DEFINITIONS — Colored Orange

Network

We will fix a single input sample $\vec{s}$ which expects output $\vec{t}$

Layer index $k = 0, \dots, L$ (i.e: there are $L+1$ layers)
Input Neuron index in layer k $i = 0, \dots, \#k-1$
Neuron index in layer k $j = 0, \dots, *k$
Output neuron index $q = 0, \dots, *L$

$\#k$ Size of layer k $\left(\#0 = 784 \quad \#L = 10\right)$

$*k = (\#k) - 1$ last index of layer k (without bias neuron)

$a_j^k = g(h_j^k)$ (Neuron j of layer k)

$a_j^0 = s_j$ (input j) $\quad a_{\#k}^k = 1$ (bias extension)

$h_j^k = \sum_{i=0}^{\#k-1} a_i^{k-1} w_{ij}^{k-1}$ $\quad g(x)$: activation function

w_{ij}^k: weight from a_i^k to a_j^{k+1}

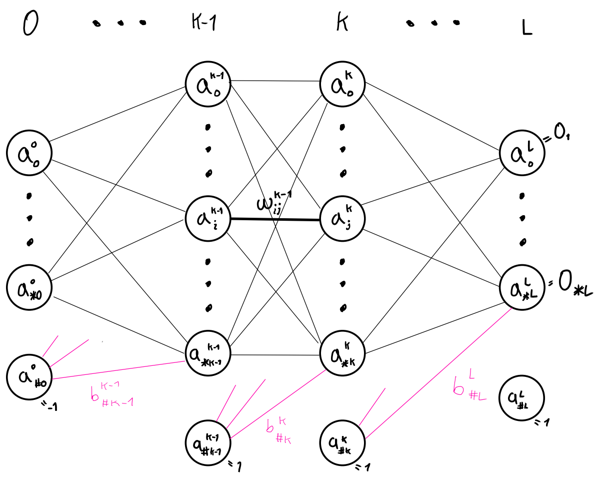
$b_j^k = w_{\#k-1,j}^{k-1}$ bias of neuron a_j^k

$O_q = a_q^L$ (Output neuron q) (we want $\vec{O} = \vec{t}$)

$$E(\vec{s}, \vec{t}) = \sum_{q=0}^{*L} (O_q - t_q)^2 / \#L$$

$$\nabla E_{ij}^k = \frac{\partial E(\vec{s}, \vec{t})}{\partial w_{ij}^k}$$ (influence of w_{ij}^k in E)

NETWORK DIAGRAM



Equations Development

Gradient

We would like to decrease the error E by tweaking w_{ij}^k

$$\nabla E_{ij}^k = \frac{\partial E(\bar{s}, \mathcal{I})}{\partial w_{ij}^k} = \frac{\partial}{\partial w_{ij}^k} \frac{1}{\#L} \sum_{q=0}^{\#L} (O_q - t_q)^2$$

$$= \frac{1}{\#L} \sum_{q=0}^{\#L} \frac{\partial (O_q - t_q)^2}{\partial w_{ij}^k}$$

$$\boxed{\nabla E_{ij}^k = \frac{2}{\#L} \cdot \sum_{q=0}^{\#L} (O_q - t_q) \frac{\partial a_q^L}{\partial w_{ij}^k}} \tag{1}$$

Given that we usually choose a g with known derivatives

our remaining unknown is $\frac{\partial a_q^l}{\partial w_{ij}^k}$ with $l = L$.

As it can be anticipated we will need to analyse $\frac{\partial a_q^l}{\partial w_{ij}^k}$ for $\begin{matrix} k = 0, \dots, L-1 \\ l = 0, \dots, L \end{matrix}$

Let's proceed our analysis by cases:

If $l = 0 \Rightarrow \frac{\partial a_q^0}{\partial w_{ij}^k} = 0 \quad (a_q^0 \equiv 0)$

Else if $q = \#l$ (bias neuron) $\Rightarrow \frac{\partial a_q^l}{\partial w_{ij}^k} = 0 \quad (a_{\#l}^l \equiv 1)$

