$$\delta_j = \sum W_{jk}\delta_k f'(h_j)$$

Then, the gradient descent step is the same as before, just with the new errors:

$$\Delta w_{ij} = \eta \delta_j^h x_i$$

where w_{ij} are the weights between the inputs and hidden layer and x_i are input unit values. This form holds for however many layers there are. The weight steps are equal to the step size times the output error of the layer times the values of the inputs to that layer

$$\Delta w_{pq} = \eta \delta_{output} V_{in}$$

Here, you get the output error, δ_{output}, by propagating the errors backwards from higher layers. And the input values, V_{in} are the inputs to the layer, the hidden layer activations to the output unit for example.

Working through an example

Let's walk through the steps of calculating the weight updates for a simple two layer network. Suppose there are two input values, one hidden unit, and one output unit, with sigmoid activations on the hidden and output units. The following image depicts this network. (**Note:** the input values are shown as nodes at the bottom of the image, while the network's output value is shown as $\hat{y}$ at the top. The inputs themselves do not count as a layer, which is why this is considered a two layer network.)