1 Backpropagation

In this problem, you will derive the equations needed to calculate the gradient of the cost function in a feedforward neural network using backpropagation. This gradient is used within learning algorithms (such as gradient descent) that train the neural network.

Recall that the output from the j^{th} neuron in layer ℓ is given by

$$a_j^{(\ell)} = g_\ell\left(z_j^{(\ell)}\right),$$

where g_ℓ is a non-linear activation function and

$$z_j^{(\ell)} = \sum_{i=1}^{n_{\ell-1}} a_i^{(\ell-1)} W_{ij}^{(\ell)} + b_j^{(\ell)},$$

with $W_{ij}^{(\ell)}$ representing the weights and $b_j^{(\ell)}$ the biases of the network. Assume that the cost function C can be expressed as a sum over the dataset $\mathcal{D}$ as

$$C = \frac{1}{|\mathcal{D}|} \sum_{\mathbf{x} \in \mathcal{D}} c(\mathbf{x}),$$

and define the notation

$$\delta_j^{(\ell)} = \frac{\partial c}{\partial z_j^{(\ell)}}.$$

a) Show that the quantity $\delta_j^{(\ell)}$ can be expressed as

$$\delta_j^{(L)} = \frac{\partial c}{\partial a_j^{(L)}}\, g_L'\left(z_j^{(L)}\right),$$

$$\delta_j^{(\ell)} = \sum_{k=1}^{n_{\ell+1}} \delta_k^{(\ell+1)} W_{kj}^{(\ell+1)\,T} g_\ell'\left(z_j^{(\ell)}\right) \qquad (\text{for } \ell < L).$$

a)

chain rule

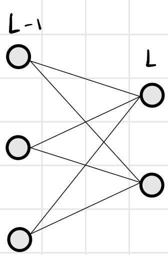
$$\delta_j^{(L)} = \frac{\partial c}{\partial z_j^{(L)}} = \frac{\partial c}{\partial a_j^{(L)}} \cdot \frac{\partial a_j^{(L)}}{\partial z_j^{(L)}}$$

on the last layer

By definition, $a_j^{(L)} = g_L(z_j^{(L)}) \Rightarrow \dfrac{\partial a_j^{(L)}}{\partial z_j^{(L)}} = g_L'(z_j^{(L)})$

$$\Rightarrow \delta_j^{(L)} = \frac{\partial c}{\partial a_j^{(L)}}\, g_L'(z_j^{(L)})$$

Observe now layer $L-1$. Schematically, we have something like



W_{kj}^{L} $\qquad$ δ_k^{L}

Then, $$\frac{\partial c}{\partial a_j^{(L-1)}} = \sum_{k=1}^{n_L} \frac{\partial z_k^{(L)}}{\partial a_j^{(L-1)}} \frac{\partial a_k^{(L)}}{\partial z_k^{(L)}} \frac{\partial c}{\partial a_k^{(L)}}$$

influenced by all neurons in the previous layer

Although this was written for the second to last layer, the same would apply to layer $L-2$, just swapping $L \to L-1$. In general, we have

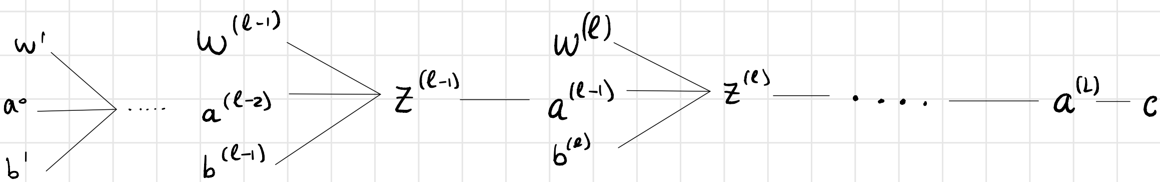
$$\frac{\partial c}{\partial a_j^{(\ell)}} = \sum_{k=1}^{n_{\ell+1}} W_{kj}^{(\ell+1)} \delta_k^{(\ell+1)}$$

$$\Rightarrow \quad \frac{\partial c}{\partial z_j^{(\ell)}} = \frac{\partial c}{\partial a_j^{(\ell)}} \overbrace{\frac{\partial a_j^{(\ell)}}{\partial z_j^{(\ell)}}}^{g_\ell'(z_j^{(\ell)})} = \sum_{k=1}^{n_{\ell+1}} W_{kj}^{(\ell+1)} \delta_k^{(\ell+1)}$$

b) Show that the partial derivatives of the cost function with respect to the network's weights and biases can be calculated as

$$\frac{\partial c}{\partial W_{ij}^{(\ell)}} = a_i^{(\ell-1)} \delta_j^{(\ell)}, \qquad \frac{\partial c}{\partial b_j^{(\ell)}} = \delta_j^{(\ell)}.$$

functions are computed in chains:



Using the chain rule:

by def.

$$\frac{\partial c}{\partial w_{ij}^{(\ell)}} = \frac{\partial z_i^{(\ell)}}{\partial w_{ij}^{(\ell)}} \frac{\partial c}{\partial z_i^{(\ell)}} = a_i^{(\ell-1)} \cdot \delta_j^{(\ell)}$$

by def.

$$\frac{\partial c}{\partial b_j^{(\ell)}} = \frac{\partial z_i^{(\ell)}}{\partial b_j^{(\ell)}} \cdot \frac{\partial c}{\partial z_i^{(\ell)}} = \delta_j^{(\ell)}$$

$\frac{\partial z}{\partial b} = 1$ since $z = wa + b$