)})$
 $(x_{test}^{(2)}, y_{test}^{(2)})$
⋮
 $(x_{test}^{(m_{test})}, y_{test}^{(m_{test})})$

Training/testing procedure for linear regression

- - Learn parameter θ from training data (minimizing training error $J(\theta)$)
70%
- Compute test set error:

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

Training/testing procedure for logistic regression

- - Learn parameter θ from training data
- - Compute test set error:
$$J_{test}(\theta) = -\frac{1}{m_{test}} \sum_{i=1}^{m_{test}} y_{test}^{(i)} \log h_\theta(x_{test}^{(i)}) + (1 - y_{test}^{(i)}) \log (1 - h_\theta(x_{test}^{(i)}))$$

Training/testing procedure for logistic regression

- - Learn parameter θ from training data
- - Compute test set error: m_{test}
- $J_{test}(\theta) = -\frac{1}{m_{test}} \sum_{i=1}^{m_{test}} y_{test}^{(i)} \log h_\theta(x_{test}^{(i)}) + (1 - y_{test}^{(i)}) \log h_\theta(x_{test}^{(i)})$
- Misclassification error (0/1 misclassification error):

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

or if $h_\theta(x) < 0.5, \quad y = 1$] error

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

Model selection

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

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

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

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

:

:

:

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

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



How well does the model generalize? Report test set

error $\underline{J_{test}(\theta^{(5)})}$. 

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

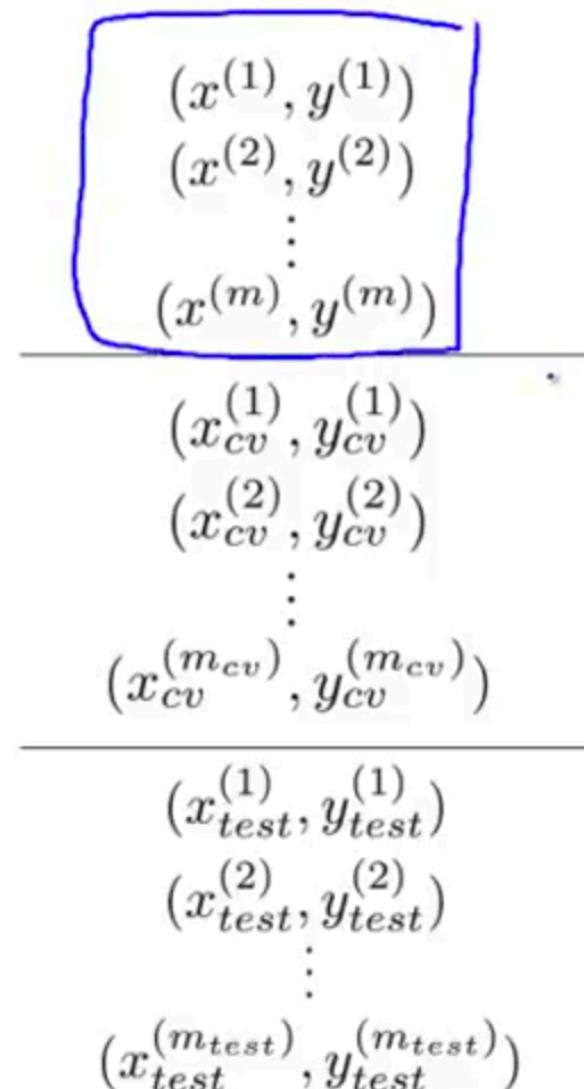
Evaluating your hypothesis

Dataset:

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

Annotations:

- Handwritten blue text: "60%" is written next to the first six rows.
- Handwritten blue bracket: A curly brace groups the first six rows under the heading "Training set".
- Handwritten blue bracket: A curly brace groups the last two rows under the heading "test set".
- Handwritten blue bracket: A curly brace groups the last three rows under the heading "Cross validation (CV)".



