$$\frac{\partial C_0}{\partial b^{(L)}} = \frac{\partial z^{(L)}}{\partial b^{(L)}} \frac{\partial a^{(L)}}{\partial z^{(L)}} \frac{\partial C_0}{\partial a^{(L)}} = 1 \cdot \text{ReLU}'(z^{(L)}) \cdot 2(a^{(L)} - y)$$

4) $\frac{\partial z^{(L)}}{\partial b^{(L)}} = 1$

5) $\frac{\partial z^{(L)}}{\partial a^{(L-1)}} = w^{(L)}$

$$\frac{\partial C_0}{\partial a^{(L-1)}} = \frac{\partial z^{(L)}}{\partial a^{(L-1)}} \frac{\partial a^{(L)}}{\partial z^{(L)}} \frac{\partial C_0}{\partial a^{(L)}} = w^{(L)} \cdot \text{ReLU}'(z^{(L)}) \cdot 2(a^{(L)} - y)$$

1) We can only change weights & biases:

$$\frac{\partial C_0}{\partial w^{(L)}} = a^{(L-1)} \cdot \text{ReLU}'(z^{(L)}) \cdot 2(a^{(L)} - y)$$

$$\frac{\partial C_0}{\partial b^{(L)}} = \text{ReLU}'(z^{(L)}) \cdot 2(a^{(L)} - y)$$

So we use the negative of the derivative to step towards 0. Further we use step of average over a batch of inputs. } negative of avg

2) We can change weights & biases of last layer to change its activation:

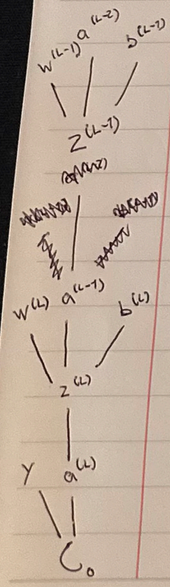
$$\frac{\partial C_0}{\partial w^{(L-1)}} = \frac{\partial z^{(L-1)}}{\partial w^{(L-1)}} \frac{\partial a^{(L-1)}}{\partial z^{(L-1)}} \frac{\partial z^{(L)}}{\partial a^{(L-1)}} \cdots \frac{\partial C_0}{\partial a^{(L)}} = \frac{\partial a^{(L-1)}}{\partial w^{(L-1)}} \frac{\partial C_0}{\partial a^{(L-1)}}$$ ← wrong

$$a^{(L-1)} = \text{ReLU}(z^{(L-1)}), \quad z^{(L-1)} = w^{(L-1)} a^{(L-2)} + b^{(L-1)}$$

$$\frac{\partial a^{(L-1)}}{\partial w^{(L-1)}} = \frac{\partial z^{(L-1)}}{\partial w^{(L-1)}} \frac{\partial a^{(L-1)}}{\partial z^{(L-1)}} = a^{(L-2)} \cdot \text{ReLU}'(z^{(L-1)})$$

stored from earlier

$$\frac{\partial C_0}{\partial w^{(L-1)}} = a^{(L-2)} \cdot \text{ReLU}'(z^{(L-1)}) \cdot w^{(L)} \cdot \text{ReLU}'(z^{(L)}) \cdot 2(a^{(L)} - y)$$

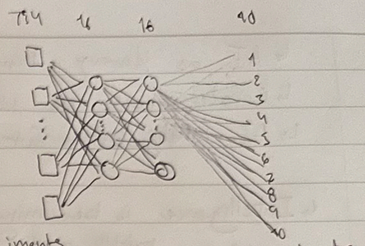


input — ReLU — ReLU — softmax

$754 \times 16 \times 16 \times 10$

$W_0 \& b_0$ → 16×754 $\quad W_1 \& b_1$ → 16×16 $\quad W_2 \& b_2$ → 10×16

maybe more complex is better b/c smooth working



$\vec{a}^{(0)}$ → determined by image inputs

$a^{(0)} = \text{ReLU}(W_0 \vec{a}^{(0)} + b_0)$ → * we could batch norm? (no later) ✓

