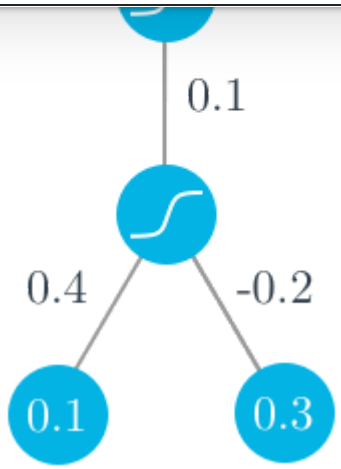

Assume we're trying to fit some binary data and the target is $y = 1$. We'll start with the forward pass, first calculating the input to the hidden unit

$$h = \sum_i w_i x_i = 0.1 \times 0.4 - 0.2 \times 0.3 = -0.02$$

and the output of the hidden unit

$$a = f(h) = \text{sigmoid}(-0.02) = 0.495.$$

Using this as the input to the output unit, the output of the network is

$$\hat{y} = f(W \cdot a) = \text{sigmoid}(0.1 \times 0.495) = 0.512.$$

With the network output, we can start the backwards pass to calculate the weight updates for both layers. Using the fact that for the sigmoid function $f'(W \cdot a) = f(W \cdot a)(1 - f(W \cdot a))$, the error term for the output unit is

$$\delta^o = (y - \hat{y}) f'(W \cdot a) = (1 - 0.512) \times 0.512 \times (1 - 0.512) = 0.122.$$

Now we need to calculate the error term for the hidden unit with backpropagation. Here we'll scale the error term from the output unit by the weight W connecting it to the hidden unit. For the hidden unit error term, $\delta_j^h = \sum_k W_{jk} \delta_k^o f'(h_j)$, but since we have one hidden unit and one output unit, this is much simpler.

$$\delta^h = W \delta^o f'(h) = 0.1 \times 0.122 \times 0.495 \times (1 - 0.495) = 0.003$$

Now that we have the errors, we can calculate the gradient descent steps. The hidden to output weight step is the learning rate, times the output unit error, times the hidden