# Advice for Applying Machine Learning

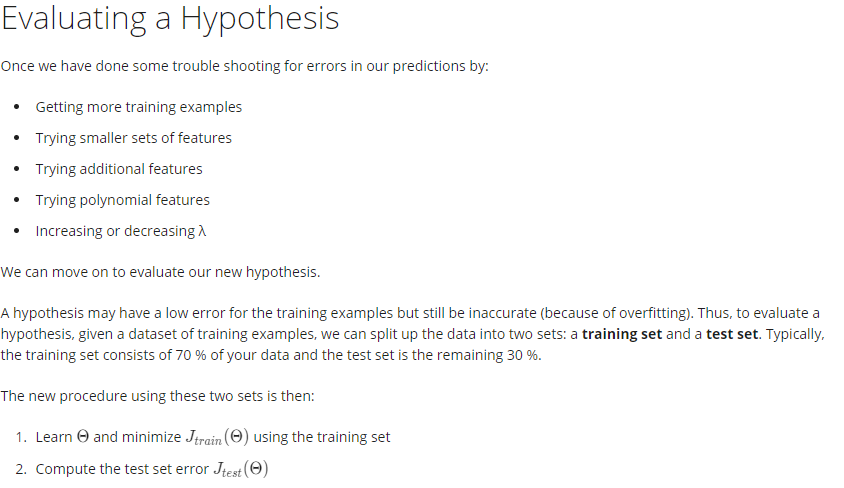
## Evaluating a Learning Algorithm

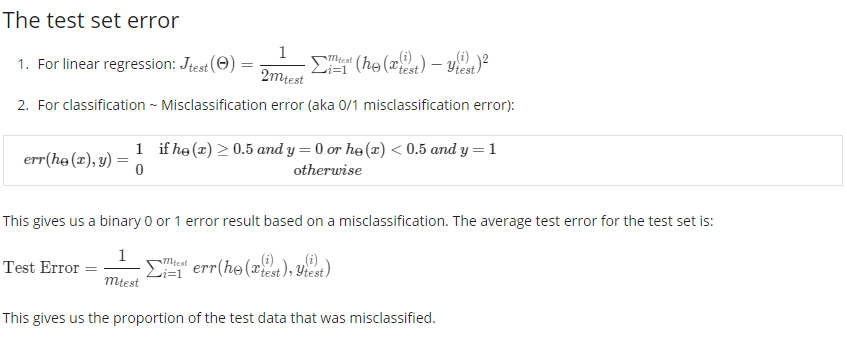
Deciding What to Try Next

* Developing or improving machine learning systems
* What to try next
* obtaining more training examples
* smaller sets of features
* getting additional features
* additional polynomial features
* decreasing λ
* increasing λ
* Diagnostics:
* What is/isn’t working
* Guidance as to how to improve performance
* Can rule out courses of action unlikely to significantly improve performance
* Take time to implement and understand

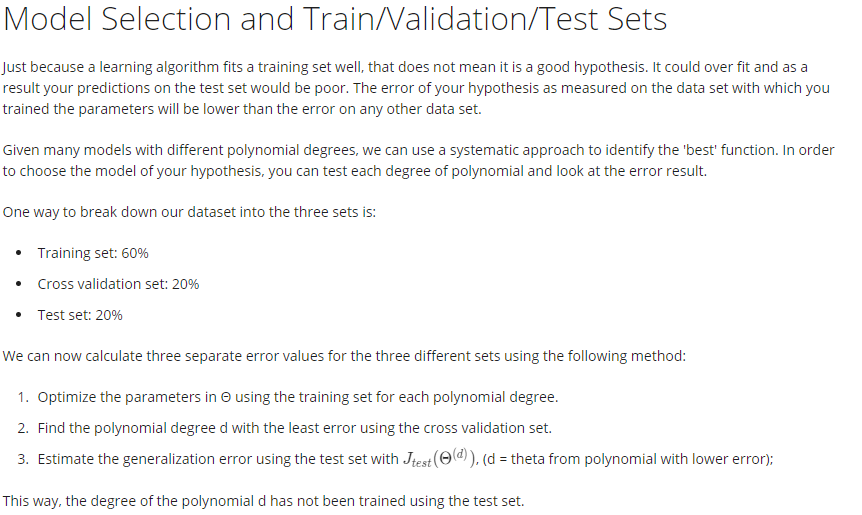
Evaluating a Hypothesis

* Failing to generalize to new examples not in training set (overfitting)
* Splitting 70/30%
* Randomize or reorder/shuffle
* Linear regression
* θ, training J, test J