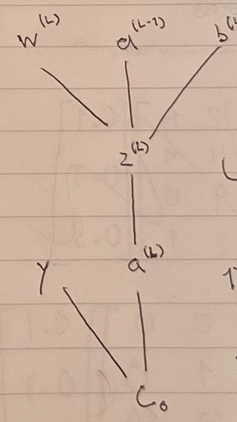
$a^{(1)} = \text{ReLU}(W_1 a^{(0)} + b_1)$

output $a^{(2)} = \text{softmax}(W_2 a^{(1)} + b_2)$ (where y is desired output)

Loss $C_0 = \sum [(y - a^{(2)})^2]^T [1]$ = 1x1 cost matrix

↳ $\text{ReLU}(x) = \begin{cases} 0 & \text{if } \leq 0 \\ x & \text{if } > 0 \end{cases}$, $\text{ReLU}' = \begin{cases} 0 & \text{if } \leq 0 \\ 1 & \text{if } > 0 \end{cases}$

$z^{(L)} = w^{(L)} a^{(L-1)} + b^{(L)}$, $a^{(L)} = \text{ReLU}(z^{(L)})$, $C_0 = (a^{(L)} - y)^2$



↳ We want sensitivity of C_0 to change in 1) $w^{(L)}$ 2) $b^{(L)}$ 3) $a^{(L-1)}$

partial derivative

↳ $\frac{\partial C_0}{\partial w^{(L)}} = \frac{\partial z^{(L)}}{\partial w^{(L)}} \frac{\partial a^{(L)}}{\partial z^{(L)}} \frac{\partial C_0}{\partial a^{(L)}}$

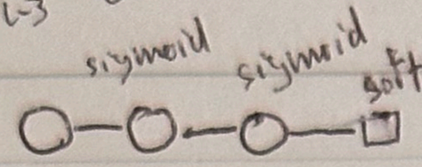
1) $\frac{\partial C_0}{\partial a^{(L)}} = 2(a^{(L)} - y)$ 2) $\frac{\partial a^{(L)}}{\partial z^{(L)}} = \text{ReLU}'(z^{(L)})$

3) $\frac{\partial z^{(L)}}{\partial w^{(L)}} = a^{(L-1)}$

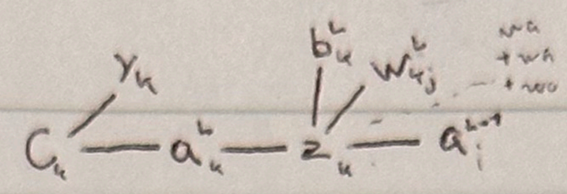
↳ Thus $\frac{\partial C_0}{\partial w^{(L)}} = a^{(L-1)} \text{ReLU}'(z^{(L)}) 2(a^{(L)} - y)$

L-3 L-2 L-1 L

sigmoid sigmoid softmax



C: is cross entropy loss



$\text{sigmoid} = \sigma_1(z_k^L) = \frac{1}{1+e^{(-z_k^L)}}$

↳ $a_k^L = \sigma(z_k^L)$

$\text{sigmoid}' = \sigma_1(z_k^L)(1-\sigma_1(z_k^L)) = a_k^L(1-a_k^L)$

$\text{softmax} = \sigma_2(z_k^L) = \frac{e^{z_k^L}}{\sum_{k=1}^{K} e^{z_k^L}}$

$\text{softmax}' = \sigma_2'(z_k^L)$ ——— ?

todo

Softmax

(1) ~~$\frac{\partial}{\partial z_k^L}\left(\frac{e^{z_k^L}}{\sum_{k=1}^{K} e^{z_k^L}}\right)$~~

(2) $a_k^L = \frac{e^{z_k^L}}{\sum_{k=1}^{K} e^{z_k^L}}$

(3) ~~$\frac{\partial a_k^L}{\partial z_k^L} \frac{\partial z_k^L}{\partial w_{kj}^L}$~~ $\frac{\partial C}{\partial w_{kj}^L} = \left(\sum_{i \text{ output neurons}} \frac{\partial C}{\partial a_i^L}\frac{\partial a_i^L}{\partial z_j^L}\right)\frac{\partial z_j^L}{\partial w_{kj}^{(L)}}$

