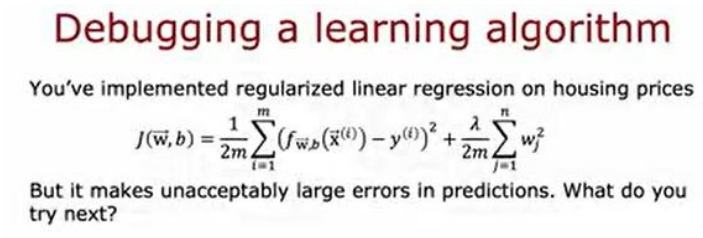
# Decide on the next trial

Hi, and welcome back. By now you've seen a lot of different learning algorithms, including linear regression, logistic regression, even deep learning, or neural networks, and next week, you'll see decision trees as well. You now have a lot of powerful tools of machine learning, but how do you use these tools effectively?

I've seen teams sometimes, say six months to build a machine learning system, that I think a more skilled team could have taken or done in just a couple of weeks. The efficiency of how quickly you can get a machine learning system to work well, will depend to a large part on how well you can repeatedly make good decisions about what to do next in the course of a machine learning project.

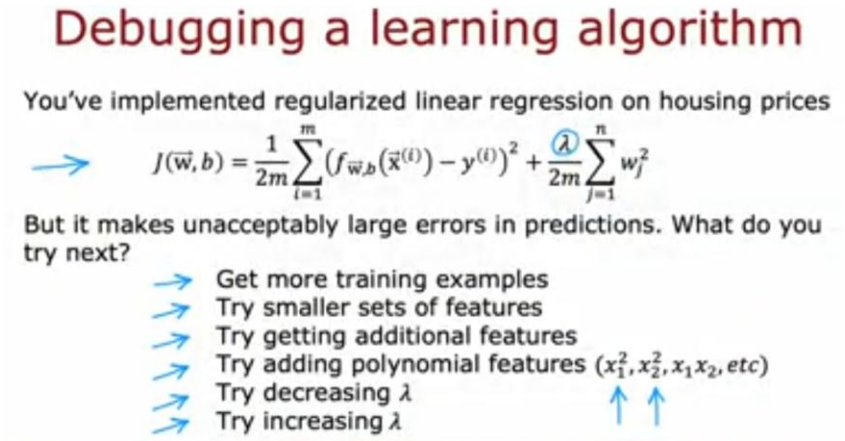
In this week, I hope to share with you a number of tips on how to make decisions about what to do next in machine learning project, that I hope will end up saving you a lot of time. Let's take a look at some advice on how to build machine learning systems.




Let's start with an example, say you've implemented regularized linear regression to predict housing prices, so you have the usual cost function for your learning algorithm, squared error plus this regularization term. But if you train the model, and find that it makes unacceptably large errors in it's predictions, what do you try next?

When you're building a machine learning algorithm, there are usually a lot of different things you could try. For example, you could decide to get more training examples since it seems having more data should help, or maybe you think maybe you have too many features, so you could try a smaller set of features.

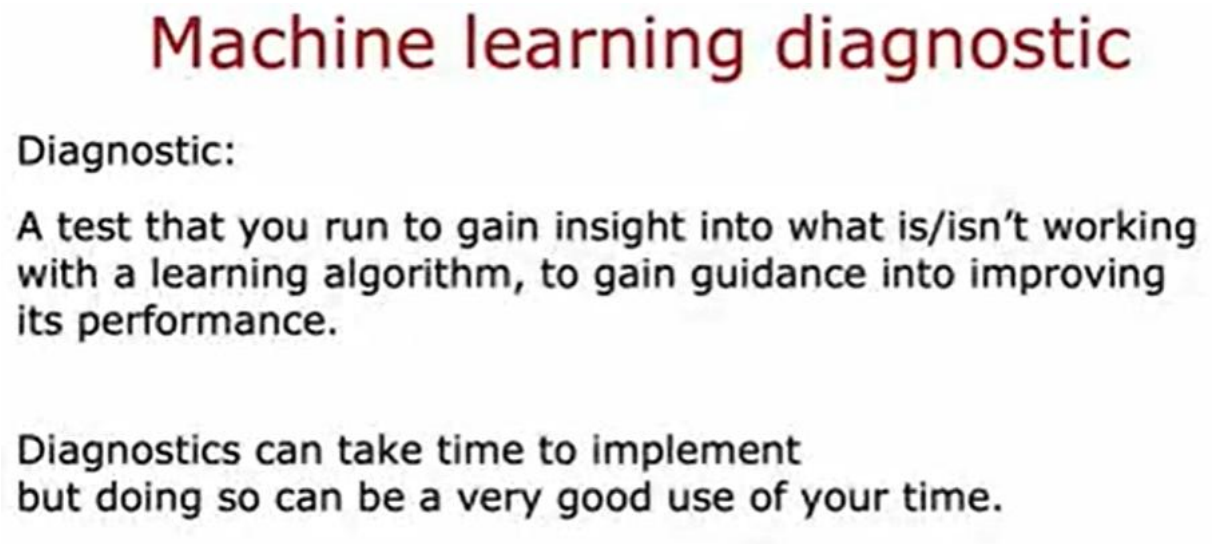
Or maybe you want to get additional features, such as finally additional properties of the houses to toss into your data, and maybe that'll help you to do better. Or you might take the existing features x_1, x_2, and so on, and try adding polynomial features x_1 squared, x_2 squared, x_1, x_2, and so on. Or you might wonder if the value of Lambda is chosen well, and you might say, maybe it's too big, I want to decrease it.



Or you may say, maybe it's too small, I want to try increasing. On any given machine learning application, it will often turn out that some of these things could be fruitful, and some of these things not fruitful. The key to being effective at how you build a machine learning algorithm will be if you can find a way to make good choices about where to invest your time.

For example, I have seen teams spend literally many many months collecting more training examples, thinking that more training data is going to help, but it turns out sometimes it helps a lot, and sometimes it doesn't. In this week, you'll learn about how to carry out a set of diagnostics.

By diagnosis, I mean a test that you can run to gain insight into what is or isn't working with learning algorithm to gain guidance into improving its performance. Some of these diagnostics will tell you things like, is it worth weeks, or even months collecting more training data, because if it is, then you can then go ahead and make the investment to get more data, which will hopefully lead to improved performance, or if it isn't then running that diagnosis could have saved you months of time.




One thing you see this week as well, is that diagnostics can take time to implement, but running them can be a very good use of your time. This week we'll spend a lot of time talking about different diagnostics you can use, to give you guidance on how to improve your learning algorithm's performance. But first, let's take a look at how to evaluate the performance of your learning algorithm.