Bias and Variance

## Regularization

**Underfitting/High bias**: Our model is not complex enough to capture the pattern in the training data well and therefore also suffers from low performance on unseen data.

**Overfitting/High variance**: If we have too many features, the learned hypothesis may fit the training set very well (J() ~= 0),but fail to generalize to new examples (predict prices on new examples).

Cause: 1. Too many parameters that lead to a model that is too complex given the underlying data.

2.

**addressing overfitting**:

1. Reduce number of features

    --- Manually select which features to keep

    --- Model selection algorithm (later in course)

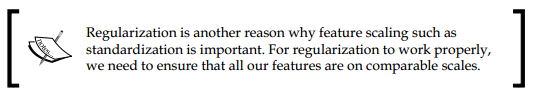
1. Regularization

--- keep all the features, but reduce magnitude/values of parameters C:\Users\phenix\AppData\Local\Temp\enhtmlclip\Image(20).png

--- Works well when we have a lot of features, each of which contributes a bit to predicting y.

--- Regularization is a very useful method to handle collinearity (high correlation among features), filter out noise from data, and eventually prevent overfitting.

L2 regularization: 



**Cost Function:**

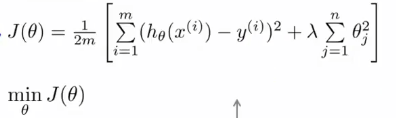
Suppose we penalize and make C:\Users\phenix\AppData\Local\Temp\enhtmlclip\Image(21).png really small.

C:\Users\phenix\AppData\Local\Temp\enhtmlclip\Image(22).png

Small values for parameters,C:\Users\phenix\AppData\Local\Temp\enhtmlclip\Image(23).png

    ---"Simplier " hypothesis

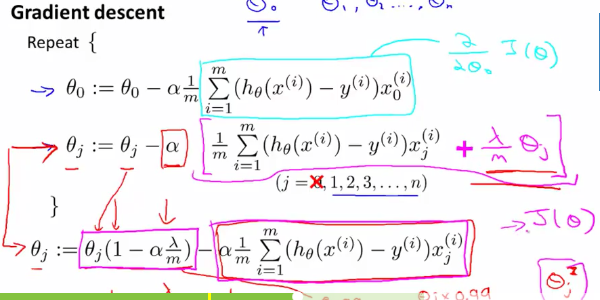
    --- Less prone to overfitting

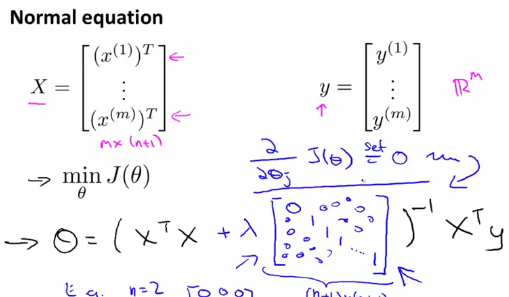


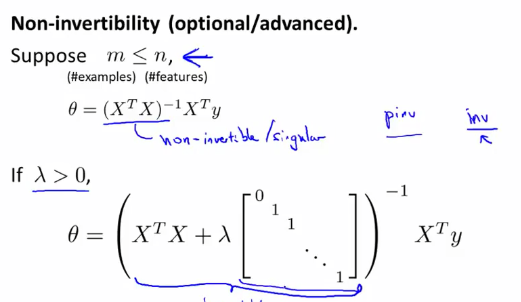
lamda is **regularization parameter**

By increasing the value of lamda, we increase the regularization strength, and the weight coefficient shrink.