(4) $\sum_{i \text{ output neurons}} \frac{\partial C}{\partial a_i^L}\frac{\partial a_i^L}{\partial z_j^L} = \sum_{i \text{ neuron}, i \neq j} a_j t_i = a_k - t_k$

(5) $\frac{\partial C}{\partial b_k^L} = \frac{\partial C}{\partial z_k^L}\frac{\partial z_k^L}{\partial b_k^L} = (a_k^L - t_k^L)\cdot 1 = (a_k^{(L)} - t_k^{(L)})$ ← whatever; bias

(6) $\frac{\partial C}{\partial w_{kj}^L} = \frac{\partial z_k^L}{\partial w_{kj}^L}\frac{\partial C}{\partial z_k^L} = a_j^{(L-1)} \cdot (a_k^{(L)} - t_k^{(L)})$ weight

(7) $\frac{\partial C}{\partial a_j^{(L-1)}} = \sum_{k=1}^{K} w_{kj}^{(L)} \cdot (a_k^{(L)} - t_k^{(L)})$ ← $\frac{\partial z_k^L}{\partial a_j^{L-1}}\frac{\partial C}{\partial z_k^L}$; forward

(8) $C = -\sum_{k=1}^{K} t_k \log(a_k^{(L)})$ → $\frac{\partial C_0}{\partial a_k^L} = \frac{-t_k}{a_k^{(L)}}$

where t_k is target and a_k is hypothesis

$\frac{\partial C}{\partial z}$?

Sigmoid

(1) $a_j^{(L-1)} = \sigma_1(z_j^{(L-1)}) = \frac{1}{1+e^{(-z_j^{(L-1)})}}$

★ On input layer use input values.

★ Not done on frontmost layer

(2) $\frac{\partial C}{\partial b_j^{(L-1)}} = \frac{\partial z_j^{(L-1)}}{\partial b_j^{(L-1)}}\frac{\partial a_j^{(L-1)}}{\partial z_j^{(L-1)}}\frac{\partial C}{\partial a_j^{(L-1)}} = 1 \cdot a_j^{(L-1)}(1-a_j^{(L-1)})\left[\frac{\partial C}{\partial a_j^{(L-1)}}\right]$ bias

(3) $\frac{\partial C}{\partial w_{jh}^{(L-1)}} = \frac{\partial z_j^{(L-1)}}{\partial w_{jh}^{(L-1)}}\frac{\partial a_j^{(L-1)}}{\partial z_j^{(L-1)}}\frac{\partial C}{\partial a_j^{(L-1)}} = a_h^{(L-2)} \cdot a_j^{(L-1)}(1-a_j^{(L-1)})\left[\frac{\partial C}{\partial a_j^{(L-1)}}\right]$ weight

(4) $\frac{\partial C}{\partial a_h^{(L-2)}} = \sum_{j=1}^{J}\frac{\partial z_j^{(L-1)}}{\partial a_h^{(L-2)}}\frac{\partial a_j^{(L-1)}}{\partial z_j^{(L-1)}}\frac{\partial C}{\partial a_j^{(L-1)}} = \sum_{j=1}^{J} w_{jh}^{(L-1)} \cdot a_j^{(L-1)}(1-a_j^{(L-1)})\left[\frac{\partial C}{\partial a_j^{(L-1)}}\right]$ forward

$$C_0 = \sum_{k=0}^{K} (a_k^{(L)} - y_k)^2 \quad , \quad a_k^{(L)} = \text{softmax}(z_k^{(L)}) = \frac{e^{z_k^{(L)}}}{\sum_{k=1}^{K} e^{z_k^{(L)}}}$$

$$\therefore C_0 \left(\sum_{n=1}^{K} \left(\frac{e^{z_k^{(L)}}}{\sum_{k=1}^{K} e^{z_k^{(L)}}} - y_k \right)^2 \right) \Bigg\} C_0 \text{ in terms of } z_k^{(L)} =$$

$$\frac{\partial C_0}{\partial w_{jk}^{(L)}} = \frac{\partial z_k^{(L)}}{\partial w_{jk}^{(L)}} \frac{\partial a_k^{(L)}}{\partial z_k^{(L)}} \frac{\partial C_0}{\partial a_k^{(L)}} = a_j^{(L-1)} \cdot \text{softmax}(z_k^{(L)}) \cdot (1 - \text{softmax}(z_k^{(L)}) \cdot 2(a_k^{(L)} - y_k)$$

