Advise for Applying Machine Learning

1 Deciding What to Try Next.

Introduction: In this section how to not waste time choosing / debugging algorithms will be explained.

Example: Say that we have a regularized linear regression implementation, and it makes unacceptably large errors. What should we do?

- More training examples? Might not be the best idea - can waste time.
- Try smaller sets of features. Might prevent overfitting.
- Try getting additional features. We can waste time - need to know if it will work.
- Try additional features.
- Decreasing, increasing λ.
- etc....

We can waste a lot of time doing this if we are not sure why our algorithm fails, so we're gonna try to see what shall be the best option, or at least rule out some of them so we don't lose time doing something that is not going to increase the prediction accuracy.

The tests done to solve this problems will be called **Machine learning diagnostic tests**.

2 Evaluating a Hypothesis

How do you tell if a hypothesis is overfitting?
Just split the data in two portions: The first portion is going to be the training set and the other one the test set (70%-30% split is a good estimation).

We train with the training data and evaluate our model with the test data. We can evaluate the model using different metrics, such as mse (eq 1) for regression or the 1/0 error for classification (eq 3).

$$J_{test}(\theta) = \frac{1}{2m_{test}} \sum_{i=1}^{m_{test}} (h_\theta(x_{test}^{(i)}) - y_{test}^{(i)})^2 \tag{1}$$

$$error(h_\theta(x), y) = \begin{cases} 1, & \text{if } h_\theta(x) \geq 0.5, y = 0 \\ & \text{or } h_\theta(x) < 0.5, y = 1 \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

$$\text{test error} = \frac{1}{m_{test}} \sum_{i=1}^{m_{test}} error(h_\theta(x_{test}^{(i)}), y^{(i)}) \tag{3}$$

3 Model Selection and Train/Validation/Test Sets

Say that we want to fit a problem. We want to decide the degree (d) of the polynomial to use (amount of features extracted from a single feature using the feature ^2, ^3, etc).

Each single polynomial will fit to different θ (d = 1 to $\theta^{(1)}$, d = 2 to $\theta^{(2)}$, etc).

To find the best d, we could minimize the test error. If we do that, though, what happens is that we would overfit the d parameter for the test set, and we don't want to do that.

3.1 Welcome, Cross Validation set.

In order to fix that problem, we simply split the data in 3: Training, CrossValidation and Test (with a typical ratio of 60% - 20% - 20%). The CV error is calculated with the same formulas, only with the CV data, obviously.

Once we have that split, we train every different model (each with a different d) and choose the one that has the lowest CV error. This way, we can still use the test error in order to know whether our model is overfitting on "real life" data.

4 Diagnosting Bias vs. Variance

We continue with the example of the previous section.

When we increase the degree of the polynomial, we get less error on the training set, which means we're overfitting. With the CV/Test sets, though, the error increases at a certain point, where the model fails to generalize the data (overfitting).

When we have a high CV error, we can have either high variance or high bias. High bias - Underfitting is when both training error and CV error are high. On the other hand, High variance - Overfitting is when training error is very low and the CV error is high.

We can see all these in figure 1.

5 Regularization and Bias/Variance

Suppose we're fitting a high order polynomial with regularization to evade the overfitting trap.

In figure 2 we can see that, given the case of a high order polynomial, when we have a large λ we are underfitting the data and when we have a small λ we're overfitting it.

An important detail here is noticing that, even though we use regularization when training, we don't use it for the evaluation cost functions, where we use the simple mse function (equation 1).

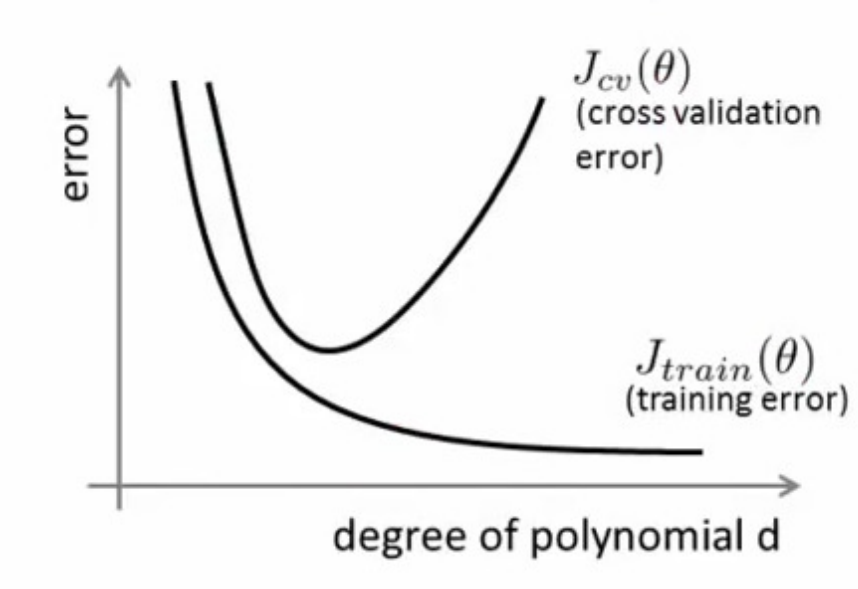

Figure 1: Error on degree of the polynomial.

