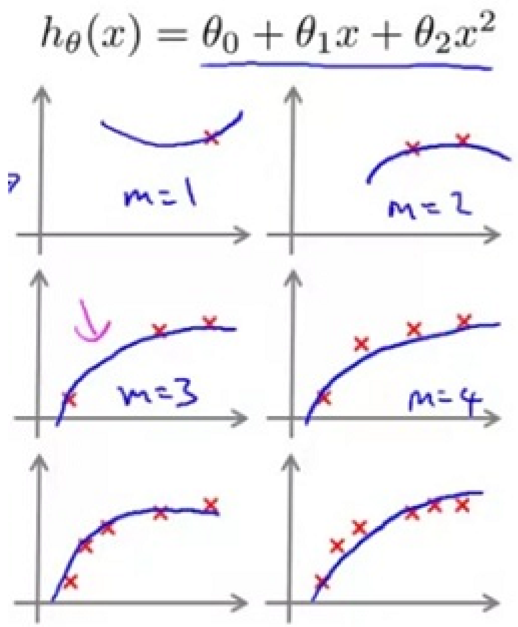
Learning Curves

Learning curves is often a very useful to plot, if either we wanted to sanity check that our algorithm is working correctly, or if we want to improve the performance of the algorithm.

To plot a learning curve, what we can do is plot J_{train} (which is the average squared error on our training set) or J_{cv} (which is the average squared error on our cross validations set) and, we're going to plot that as a function of m (the number of training examples), so m is usually a constant (like a 100 training examples) and what we're going to do is, so artificially reduce m and recalculate errors with the smaller training set sizes.

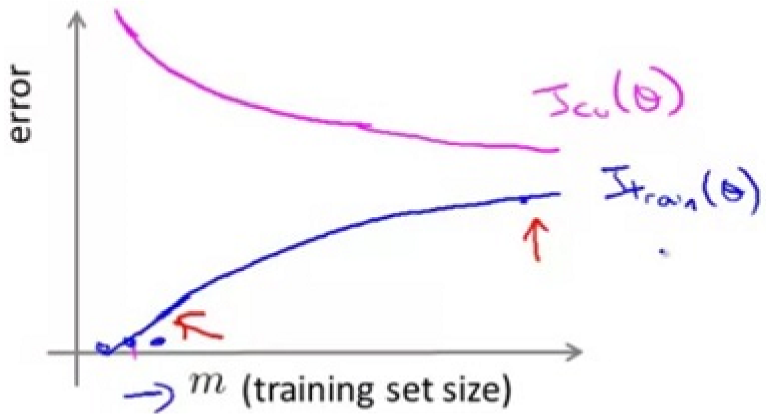


- J_{train}
  - Error on smaller sample sizes is smaller (when we have very few training examples, it's pretty easy to fit every single one of our training examples perfectly and so our error is going to be small).
  - So as m grows error grows (when m is larger then it's more harder to train all our training examples perfectly and so our training set error becomes more larger)

How about the cross validation error?

- J_{cv}
  - When we have a very small training set, we're not going to generalize well.
  - But as training set grows our hypothesis generalize better to new examples (when we get a larger training set, we're starting to get hypotheses that maybe fit the data somewhat better).
  - So cv error will decrease as m increases.

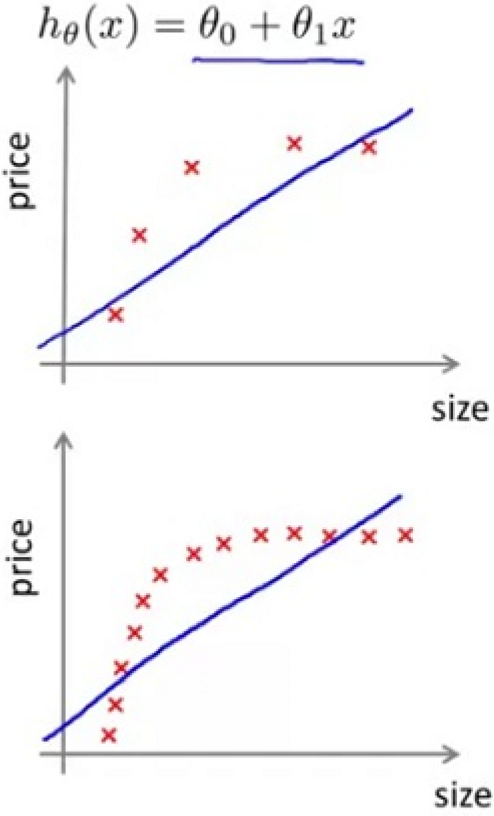
So our cross validation error will tend to decrease as our training set size increases because the more data we have, the better we do at generalizing to new examples.