**Regularized Linear Regression**







### Advice for Applying Machine Learning

#### Debugging a learning algorithm

  -- Get more training examples

  -- Try smaller sets of features

  -- Try getting additional features

  -- Try adding polynomial features

  -- Try descreasing 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 a Hypothesis**

70% training set ,30% test set

  Trianing/testing proceduce for linear regression

 -- Learning parameter C:\Users\phenix\AppData\Local\Temp\enhtmlclip\Image(31).png from training data (minimizing training error J ())

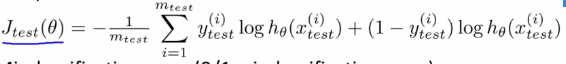
--compute test set error:



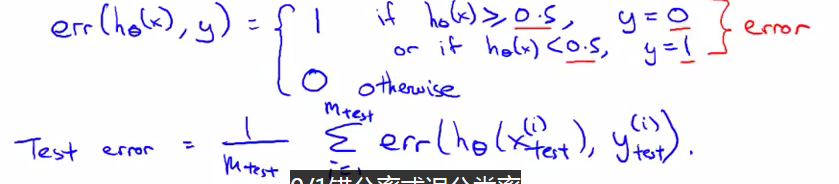
Training/testing proceduce for logistic regression

 -- Learning  from training data

  -- Compute test set error:

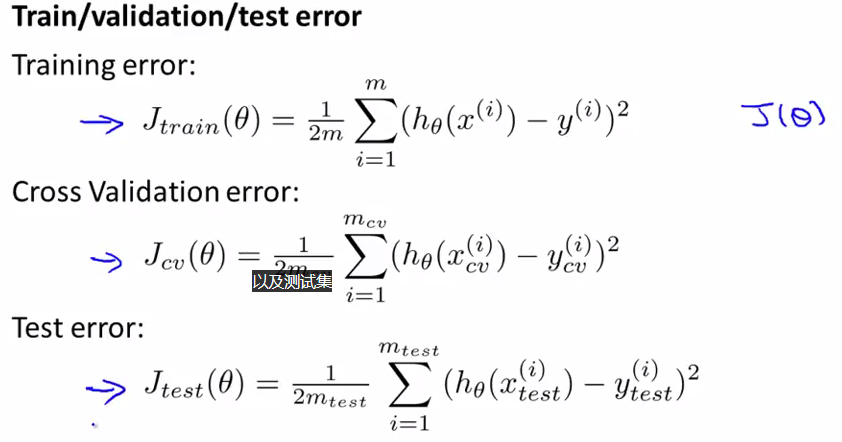


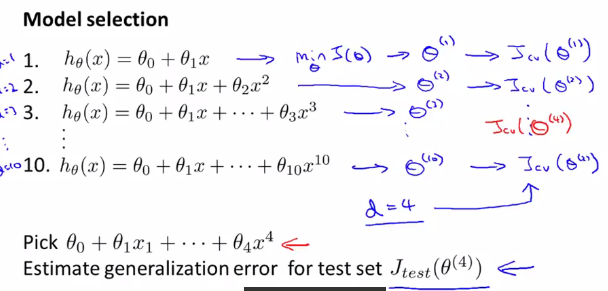
-- Misclassification error (0/1 misclassification error):

