

# Week 3

## Advice for Machine Learning

### Debugging

### Evaluating a model

#### Computing Test and Train set error

#### Cross-validation set

### Bias and Variance

## ML Development Process

### Iterative Loop

#### Example with Spam Classification

### Adding Data

## Advice for Machine Learning

### Debugging

## Debugging a learning algorithm

You've implemented regularized linear regression on housing prices

$$\rightarrow J(\vec{w}, b) = \frac{1}{2m} \sum_{i=1}^m (f_{\vec{w}, b}(\vec{x}^{(i)}) - y^{(i)})^2 + \frac{\lambda}{2m} \sum_{j=1}^n w_j^2$$

But it makes unacceptably large errors in predictions. What do you try next?

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

### Evaluating a model

- Splitting datasets into train set and test set (Prevent overfitting etc.)

## Evaluating your model

Dataset:

	size	price	
70%	2104	400	$\left. \begin{array}{l} \text{training set} \\ m_{train} = \text{no. training examples} \\ = 7 \end{array} \right\} \rightarrow \begin{array}{l} (x^{(1)}, y^{(1)}) \\ (x^{(2)}, y^{(2)}) \\ \vdots \\ (x^{(m_{train})}, y^{(m_{train})}) \end{array}$
	1600	330	
	2400	369	
	1416	232	
	3000	540	
	1985	300	
	1534	315	
30%	1427	199	$\left. \begin{array}{l} \text{test set} \\ m_{test} = \text{no. test examples} \\ = 3 \end{array} \right\} \rightarrow \begin{array}{l} (x_{test}^{(1)}, y_{test}^{(1)}) \\ \vdots \\ (x_{test}^{(m_{test})}, y_{test}^{(m_{test})}) \end{array}$
	1380	212	
	1494	243	

### Computing Test and Train set error

- For Linear Regression:
- Notice that the regularization term is removed

## Train/test procedure for linear regression (with squared error cost)

Fit parameters by minimizing cost function  $J(\vec{w}, b)$

$$\rightarrow J(\vec{w}, b) = \left[ \frac{1}{2m_{\text{train}}} \sum_{i=1}^{m_{\text{train}}} (f_{\vec{w}, b}(\vec{x}^{(i)}) - y^{(i)})^2 + \frac{\lambda}{2m_{\text{train}}} \sum_{j=1}^n w_j^2 \right]$$

Compute test error:

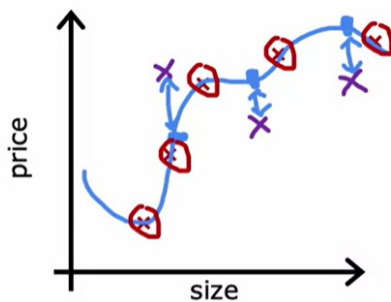
$$J_{\text{test}}(\vec{w}, b) = \frac{1}{2m_{\text{test}}} \left[ \sum_{i=1}^{m_{\text{test}}} (f_{\vec{w}, b}(\vec{x}_{\text{test}}^{(i)}) - y_{\text{test}}^{(i)})^2 \right] \quad \text{with } \sum_{j=1}^n w_j^2 \text{ crossed out}$$

Compute training error:

$$J_{\text{train}}(\vec{w}, b) = \frac{1}{2m_{\text{train}}} \left[ \sum_{i=1}^{m_{\text{train}}} (f_{\vec{w}, b}(\vec{x}_{\text{train}}^{(i)}) - y_{\text{train}}^{(i)})^2 \right]$$

- Check cost of test set to see whether model is effective

## Train/test procedure for linear regression (with squared error cost)



$x = \text{train}$

$x = \text{test}$

$J_{\text{train}}(\vec{w}, b)$  will be low

$J_{\text{test}}(\vec{w}, b)$  will be high

- For Classification problems:
  - You can choose to compute the loss, or compute the fraction of the train/test set that was misclassified

## Train/test procedure for classification problem

0/1

Fit parameters by minimizing  $J(\vec{w}, b)$  to find  $\vec{w}, b$

E.g.,

$$J(\vec{w}, b) = -\frac{1}{m_{\text{train}}} \sum_{i=1}^{m_{\text{train}}} [y^{(i)} \log(f_{\vec{w}, b}(\vec{x}^{(i)})) + (1 - y^{(i)}) \log(1 - f_{\vec{w}, b}(\vec{x}^{(i)}))] + \frac{\lambda}{2m_{\text{train}}} \sum_{j=1}^n w_j^2$$

Compute test error:

$$J_{\text{test}}(\vec{w}, b) = -\frac{1}{m_{\text{test}}} \sum_{i=1}^{m_{\text{test}}} [y_{\text{test}}^{(i)} \log(f_{\vec{w}, b}(\vec{x}_{\text{test}}^{(i)})) + (1 - y_{\text{test}}^{(i)}) \log(1 - f_{\vec{w}, b}(\vec{x}_{\text{test}}^{(i)}))]$$

Compute train error:

$$J_{\text{train}}(\vec{w}, b) = -\frac{1}{m_{\text{train}}} \sum_{i=1}^{m_{\text{train}}} [y_{\text{train}}^{(i)} \log(f_{\vec{w}, b}(\vec{x}_{\text{train}}^{(i)})) + (1 - y_{\text{train}}^{(i)}) \log(1 - f_{\vec{w}, b}(\vec{x}_{\text{train}}^{(i)}))]$$

## Train/test procedure for classification problem

fraction of the test set and the fraction of the train set  
that the algorithm has misclassified.

$$\hat{y} = \begin{cases} 1 & \text{if } f_{\vec{w}, b}(\vec{x}^{(i)}) \geq 0.5 \\ 0 & \text{if } f_{\vec{w}, b}(\vec{x}^{(i)}) < 0.5 \end{cases}$$

count  $\hat{y} \neq y$

$J_{\text{test}}(\vec{w}, b)$  is the fraction of the test set that has been misclassified.