~ High bias ~

What do these curves look like if you have **high bias**? Suppose our hypothesis has high bias, to explain this we're going to use a, set an example, of fitting a straight line to data that can't really be fit well by a straight line.

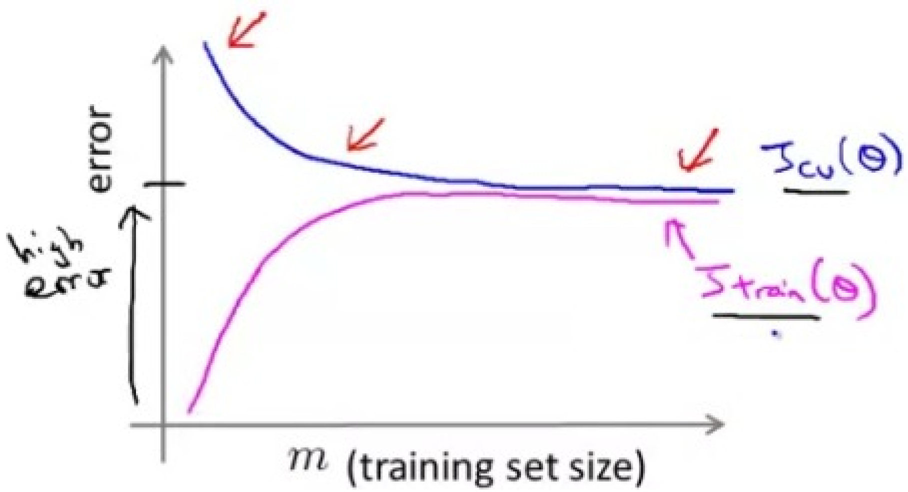
And what happens if we have a lot more training examples, what we find is that, we end up with a pretty much the same straight line and the stright line just cannot fit the data and getting a ton more data, the straight line is not going to change that much (no increase in data will help it fit).



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 $J_{train}(\Theta) \approx J_{CV}(\Theta)$.

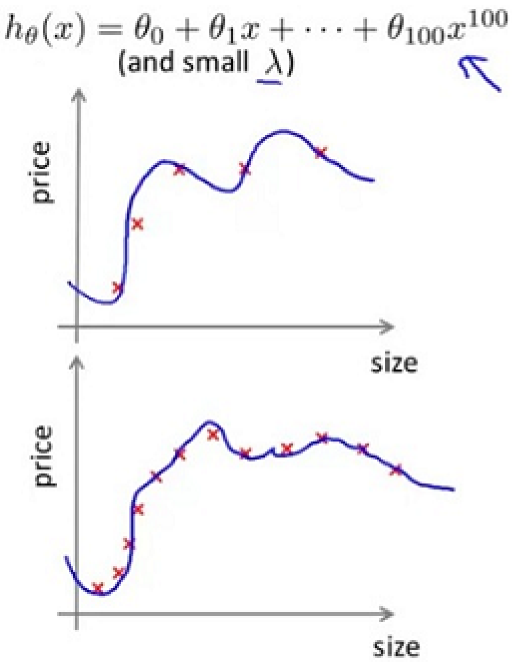
- So the performance of the cross validation and training set end up being similar (but very poor).
- The problem with high bias is because cross validation and training error are both high.
- Also implies that if a learning algorithm as high bias as we get more examples the cross validation error doesn't decrease.



