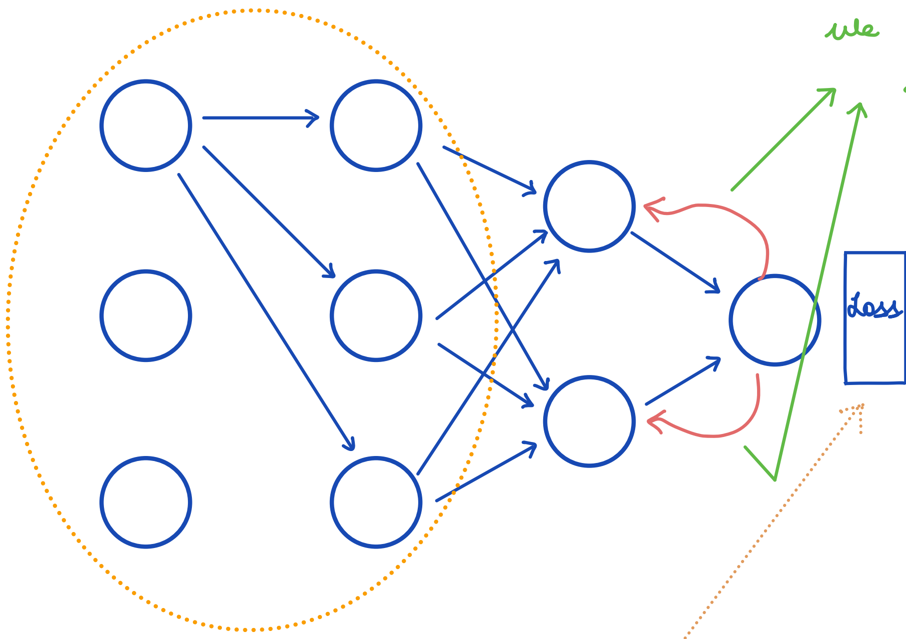
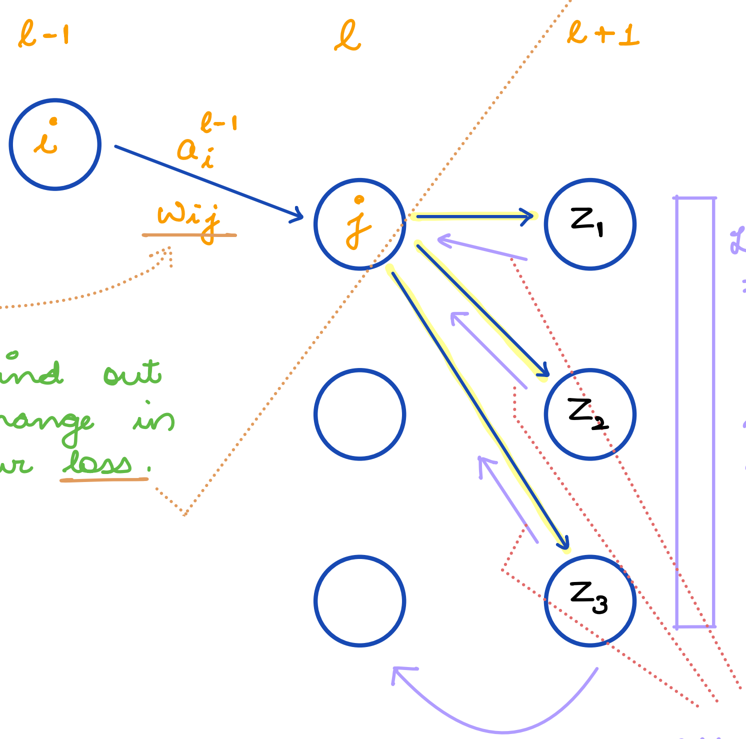
Case 2 — Hidden Layer

we want to figure out how weights associated with this will get updated.



we have already seen how these weights will get updated.



we want to find out how small change in w_{ij} affects our loss.

Let us assume that we know error upto this layer. we will see how to back propagate this error to previous layers.

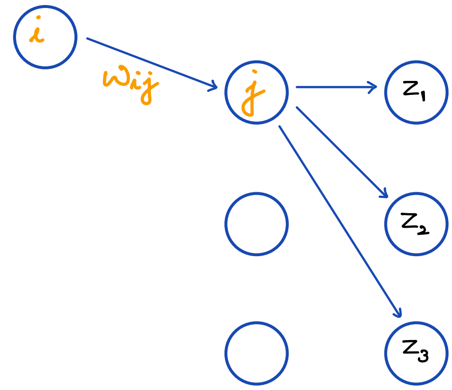
All these 3 neurons will backpropagate some loss.