Train/validation/test error

Training error:

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

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

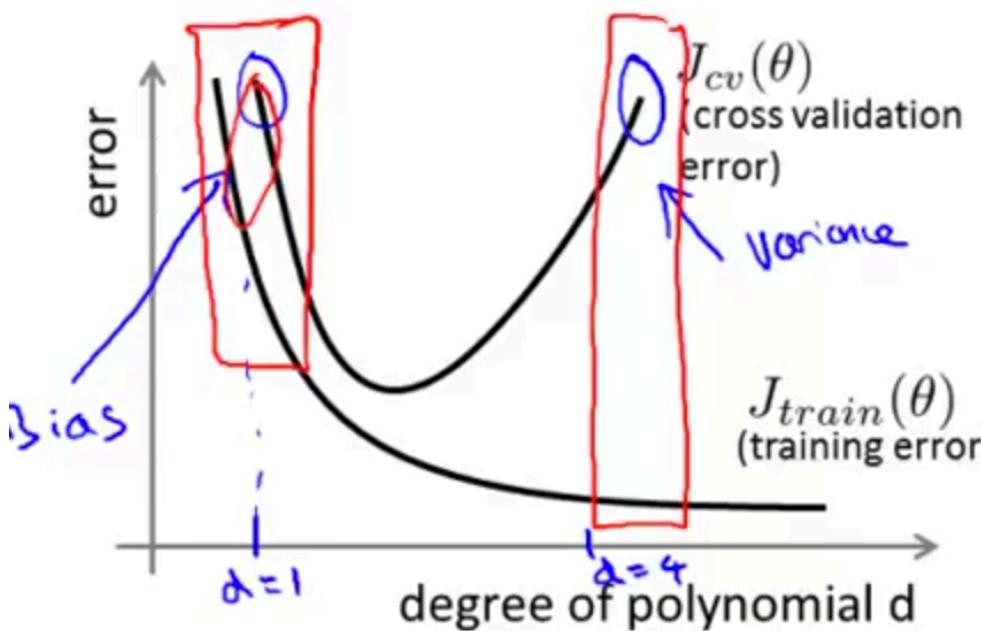
1. $h_{\theta}(x) = \theta_0 + \theta_1 x \rightarrow \min_{\theta} J(\theta) \rightarrow \theta^{(1)} \rightarrow J_{cv}(\theta^{(1)})$
2. $h_{\theta}(x) = \theta_0 + \theta_1 x + \theta_2 x^2 \rightarrow \theta^{(2)} \rightarrow J_{cv}(\theta^{(2)})$
3. $h_{\theta}(x) = \theta_0 + \theta_1 x + \dots + \theta_3 x^3 \rightarrow \theta^{(3)} \rightarrow J_{cv}(\theta^{(3)})$
- ⋮
10. $h_{\theta}(x) = \theta_0 + \theta_1 x + \dots + \theta_{10} x^{10} \rightarrow \theta^{(10)} \rightarrow J_{cv}(\theta^{(10)})$
- $d=4$

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

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

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



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 $\lambda = 0$ 
2. Try $\lambda = \underline{0.01}$
3. Try $\lambda = \underline{0.02}$
4. Try $\lambda = \underline{0.04}$
5. Try $\lambda = 0.08$
 \vdots
12. Try $\lambda = 10$