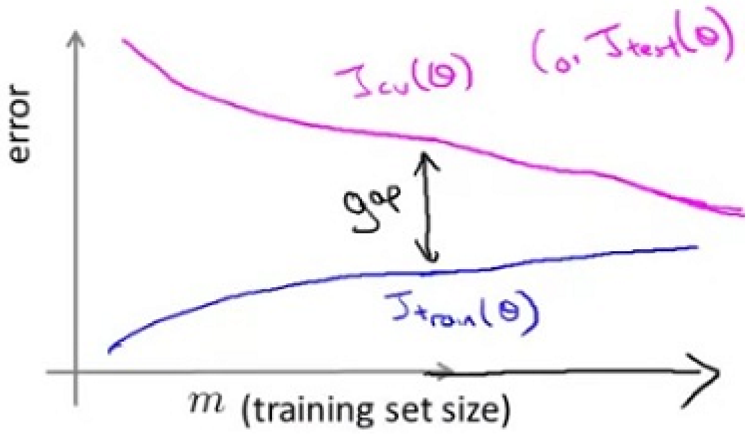
Concretely, if a learning algorithm is suffering from high bias, getting more training data will not (by itself) help much. So knowing if our learning algorithm is suffering from high bias seems like a useful thing to know because this can prevent us from wasting a lot of time collecting more training data where it might just not end up being helpful.

~ High variance ~

Next let us look at the setting of a learning algorithm that may have high variance. In this example suppose we have a hypotheses with a high order polynomial, if we have a very small training set like five training example and we're fitting with a very high order polynomial and we're using a fairly small value of λ, then we'll end up, fitting the data very well that with a function that overfits, and as the training set size increases a bit, we may still be overfitting this data a little bit but it also becomes slightly harder to fit this data set perfectly.



- J_{train}
  - When the training set is small, training error is small too
  - As training set sizes increases, the J_{train} increases (when the set size increases becomes slightly harder to fit the data set perfectly, and so, we'll find that J_{train} slowly increases).
- J_{cv}
  - Error remains high, even when you have a moderate number of examples
  - Because the problem with high variance (overfitting) is our model doesn't generalize
- An indicative diagnostic that we have high variance is that there's a big gap between training error and cross validation error



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.

Video Question: In which of the following circumstances is getting more training data likely to significantly help a learning algorithm's performance?

- Algorithm is suffering from high bias.

> Algorithm is suffering from high variance.

> $J_{\text{CV}}(\theta)$ (cross validation error) is much larger than $J_{\text{train}}(\theta)$ (training error).

- $J_{\text{CV}}(\theta)$ (cross validation error) is about the same as $J_{\text{train}}(\theta)$ (training error).

Summary

Training an algorithm on a very few number of data points (such as 1, 2 or 3) will easily have 0 errors because we can always find a quadratic curve that touches exactly those number of points. Hence:

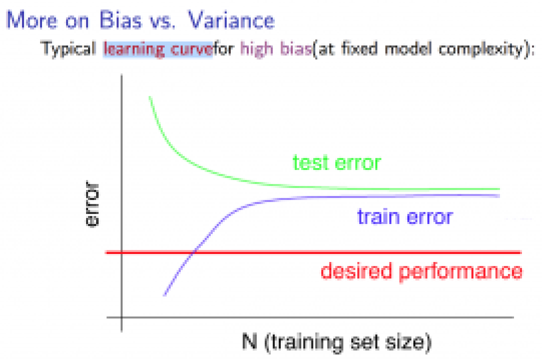
- As the training set gets larger, the error for a quadratic function increases.
- The error value will plateau out after a certain m, or training set size.

Experiencing 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 $J_{train}(\Theta) \approx J_{CV}(\Theta)$.

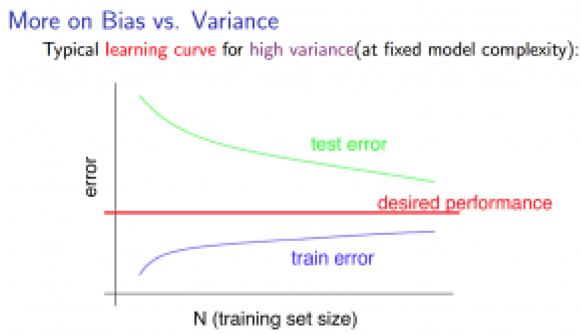
If a learning algorithm is suffering from **high bias**, getting more training data will not (**by itself**) help much.



Experiencing 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.