Backpropagation with Sigmoid and Binary Cross-Entropy (No Biases)

1. Forward definitions (no biases)

For layer ℓ and unit j with inputs $a_i^{\ell-1}$:

$$z_j^\ell \;=\; \sum_i W_{ji}^\ell \, a_i^{\ell-1}, \qquad a_j^\ell \;=\; \sigma\bigl(z_j^\ell\bigr), \qquad \sigma(u) = \frac{1}{1+e^{-u}}, \;\; \sigma'(u) = \sigma(u)\bigl(1-\sigma(u)\bigr).$$

2. Loss at the output (binary cross-entropy)

For the output layer L with independent sigmoid outputs a_j^L and targets t_j:

$$\mathcal{L} \;=\; -\sum_j \Bigl(t_j \log a_j^L + (1-t_j)\log\bigl(1-a_j^L\bigr)\Bigr).$$

3. Gradient at the output layer

We seek the derivative w.r.t. the *pre-activation* z_j^L:

$$\delta_j^L \;\equiv\; \frac{\partial \mathcal{L}}{\partial z_j^L}.$$

Apply the chain rule:

$$\frac{\partial \mathcal{L}}{\partial z_j^L} \;=\; \frac{\partial \mathcal{L}}{\partial a_j^L} \cdot \frac{\partial a_j^L}{\partial z_j^L}.$$

Each factor is

$$\frac{\partial \mathcal{L}}{\partial a_j^L} = -\frac{t_j}{a_j^L} + \frac{1-t_j}{1-a_j^L}, \qquad \frac{\partial a_j^L}{\partial z_j^L} = \sigma'\bigl(z_j^L\bigr) = a_j^L\bigl(1-a_j^L\bigr).$$

Multiplying and simplifying gives the familiar result

$$\boxed{\delta_j^L \;=\; a_j^L - t_j}$$

(i.e., the sigmoid derivative cancels with the BCE term via the chain rule).

4. Propagating to a hidden layer via the chain rule

Start from the forward relation of the next layer:

$$z_k^{\ell+1} \;=\; \sum_j W_{kj}^{\ell+1} \, a_j^\ell.$$

For a hidden unit j in layer ℓ,

$$\frac{\partial \mathcal{L}}{\partial z_j^\ell} = \sum_k \underbrace{\frac{\partial \mathcal{L}}{\partial z_k^{\ell+1}}}_{\delta_k^{\ell+1}} \underbrace{\frac{\partial z_k^{\ell+1}}{\partial a_j^\ell}}_{W_{kj}^{\ell+1}} \underbrace{\frac{\partial a_j^\ell}{\partial z_j^\ell}}_{\sigma'(z_j^\ell)} = \sum_k \delta_k^{\ell+1} W_{kj}^{\ell+1} \sigma'(z_j^\ell).$$

Using $\sigma'(z_j^\ell) = a_j^\ell(1 - a_j^\ell)$, the hidden-layer delta becomes

$$\boxed{\delta_j^\ell = \Big(\sum_k W_{kj}^{\ell+1} \delta_k^{\ell+1}\Big) a_j^\ell(1 - a_j^\ell)}.$$

5. Gradients w.r.t. the weights (no biases)

From $z_j^\ell = \sum_i W_{ji}^\ell a_i^{\ell-1}$,

$$\frac{\partial \mathcal{L}}{\partial W_{ji}^\ell} = \frac{\partial \mathcal{L}}{\partial z_j^\ell} \cdot \frac{\partial z_j^\ell}{\partial W_{ji}^\ell} = \delta_j^\ell a_i^{\ell-1}.$$

A plain gradient-descent update with learning rate η is

$$W_{ji}^\ell \leftarrow W_{ji}^\ell - \eta \frac{\partial \mathcal{L}}{\partial W_{ji}^\ell} = W_{ji}^\ell - \eta \delta_j^\ell a_i^{\ell-1}.$$

6. Minimal backprop recipe (index form)

1. Forward: compute all z_j^ℓ and a_j^ℓ for $\ell = 1, \ldots, L$.
2. Output deltas: $\delta_j^L = a_j^L - t_j$.
3. For $\ell = L - 1$ down to 1: $\delta_j^\ell = \Big(\sum\limits_k W_{kj}^{\ell+1} \delta_k^{\ell+1}\Big) a_j^\ell(1 - a_j^\ell)$.
4. Gradients: $\dfrac{\partial \mathcal{L}}{\partial W_{ji}^\ell} = \delta_j^\ell a_i^{\ell-1}$.