$J_{\text{train}}(\vec{w}, b)$  is the fraction of the train set that has been misclassified.

## Cross-validation set

- Helps with model selection and fine-tuning parameters
- Pick the model with the lowest cross validation error

## Training/cross validation/test set

Training error:  $J_{train}(\vec{w}, b) = \frac{1}{2m_{train}} \left[ \sum_{i=1}^{m_{train}} (f_{\vec{w}, b}(\vec{x}^{(i)}) - y^{(i)})^2 \right]$

Cross validation error:  $J_{cv}(\vec{w}, b) = \frac{1}{2m_{cv}} \left[ \sum_{i=1}^{m_{cv}} (f_{\vec{w}, b}(\vec{x}_{cv}^{(i)}) - y_{cv}^{(i)})^2 \right]$  (validation error, dev error)

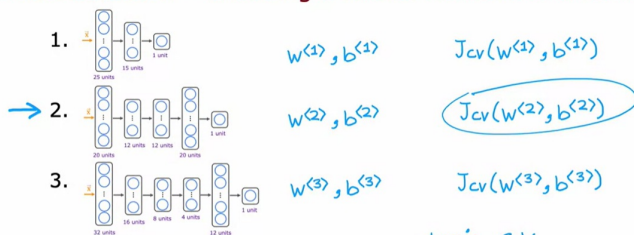
Test error:  $J_{test}(\vec{w}, b) = \frac{1}{2m_{test}} \left[ \sum_{i=1}^{m_{test}} (f_{\vec{w}, b}(\vec{x}_{test}^{(i)}) - y_{test}^{(i)})^2 \right]$

### Model selection

$d=1$  1.  $f_{\vec{w}, b}(\vec{x}) = w_1x + b$   $w^{<1>, b^{<1>}} \rightarrow J_{cv}(w^{<1>, b^{<1>})$   
 $d=2$  2.  $f_{\vec{w}, b}(\vec{x}) = w_1x + w_2x^2 + b$   $\rightarrow J_{cv}(w^{<2>, b^{<2>})$   
 $d=3$  3.  $f_{\vec{w}, b}(\vec{x}) = w_1x + w_2x^2 + w_3x^3 + b$   $\vdots$   
 $\vdots$   
 $d=10$  10.  $f_{\vec{w}, b}(\vec{x}) = w_1x + w_2x^2 + \dots + w_{10}x^{10} + b$   $J_{cv}(w^{<10>, b^{<10>})$

$\rightarrow$  Pick  $w_1x + \dots + w_4x^4 + b$  ( $J_{cv}(w^{<4>, b^{<4>})$ )  
 Estimate generalization error using test the set:  $J_{test}(w^{<4>, b^{<4>})$

### Model selection – choosing a neural network architecture



Pick  $w^{<2>, b^{<2>}}$

+ train, CV

Estimate generalization error using the test set:  $J_{test}(w^{<2>, b^{<2>})$

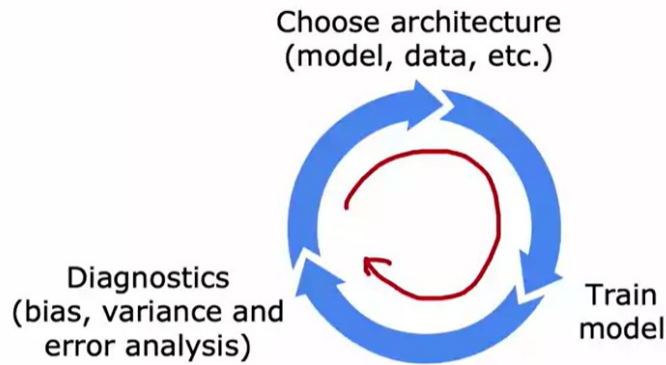
## Bias and Variance

{TBC}

## ML Development Process

### Iterative Loop

# Iterative loop of ML development



## Example with Spam Classification

- You can also manually spot the errors (Printing the values etc) checking the words where the model fails and misclassifies

## Building a spam classifier

Supervised learning:  $\vec{x}$  = features of email

$y$  = spam (1) or not spam (0)

Features: list the top 10,000 words to compute  $x_1, x_2, \dots, x_{10,000}$

$$\vec{x} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \\ 0 \\ \vdots \end{bmatrix} \begin{matrix} a \\ andrew \\ buy \\ deal \\ discount \\ \vdots \end{matrix}$$

From: cheapsales@buystufffromme.com  
To: Andrew Ng  
Subject: Buy now!

Deal of the week! Buy now!  
Rolex w4tchs - \$100  
Medlcine (any kind) - £50  
Also low cost M0rgages  
available.

## Adding Data