Basic Concepts

Errors in your predictions can be troubleshooted by:

- Getting more training examples
- Trying smaller sets of features
- Trying additional features
- Trying polynomial features
- Increasing or decreasing λ

Evaluating a Hypothesis

- A hypothesis may have low error for the training examples but still be inaccurate (because of overfitting).
- With a given dataset of training examples, we can split up the data into two sets: a training set and a test set.
- The new procedure using these two sets is then:
  - Learn W and minimize error using the training set
  - Compute the test set error

Model Selection and Train/Validation/Test Sets

- Just because a learning algorithm fits a training set well, that does not mean it is a good hypothesis.
- The error of your hypothesis as measured on the data set with which you trained the parameters will be lower than any other data set.
- In order to choose the model of your hypothesis, you can test each degree of polynomial and look at the error result.

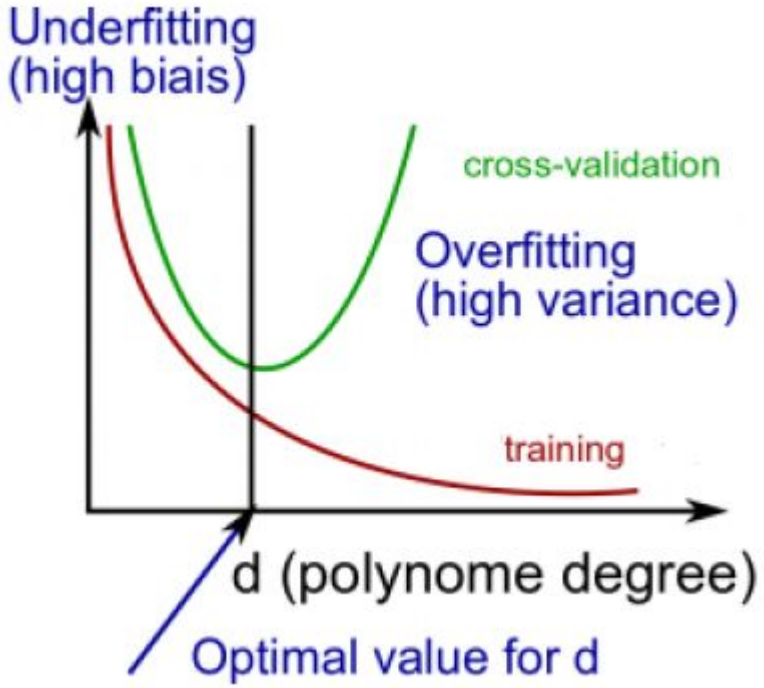
Use of the Cross Validation (CV) set

- To solve this, we can introduce a third set, the Cross Validation Set, to serve as an intermediate set that we can train with.
- Then our test set will give us an accurate, non-optimistic error.
- One example way to break down our dataset into the three sets is:
  - Training set: 60%
  - Cross validation set: 20%
  - Test set: 20%
- We can now calculate three separate error values for the three different sets.
- With the Validation Set
  - Optimize the parameters using the training set.
  - Find the set of hyper-parameters with the least error using the cross validation set.
  - Estimate the generalization error using the test set

Diagnosing Bias vs. Variance

- We examine the relationship between the degree of the polynomial d and the underfitting or overfitting of our hypothesis.
- We need to distinguish whether bias or variance is the problem contributing to bad predictions.
- High bias is underfitting and high variance is overFitting. We need to find a golden mean between these two.
- The training error will tend to decrease as we increase the degree d of the polynomial.
- At the same time, the cross validation error will tend to decrease as we increase d up to a point, and then it will increase as d is increased, forming a convex curve.
- High bias (underfitting): both training and validation error will be high and also Training error ≈ Validation error
- High variance (overfitting): training error will be low and validation error much grater than training error.

