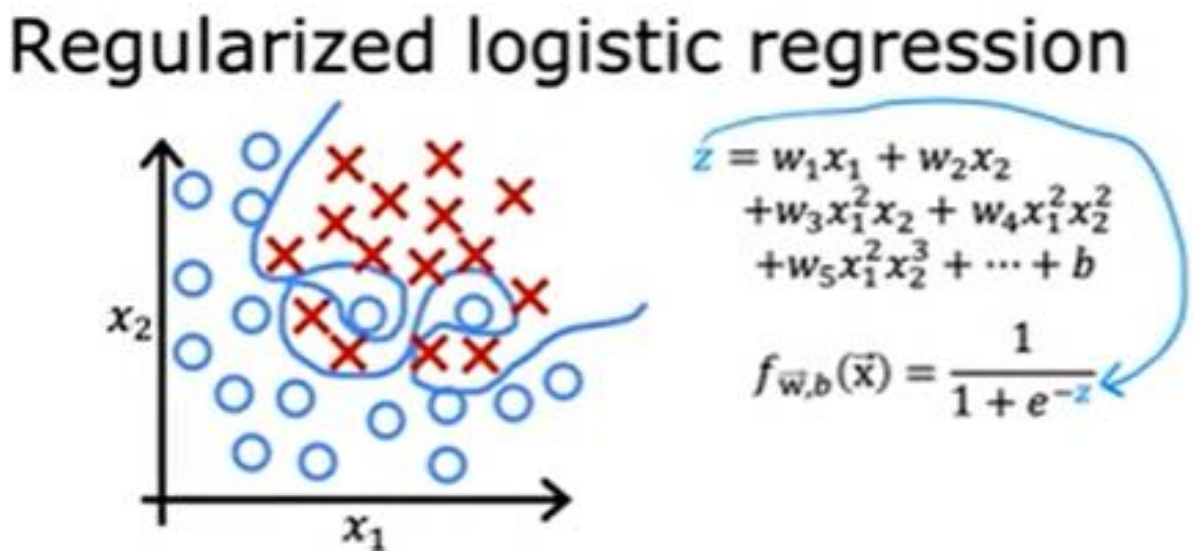
Régression logistique régularisée

In this video, you see how to implement regularized logistic regression. Just as the gradient update for logistic regression has seemed surprisingly similar to the gradient update for linear regression, you find that the gradient descent update for regularized logistic regression will also look similar to the update for regularized linear regression. Let's take a look. Here is the idea.

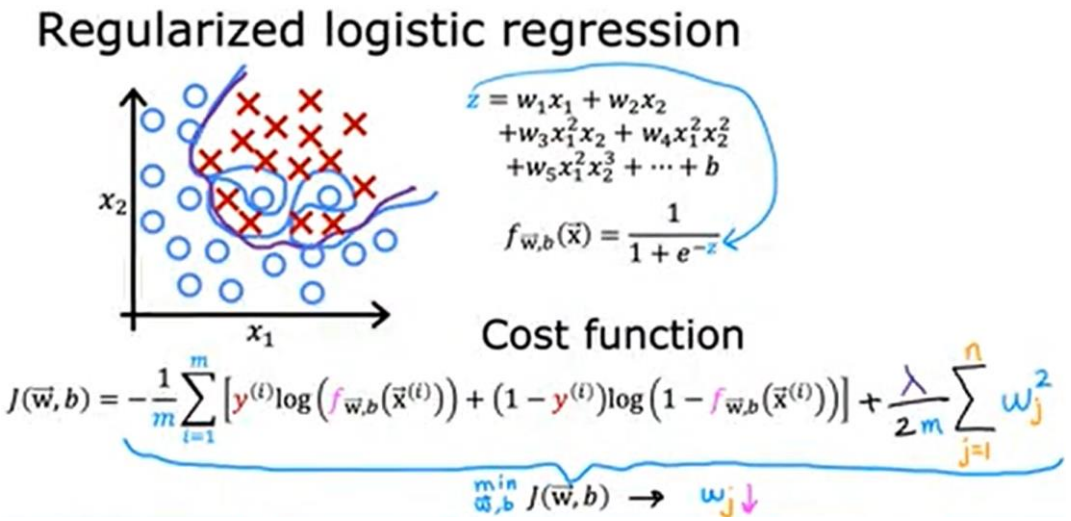
We saw earlier that logistic regression can be prone to overfitting if you fit it with very high order polynomial features like this. Here, z is a high order polynomial that gets passed into the sigmoid function like so to compute f. In particular, you can end up with a decision boundary that is overly complex and overfits as training set.



More generally, when you train logistic regression with a lot of features, whether polynomial features or some other features, there could be a higher risk of overfitting. This was the cost function for logistic regression. If you want to modify it to use regularization, all you need to do is add to it the following term. Let's add lambda to regularization parameter over 2m times the sum from j equals 1 through n, where n is the number of features as usual of wj squared.

When you minimize this cost function as a function of w and b, it has the effect of penalizing parameters w_1, w_2 through w_n, and preventing them from being too large. More generally, when you train logistic regression with a lot of features, whether polynomial features or some other features, there could be a higher risk of overfitting. This was the cost function for logistic regression. If you want to modify it to use regularization, all you need to do is add to it the following term.

Let's add lambda to regularization parameter over 2m times the sum from j equals 1 through n, where n is the number of features as usual of wj squared. When you minimize this cost function as a function of w and b, it has the effect of penalizing parameters w_1, w_2 through w_n, and preventing them from being too large.



How can you actually minimize this cost function j of wb that includes the regularization term? Well, let's use gradient descent as before. Here's a cost function that you want to minimize. To implement gradient descent, as before, we'll carry out the following simultaneous updates over wj and b. These are the usual update rules for gradient descent. Just like regularized linear regression, when you compute where there are these derivative terms, the only thing that changes now is that the derivative respect to wj gets this additional term, lambda over m times wj added here at the end.

Again, it looks a lot like the update for regularized linear regression. In fact is the exact same equation, except for the fact that the definition of f is now no longer the linear function, it is the logistic function applied to z. Similar to linear regression, we will regularize only the parameters w, j, but not the parameter b, which is why there's no change the update you will make for b.