Here is how you can decompose the derivative:

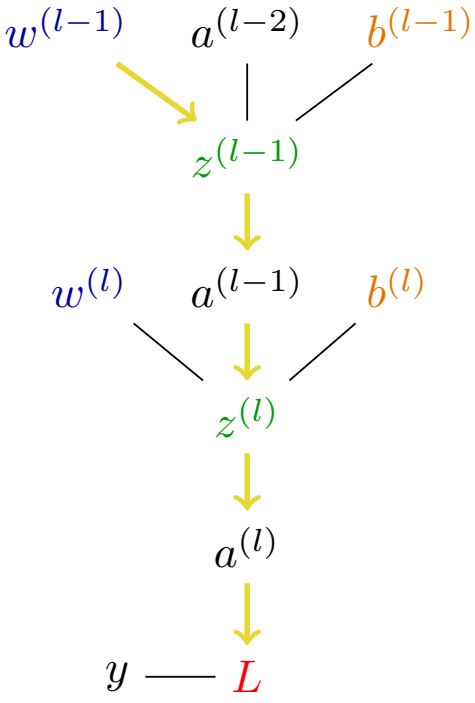
$$\frac{\partial L}{\partial w^{(l-1)}} = \frac{\partial z^{(l-1)}}{\partial w^{(l-1)}} \frac{\partial a^{(l-1)}}{\partial z^{(l-1)}} \underbrace{\frac{\partial z^{(l)}}{\partial a^{(l-1)}} \frac{\partial a^{(l)}}{\partial z^{(l)}} \frac{\partial L}{\partial a^{(l)}}}_{\frac{\partial L}{\partial a^{(l-1)}}}$$

By following the dependencies through our tree and multiplying together a long series of partial derivatives, we now **can calculate the derivative of the cost with respect to any weight or bias of the entire network**. *We are simply applying the same idea of the chain rule that we have always used!*

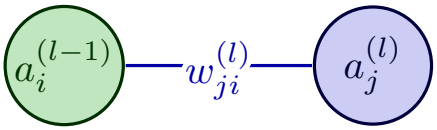
And since we can get any derivative, we can calculate the entire gradient vector:

$$\Delta_w L = \left[\frac{\partial L}{\partial w^{(1)}}, \frac{\partial L}{\partial b^{(1)}}, \ldots, \frac{\partial L}{\partial w^{(l)}}, \frac{\partial L}{\partial b^{(l)}} \right]$$

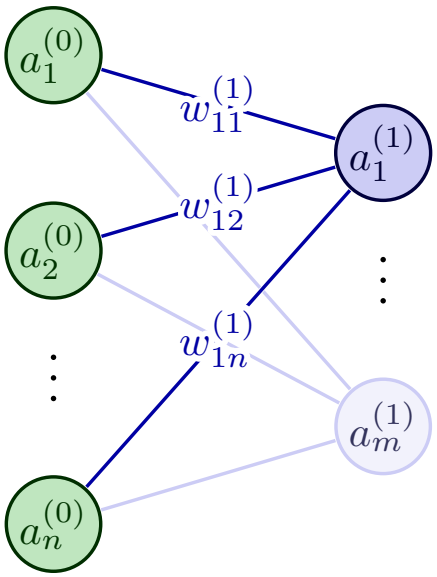
The job is done! At least for this network, now we must add more neurons for each layer. *I know, you are crying inside, but actually, not much changes when we give the layers more neurons: it's just a few more indexes to keep track of* 😄.



Calculations with 3 Layers



Neural Network Indexing



Example of Indexing

When dealing with neural networks with multiple neurons per layer, we have to change the notations a bit.

The activation of each neuron will be denoted with a subscript indicating its position within the layer. Thus, **the superscript of each neuron indicates which layer it is in, while the subscript indicates the specific neuron**.

The weights also need a refresh: additional indices are required to specify their location. In addition to the superscript representing the layer, **two subscripts indicate the edge weight connecting the neuron in layer i to the neuron in layer j**.

I know, these "ji" indices, being backwards, might seem strange or unconventional at first, but they align with the way the weight matrix is usually indexed.

On the left are two images: one formally represents how the nodes and edges of the network are represented, while the other provides an example with two dummy layers, named 0 and 1, with n and m nodes, respectively.