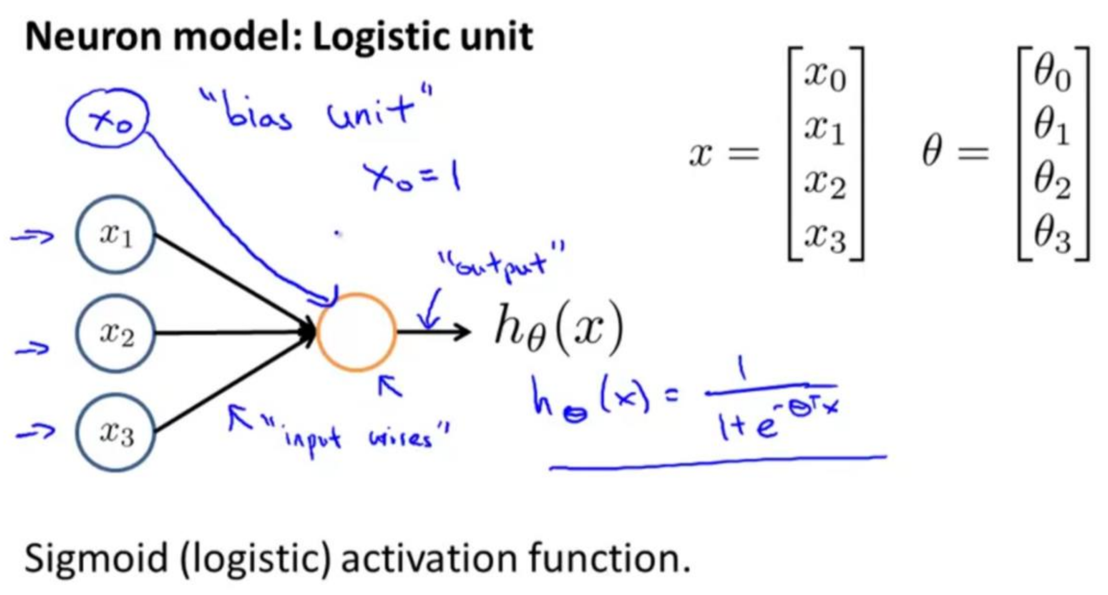
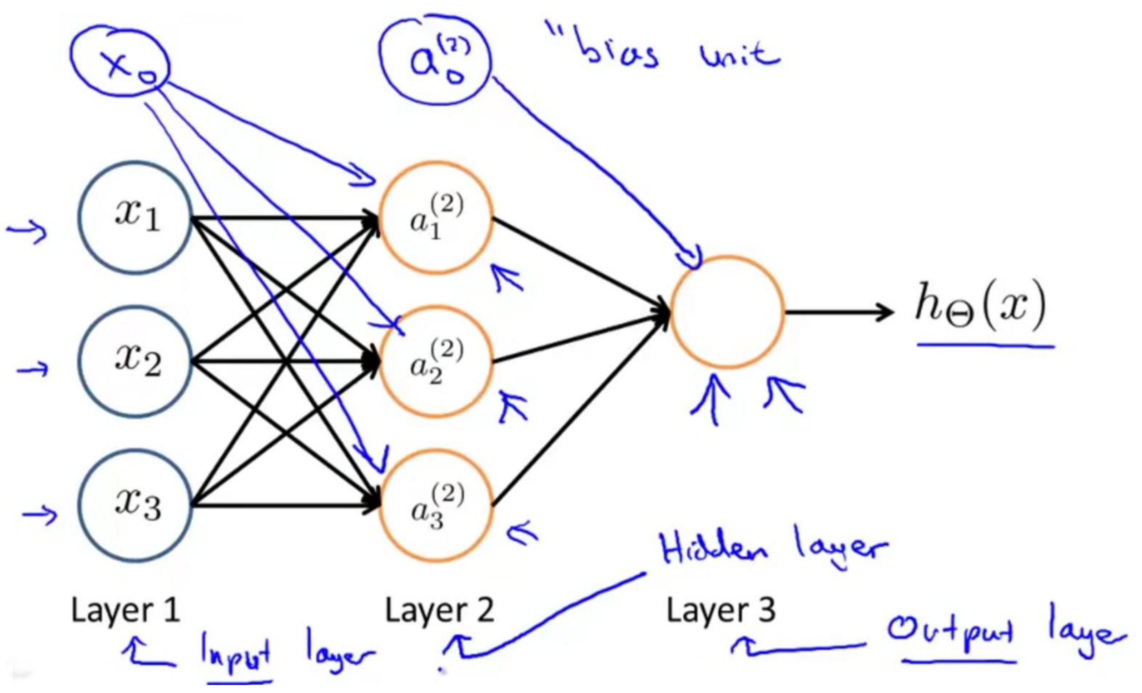
Week 4 –Neural network

