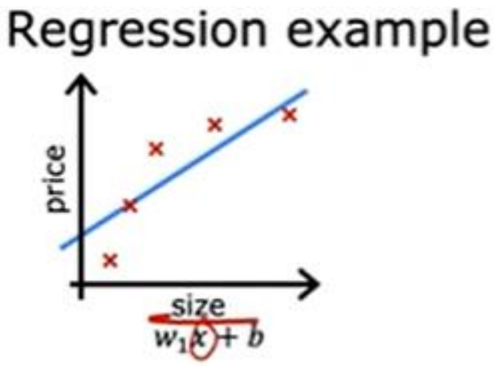
Le problème de l'overfitting

Now you've seen a couple of different learning algorithms, linear regression and logistic regression. They work well for many tasks. But sometimes in an application, the algorithm can run into a problem called overfitting, which can cause it to perform poorly. What I like to do in this video is to show you what is overfitting, as well as a closely-related, almost opposite problem called underfitting.

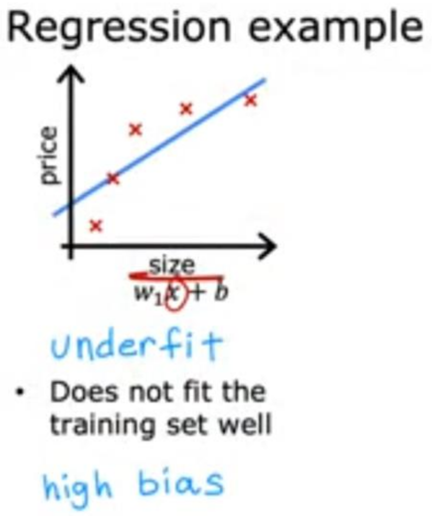
In the next videos after this, I'll share with you some techniques for accuracy overfitting. In particular, there's a method called regularization. Very useful technique. I use it all the time. Then regularization will help you minimize this overfitting problem and get your learning algorithms to work much better. Let's take a look at what is overfitting? To help us understand what is overfitting. Let's take a look at a few examples.



Let's go back to our original example of predicting housing prices with linear regression. Where you want to predict the price as a function of the size of a house. To help us understand what is overfitting, let's take a look at a linear regression example. I'm going to go back to our original running example of predicting housing prices with linear regression.

Suppose your data-set looks like this, with the input feature x being the size of the house, and the value, y that you're trying to predict the price of the house. One thing you could do is fit a linear function to this data. If you do that, you get a straight line fit to the data that maybe looks like this. But this isn't a very good model. Looking at the data, it seems pretty clear that as the size of the house increases, the housing process flattened out.

This algorithm does not fit the training data very well. The technical term for this is the model is underfitting the training data. Another term is the algorithm has high bias. You may have read in the news about some learning algorithms really, unfortunately, demonstrating bias against certain ethnicities or certain genders.

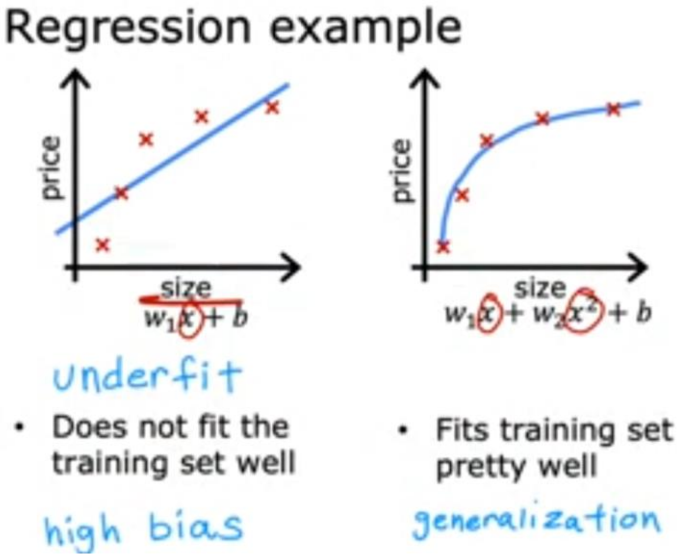


Regression example

- Does not fit the training set well

In machine learning, the term bias has multiple meanings. Checking learning algorithms for bias based on characteristics such as gender or ethnicity is absolutely critical. But the term bias has a second technical meaning as well, which is the one I'm using here, which is if the algorithm has underfit the data, meaning that it's just not even able to fit the training set that well. There's a clear pattern in the training data that the algorithm is just unable to capture.

