# Model Selection
# AI4003-Applied Machine Learning

Dr. Mohsin Kamal

Department of Electrical Engineering
National University of Computer and Emerging Sciences, Lahore, Pakistan

**Debugging a learning algorithm:**

Suppose you have implemented regularized linear regression to predict output.

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

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

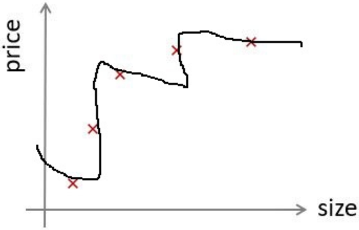
- Get more training examples
- Try smaller sets of features
- Try getting additional features
- Try adding polynomial features ($x_1^2, x_2^2, x_1x_2$ etc.)
- Try decreasing $\lambda$
- Try increasing $\lambda$

**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.

## 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 the house
$x_2$ = No. of bedrooms
$x_3$ = No. of floors
$x_4$ = Age of the house
$\vdots$
$x_{100}$ = Kitchen size

## Evaluating your hypothesis

Dataset:

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

$$\longrightarrow \quad \begin{array}{c} (x^{(1)}, y^{(1)}) \\ (x^{(2)}, y^{(2)}) \\ \vdots \\ (x^{(m)}, y^{(m)}) \end{array}$$

$$\longrightarrow \quad \begin{array}{c} (x_{test}^{(1)}, y_{test}^{(1)}) \\ (x_{test}^{(2)}, y_{test}^{(2)}) \\ \vdots \\ (x_{test}^{(m_{test})}, y_{test}^{(m_{test})}) \end{array}$$

# Training/testing procedure for linear regression

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

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

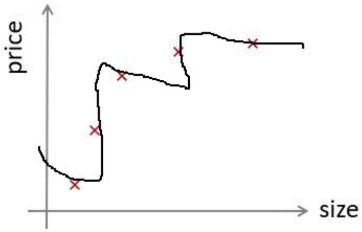
# Training/testing procedure for logistic regression

- Learn parameter $\theta$ from training data
- Compute test set error:

$$J(\theta) = -\frac{1}{m_{test}} \sum_{i=1}^{m_{test}} y_{test}^{(i)} \log h_\theta(x_{test}^{(i)}) + (1 - y_{test}^{(i)}) \log(h_\theta(x_{test}^{(i)}))$$

- Misclassification error (0/1 misclassification error):
  - $err(h_\theta, y) = 1 \quad if \;\; h_\theta \geq 0.5, y = 0$
  - $err(h_\theta, y) = 1 \quad if \;\; h_\theta < 0.5, y = 1$
  - $err(h_\theta, y) = 0 \quad otherwise$
  - $Test\ error = \frac{1}{m_{test}} \sum_{i=1}^{m_{test}} err(h_\theta(x_{test}^{(i)}, y_{test}^{(i)})$

# Overfitting Example




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

Once parameters $\theta_0, \theta_1, \cdots, \theta_4$ 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. $h_\theta = \theta_0 + \theta_1 x$
2. $h_\theta = \theta_0 + \theta_1 x + \theta_2 x^2$
3. $h_\theta = \theta_0 + \theta_1 x + \cdots + \theta_3 x^3$

   $\vdots$

10. $h_\theta = \theta_0 + \theta_1 x + \cdots + \theta_{10} x^{10}$

Lets assume that we choose $\theta_0 + \cdots + \theta_5 x^5$ because it gives low value for $J_{train}(\theta)$

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 |
|---|---|---|
| Training set | 2104 | 400 |
| | 1600 | 330 |
| | 2400 | 369 |
| | 1416 | 232 |
| | 3000 | 540 |
| | 1985 | 300 |
| Cross validation set | 1534 | 315 |
| | 1427 | 199 |
| Test set | 1380 | 212 |
| | 1494 | 243 |