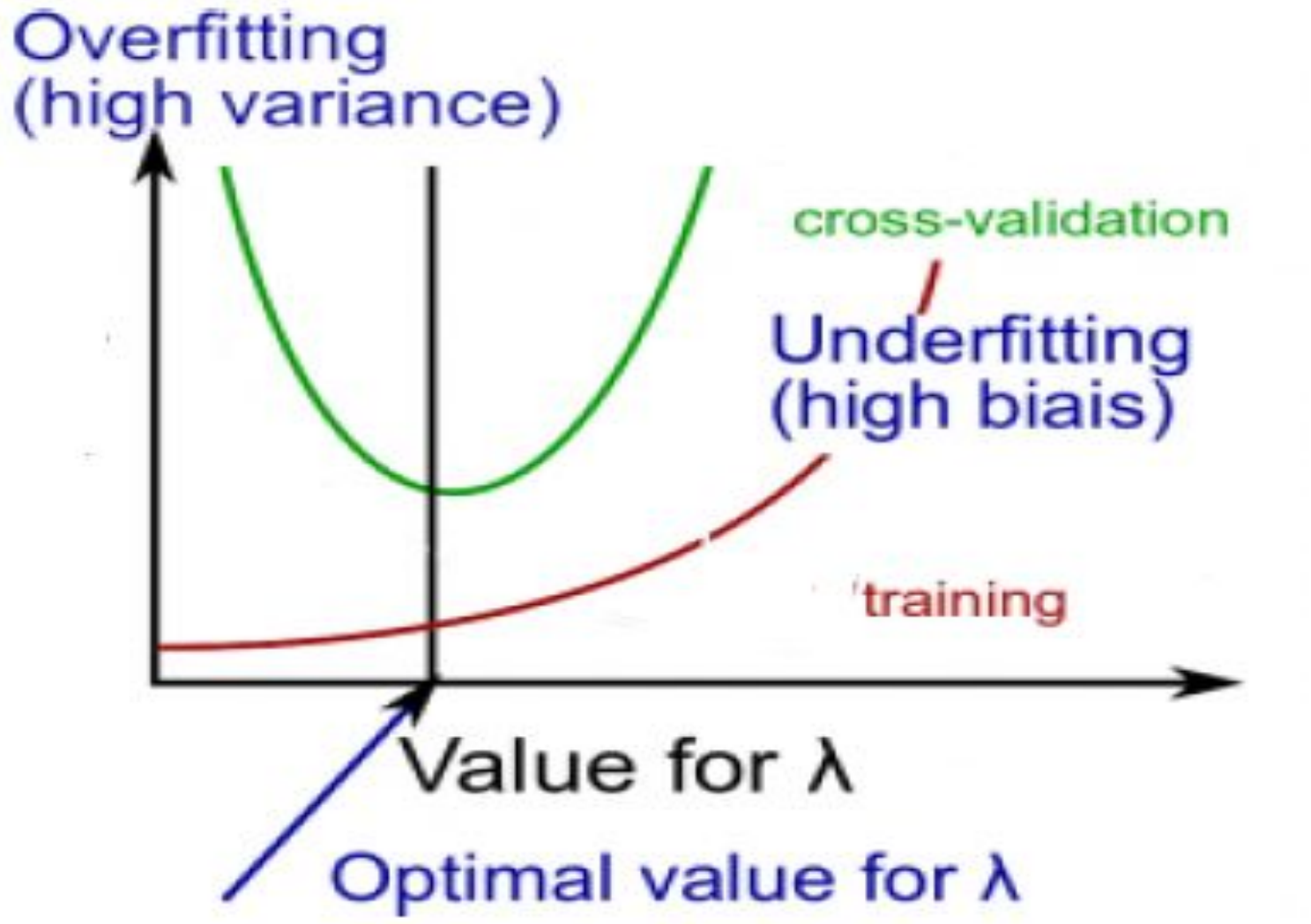
The is represented in the figure below:



Regularization and Bias/Variance

- Instead of looking at the degree d contributing to bias/variance, now we will look at the regularization parameter λ.
  - Large λ: High bias (underfitting)
  - Intermediate λ: just right
  - Small λ: High variance (overfitting)
- A large lambda heavily penalizes all the W parameters, which greatly simplifies the line of our resulting function, so causes underfitting.
- The relationship of λ to the training set and the variance set is as follows:
  - Low λ: Training error is low and CV error is high (high variance/overFitting).
  - Intermediate λ: Training error and CV error are somewhat low and Training error ≈ CV error.
- Large λ: both Training error and CV error will be high (underfitting /high bias)

The figure below illustrates the relationship between lambda and the hypothesis:



In order to choose the model and the regularization λ, we need:

1. Create a list of lambda (i.e. $\lambda \in \{0,0.01,0.02,0.04,0.08,0.16,0.32,0.64,1.28,2.56,5.12,10.24\}$);
2. Select a lambda to compute;
3. Create a model set like degree of the polynomial or others;
4. Select a model to learn W;
5. Learn the parameter W for the model selected.
6. Compute the train error using the learned W (computed with λ).
7. Compute the cross validation error using the learned W (computed with λ).
8. Do this for the entire model set and lambdas, then select the best combo that produces the lowest error on the cross validation set;
9. Now visualize to help you understand your decision, you can plot to the figure like above with: (λ x Training Error) and (λ x CV Error);
10. Now using the best combo W and λ, apply it on test data to see if it has a good generalization of the problem.
11. To help decide the best polynomial degree and λ to use, we can diagnose with the learning curves, that is the next subject.

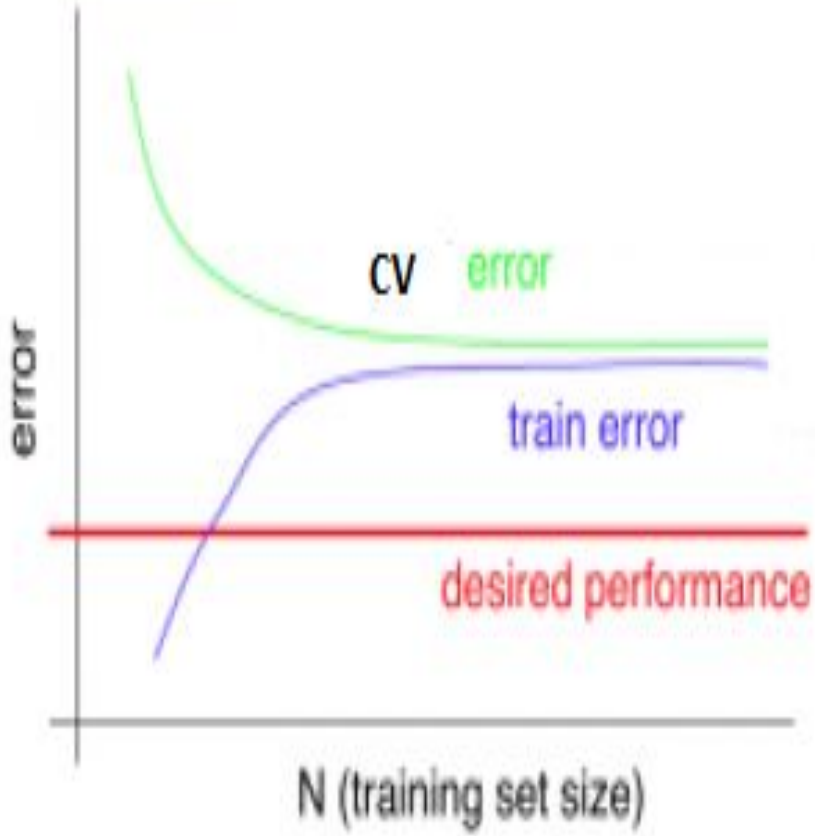
Learning Curves

- Training 3 examples will easily have 0 errors because we can always find a quadratic curve that exactly touches 3 points.
  - As the training set gets larger, the error for a quadratic function increases.
  - The error value will plateau out after a certain training set size (m).
- With high bias
  - Low training set size: causes Training Error to be low and CV error to be high.
  - Large training set size: causes both Training Error and CV error to be high with Training Error ≈ CV error .
- If a learning algorithm is suffering from high bias, getting more training data will not (by itself) help much.

