[Summary]Troubleshoot

Deciding What to Try Next

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

Don't just pick one of these avenues at random. We'll explore diagnostic techniques for choosing one of the above solutions in the following sections.

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:

1. Learn Θ and minimize $J_{train}(\Theta)$ using the training set
2. Compute the test set error $J_{test}(\Theta)$

The test set error

1. For linear regression: $J_{test}\left(\Theta\right) = \frac{1}{2m_{test}}\sum_{i=1}^{m_{test}}\left(\mathrm{h}_\theta\left(x_{test}^i\right) - y_{test}^i\right)^2$
2. For classification ~ Misclassification error (aka 0/1 misclassification error):

$$err(\mathrm{h}_\theta(x), y) = \begin{matrix} 1\ if\ \mathrm{h}_\theta(x) \geq 0.5\ and\ y=0\ or\ \mathrm{h}_\theta(x) < 0.5\ and\ y=1 \\ 0\ otherwise \end{matrix}$$

This gives us a binary 0 or 1 error result based on a misclassification.

The average test error for the test set is

$$Test\ Error = \frac{1}{m_{test}}\sum_{i=1}^{m_{test}} err(\mathrm{h}_\theta\left(x_{test}^i\right),\ y_{test}^i)$$

This gives us the proportion of the test data that was misclassified.

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 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 d 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 (note: this method presumes we do not also use the CV set for regularization)

Trained one variable, d, or the degree of the polynomial.

1. Optimize the parameters in Θ using the training set for each polynomial degree.
2. Find the polynomial degree d with the least error using the cross validation set.
3. Estimate the generalization error using the test set with $J_{test}(\Theta^{d})$, (d = theta from polynomial with lower error);

This way, the degree of the polynomial d has not been trained using the test set.

Diagnosing Bias vs. Variance

In this section 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.

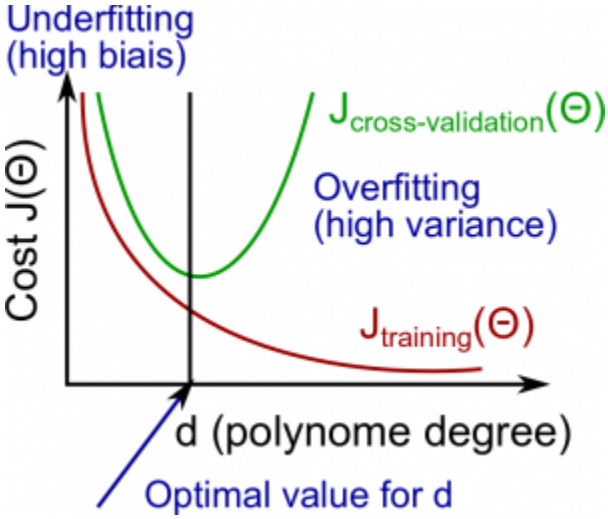
The training error will tend to **decrease** as we increase the degree d of the polynomial.

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 $J_{train}(\Theta)$ and $J_{CV}\left(\Theta\right)$ will be high. Also, $J_{CV}\left(\Theta\right) \approx J_{train}(\Theta)$.

High variance (overfitting): $J_{train}(\Theta)$ will be low and $J_{CV}\left(\Theta\right)$ will be much greater than $J_{train}(\Theta)$.

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 Θ parameters, which greatly simplifies the line of our resulting function, so causes underfitting.

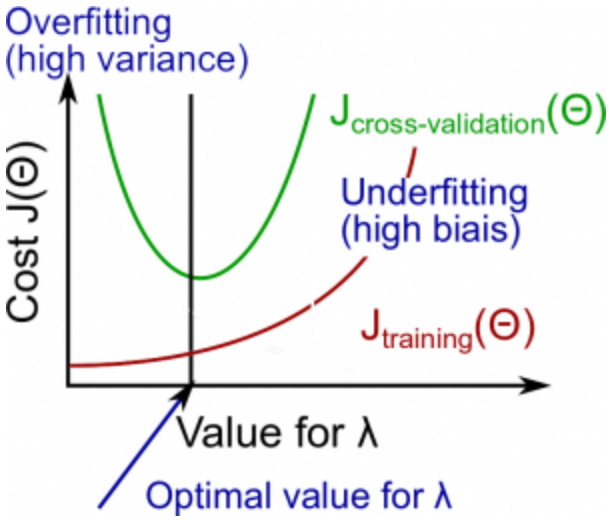
The relationship of λ to the training set and the variance set is as follows:

Low λ: $J_{train}(\Theta)$ is low and $J_{CV}\left(\Theta\right)$ is high (high variance/overfitting).