Choosing the regularization parameter λ

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

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

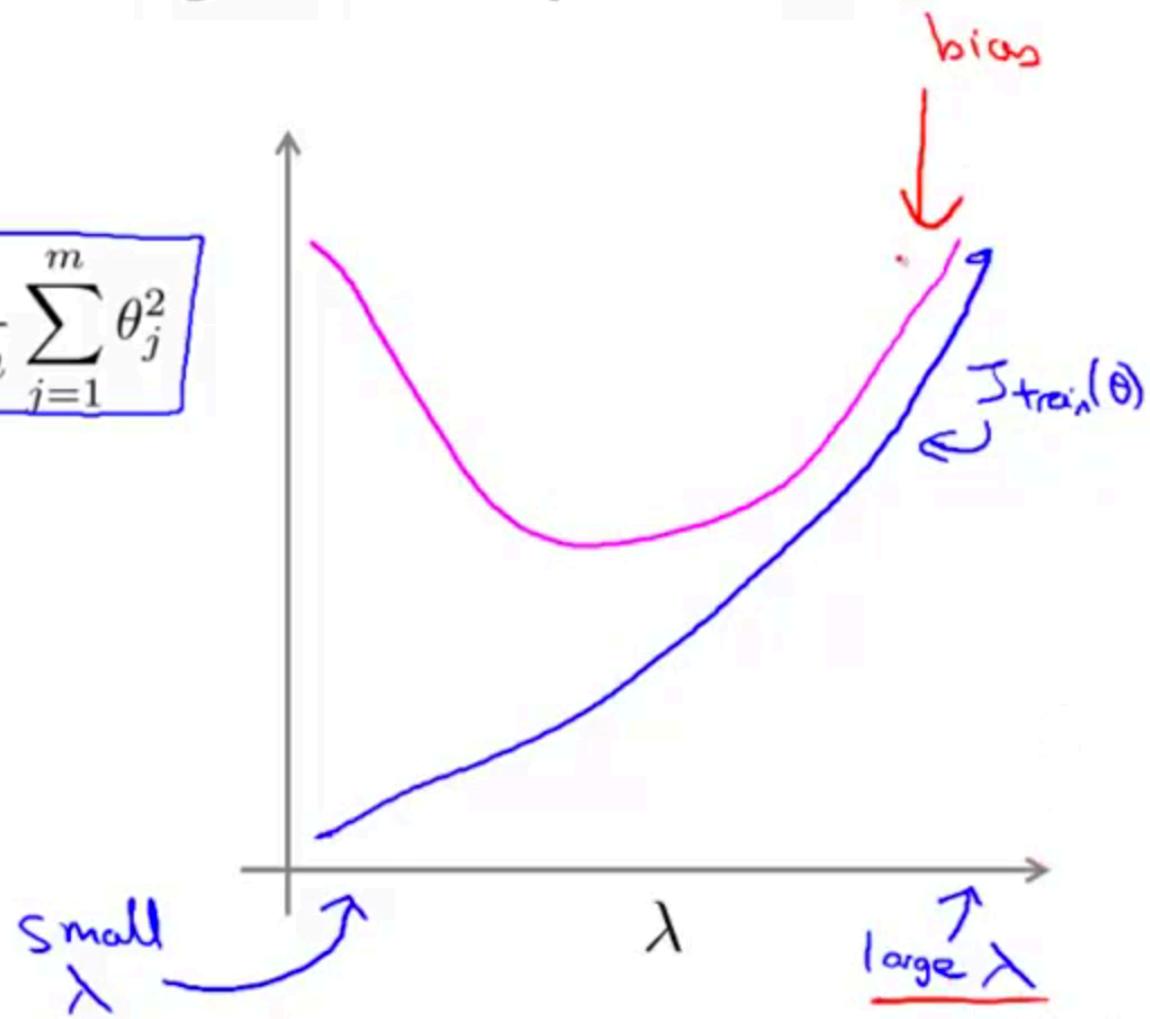
1. Try $\lambda = 0$ $\xrightarrow{\uparrow}$ $\min_{\theta} J(\theta) \rightarrow \theta^{(1)}$
2. Try $\lambda = 0.01$ $\xrightarrow{\quad}$ $\min_{\theta} J(\theta) \rightarrow \theta^{(2)}$
3. Try $\lambda = 0.02$ $\xrightarrow{\quad}$ $\theta^{(3)}$
4. Try $\lambda = 0.04$ \vdots
5. Try $\lambda = 0.08$
6. Try $\lambda = 0.16$
7. Try $\lambda = 0.32$
8. Try $\lambda = 0.64$
9. Try $\lambda = 1.28$
10. Try $\lambda = 2.56$
11. Try $\lambda = 5.12$
12. Try $\lambda = 10.24$ $\xrightarrow{\quad}$ $\theta^{(12)}$