$$\text{or} = \frac{\partial z_k^{(L)}}{\partial w_{kj}^{(L)}} \cdot \frac{\partial C_0}{\partial z_k^{(L)}} = a_j^{(L-1)} \cdot \Big[\underbrace{\sigma(z_k^{(L)})(1-\sigma(z_k^{(L)}) \cdot 2(a_k^{(L)} - y_k)}_{\text{activation}} + \ldots + \underbrace{-\sigma(z_n^{(L)}) \sigma(z_k^{(L)}) 2(a_n^{(L)} - y_n)}_{\text{activation}} \Big]$$

★ right

1) Forward pass & store activations in vectors for each layer } after activation function
2) Calculate derivative of loss w.r.t. ~~expected~~ each neuron & store in vector $(2(a_k^{(L)} - y_k))$ } similar to $O(n)$ step
2.1) Mult by neuron's activation (this is ~~$a_k^{(L)} \cdot 2(a_k^{(L)} - y_k)$~~) store in vector
2.2) Now for each neuron mult its activation thing by $(1 - a_k^{(L)})$ add ~~$-a_k^{(L)}$~~ · nothing of all other vectors $\left(\frac{\partial C_0}{\partial z_k^{(L)}}\right)$ $O(n)$
3) do $\frac{\partial C_0}{\partial w_{kj}^{(L)}} = a_j^{(L-1)} \cdot$ (activation thing) (store gradient +=)

first layer

4) do $\frac{\partial C_0}{\partial b_k^{(L)}} = \frac{\partial z_k^{(L)}}{\partial b_k^{(L)}} \cdot \frac{\partial C_0}{\partial z_k^{(L)}} = 1 \cdot$ (activation thing) store bias gradient += $O(n)$ $O(n \times m)$
3) Contribute to activation thing of next layer $\frac{\partial C_0}{\partial a_j^{(L-1)}} = \frac{\partial z_k^{(L)}}{\partial a_j^{(L-1)}} \frac{\partial C_0}{\partial z_k^{(L)}} = w_{kj}^{(L)} \cdot$ (act thing) (store in same array NOT)

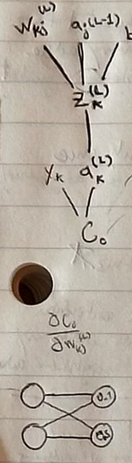
store activation array for

6) same thing 1) bias 2) weights 3) contribute to next
7) on last one no #3, make sure to use clean/formatted arrays

1 × negative activation of self 2 add $(1 - a_k^{(L)})$ doni 1

repurpose to $\frac{\partial C_0}{\partial a_k^{(L)}}$

activation thing 1 ↗



$e(\Sigma - e)$

Activation Functions:

$$\text{ReLU} = \begin{cases} 0 & \text{if } \leq 0 \\ x & \text{if } > 0 \end{cases} \quad , \quad \text{ReLU}' = \begin{cases} 0 & \text{if } \leq 0 \\ 1 & \text{if } > 0 \end{cases}$$

$$\text{softmax} = \frac{e^{(z_k^{(L)})}}{h^{(L)}} \quad , \quad \text{softmax}(z_n^{(L)}) = \partial(z_k^{(L)}) = \frac{e^{z_k^{(L)}}}{\sum e^{z_k^{(L)}}}$$

$$h^{(L)} = \sum_{k=0}^{n=K-1} e^{(z_k^{(L)})} = e^{z_0^{(L)}} + e^{z_1^{(L)}} + e^{z_2^{(L)}} + \ldots + e^{z_K^{(L)}}$$

$$\frac{d}{dx} \frac{f(x)}{g(x)} = \frac{f'(x)g(x) - f\cancel{o}g'(x)}{(g(x))^2}$$