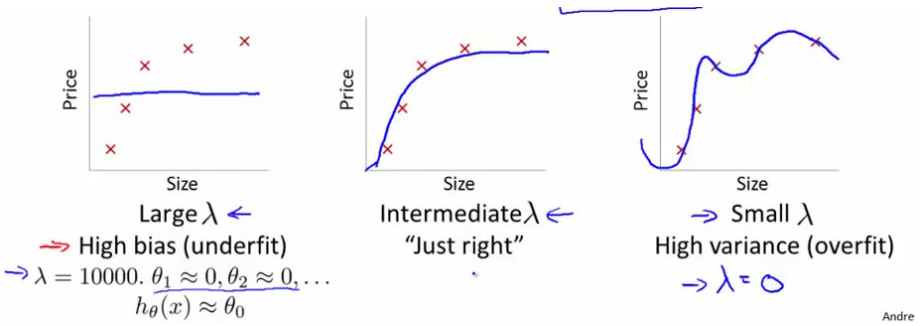

Figure 2: Different λ values comparisson on a high degree polynomial.

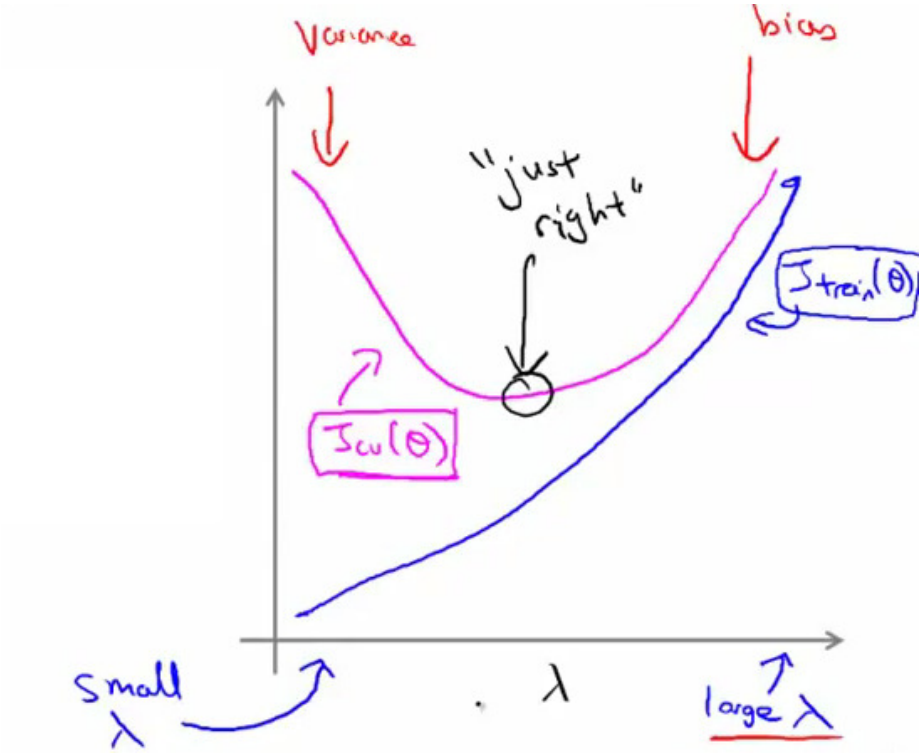

Figure 3: CV and Train error functions related to λ values.

In order to find the correct λ value, we train a model with every different value and pick up the one that gives the min CV error function. We can then evaluate the chosen θ with the test error function to see how well it's working (figure 3).

6 Learning Curves

Tool to diagnose a algorithm and find out if it suffers from bias or variance.

Artificially reduce training set size and train with that. On very small datasets the training error will be very small because it will fit perfectly. As the size of the dataset increases, though, the training error increases.

With the CV error, it starts with huge errors when training with a small dataset and then the error decreases as the training set size increases.

We can see the "perfect" graph we should expect if our model isn't overfitting or underfitting the data in figure 4.

If we have high bias (underfitting), the cross validation error will end up close to the train error (figure 5). This means that if we have a high bias, getting more data doesn't help much.