Bias/variance as a function of the regularization parameter λ

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

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

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



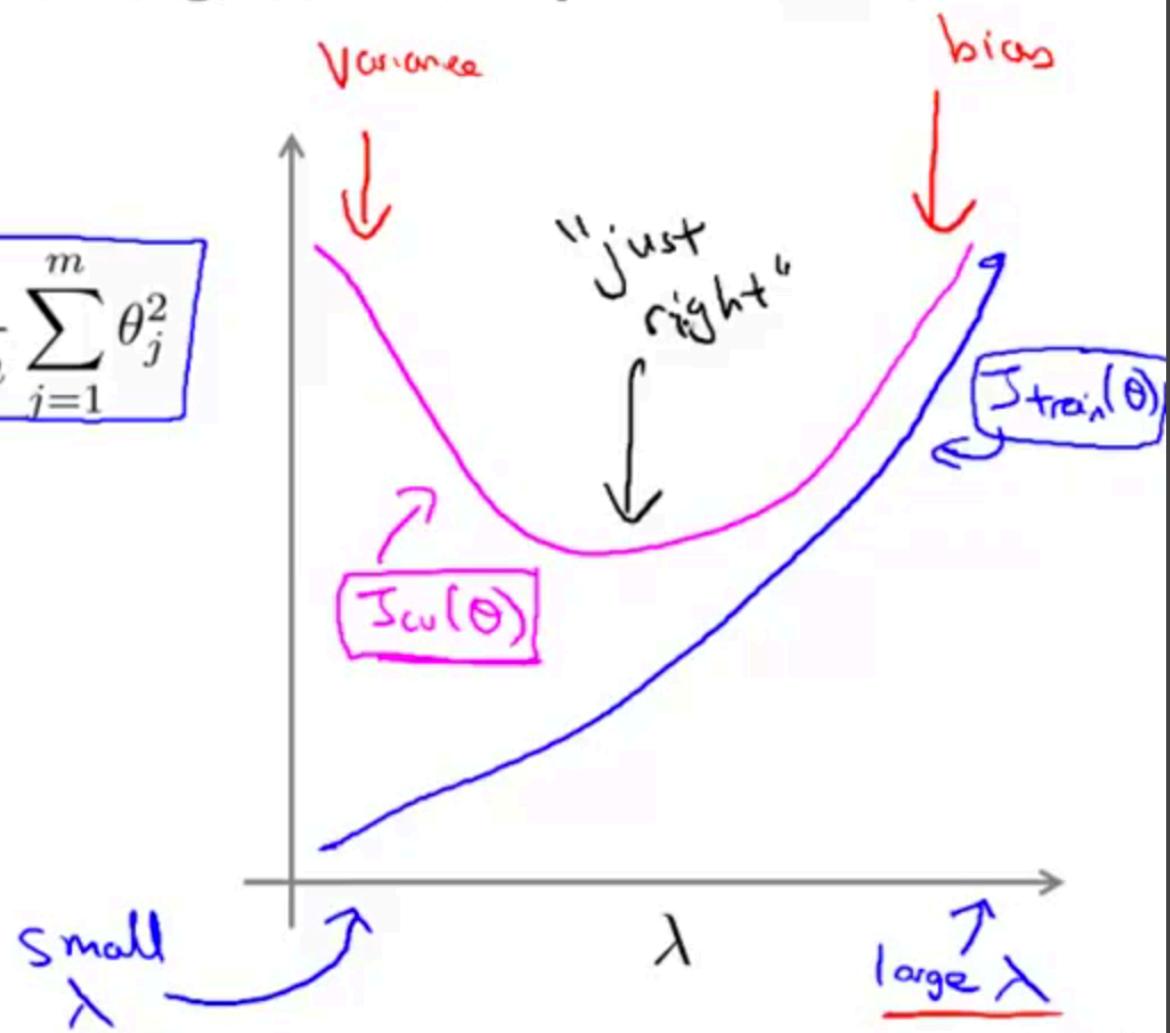
Andrew Ng

Bias/variance as a function of the regularization parameter λ

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

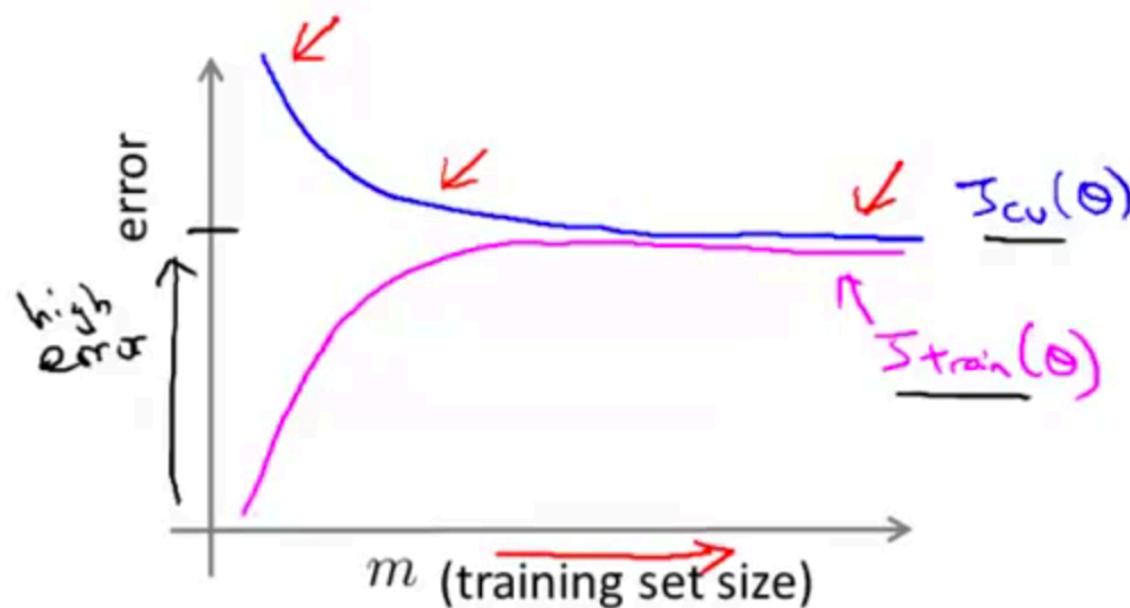
$$\rightarrow \underline{J_{train}(\theta)} = \frac{1}{2m} \sum_{i=1}^m (h_\theta(x^{(i)}) - y^{(i)})^2$$

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

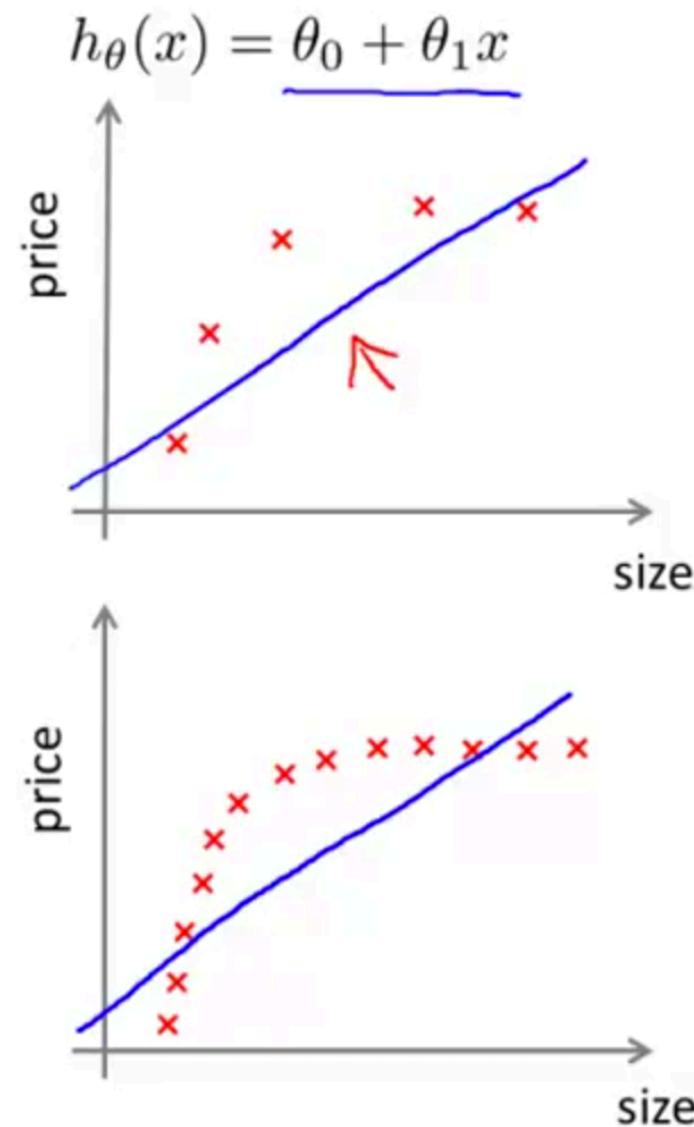


Andrew Ng

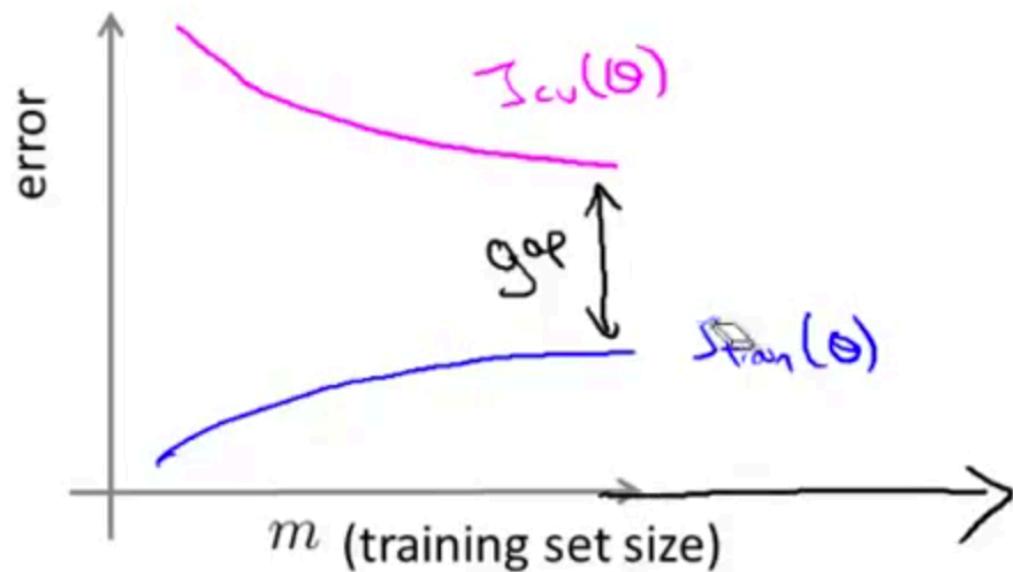
High bias



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



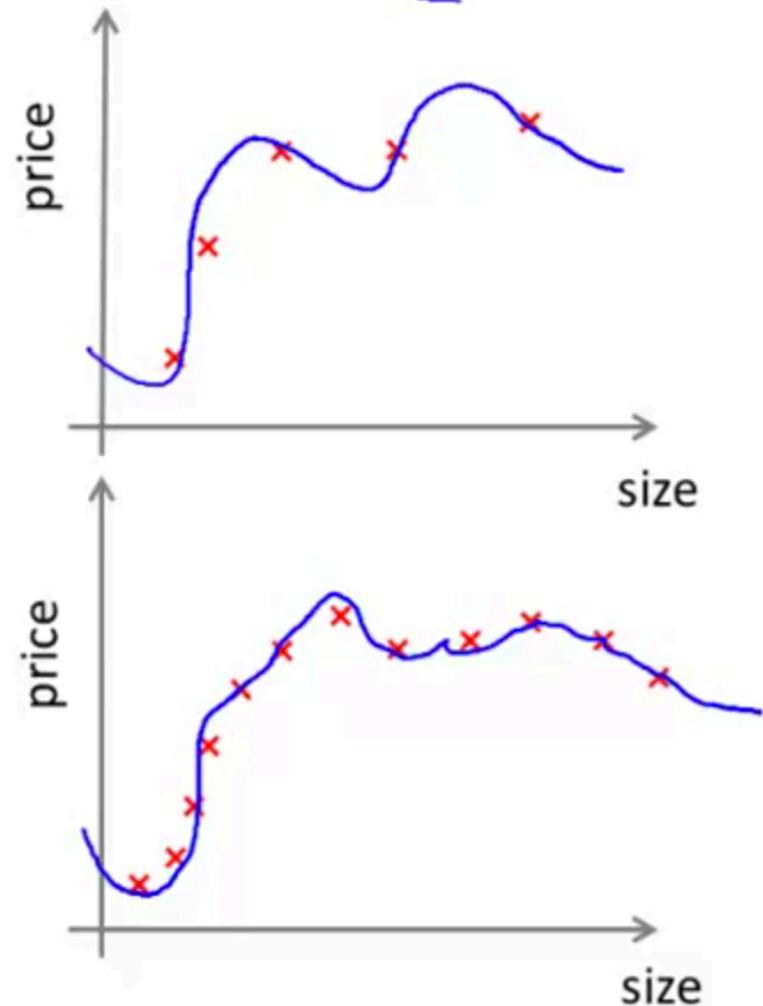
High variance



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

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

(and small λ)



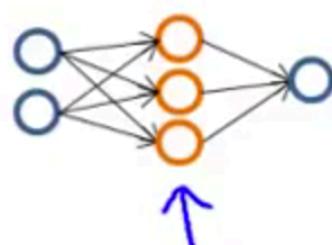
Debugging a learning algorithm:

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 λ → fixes high bias
- Try increasing λ → fixes high variance .