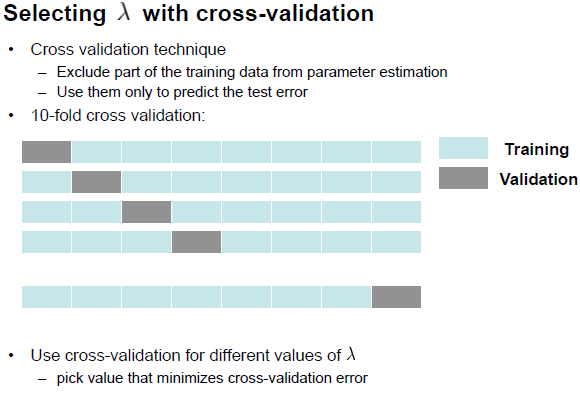


**Model Selection and Training/validation/Test Sets**

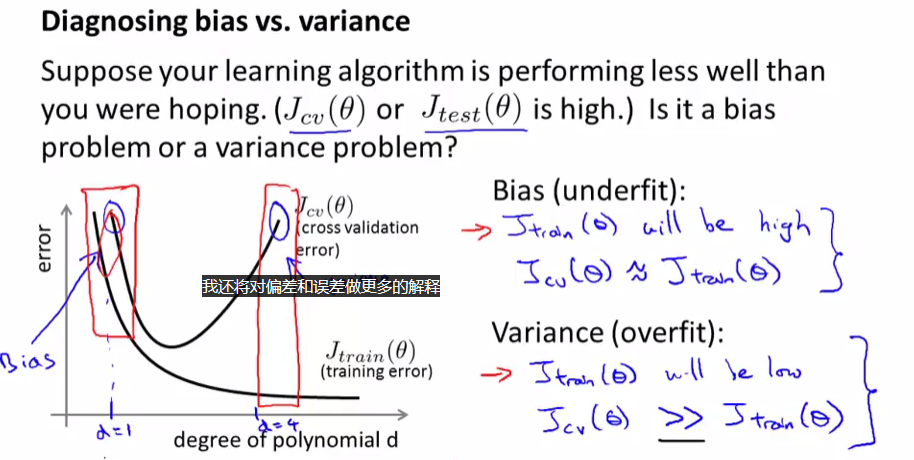
60% Training Set, 20% crsoss Validataion set , 20% test set