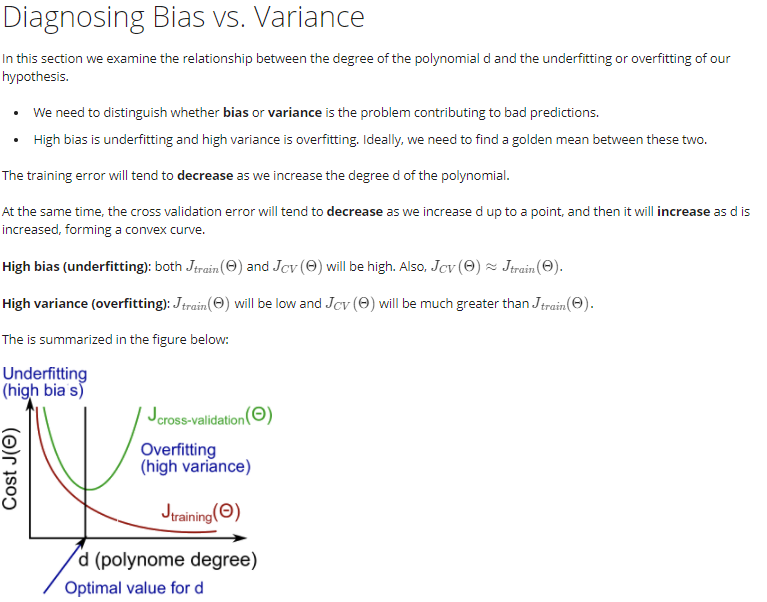


Model Selection and Train/Validation/Test Sets



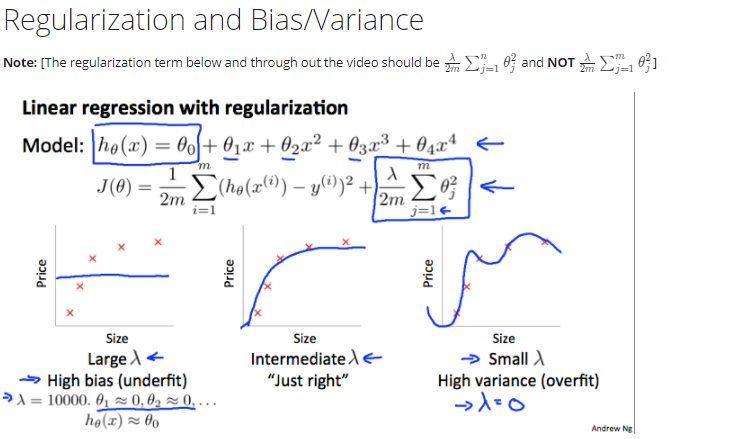
## Bias vs. Variance

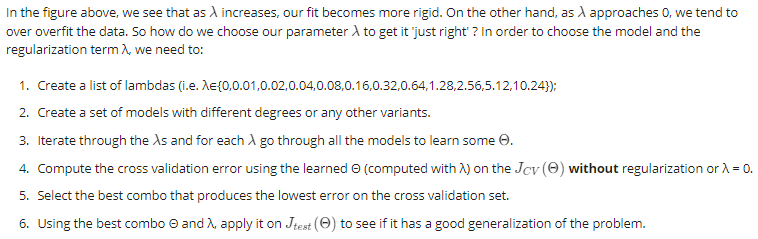
Diagnosing Bias vs. Variance



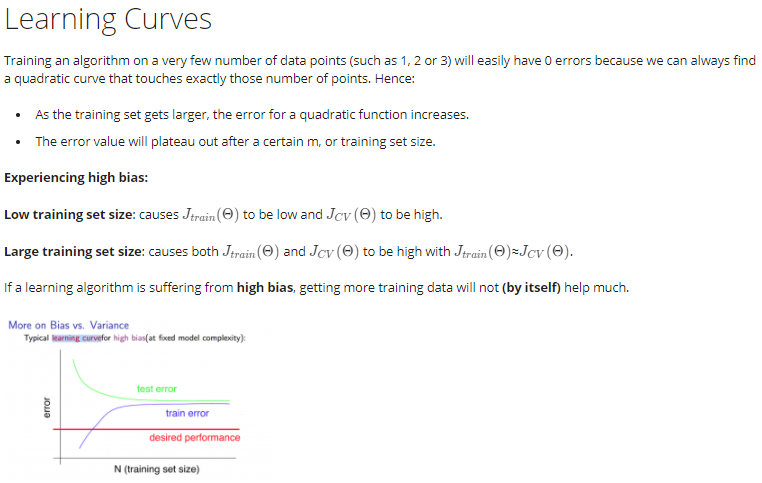
Regularization and Bias/Variance

