Derivation of backpropagation generalized

Machine Learning Course A.A. 2022/2023

Filippo Fantinato 2041620

November 11, 2022

1 Exercise

Extend the derivation of Backpropagation to more than one hidden layers. [1]

During the class about backpropagation we saw that for a 3 layers network the derivation is the following:

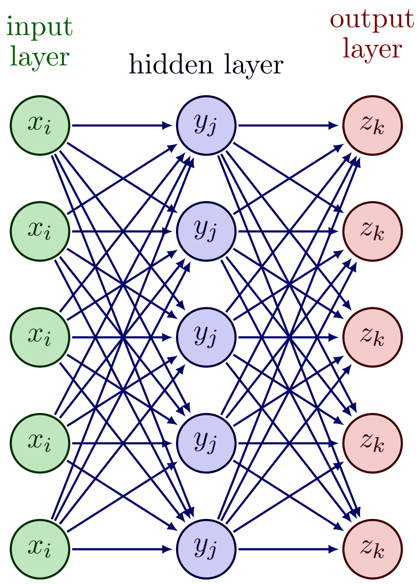

Figure 1: Example NN with one hidden layer

[1] Note that I retrieved all information from this site: https://blog.yani.ai/backpropagation/

$$\frac{\partial E}{\partial W_{kj}} = -\frac{1}{cN}\sum_{s=1}^{N}\delta_k^{(s)} y_j^{(s)}$$

$$\frac{\partial E}{\partial W_{ji}} = -\frac{1}{cN}\sum_{s=1}^{N}\delta_j^{(s)} x_i^{(s)}$$

where:

$$\delta_k^{(s)} = z_k(1-z_k)(t_k^{(s)} - z_k^{(s)})$$

$$\delta_j^{(s)} = y_j(1-y_j)\sum_{k=1}^{size(k)}\delta_k^{(s)} w_{kj}$$

and the activation function is Sigmoid.

As we can see, the delta value of the hidden unit is computed exploiting the delta value of all output units. Therefore, generalizing, the calculation of the delta value for the layer i there is a recursive step which involves the computation of all delta values for each adjacent unit in the layer $i+1$.

So we can define the computation of the delta value as follows:

$$\delta_i^{(s)} = \begin{cases} f'(z_i^{(s)})(t_i^{(s)} - z_i^{(s)}) & \text{if } i \text{ is an output layer} \\ f'(v_i^{(s)})\sum_{k=1}^{size(i+1)}\delta_k^{(s)} w_{ki} & \text{otherwise} \end{cases}$$

where $v_i^{(s)}$ is the output of the hidden layer i w.r.t. the example s, $size(i+1)$ is the number of units in the layer $i+1$ and f' is the derivative of the chosen activation function.

Adopting the Sigmoid as activation function, we end up having the same recursive formula to compute the delta value as we saw during the lesson.

Summarizing, the derivative of a hidden layer i is

$$\frac{\partial E}{\partial W_{ji}} = -\frac{1}{cN}\sum_{s=1}^{N}\delta_j^{(s)} x_i^{(s)}$$

where $\delta_j^{(s)}$ is computed recursively as explained above.