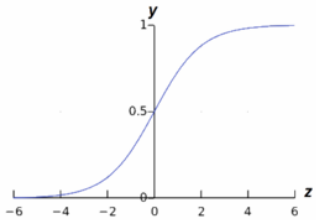
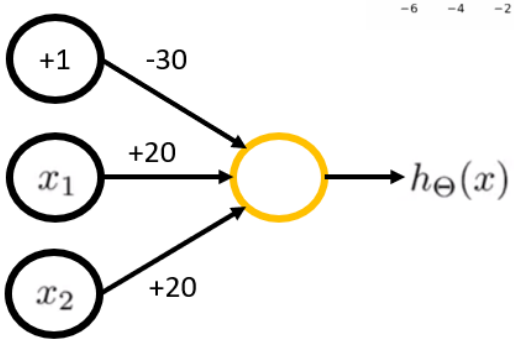
Neural Network Logic Gates

Different networks and activation functions can be combined to create even more complex networks creating a series of logic such as AND, NOT, OR, XNOR, etc.

AND

The AND gate is 1 only if all inputs, x values, are 1 else 0. Using the Sigmoid function as used in Logistic Regression where a value ≥ 0.5 is 1 else 0.





x_1	x_2	$h_\Theta(x)$
0	0	$g(-30) \approx 0$
0	1	$g(-10) \approx 0$
1	0	$g(-10) \approx 0$
1	1	$g(10) \approx 1$

$$h_\Theta(x) = g(-30 + 20x_1 + 20x_2)$$

This can be broken down as follows:

> $$z_1^{(2)} = -30 + (20 \cdot 0) + (20 \cdot 0) = -30 \therefore g(z_1^{(2)}) \approx 0$$
> $$z_2^{(2)} = -30 + (20 \cdot 0) + (20 \cdot 1) = -10 \therefore g(z_2^{(2)}) \approx 0$$
> $$z_3^{(2)} = -30 + (20 \cdot 1) + (20 \cdot 0) = -10 \therefore g(z_3^{(2)}) \approx 0$$
> $$z_4^{(2)} = -30 + (20 \cdot 1) + (20 \cdot 1) = 10 \therefore g(z_4^{(2)}) \approx 1$$

Therefore, given the above example, the classification is positive (1) if and only if x^1 and x^2 equals 1.

OR

The OR gate is 1 if any of the inputs, x values, are 1 else 0. Using the Sigmoid function as used in Logistic Regression where a value ≥ 0.5 is 1 else 0.

XNOR

XNOR is a simpler way of saying NOT (x_1 XOR x_2)