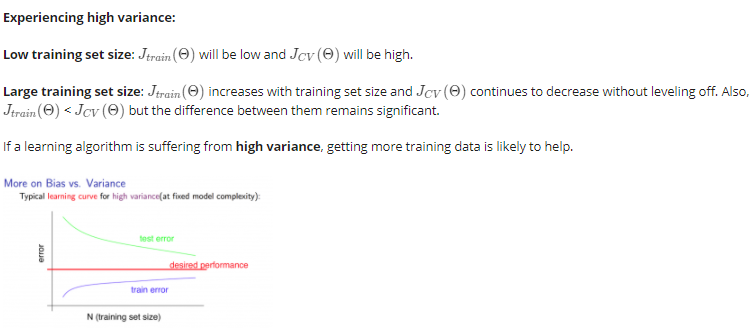
* set of models? min J?





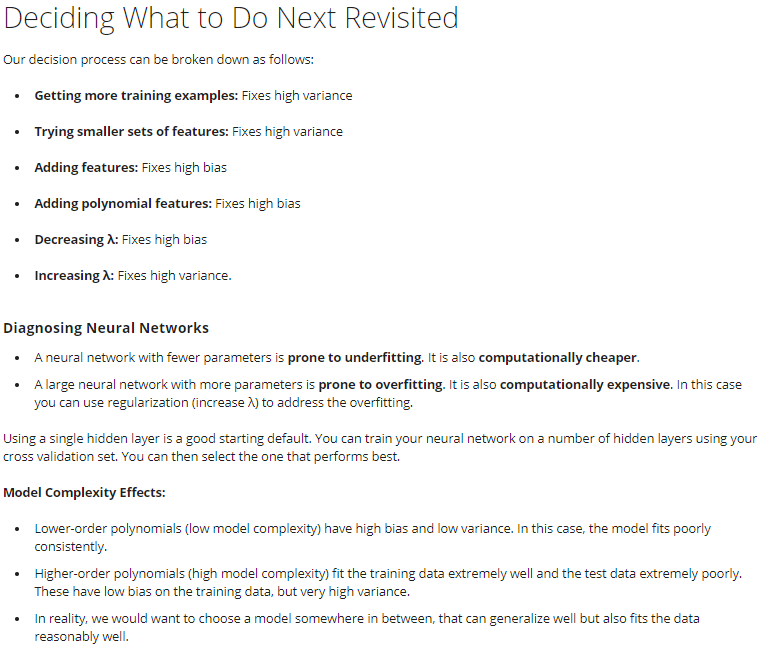
Learning Curves





Deciding What to Do Next Revisited

* Better to use more neural network hidden units then address overfitting (increase λ) than have underfitting

