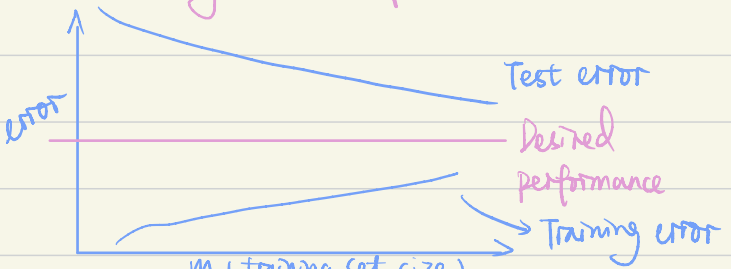
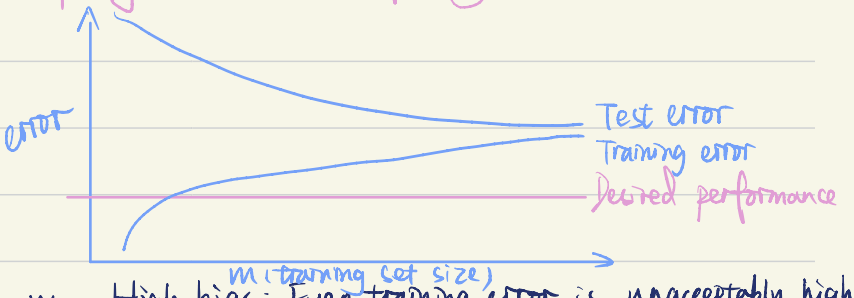
Lecture 13 Debugging ML Models and Error Analysis

1. Debugging learning algorithms
   - Example 1: Anti-spam; 100 words as features
     Use Logistic regression with regularization + gradient descent
     $\Rightarrow$ 20% test error; unacceptably high
     $\max_\theta \sum_{i=1}^{m} \log p(y^{(i)} | x^{(i)}, \theta) - \lambda \|\theta\|^2$

$\Rightarrow$ Diagnostic for bias vs. variance { high variance: overfitting; high bias: underfitting }



High Variance: Test error still decreases as m increases; Large gap between training & test error.



High bias: Even training error is unacceptably high. Small gap between training & test error.

Fixes to try:

Fix high variance	Fix high bias
· Try getting more training examples	· Try a larger set of features
· Try a smaller set of features	· Try email header features

- Example 2: Logistic regression gets 2% error on spam, and 2% error on non-spam (unacceptably error on non-spam)
  SVM gets 10% error on spam, and 0.01% error on non-spam (acceptable performance)
  But you want to use logistic regression because of computational limit

The question equals to: 1) Is gradient descent converging?
2) Are you optimizing the right function?

⇒ care about weighted accuracy: $a(\theta) = \max_{\theta} w^{(i)} I\{h_\theta(x^{(i)}) = y^{(i)}\}$

$$a(\theta_{svm}) > a(\theta_{lgr})$$

Diagnose:

Case 1: $a(\theta_{svm}) > a(\theta_{lgr})$ and $J(\theta_{svm}) > J(\theta_{lgr})$

$\Rightarrow \theta_{lgr}$ fails to maximize J $\Rightarrow$ not converging, more iteration

Case 2: $a(\theta_{svm}) > a(\theta_{lgr})$ and $J(\theta_{svm}) < J(\theta_{lgr})$

$\Rightarrow$ Lgr succeeded at maximizing $J(\theta)$ $\Rightarrow$ $J(\theta)$ is the wrong function to be maximizing

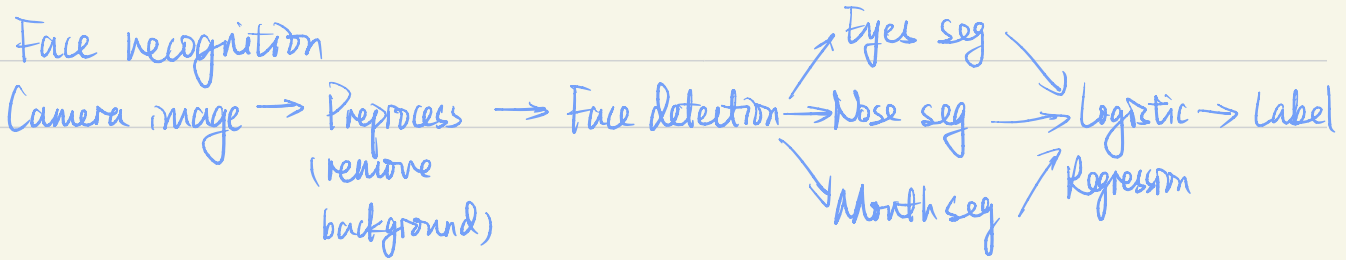
Fixes to try:

Fix optimization algorithm
- Run gradient descent more iterations
- Try Newton's Method

Fix optimization objective
- Use a different value of λ
- Try using SVM

2. Error Analysis

Example: Face recognition



Solution: How much error is attributable to each of the components?

Plug in ground-truth for each component, and see how accuracy changes

Component	Accuracy
Overall System	85%
Preprocess	85.1%
✡ Face detection	91%
Eyes seg	95%
Nose seg	96%
Mouth seg	97%
Logistic Regression	100%

3. Ablative Analysis

Remove components from your system one at a time, to see how it breaks.

Components	Accuracy	
Overall Systems	99.9%	
Spelling correction	99.8%	
Sender host features	98.9%	
Email header features	98.9%	
☆ Email text parser features	95%	
Javascript parser	94.5%	
Features from images	94.0%	baseline