Else if $k \geq l \Rightarrow \frac{\partial a_q^l}{\partial w_{ij}^k} = 0$ (w_{ij}^k is posterior to a_q^l in the network)

Else if $k = l-1 \Rightarrow \frac{\partial a_q^l}{\partial w_{ij}^k} = \underbrace{\frac{\partial g(h_q^l)}{\partial h_q^l}}_{g'(h_q^l)} \cdot \frac{\partial h_q^l}{\partial w_{ij}^k}$

And we have

$$\frac{\partial h_q^l}{\partial w_{ij}^k} = \frac{\partial \sum_{r=0}^{\#l-1} a_r^{l-1} w_{rq}^{l-1}}{\partial w_{ij}^{l-1}} = \cancelto{0}{\frac{\partial \sum_{r=0, r \neq i}^{\#k} a_r^k w_{rq}^k}{\partial w_{ij}^k}} + \frac{\partial a_i^k w_{iq}^k}{\partial w_{ij}^k} \quad \text{therefore:}$$

If $j = q \Rightarrow \frac{\partial a_q^l}{\partial w_{ij}^k} = g'(h_q^l) \cdot a_i^k$

If $j \neq q \Rightarrow \frac{\partial a_q^l}{\partial w_{ij}^k} = g'(h_q^l) \cdot 0 = 0$

Else if $k < l-1 \Rightarrow \frac{\partial h_q^l}{\partial w_{ij}^k} = \frac{\partial \sum_{r=0}^{\#l-1} a_r^{l-1} w_{rq}^{l-1}}{\partial w_{ij}^k} = \sum_{r=0}^{\#l-1} w_{rq}^{l-1} \frac{\partial a_r^{l-1}}{\partial w_{ij}^k}$ * (bias term is 0)

$$\Rightarrow \frac{\partial a_q^l}{\partial w_{ij}^k} = g'(h_q^l) \cdot \sum_{r=0}^{\#l-1} w_{rq}^{l-1} \frac{\partial a_r^{l-1}}{\partial w_{ij}^k}$$

In conclusion we can now recursively express $\frac{\partial a_q^l}{\partial w_{ij}^k}$:

$$\boxed{\frac{\partial a_q^l}{\partial w_{ij}^k} = g'(h_q^l) \cdot \begin{cases} 0 & \text{if } l = 0 \text{ or } k \geq l \text{ or } q = \#l \text{ or } (k = l-1 \text{ and } j \neq q) \\ a_i^k & \text{if } k = l-1 \text{ and } j = q \\ \sum_{r=0}^{\#l-1} w_{rq}^{l-1} \cdot \frac{\partial a_r^{l-1}}{\partial w_{ij}^k} & \text{else} \end{cases}} \tag{2}$$

Implementation Details

Is important to note that (2) holds for valid values of l and k, i.e: $l, k \in \{0, \ldots, L\}$. The implementation will need to perform checks to ensure l and k are not out of range.

Also, while the cases for $q = \#l$ and $k \geq l$ are theoretically correct, when doing backpropagation those cases should not be reached, therefore we will assert so as well.

It will be also important to add an extra "Else if $k = l-2$" case after the "Else if $k = l-1$" guard. While this case is not necessary for completeness sake, it will provide a significative performance gain for our initial recursive and slow implementation.

Else if $k = l-2$:

$$\frac{\partial h_q^l}{\partial w_{ij}^k} = \frac{\partial \sum_{r=0}^{\#l-1} a_r^{l-1} w_{rq}^{l-1}}{\partial w_{ij}^{l-2}} = \sum_{r=0}^{\#l-1} w_{rq}^{l-1} \cdot \frac{\partial a_r^{l-1}}{\partial w_{ij}^{l-2}}$$

and because of our previous case considering r and j this is

$$= w_{jq}^{l-1} \cdot g'(h_j^{l-1}) \cdot a_i^k \quad \text{therefore}$$

$$\boxed{k = l-2 \Rightarrow \frac{\partial a_q^l}{\partial w_{ij}^k} = g'(h_q^l) \cdot w_{jq}^{l-1} \cdot g'(h_j^{l-1}) \cdot a_i^k} \quad (2')$$

Matricization

Now let's translate our equations to matrices so that we can use numpy matrix operations and get a significant speed boost.