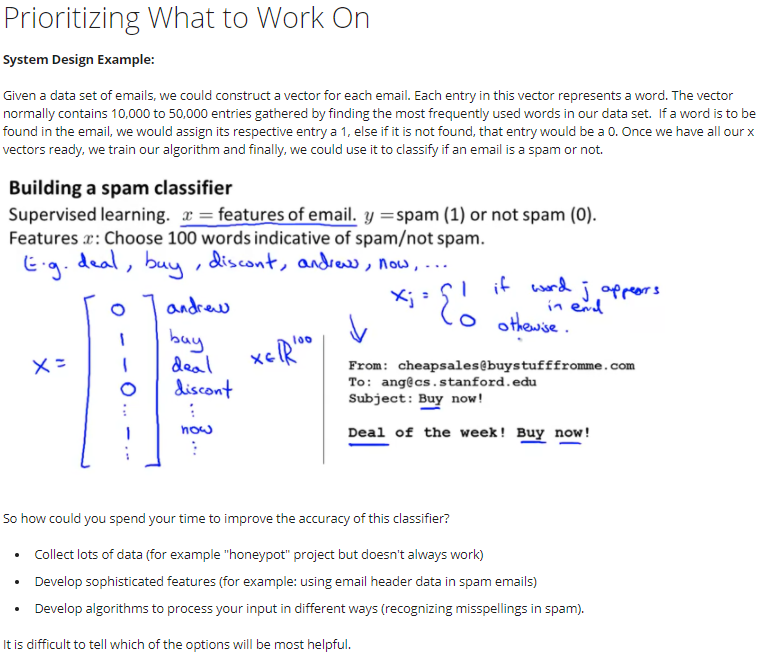


# Machine Learning System Design

## Building a Spam Classifier

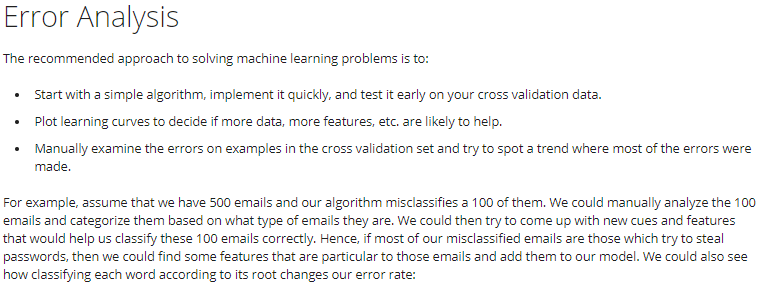
Prioritizing What to Work On

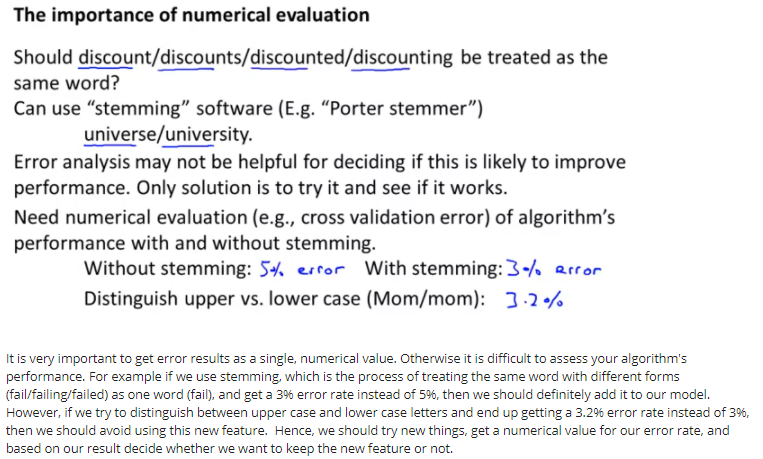
* *Machine learning system design*
* *Spam classification example*



Error Analysis

* Premature optimization
* Numerical evaluation





## Handling Skewed Data

Error Metrics for Skewed Classes

* Precision
* Of all predicted y = 1, what fraction actually was pos
* Recall
* Of all actual pos, what fraction did we detect/correctly predict
* y = 1: presence of rare case to detect

#### Trading Off Precision and Recall

* Predict 1 if h >= threshold
* Higher precision, lower recall
* Predict y = 1 only if very confident
* Higher recall, lower precision
* Avoiding missing too many cases (false negatives)
* F1 Score (F Score)

## Using Large Data Sets

#### Data for Machine Learning

* Designing high accuracy learning system
* Classifying between confusable words
* Algorithms
* Perception (Logistic regression)
* Winnow
* Memory-based
* Naïve Bayes
* Best algorithm vs. most data
* Large data rationale
* Features: given input *x*, can human expert confidently predict *y*?
* Many parameters: features/hidden units (low bias)
* small Jtrain
* very large training set: unlikely to overfit (low variance)
* Jtrain ~ Jtest (small)
* Can we obtain a large training set?

*Data: training examples*

*Parameters: features*