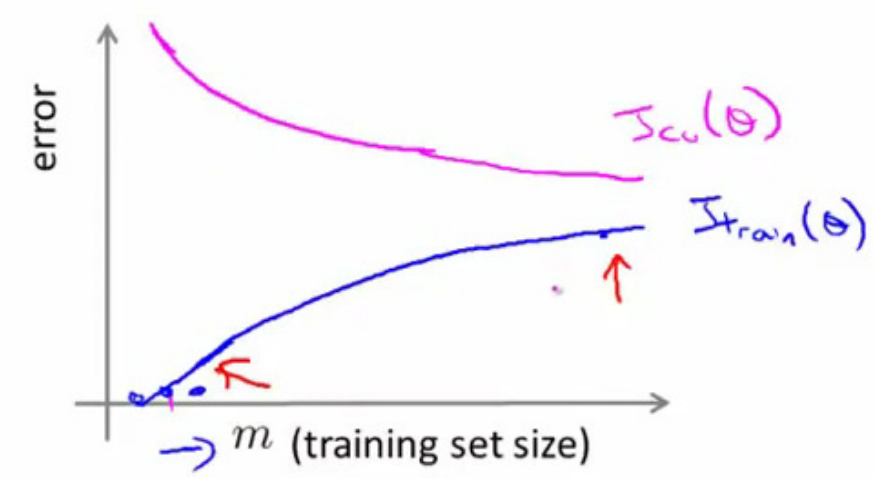

Figure 4: Error on training set size without both variance and bias.

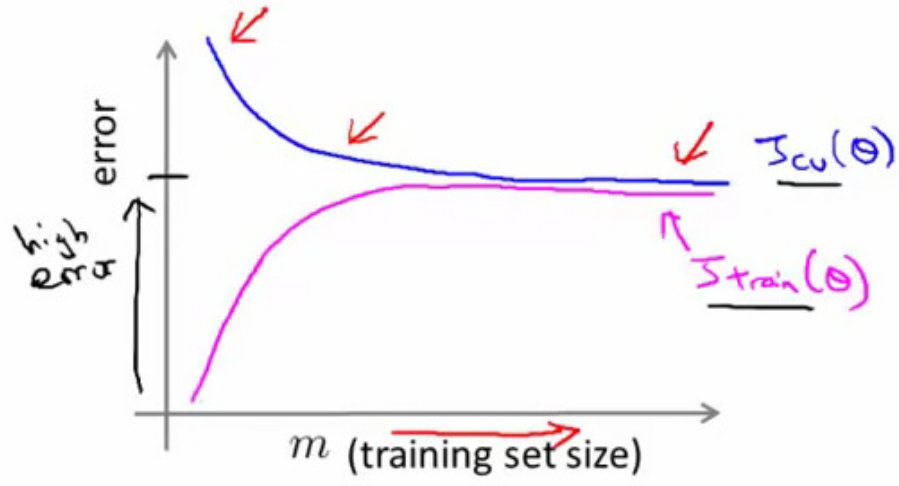

Figure 5: Error on training set size with bias (underfitting).

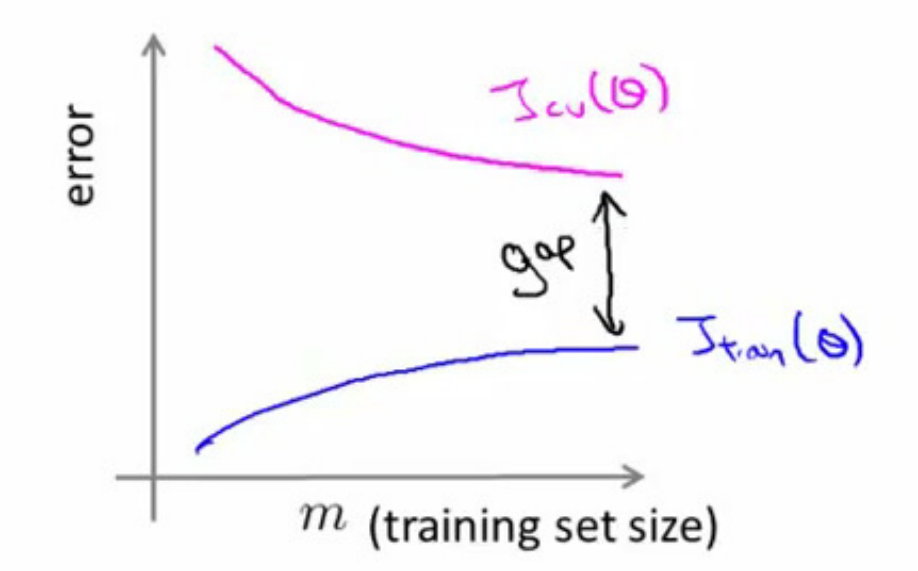

Figure 6: Error on training set size with variance (overfitting).

If we have high variance (overfitting), the model will have a very low training error and very high CV error (figure 6). In this case, getting more data is likely to help.

7 Deciding What to Do Next Revisited

- Getting more training examples → Fixes high variance.
- Smaller sets of features → Fixes high variance.
- Additional features → Fixes high bias.
- Adding polynomial features → Fixes high bias.
- Decreasing λ → Fixes high bias.
- Increasing λ → Fixes high variance.

7.1 Neural Networks

A small neural network is prone to underfitting and a large one is prone to overfitting.