Model 1 – Logistic unit

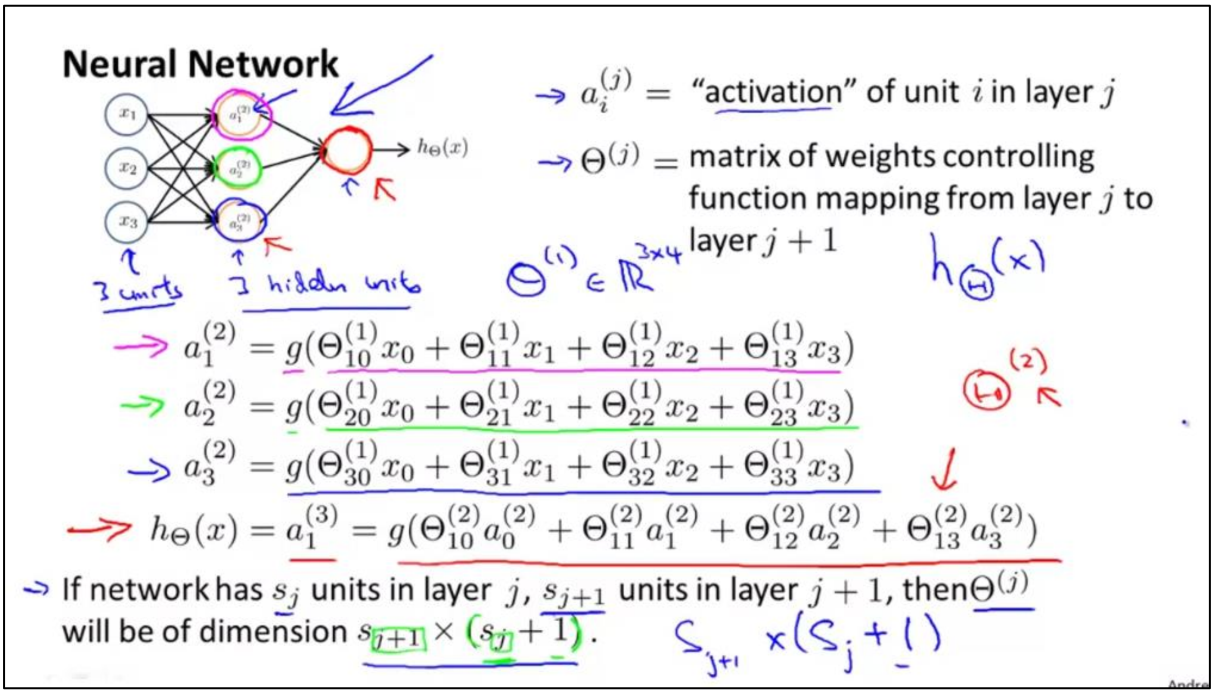


Theta = parameters = Weights, terminology the same

Neural Network:



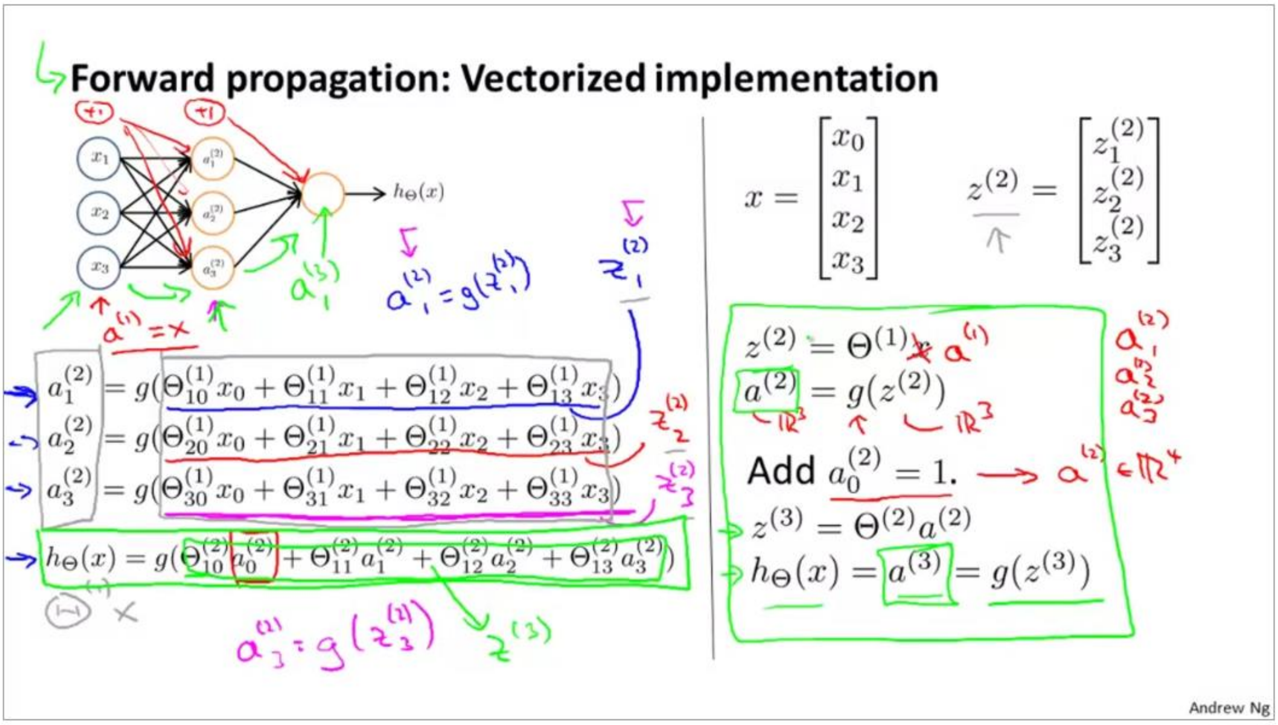
Neural Network representation



Our input nodes (layer 1), also known as the "input layer", go into another node (layer 2), which finally outputs the hypothesis function, known as the "output layer".

We can have intermediate layers of nodes between the input and output layers called the "hidden layers."