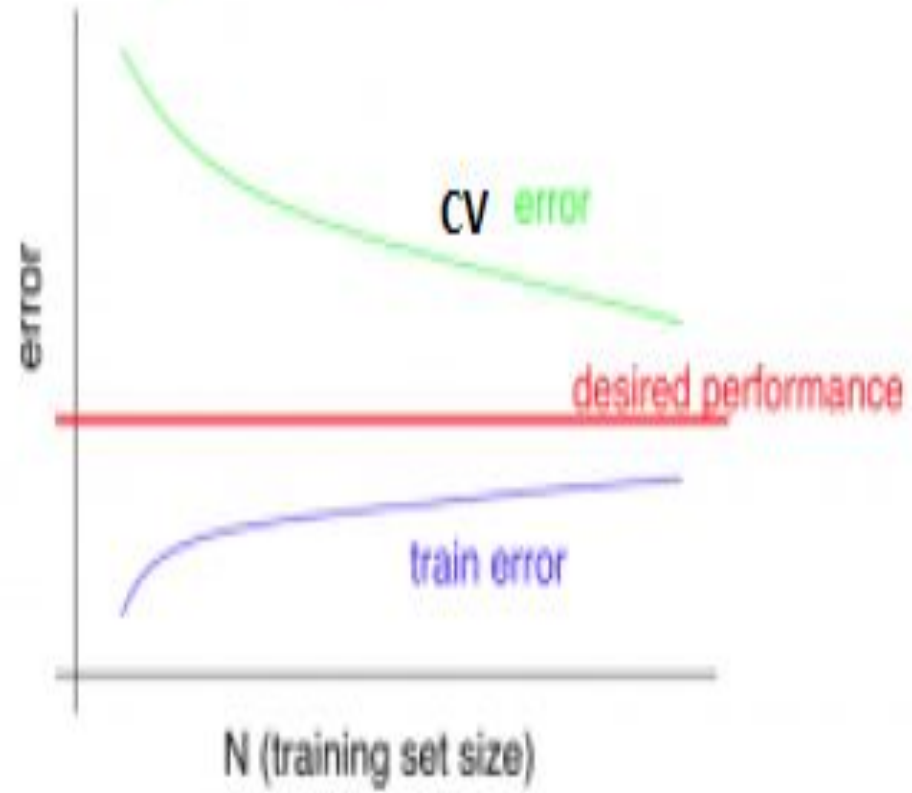
Learning Curves

- For high variance, we have the following relationships in terms of the training set size:
  - Low training set size: Training Error will be low and CV error will be high.
  - Large training set size: Training Error increases with training size and CV error continues to decrease without leveling of. Also Training error < CV error but difference between them remains significant.
- If a learning algorithm is suffering from high variance, getting more training data is likely to help.

Typical learning curvefor high bias(at fixed model complexity):



Typical learning curve for high variance(at fixed model complexity):



Deciding What to Do Next Revisited

- Our decision process can be broken down as follows:
  - Getting more training examples
    - Fixes high variance
  - Trying smaller sets of features
    - Fixes high variance
  - Adding features
    - Fixes high bias
  - Adding polynomial features
    - Fixes high bias
  - Decreasing λ
    - Fixes high bias
  - Increasing λ
    - Fixes high variance

Diagnosing Neural Networks

- A neural network with fewer parameters is prone to underfitting. It is also computationally cheaper.
- A large neural network with more parameters is prone to overfitting. It is also computationally expensive. In this case you can use regularization (increase λ) to address the overfitting.
- Using a single hidden layer is a good starting default. You can train your neural network on a number of hidden layers using your cross validation set.

