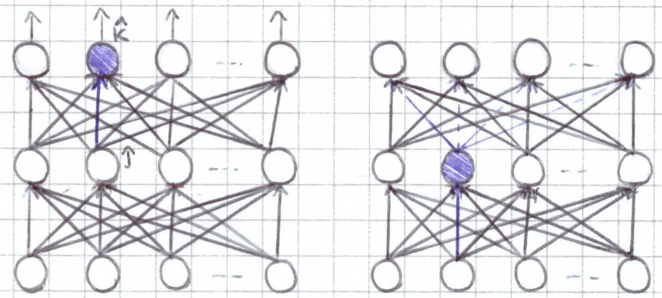
Lecture 09 Cristiano Panighel 2058351

- Extend the derivation of Backprop to more than one hidden layer

During the lecture we have seen that



$$\frac{\partial E}{\partial w_{kj}} = \frac{1}{CN} \sum_{s=1}^{N} \delta_k^{(s)} y_j^{(s)}$$

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

with

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

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

To define the Backpropagation algorithm on neural networks with more than one hidden layer, we must extend the definition of δ from the previous hidden layer to the final one.

We can consider the definition of δ for the units in the hidden layer l and, by recursion, define δ on $h<l$

$$\delta_{j_h}^{(s)} = \begin{cases} y_{j_l}^{(s)}(1-y_{j_l}^{(s)}) \sum_{k=1}^{C} w_{k,j_l} \delta_k^{(s)} & h=l \\ y_{j_h}^{(s)}(1-y_{j_h}^{(s)}) \sum_{j_{h+1}=1}^{N_{H,h+1}} w_{j_{h+1},j_h} \delta_{j_{h+1}}^{(s)} & 0<h<l \end{cases}$$

So, we can formalize the Backpropagation algorithm as follows:

- Initialize all the weights with small random values;
- Until the end condition is not verified, for each $s=(x,t) \in S$
  - With x as sample, compute the vectors y and z;
  - for each output k,

$$\delta_k^{(s)} = (t_k^{(s)} - z_k^{(s)}) z_k^{(s)} (1-z_k^{(s)})$$
$$\Delta w_{k,j_l} = \delta_k^{(s)} y_{j_l}^{(s)}$$

  - for each hidden unit j_h with $h>1$

$$\delta_{j_h}^{(s)} = \begin{cases} y_{j_l}^{(s)}(1-y_{j_l}^{(s)}) \sum_{k=1}^{C} w_{k,j_l} \delta_k^{(s)} & h=l \\ y_{j_h}^{(s)}(1-y_{j_h}^{(s)}) \sum_{j_{h+1}=1}^{N_{H,h+1}} w_{j_{h+1},j_h} \delta_{j_{h+1}}^{(s)} & 1<h<l \end{cases}$$

$$\Delta w_{j_h,j_{h-1}} = \delta_{j_h}^{(s)} y_{j_{h-1}}^{(s)}$$

  - for each hidden unit j_1

$$\delta_{j_1}^{(s)} = y_{j_1}^{(s)}(1-y_{j_1}^{(s)}) \sum_{j_2=1}^{N_{H,2}} w_{j_2,j_1} \delta_{j_2}^{(s)}$$

$$\Delta w_{j_1,i} = \delta_{j_1}^{(s)} x_i^{(s)}$$

  - Upload all the weights $w_{sq} \leftarrow w_{sq} + \eta \Delta w_{sq}$