

This example assumes that the layers are numbered from 1 to n, with the first layer being the first hidden layer and the nth layer being the output layer. It also assumes the costfunction $\sum \frac{1}{2}(target - out)^2$ and that the sigmoid function ($sig(x) = \frac{1}{1+e^{-x}}$) is used as an activation function and uses the given example of a 2-4-3-2 network.

The terminology and naming of neurons outputs etc. is mostly from this amazing blogpost and is a prerequisite for understanding the following calculations.

Now if you want to calculate the gradient for a weight of the outputlayer, for example w_{31} connecting neurons c_1 and d_1 , this is fairly simple. Weight w_{31} is right in the beginning (or rather at the end of the network, but since we are trying to do backpropagation the end is our beginning), most of the network becomes irrelevant when calculating the (partial) derivative. Since our intention is to minimize the error we will need to calculate the derivative of the error function E_{total} with respect to w_{31} . When following the chain rule to calculate the derivative we get:

$$\frac{\partial E_{total}}{\partial w_{31}} = \frac{\partial E_{total}}{\partial out_{d1}} * \frac{\partial out_{d1}}{\partial net_{d1}} * \frac{\partial net_{d1}}{\partial w_{31}} \quad (1)$$

Because I had some trouble understanding how the left hand side of the equation equals the right hand side at first and this is essential to understanding anything following this equation, I will explain it in more detail before proceeding.

For that let's split up the Error function into its components. In this particular example we begin with $E_{total} = \sum \frac{1}{2}(target - out)^2$. Having only two output neurons this equals $E_{total} = \frac{1}{2}(target_{d1} - out_{d1})^2 + \frac{1}{2}(target_{d2} - out_{d2})^2$. The first part of the sum is the error E_{d1} for neuron d_1 and the second is the error E_{d2} for neuron d_2 .

We want to derive with respect to w_{31} and not the output of some neuron so let's split those up again. Then we would get $out_{d1} = sig(net_{d1})$ and $out_{d2} = sig(net_{d2})$.

To have w_{31} in our equation we need to split it up once more and get $net_{d1} = w_{31} * out_{c1} + w_{xx} * out_{c2} + w_{xx} * out_{c3}$ and $net_{d2} = w_{xx} * out_{c1} + w_{xx} * out_{c2} + w_{xx} * out_{c3}$. I have omitted stating the correct indices for anything but w_{31} because, as we will see soon, they become irrelevant when deriving with respect to w_{31} .

Putting it back together we get

$$E_{d_1} = \frac{1}{2}(target_{d_1} - sig(net_{d_1} = w_{31} * out_{c_1} + w_{xx} * out_{c_2} + w_{xx} * out_{c_3}))^2 \quad (2)$$