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

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

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

# 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. $h_\theta = \theta_0 + \theta_1 x$
2. $h_\theta = \theta_0 + \theta_1 x + \theta_2 x^2$
3. $h_\theta = \theta_0 + \theta_1 x + \cdots + \theta_3 x^3$

$\vdots$

10. $h_\theta = \theta_0 + \theta_1 x + \cdots + \theta_{10} x^{10}$

Let's assume that we get minimum cross validation error on polynomial order = 4
We pick $\theta_0 + \theta_1 x_1 + \cdots + \theta_4 x^4$
Estimate generalization error for test set $J_{test}(\theta^{(4)})$

# Bias/variance

## Bias/variance


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

High bias (underfit) | "Just right" | High variance (overfit)

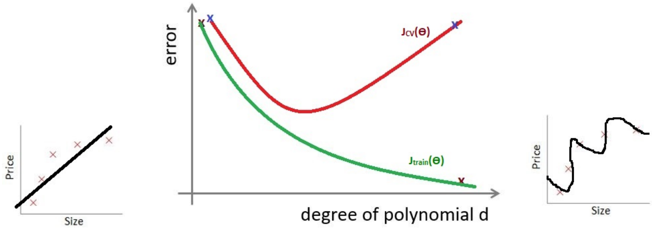
# Bias/variance

**Training error:**

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

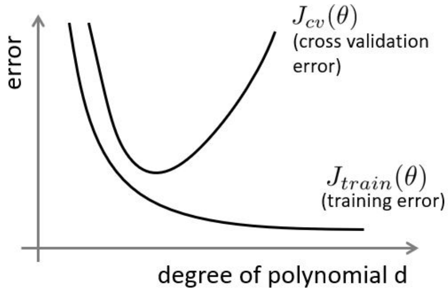
**Cross validation error:**

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



# 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?



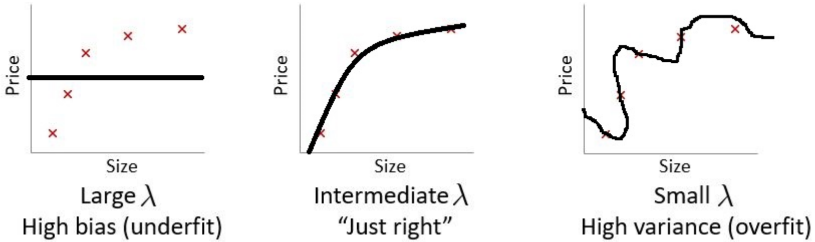

Bias (underfit):
$J_{train}(\theta)$ will be high
$J_{cv}(\theta) \approx J_{train}(\theta)$

Variance (overfit):
$J_{train}(\theta)$ will be low
$J_{cv}(\theta) >> J_{train}(\theta)$

# Linear regression with regularization

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

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



# Choosing the regularization parameter $\lambda$

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

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

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

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

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

# Choosing the regularization parameter $\lambda$

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

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

1 Try: $\lambda = 0$

2 Try: $\lambda = 0.01$

3 Try: $\lambda = 0.02$

4 Try: $\lambda = 0.04$

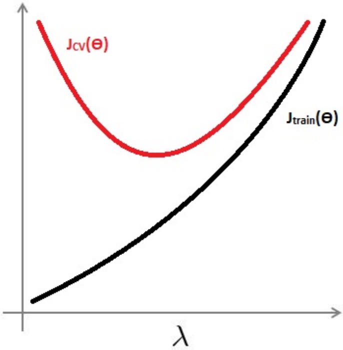
5 Try: $\lambda = 0.08$

$\vdots$

12 Try: $\lambda \approx 10$

For example $\theta^{(5)}$ gives low value for $J_{cv}(\theta^{(5)})$, then compute $J_{test}(\theta^{(5)})$

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




$$J(\theta) =$$

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

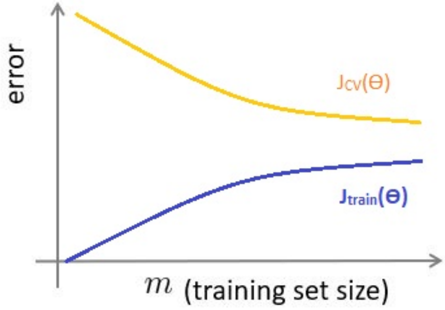
$$J_{train}(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2$$

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

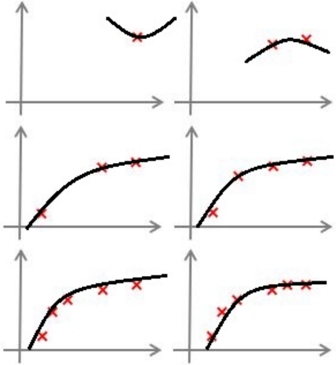
# Learning curves

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

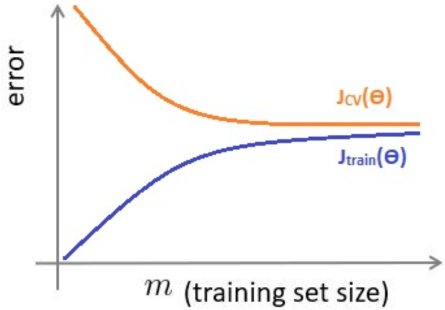
$$J_{cv}(\theta) = \frac{1}{2m_{cv}}\sum_{i=1}^{m_{cv}}(h_\theta(x_{cv}^{(i)}) - y_{cv}^{(i)})^2$$




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

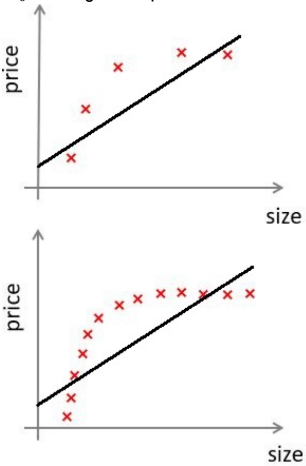


# High bias

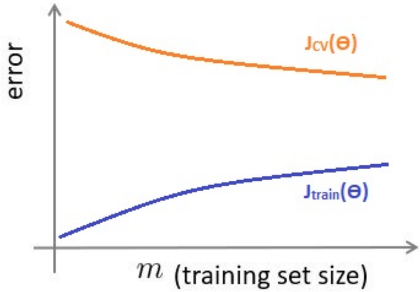



If a learning algorithm is suffering from high bias, getting more training data will not (by itself) help much.

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

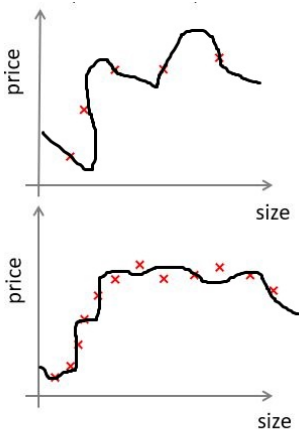


# High variance




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

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



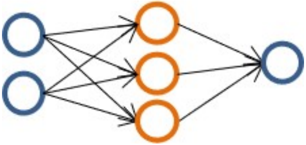
**Debugging a learning algorithm:**

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 $\rightarrow$ fixes high variance
- Try smaller sets of features $\rightarrow$ fixes high variance
- Try getting additional features $\rightarrow$ fixes high bias
- Try adding polynomial features ($x_1^2, x_2^2, x_1 x_2$ etc.) $\rightarrow$ fixes high bias
- Try decreasing $\lambda$ $\rightarrow$ fixes high bias
- Try increasing $\lambda$ $\rightarrow$ fixes high variance

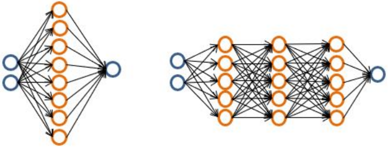
# Neural networks and overfitting

"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.