Prioritizing What to Work On

- Different ways we can approach a machine learning problem:
  - Collect lots of data
  - Develop sophisticated features (for example: using email header data in spam emails)
  - Develop algorithms to process your input in different ways (recognizing misspellings in spam).
- It is difficult to tell which of the options will be helpful.

Error Analysis

- The recommended approach to solving machine learning problems is:
  - Start with a simple algorithm, implement it quickly, and test it early.
  - Plot learning curves to decide if more data, more features, etc. will help
  - Error analysis: manually examine the errors on examples in the cross validation set and try to spot a trend.
- It's important to get error results as a single, numerical value. Otherwise it is difficult to assess your algorithm's performance.
- You may need to process your input before it is useful. For example, if your input is a set of words, you may want to treat the same word with different forms (fail/failing/failed) as one word, so must use "stemming software" to recognize them all as one.

Error Metrics for Skewed Classes

- It is sometimes difficult to tell whether a reduction in error is actually an improvement of the algorithm.
- For example: In predicting a cancer diagnoses where 0.5% of the examples have cancer, we find our learning algorithm has a 1% error. However, if we were to simply classify every single example as a 0, then our error would reduce to 0.5% even though we did not improve the algorithm.
- This usually happens with skewed classes; that is, when our class is very rare in the entire data set.
- Or to say it another way, when we have lot more examples from one class than from the other class.
- For this we can use Precision/Recall.
  - Predicted: 1, Actual: 1 --- True positive
  - Predicted: 0, Actual: 0 --- True negative
  - Predicted: 0, Actual, 1 --- False negative
  - Predicted: 1, Actual: 0 --- False positive

Precision, Recall and F-measure

- Precision: of all patients we predicted where y=1, what fraction actually has cancer? TP/Total no. of predicted positive OR TP/(TP+FP)
- Recall: Of all the patients that actually have cancer, what fraction did we correctly detect as having cancer? TP/ Total no. of actual positive OR TP/(TP+FN)
- These two metrics give us a better sense of how our classifier is doing.
- We want both precision and recall to be high.
- In the example at the beginning of the section, if we classify all patients as 0, then our recall will be 0/(0+FP), so despite having a lower error percentage, we can quickly see it has worse recall.

Trading Of Precision and Recall

- We might want a confident prediction of two classes using logistic regression. One way is to increase our threshold:
  - Predict 1 if: y >= 0.7
  - Predict 0 if: y < 0.7
- This way, we only predict cancer if the patient has a 70% chance.
- Doing this, we will have higher precision but lower recall.
- In the opposite example, we can lower our threshold:
  - Predict 1 if: y >= 0.3
  - Predict 0 if: y < 0.3
- That way, we get a very safe prediction.
- This will cause higher recall but lower precision.
- The greater the threshold, the greater the precision and the lower the recall.
- The lower the threshold, the greater the recall and the lower the precision.
- In order to turn these two metrics into one single number, we can take the F value.

F-measure

- One way is to take the average:
  - P + R / 2
- This does not work well. If we predict all y=0 then that will bring the average up despite having 0 recall. If we predict all examples as y=1, then the very high recall will bring up the average despite having 0 precision.
- A better way is to compute the F Score (or F1 score):
  - 2PR/(P+R)
- In order for the F Score to be large, both precision and recall must be large.
- We want to train precision and recall on the cross validation set so as not to bias our test set.

References:

- http://www.cedar.bu□alo.edu/~srihari/CSE555/Chap9.Part2.pdf
- http://blog.stephenpurpura.com/post/13052575854/managing-bias-variance-tradeo□-in-machine-learning
- http://www.cedar.bu□alo.edu/~srihari/CSE574/Chap3/Bias-Variance.pdf