Forward Propagation: Vectorised implementation



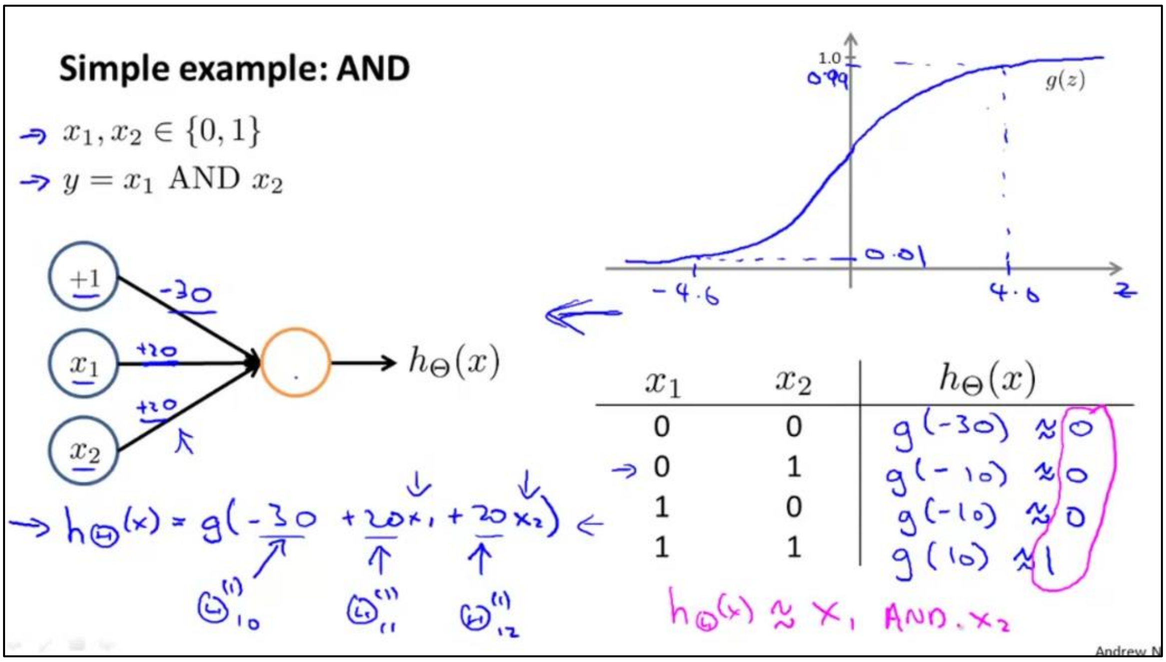
$z^{(j+1)} = \Theta^{(j)} a^{(j)}$

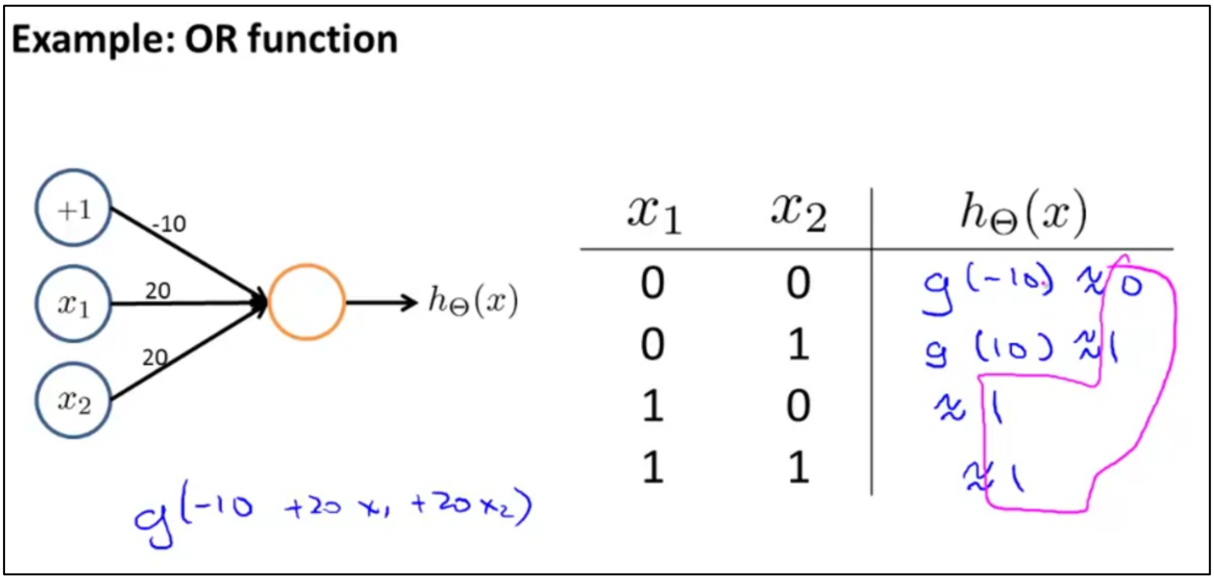
We get this final z vector by multiplying the next theta matrix after $\Theta^{(j-1)}$ with the values of all the activation nodes we just got. This last theta matrix $\Theta^{(j)}$ will have only **one row** which is multiplied by one column $a^{(j)}$ so that our result is a single number. We then get our final result with:

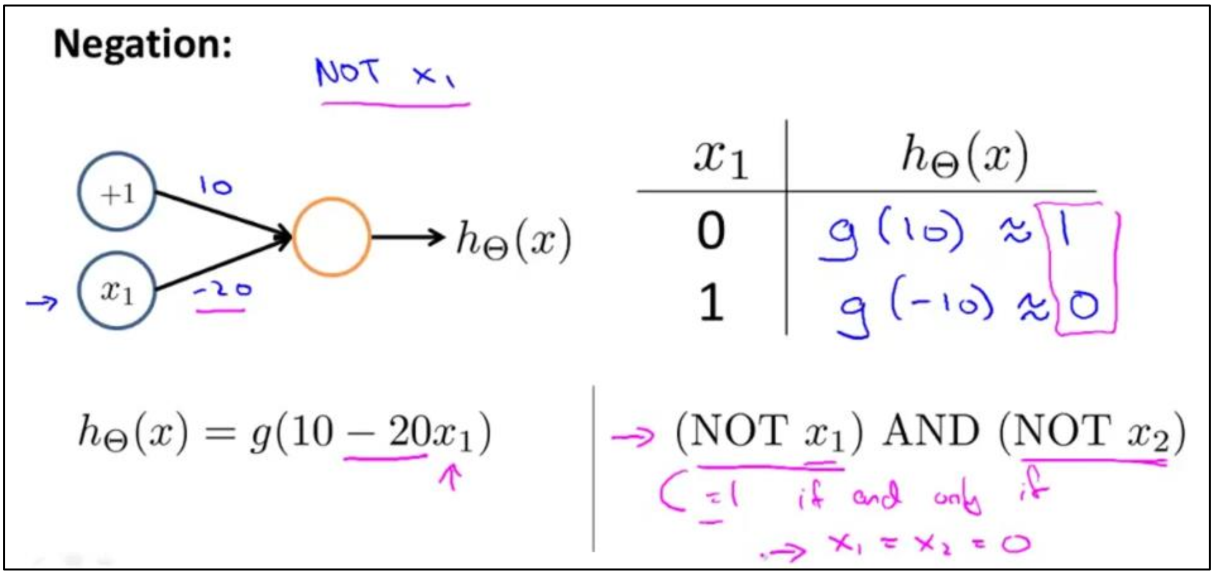
$h_\Theta(x) = a^{(j+1)} = g(z^{(j+1)})$