There's a clear pattern in the training data that the algorithm is just unable to capture. Another way to think of this form of bias is as if the learning algorithm has a very strong preconception, or we say a very strong bias, that the housing prices are going to be a completely linear function of the size despite data to the contrary. This preconception that the data is linear causes it to fit a straight line that fits the data poorly, leading it to underfitted data.

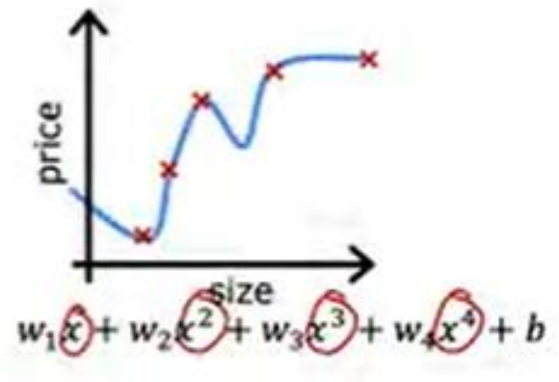


Regression example

- Does not fit the training set well
- Fits training set pretty well

Now, let's look at a second variation of a model, which is if you insert for a quadratic function at the data with two features, x and x^2, then when you fit the parameters W1 and W2, you can get a curve that fits the data somewhat better. Maybe it looks like this. Also, if you were to get a new house, that's not in this set of five training examples.

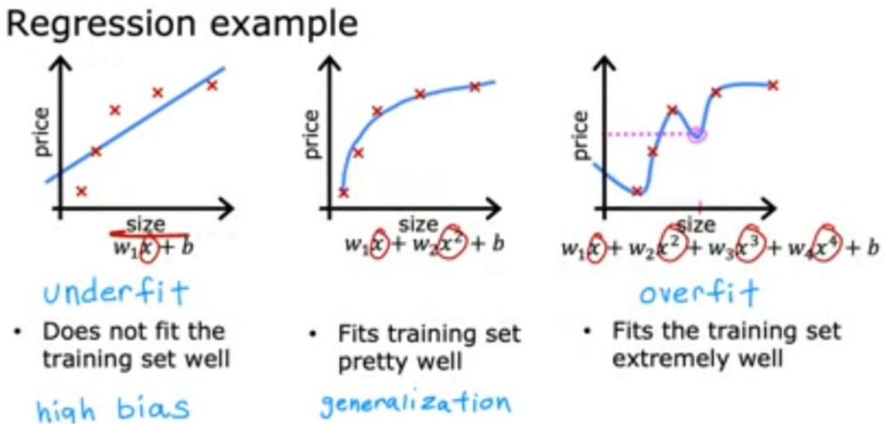
This model would probably do quite well on that new house. If you're real estate agents, the idea that you want your learning algorithm to do well, even on examples that are not on the training set, that's called generalization. Technically we say that you want your learning algorithm to generalize well, which means to make good predictions even on brand new examples that it has never seen before. These quadratic models seem to fit the training set not perfectly, but pretty well. I think it would generalize well to new examples. Now let's look at the other extreme.



What if you were to fit a fourth-order polynomial to the data? You have x, x^2, x^3, and x^4 all as features. With this fourth for the polynomial, you can actually fit the curve that passes through all five of the training examples exactly. You might get a curve that looks like this. This, on one hand, seems to do an extremely good job fitting the training data because it passes through all of the training data perfectly.

In fact, you'd be able to choose parameters that will result in the cost function being exactly equal to zero because the errors are zero on all five training examples. But this is a very wiggly curve, its going up and down all over the place. If you have this whole size right here, the model would predict that this house is cheaper than houses that are smaller than it.

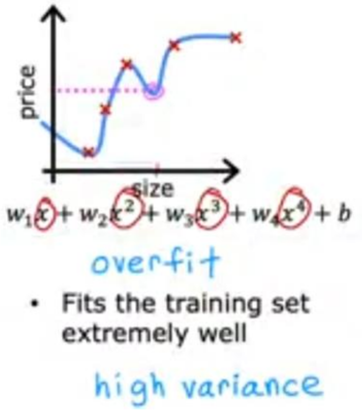
We don't think that this is a particularly good model for predicting housing prices. The technical term is that we'll say this model has overfit the data, or this model has an overfitting problem.



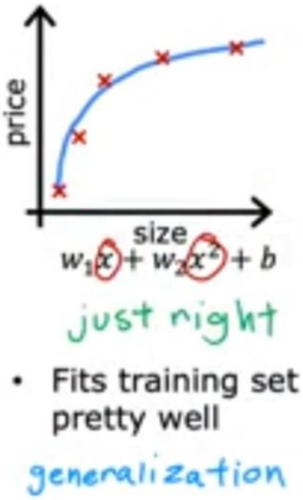
Because even though it fits the training set very well, it has fit the data almost too well, hence is overfit. It does not look like this model will generalize to new examples that's never seen before. Another term for this is that the algorithm has high variance. In machine learning, many people will use the terms over-fit and high-variance almost interchangeably.

We'll use the terms underfit and high bias almost interchangeably. The intuition behind overfitting or high-variance is that the algorithm is trying very hard to fit every single training example. It turns out that if your training set were just even a little bit different, say one holes was priced just a little bit more little bit less, then the function that the algorithm fits could end up being totally different.

If two different machine learning engineers were to fit this fourth-order polynomial model, to just slightly different datasets, they couldn't end up with totally different predictions or highly variable predictions. That's why we say the algorithm has high variance.

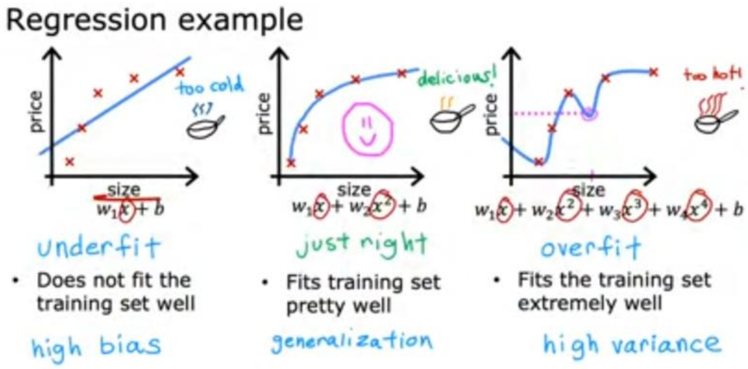


Contrasting this rightmost model with the one in the middle for the same house, it seems, the middle model gives them much more reasonable prediction for price. There isn't really a name for this case in the middle, but I'm just going to call this just right, because it is neither underfit nor overfit.



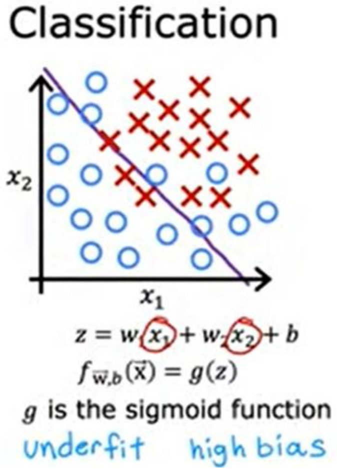
You can say that the goal machine learning is to find a model that hopefully is neither underfitting nor overfitting. In other words, hopefully, a model that has neither high bias nor high variance. When I think about underfitting and overfitting, high bias and high variance. I'm sometimes reminded of the children's story of Goldilocks and the Three Bears in this children's tale, a girl called Goldilocks visits the home of a bear family. There's a bowl of porridge that's too cold to taste and so that's no good. There's also a bowl of porridge that's too hot to eat. That's no good either. But there's a bowl of porridge that is neither too cold nor too hot. The temperature is in the middle, which is just right to eat.

To recap, if you have too many features like the fourth-order polynomial on the right, then the model may fit the training set well, but almost too well or overfit and have high variance. On the flip side if you have too few features, then in this example, like the one on the left, it underfits and has high bias. In this example, using quadratic features x and x squared, that seems to be just right.

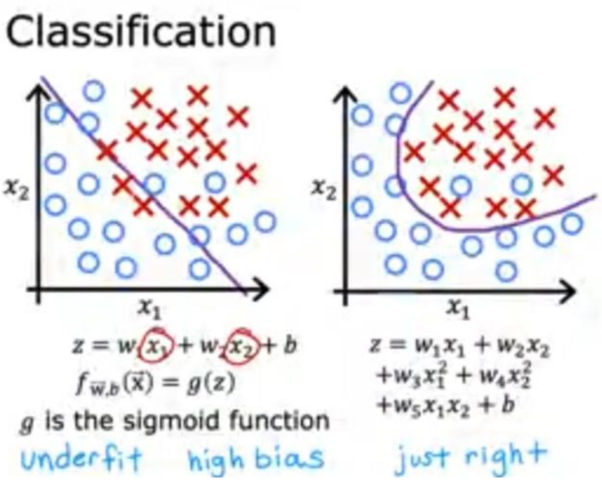


So far we've looked at underfitting and overfitting for linear regression model. Similarly, overfitting applies a classification as well. Here's a classification example with two features, x_1 and x_2, where x_1 is maybe the tumor size and x_2 is the age of patient.

We're trying to classify if a tumor is malignant or benign, as denoted by these crosses and circles, one thing you could do is fit a logistic regression model. Just a simple model like this, where as usual, g is the sigmoid function and this term here inside is z. If you do that, you end up with a straight line as the decision boundary. This is the line where z is equal to zero that separates the positive and negative examples. This straight line doesn't look terrible. It looks okay, but it doesn't look like a very good fit to the data either. This is an example of underfitting or of high bias.

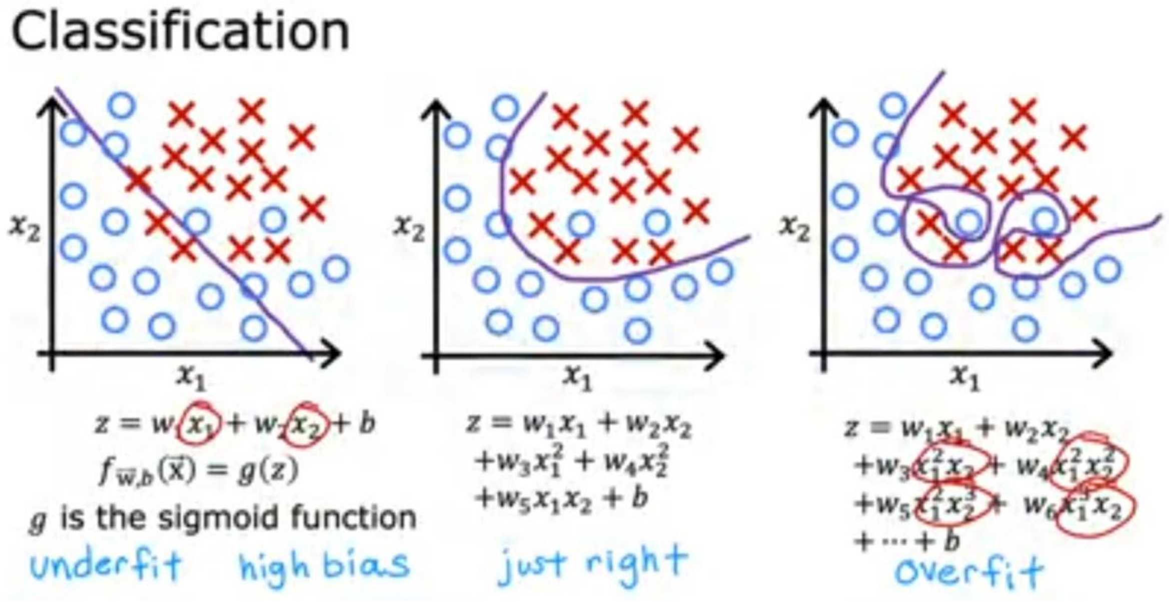


Let's look at another example. If you were to add to your features these quadratic terms, then z becomes this new term in the middle and the decision boundary, that is where z equals zero can look more like this, more like an ellipse or part of an ellipse. This is a pretty good fit to the data, even though it does not perfectly classify every single training example in the training set. Notice how some of these crosses get classified among the circles. But this model looks pretty good. I'm going to call it just right. It looks like this generalized pretty well to new patients.


Classification
x_2
x_1
$z = w_1 x_1 + w_2 x_2 + b$
$f_{\vec{w},b}(\vec{x}) = g(z)$
g is the sigmoid function
underfit high bias
$z = w_1 x_1 + w_2 x_2$
$+ w_3 x_1^2 + w_4 x_2^2$
$+ w_5 x_1 x_2 + b$
just right

Finally, at the other extreme, if you were to fit a very high-order polynomial with many features like these, then the model may try really hard and contoured or twist itself to find a decision boundary that fits your training data perfectly. Having all these higher-order polynomial features allows the algorithm to choose this really over the complex decision boundary.

If the features are tumor size in age, and you're trying to classify tumors as malignant or benign, then this doesn't really look like a very good model for making predictions. Once again, this is an instance of overfitting and high variance because its model, despite doing very well on the training set, doesn't look like it'll generalize well to new examples.


Classification
x_2
x_1
$z = w_1 x_1 + w_2 x_2 + b$
$f_{\vec{w},b}(\vec{x}) = g(z)$
g is the sigmoid function
underfit high bias
$z = w_1 x_1 + w_2 x_2$
$+ w_3 x_1^2 + w_4 x_2^2$
$+ w_5 x_1 x_2 + b$
just right
$z = w_1 x_1 + w_2 x_2$
$+ w_3 x_1^2 x_2 + w_4 x_1^2 x_2^2$
$+ w_5 x_1^2 x_2^3 + w_6 x_1^3 x_2$
$+ \cdots + b$
overfit