1. Feedforward neural networks

1.3b. The Backpropagation algorithm (2)

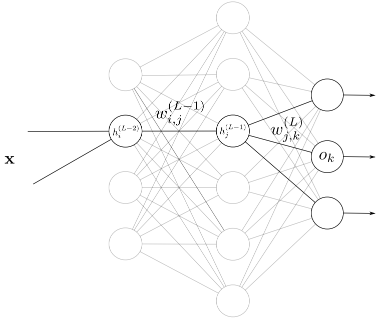
Manel Martínez-Ramón
Meenu Ajith
Aswathy Rajendra Kurup

Gradient with respect to hidden weights

- Using the same reasoning as before, we compute the gradient of the cost function $J_{ML}(\mathbf{y}, \mathbf{f}(\mathbf{x}))$ with respect to weight $w_{i,j}^{(L-1)}$:

$$\frac{d}{dw_{i,j}^{(L-1)}} J_{ML}\left(\mathbf{y}, \underbrace{\mathbf{o}\left({\mathbf{W}^{(L)}}^\top \boldsymbol{\phi}\left({\mathbf{W}^{(L-1)}}^\top \mathbf{h}^{(L-2)}\right)\right)}_{\mathbf{f}(\mathbf{x})}\right) \tag{1}$$

- First, we determine what elements are connected to $w_{i,j}^{(L-1)}$, and this can be done graphically.



- Elements involved in the computation of the derivative with respect to $w_{i,j}^{(L-1)}$:
  - Weights $w_{j,k}^{(L)}$ and outputs o_k.
  - Node $h_j^{(L-1)}$.
  - Node $h_i^{(L-2)}$.

Chain rule
Hidden layer

- Specifically, the elements of the chain are

$$
\begin{aligned}
o_k = o(z_k^{(L)}), &\quad z_k^{(L)} = \mathbf{w}_k^{(L)} \mathbf{h}^{(L-1)}, \ \forall k \\
h_j^{(L-1)} = \phi\left(z_j^{(L-1)}\right), &\quad z_j^{(L-1)} = \mathbf{w}_j^{(L-1)} \mathbf{h}^{(L-2)}
\end{aligned}
\tag{2}
$$

  where $\mathbf{w}_j^{(L-1)}$ contains the element of interest $w_{i,j}^{(L-1)}$.

- We can compute the derivative with respect to $w_{i,j}^{(L-1)}$:

$$
\frac{d}{dw_{i,j}^{(L-1)}} J_{ML}(\mathbf{y}, \mathbf{f}(\mathbf{x})) = \sum_k \underbrace{\frac{\delta J_{ML}}{do_k} \frac{do_k}{dz_k^{(L)}}}_{\delta_k^{(L)}} \underbrace{\frac{dz_k^{(L)}}{dh_j^{(L-1)}}}_{w_{j,k}^{(L)}} \underbrace{\frac{dh_j^{(L-1)}}{dz_j^{L-1}}}_{\phi'} \underbrace{\frac{dz_j^{(L-1)}}{dw_{i,j}^{(L-1)}}}_{h_i^{L-2}}
\tag{3}
$$

Chain rule

Hidden layer

- Taking into account the expressions in Eqs. (2) and (3), this turns into equation

$$\frac{d}{dw_{i,j}^{(L-1)}} J_{ML}(\mathbf{y}, \mathbf{f}(\mathbf{x})) = \underbrace{\sum_k \delta_k^{(L)} w_{k,j}^{(L)} \phi' \left(z_j^{(L-1)} \right)}_{\delta_j^{L-1}} h_i^{L-2} = h_i^{(L-2)} \delta_j^{(L-1)} \tag{4}$$

- Expression $\sum_k \delta_k^{(L)} w_{k,j}^{(L)}$ is the element j of vector $\mathbf{W}^{(L)} \boldsymbol{\delta}^{(L)}$
- This vector is elementwise multiplied with the elements of $\boldsymbol{\phi}' \left(\mathbf{z}^{(L-1)} \right)$.

Weight update
Hidden layer

- In summary, from

$$\frac{d}{dw_{i,j}^{(L-1)}} J_{ML}(\mathbf{y}, \mathbf{f}(\mathbf{x})) = \sum_k \delta_k^{(L)} w_{k,j}^{(L)} \phi' \left(z_j^{(L-1)} \right) h_i^{L-2} = h_i^{(L-2)} \delta_j^{(L-1)}$$

- we define

$$\boldsymbol{\delta}^{(L-1)} = \mathbf{W}^{(L)} \boldsymbol{\delta}^{(L)} \odot \phi' \left(\mathbf{z}^{(L-1)} \right) \tag{5}$$

- and

$$\nabla_{\mathbf{W}^{L-1}} J_{ML} \left(\mathbf{y}, \mathbf{f} \left(\mathbf{x} \right) \right) = \mathbf{h}^{(L-2)} \boldsymbol{\delta}^{(L-1)^\top} \tag{6}$$

- Finally

$$\mathbf{W}^{(L-1)} \leftarrow \mathbf{W}^{(L-1)} - \mu \mathbf{h}^{(L-2)} \boldsymbol{\delta}^{(L-1)^\top} \tag{7}$$

Weight update

Hidden layer

- The process can be iterated down to the input layer, with the same result, and therefore the update of weight matrix $\mathbf{W}^{(l-1)}$ is

$$\mathbf{W}^{(l-1)} \leftarrow \mathbf{W}^{(l-1)} - \mu \mathbf{h}^{(l-2)} \boldsymbol{\delta}^{(l-1)\top} \tag{8}$$

where

$$\boldsymbol{\delta}^{(l-1)} = \mathbf{W}^{(l)} \boldsymbol{\delta}^{(l)} \odot \phi' \left(\mathbf{z}^{(l-1)} \right) \tag{9}$$

where to start and end the process, we need

$$\begin{aligned} \boldsymbol{\delta}^{(L)} &= \nabla_{\mathbf{o}} J_{ML}(\mathbf{y}, \mathbf{o}) \odot \mathbf{o}' \\ \mathbf{h}^{(0)} &= \mathbf{x} \quad \text{(Input layer)} \end{aligned} \tag{10}$$