$$a_k^{(L)} = \text{softmax}(z_k^{(L)}) \quad , \quad \frac{\partial a_k^{(L)}}{\partial z_k^{(L)}} = \frac{\partial}{\partial z_k^{(L)}} \frac{e^{z_k^{(L)}}}{e^{z_0^{(L)}} + e^{z_1^{(L)}} + \ldots + e^{z_K^{(L)}}} = \frac{e^{z_k^{(L)}} \sum_{k=1}^{K} e^{z_k^{(L)}} - e^{z_k^{(L)}} e^{z_k^{(L)}}}{\left[\sum_{k=1}^{K} e^{z_k^{(L)}} \right]^2}$$

$$\frac{\partial C_0}{\partial w_{kj}^{(L)}} = \frac{\partial z_k^{(L)}}{\partial w_{kj}^{(L)}} \frac{\partial a_k^{(L)}}{\partial z_k^{(L)}} \frac{\partial C_0}{\partial a_k^{(L)}} = \frac{e^{z_k^{(L)}}}{\sum_{k=1}^{K} e^{z_k^{(L)}}} \left(1 - \frac{e^{z_k^{(L)}}}{\sum_{i=1}^{K} e^{z_k^{(L)}}} \right) \cdot a_j^{(L-1)} \cdot 2(a_k^{(L)} - y) = \frac{e^{z_k^{(L)}}}{\sum_{k=1}^{K} e^{z_k^{(L)}}} \left(1 - \frac{e^{z_k^{(L)}}}{\sum_{k=1}^{K} e^{z_k^{(L)}}} \right)$$

same

$$= \partial(z_k^{(L)}) (1 - \partial(z_k^{(L)}))$$

↳ I need to consider overall cost of changing weights; this means I need to sum & check others

where: $h \neq k$

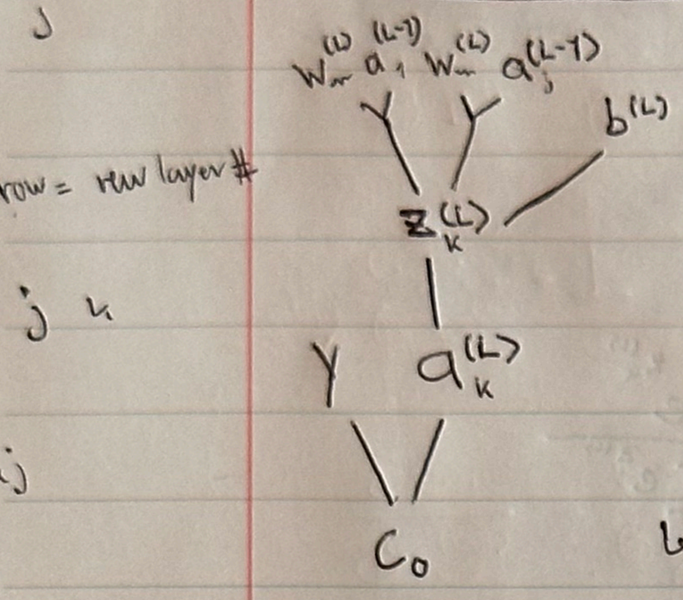
$$\frac{\partial C_0}{\partial z_{kj}^{(L)}} = \frac{\partial}{\partial z_{kj}^{(L)}} \left[\frac{e^{z_h^{(L)}}}{\sum_{k=1}^{K} e^{z_k^{(L)}}} \right] = \frac{\partial}{\partial z_k^{(L)}} e^{z_h^{(L)}} \left[e^{z_1^{(L)}} + \ldots + e^{z_k^{(L)}} \right]^{-1} =$$

$$\frac{e^{z_h^{(L)}}}{e^{z_1^{(L)}} + \ldots + e^{z_k^{(L)}}} = \frac{0 \cdot \sum_{k=1}^{K} e^{z_k^{(L)}} - e^{z_h^{(L)}} e^{z_k^{(L)}}}{\left(\sum_{k=1}^{K} e^{z_k^{(L)}} \right)^2} = -\sigma(e^{z_h^{(L)}}) \sigma(e^{z_k^{(L)}})$$

W is $k \times j$ matrix

$W^{(L)}_{jk}$ → connection, edge in Layer L-1
↳ current node in layer L

Additional Neurons:



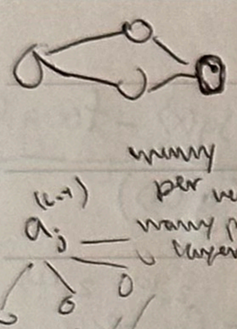
$$z^{(L)}_k = W^{(L)}_{jk1} a^{(L-1)}_1 + W^{(L)}_{jk2} a^{(L-1)}_2 + \dots + W^{(L)}_{jkj} a^{(L-1)}_j + b^{(L)}_k$$

$$C_0 = \sum \left[(a^{(L)}_k - y_k)^2 \right] \longrightarrow$$

$$a^{(L)}_k = \text{ReLU}(z^{(L)}_k) \quad \left\{ \text{ReLU}' = \begin{cases} 0 & \text{if } \leq 0 \\ 1 & \text{if } > 0 \end{cases} \right.$$

} flip ifks

↳ Gradient is avg, after ~~are~~ sums



many per node, many per layer

$$\frac{\partial C_0}{\partial w^{(L)}_{kj}} = \frac{\partial z^{(L)}_k}{\partial w^{(L)}_{kj}} \frac{\partial a^{(L)}_k}{\partial z^{(L)}_k} \frac{\partial C_0}{\partial a^{(L)}_k} = a^{(L-1)}_j \cdot \text{ReLU}'(z^{(L)}_k) \cdot 2(a^{(L)}_k - y_k)$$

} repeated for every edge
↳ store as gradient!

1 per node, many per layer

$$\frac{\partial C_0}{\partial b^{(L)}_k} = \frac{\partial z^{(L)}_k}{\partial b^{(L)}_k} \frac{\partial a^{(L)}_k}{\partial z^{(L)}_k} \frac{\partial C_0}{\partial a^{(L)}_k} = 1 \cdot \text{ReLU}'(z^{(L)}_k) \cdot 2(a^{(L)}_k - y_k)$$

} only once
↳ store as gradient!

h i j k

many per node, many per layer

$$\frac{\partial C_0}{\partial a^{(L-1)}_j} = \frac{\partial z^{(L)}_k}{\partial a^{(L-1)}_j} \frac{\partial a^{(L)}_k}{\partial z^{(L)}_k} \frac{\partial C_0}{\partial a^{(L)}_k} = w^{(L)}_{kj} \cdot \text{ReLU}'(z^{(L)}_k) \cdot 2(a^{(L)}_k - y_k)$$

} only for this ★ node!
for real activation sum all weights w/ other neighbor nodes

$$\frac{\partial C_0}{\partial w^{(L-1)}_{ji}} = \frac{\partial z^{(L-1)}_j}{\partial w^{(L-1)}_{ji}} \frac{\partial a^{(L-1)}_j}{\partial z^{(L-1)}_j} \frac{\partial C_0}{\partial a^{(L-1)}_j} = a^{(L-2)}_i \cdot \text{ReLU}'(z^{(L-1)}_j) \cdot \boxed{\frac{\partial C_0}{\partial a^{(L-1)}_j}}$$

sum

} ofc repeat for all edges

$$\frac{\partial C_0}{\partial b^{(L-1)}_j} = \frac{\partial z^{(L-1)}_j}{\partial b^{(L-1)}_j} \frac{\partial a^{(L-1)}_j}{\partial z^{(L-1)}_j} \frac{\partial C_0}{\partial a^{(L-1)}_j} = 1 \cdot \text{ReLU}'(z^{(L-1)}_j) \cdot \boxed{\frac{\partial C_0}{\partial a^{(L-1)}_j}}$$

$$\frac{\partial C_0}{\partial a^{(L-2)}_i} = \frac{\partial z^{(L-1)}_j}{\partial a^{(L-2)}_i} \frac{\partial a^{(L-1)}_j}{\partial z^{(L-1)}_j} \frac{\partial C_0}{a^{(L-1)}_j} = w^{(L-1)}_{ji} \cdot \text{ReLU}'(z^{(L-1)}_j) \cdot \boxed{\frac{\partial C_0}{\partial a^{(L-1)}_j}}$$

↳ For last layer (frontmost) - don't calc prev layer's influence
↳ At end we have gradient for biases & weights; these accumulate & divide at end by batch size