Let's define:

$$a^\ell := \left[a_1 \cdots a_{\#\ell} \right] = \left[a_i^\ell \right]_i \quad \text{w/o bias } i = 0, \dots, \#k$$

$$w^k := \begin{bmatrix} w_{11}^k & \cdots & w_{1,\#k+1}^k \\ \vdots & \ddots & \vdots \\ w_{\#k,1}^k & \cdots & w_{\#k,\#k+1}^k \end{bmatrix} = \left[w_{ij}^k \right]_{ij}$$

$k = 0, \dots, L-1$

$i = 0, \dots, (\#k)+1 \rightarrow \text{bias}$

$j = 0, \dots, \#(k+1)$

$$\nabla E^k := \begin{bmatrix} \frac{\partial E}{\partial w_{11}^k} & \cdots & \frac{\partial E}{\partial w_{1,\#k+1}^k} \\ \vdots & \ddots & \vdots \\ \frac{\partial E}{\partial w_{\#k,1}^k} & \cdots & \frac{\partial E}{\partial w_{\#k,\#k+1}^k} \end{bmatrix} \overset{(1)}{=} \left[\frac{2}{\#L} \cdot \sum_{q=0}^{\#L} (o_q - t_q) \frac{\partial a_q^L}{\partial w_{ij}^k} \right]_{ij}$$

$$A_k^{\ell q} := \left[\frac{\partial a_q^\ell}{\partial w_{ij}^k} \right]_{ij} \Rightarrow \boxed{\nabla E^k = \frac{2}{\#L} \cdot \sum_{q=0}^{\#L} (o_q - t_q) A_k^{Lq}} \quad (3)$$

Now let's use (2) to obtain an expression by cases for $A_k^{\ell q}$

If $\ell = k+1 \Rightarrow A_k^{\ell q} = \left[\frac{\partial a_q^{k+1}}{\partial w_{ij}^k} \right]_{ij} \overset{(2)}{=} g'(h_q^\ell) \left[\begin{cases} 0 & \text{if } j \neq q \\ a_i^k & \text{if } j = q \end{cases} \right]_{ij}$ (4)

Else if $\ell = k+2 \Rightarrow A_k^{\ell q} = \left[\frac{\partial a_q^{k+2}}{\partial w_{ij}^k} \right]_{ij} \overset{(2')}{=} \left[g'(h_q^\ell)\, w_{jq}^{k+1}\, g'(h_j^{k+1})\, a_i^k \right]_{ij}$ (5)

Else if $\ell > k+1 \Rightarrow A_k^{\ell q} = \left[\frac{\partial a_q^\ell}{\partial w_{ij}^k} \right]_{ij} \overset{(2)}{=} \left[g'(h_q^\ell) \sum_{r=0}^{\#\ell-1} w_{rq}^{\ell-1} \cdot \frac{\partial a_r^{\ell-1}}{\partial w_{ij}^k} \right]_{ij}$ (6)

$$= g'(h_q^\ell) \sum_{r=0}^{\#\ell-1} w_{rq}^{\ell-1} A_k^{\ell-1,r}$$

In summary:

(3) $\nabla E^k = \frac{2}{\#L} \cdot \sum_{q=0}^{\#L} (o_q - t_q) A_k^{Lq}$

(4) $A_k^{k+1,q} = g'(h_q^\ell) \left[\begin{cases} 0 & \text{if } j \neq q \\ a_i^k & \text{if } j = q \end{cases} \right]_{ij}$

(5) $A_k^{k+2,q} = g'(h_q^\ell) \left[w_{jq}^{k+1}\, g'(h_j^{k+1})\, a_i^k \right]_{ij}$

(6) $A_k^{\ell,q} = g'(h_q^\ell) \sum_{r=0}^{\#\ell-1} w_{rq}^{\ell-1} A_k^{\ell-1,r}$ $\quad \ell > k+1$