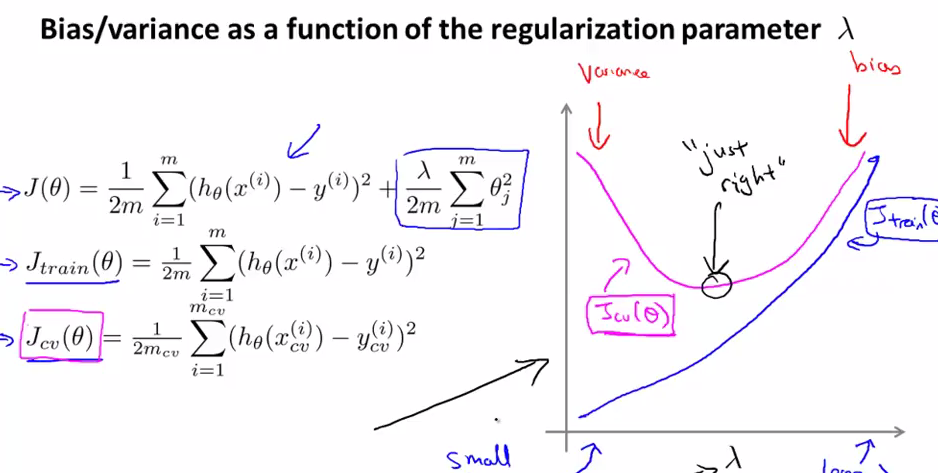




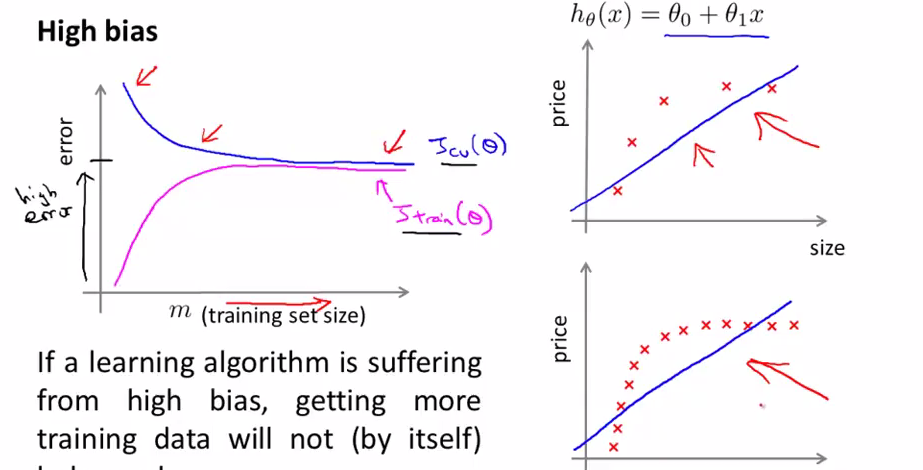
Diagnosing Bias vs Variance

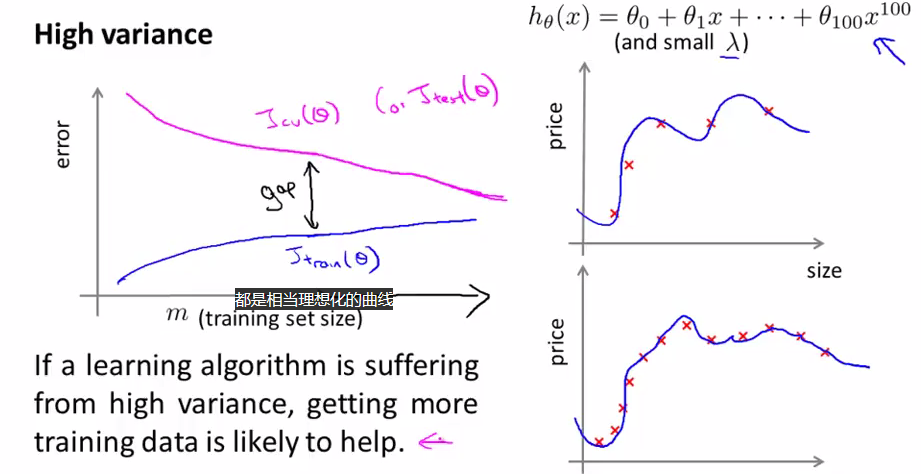


Regularization and Bias/Variance

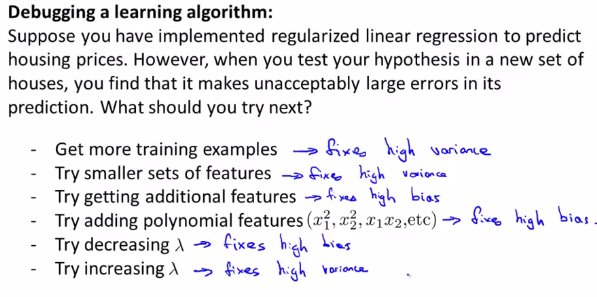


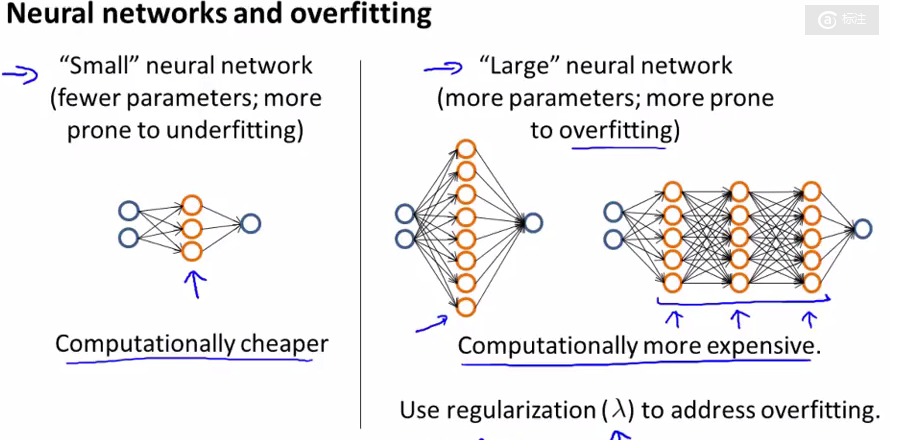
Learning Curves





Deciding What to Do Next Revisited





Feature scaling(特征缩放)

     Idea: Make sure features are on a similar scale.

     Get every feature into approximately a -1<= xi<=1 range

Mean normalization（规范化）

     Replace xi with xi-ui to make features have approximately zero mean (Do no apply to x0 =1

Reference:

1. Machine learning , Ng video.
2. Python machine learning, book.