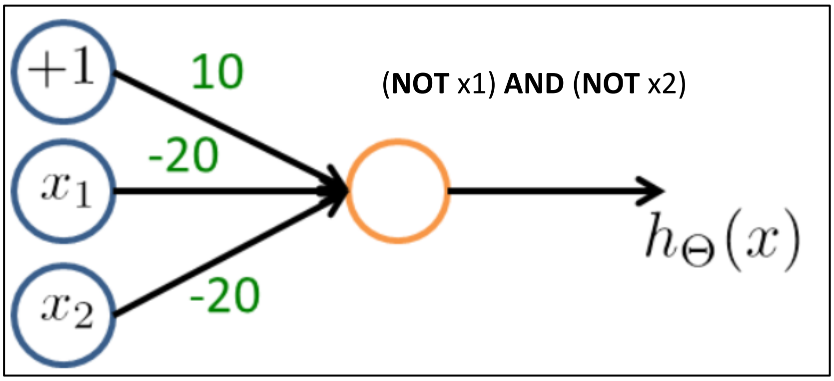
Notice that in this **last step**, between layer j and layer j+1, we are doing **exactly the same thing** as we did in logistic regression. Adding all these intermediate layers in neural networks allows us to more elegantly produce interesting and more complex non-linear hypotheses.

Simple example

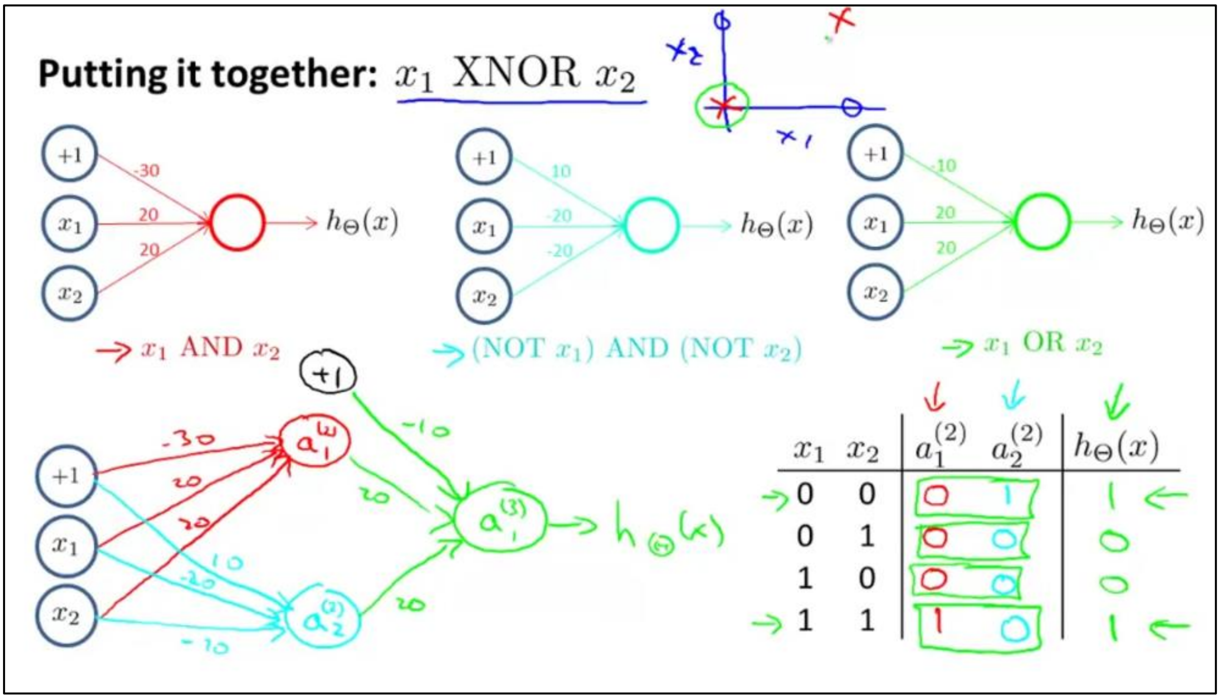








Constructing XNOR gate



Multiclass Classification – One vs All

To classify data into multiple classes, we let our hypothesis function return a vector of values. Say we wanted to classify our data into one of four categories. We will use the following example to see how this classification is done. This algorithm takes as input an image and classifies it accordingly: