L = number of layers in a network.

l = a particular layer in the network : $1 \leq l \leq L$

a_i^l = value of the ith node in layer l

w_{ij}^l = The weight of the connection between a_j^{l-1} and a_i^l

b_i^l = bias of node i in the layer l.

y_i = the expected outcome from a given input.

Generating the output

$$\sigma(x) = \frac{1}{1+e^{-x}}$$ ← keeps our nodes between 0 & 1.

Let $$z_i^l = w_{ij}^l a_j^{l-1} + b_i^l$$ ← This will be useful later.

$$\Rightarrow a_i^l = \sigma(z_i^l)$$

Cost function

$$C = \frac{1}{n_L} \sum_i^{n_L} (a_i^L - y_i)^2$$ n_L = length of layer L.

$$\nabla C = \left(\frac{\partial C}{\partial w_{ij}^l}, \frac{\partial C}{\partial b_j^l} \right) \quad \forall l, \ 1 \leq l \leq L$$

Essentially one big vector to contain the change in cost with respect to every weight and bias in the network.

$$\frac{\partial C}{\partial w_{AB}^l} = \frac{\partial C}{\partial a^L} \frac{\partial a^L}{\partial a_A^l} \frac{\partial a_A^l}{\partial w_{AB}^l} = \overset{(1)}{\frac{\partial C}{\partial a^L}} \ \overset{(2)}{\frac{\partial a^L}{\partial a^{l+1}}} \ \overset{(3)}{\frac{\partial a^{l+1}}{\partial a_A^l}} \ \overset{(4)}{\frac{\partial a_A^l}{\partial w_{AB}^l}}$$

Note: I'm using A, B as fixed indices, so this formula only shows the cost w.r.t ONE specific weight.

Having got this formula, we can calculate each component separately.

① $$\frac{\partial C}{\partial a_i^L} = \frac{1}{n_L}\sum \frac{\partial}{\partial a_i^L}(a_i^L - y_i)^2 = \frac{2}{n_L}(a_i^L - y_i)$$

↑ This is a vector. It helps a lot to visualise what it means in the context of the network.

Matrix

② $$\frac{\partial a_i^L}{\partial a_j^{l+1}} = \underbrace{\frac{\partial a_i^L}{\partial a^{L-1}}\frac{\partial a^{L-1}}{\partial a^{L-2}}\cdots\frac{\partial a^{l+2}}{\partial a_j^{l+1}}}_{\text{These are all matrices.}} \overset{*}{=} (\sigma'(z_i^L)w_{ip}^L)(\sigma'(z_p^{L-1})w_{pm}^{L-1})\cdots(\sigma'(z_q^{l+2})w_{jq}^{l+2})$$

$$* \quad \frac{\partial a_m^{l+1}}{\partial a_p^l} = \frac{\partial}{\partial a_p^l}(\sigma(w_{mp}^{l+1}a_p^l + b_m^{l+1})) = \underbrace{\sigma'(z_m^{l+1})w_{mp}^{l+1}}$$

↑ A matrix with m columns and p rows.

That's the worst one done. Phew! The indices are very tricky to code.

③ $$\frac{\partial a_j^{l+1}}{\partial a_A^l} = \frac{\partial}{\partial a_A^l}(\sigma(w_{jA}^{l+1}a_A^l + b_j^{l+1})) = (\sigma'(z_j^{l+1})w_{jA}^{l+1})$$

↑ vector.

Again, it [redacted] is helpful to think of it visually; How does each node in layer $l+1$ change w.r.t the node a_A^l.

④ $$\frac{\partial a_A^l}{\partial w_{AB}^l} = \frac{\partial}{\partial w_{AB}^l}\sigma(w_{AB}^l a_B^{l-1} + b_A^l)) = \sigma'(z_A^l)a_B^{l-1}$$ ← Scalar.

Quick Remark: $a_A^l = \sigma(w_{Ai}^l a_i^{l-1} + b_A^l)$ [redacted] $0 \le i < n_{L-1}$

but since, $$\frac{\partial w_{Ai}^l}{\partial w_{AB}^l} = \begin{cases} 1 & \text{if } i = B \\ 0 & \text{otherwise} \end{cases}$$

I've left it out.

Most of this, except eq ④, is the same for $\frac{\partial C}{\partial b^l}$.