Intermediate λ: $J_{train}(\Theta)$ and $J_{CV}\left(\Theta\right)$ are somewhat low and $J_{train}(\Theta) \approx J_{CV}\left(\Theta\right)$.

Large λ: both $J_{train}(\Theta)$ and $J_{CV}\left(\Theta\right)$ 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 lambdas (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. Create a set of models with different degrees or any other variants.
3. Iterate through the λs and for each λ go through all the models to learn some Θ.
4. Compute the cross validation error using the learned Θ (computed with λ) on the $J_{CV}(\Theta)$ without regularization or λ = 0.
5. Select the best combo that produces the lowest error on the cross validation set.
6. Using the best combo Θ and λ, apply it on $J_{test}(\Theta)$ to see if it has a good generalization of the problem.

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 m, or training set size.

With high bias

Low training set size: causes $J_{train}(\Theta)$ to be low and $J_{CV}(\Theta)$ to be high.

Large training set size: causes both $J_{train}(\Theta)$ and $J_{CV}(\Theta)$ to be high with $J_{train}(\Theta) \approx J_{CV}(\Theta)$

If a learning algorithm is suffering from **high bias**, getting more training data **will not (by itself) help much**.

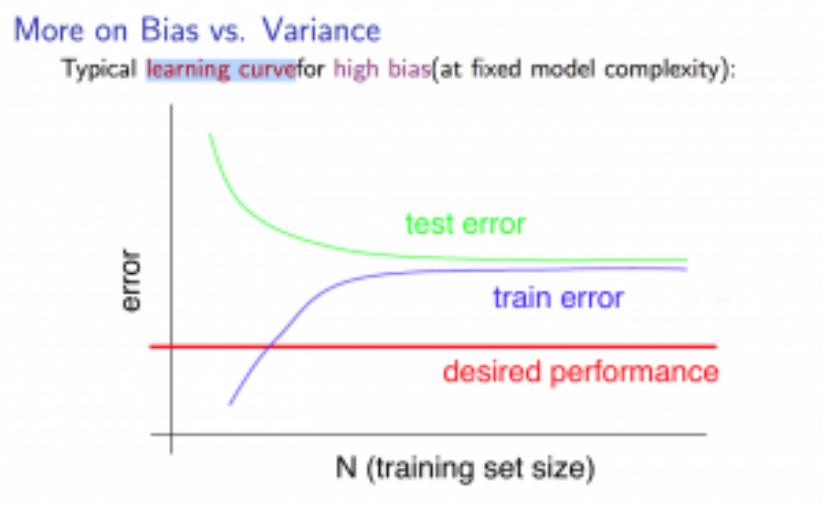
For high variance, we have the following relationships in terms of the training set size:

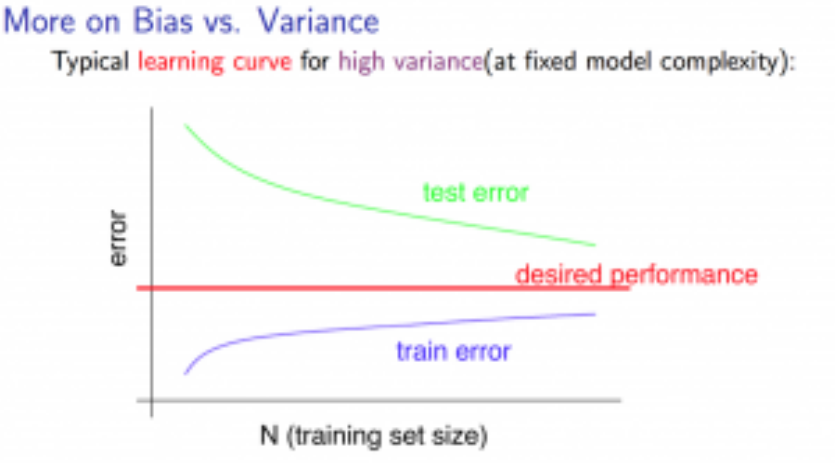
With high variance

Low training set size: $J_{train}(\Theta)$ will be low and $J_{CV}(\Theta)$ will be high.

Large training set size: $J_{train}(\Theta)$ increases with training set size and $J_{CV}(\Theta)$ continues to decrease without leveling off. Also, $J_{train}(\Theta) < J_{CV}(\Theta)$ but the difference between them remains significant.

If a learning algorithm is suffering from **high variance**, getting more training data is **likely to help.**





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.