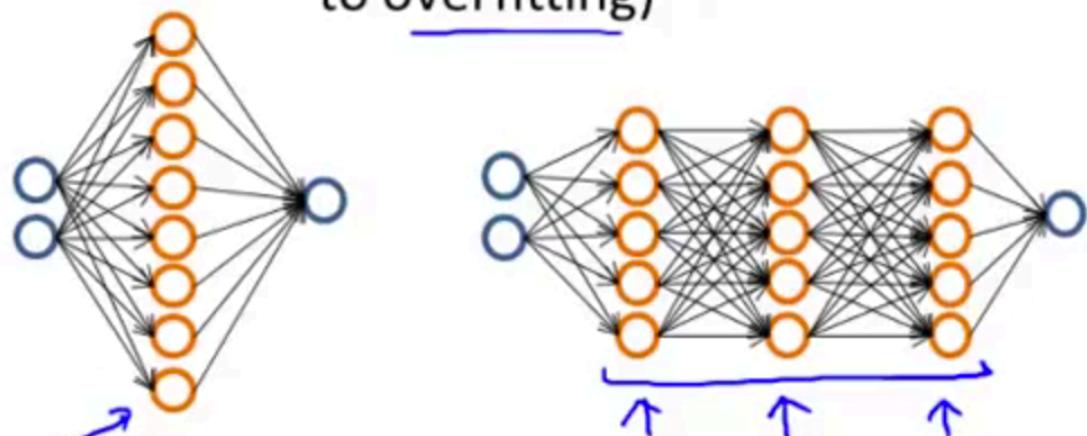
Neural networks and overfitting

→ “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 (λ) to address overfitting.

$$J_{c_0}(\Theta) \quad \uparrow$$

Building a spam classifier

Supervised learning. $x = \text{features of email}$. $y = \text{spam (1) or not spam (0)}$.

Features x : Choose 100 words indicative of spam/not spam.

E.g. deal, buy, discount, andrew, now, ...

$$x = \begin{bmatrix} 0 & \text{andrew} \\ 1 & \text{buy} \\ 1 & \text{deal} \\ 0 & \text{discount} \\ \vdots & \vdots \\ 1 & \text{now} \\ \vdots & \vdots \end{bmatrix} \quad x \in \mathbb{R}^{100}$$



From: cheapsales@buystufffromme.com
To: ang@cs.stanford.edu
Subject: Buy now!

Deal of the week! Buy now!

Building a spam classifier

Supervised learning. x = features of email. y = spam (1) or not spam (0).

Features x : Choose 100 words indicative of spam/not spam.

E.g. deal, buy, discount, andrew, now, ...

$$x = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ \vdots \\ 1 \\ \vdots \end{bmatrix} \begin{matrix} \text{andrew} \\ \text{buy} \\ \text{deal} \\ \text{discount} \\ \vdots \\ \text{now} \\ \vdots \end{matrix} \quad x \in \mathbb{R}^{100}$$

$$x_j = \begin{cases} 1 & \text{if word } j \text{ appears} \\ 0 & \text{otherwise.} \end{cases}$$

From: cheapsales@buystufffromme.com
To: ang@cs.stanford.edu
Subject: Buy now!

Deal of the week! Buy now!

Note: In practice, take most frequently occurring n words (10,000 to 50,000) in training set, rather than manually pick 100 words.

Building a spam classifier

How to spend your time to make it have low error?

- Collect lots of data
 - E.g. “honeypot” project.
- Develop sophisticated features based on email routing information (from email header).
- Develop sophisticated features for message body, e.g. should “discount” and “discounts” be treated as the same word? How about “deal” and “Dealer”? Features about punctuation?
- Develop sophisticated algorithm to detect misspellings (e.g. m0rtgage, med1cine, w4tches.)