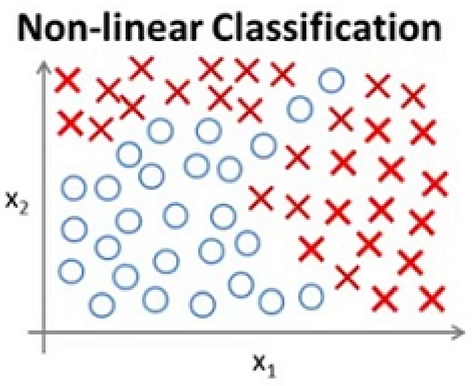
Non-linear Hypotheses

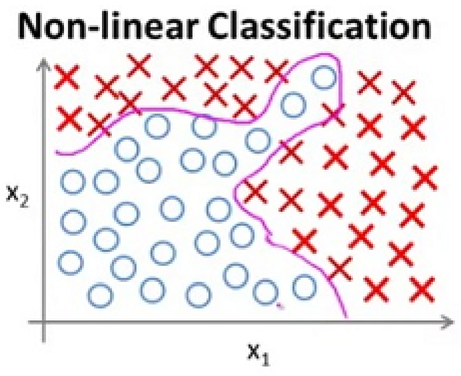
Neutral networks is actually a pretty old idea, but had fallen out of favor for a while. But today, it is the state of the art technique for many different machine learning problems. So why do we need yet another learning algorithm? We already have linear regression and we have logistic regression, **why do we need neural networks?**.

Let's consider a supervised learning classification problem where we have a training set. If we want to apply logistic regression to this problem, one thing we could do is apply logistic regression with a lot of nonlinear features.



$$g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 \\ + \theta_3 x_1 x_2 + \theta_4 x_1^2 x_2 \\ + \theta_5 x_1^3 x_2 + \theta_6 x_1 x_2^2 + \ldots)$$

And, if we include enough polynomial terms then, maybe we can get a hypotheses that separates the positive and negative examples.



This particular method works well when we have only, two features - x_1 and x_2 - because we can then include all those polynomial terms of x_1 and x_2. But for many interesting machine learning problems would have a lot more features than just two.

Let's suppose we have a housing classification problem rather than a regression problem, if we have different features of a house, and we want to predict what are the odds that our house will be sold within the next six months, so that will be a classification problem.

x_1 (size), x_2 (# bedrooms), x_3 (# floors), x_4 (age), ..., x_{100} where $n = 100$

For a problem like this, if we were to include all the quadratic terms, even all of the quadratic that is the second or the polynomial terms, there would be a lot of them.

x_1^2, $x_1 x_2$, $x_1 x_3$, $x_1 x_4$, ..., $x_1 x_{100}$

x_2^2, $x_2 x_3$, $x_2 x_4$, $x_2 x_5$, ..., $x_2 x_{100}$

...

For the case of $n = 100$, we end up with about 5000 features. And, asymptotically, the number of quadratic features grows roughly as $O(n^2)$, where n is the number of the original features, like x_1 through x_{100} that we had. And its actually closer to $\frac{n^2}{2}$.

So including all the quadratic features doesn't seem like it's maybe a good idea, because that is a lot of features and **we might up overfitting the training set**, and it can also be computationally expensive, to be working with that many features. One thing we could do is include only a subset of these, so if you include only the features like:

$x_1^2, x_2^2, x_3^2, ..., x_{100}^2,$

Then the number of features is much smaller. Here we have only 100 such quadratic features, but this is not enough features and certainly won't let us fit the data set.

If you were to include the cubic, or third order known of each others,

$x_1^2 x_2 x_3, x_1 x_2, x_{10} x_{11} x_{17}, ...$

We can imagine there are gonna be a lot of these features. In fact, they are going to $O(n^3)$, and for example, if there are 100 features, we end up with on the order of about $170,000$, and so including these higher auto-polynomial features when our original feature set end is large this really dramatically blows up our feature space and this doesn't seem like a good way to come up with additional features with which to build none many classifiers when n is large. **For many machine learning problems, n (# of features) will be pretty large.**

Video Question: Suppose you are learning to recognize cars from 100×100 pixel images (grayscale, not RGB). Let the features be pixel intensity values. If you train logistic regression including all the quadratic terms $(x_i x_j)$ as features, about how many features will you have?

- $5,000$
- $100,000$

> 50 million (5×10^7)

- 5 billion (5×10^9)

So, simple logistic regression together with adding in maybe the quadratic or the cubic features - **that's just not a good way to learn complex nonlinear hypotheses when n is large** because you just end up with too many features.