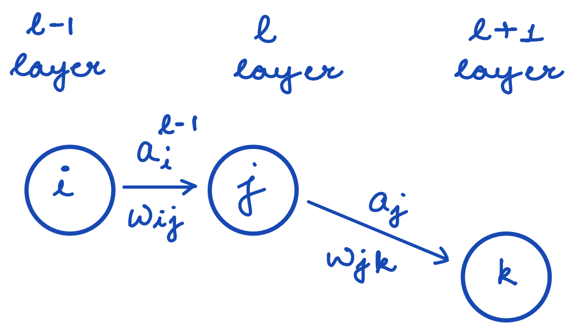
$$\frac{\partial L}{\partial w_{ij}^{l}} = \sum_k \underbrace{\frac{\partial L}{\partial z_k^{l+1}}}_{\delta_k^{l+1}} \cdot \underset{1}{\frac{\partial z_k^{l+1}}{\partial a_j}} \cdot \underset{2}{\frac{\partial a_j}{\partial z_j}} \cdot \underset{3}{\frac{\partial z_j}{\partial w_{ij}}}$$

$$w_{ij} \to z_j \to a_j \to \begin{matrix} \vdots \\ z_k \\ \vdots \end{matrix} \to L$$

(affect loss)

how loss changes according to w_{ij} will depend on how loss changes according to z_k * how z_k of $l+1$ changes according to a_j * how a_j changes according to z_j * how z_k changes according to w_{ij}.





$$z_k^{l+1} = \sum_j w_{jk} a_j^l$$

① $\dfrac{\partial z_k^{l+1}}{\partial a_j^l}$

$$z_k^{l+1} = \sum_j w_{jk} \cdot a_j^l$$

$$\frac{\partial z_k^{l+1}}{\partial a_j^l} = w_{jk}$$

② $\dfrac{\partial a_j}{\partial z_j}$

$$a_j = \sigma(z_j)$$

$$\frac{\partial a_j}{\partial z_j} = \sigma'(z_j)$$

③ $\dfrac{\partial z_j}{\partial w_{ij}}$

$$z_j = \sum_i w_{ij} \, a_i^{l-1}$$

$$\frac{\partial z_j}{\partial w_{ij}} = a_i^{l-1}$$

$$\frac{\partial L}{\partial w_{ij}^{l}} = \sum_k \frac{\partial L}{\partial z_k^{l+1}} \cdot \frac{\partial z_k^{l+1}}{\partial a_j} \cdot \frac{\partial a_j}{\partial z_j} \cdot \frac{\partial z_j}{\partial w_{i,j}}$$

$$= \boxed{\sum_k \left(\delta_k^{l+1} \cdot w_{jk}\right) \cdot \sigma'(z_j^{l})} \cdot a_i^{l-1}$$

$$\delta_j^{l} = \sum_k \left(\delta_k \cdot w_{jk}\right) \cdot \sigma'(z_j^{l})$$

$$\frac{\partial L}{\partial w_{ij}} = \delta_j^{l} \cdot a_i^{l-1}$$

Bias Update Rule

$$\frac{\partial L}{\partial b} = \underbrace{\sum_k \frac{\partial L}{\partial z_k^{l+1}} \cdot \frac{\partial z_k^{l+1}}{\partial a_j} \cdot \frac{\partial a_j}{\partial z_j}}_{\delta_j^{l}} \cdot \frac{\partial z_j}{\partial b}$$

$$\frac{\partial L}{\partial b} = \delta_j^{l} \cdot 1$$

$$\delta_j^{l} = \sum_k \left(\delta_k^{l+1} \cdot w_{jk}^{l+1}\right) \odot \sigma'(z_j^{l})$$

Final Result for Hidden Neuron:

$$\frac{\partial L}{\partial w_{ij}} = \delta_j^{l} \, a_i^{l-1}$$

$$\frac{\partial L}{\partial b} = \delta_j^{l}$$

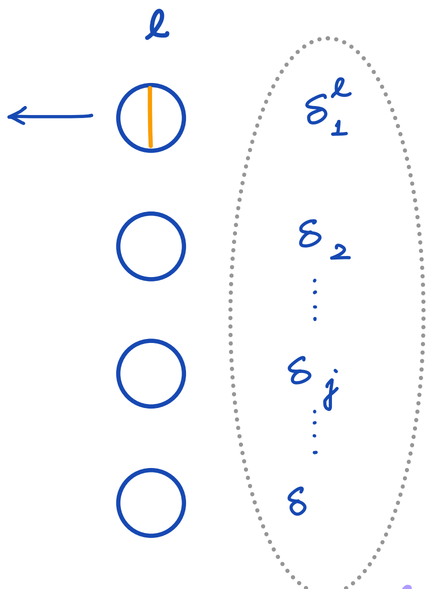
$$\delta_j^{l} = \sum_k \left(w_{jk}^{l+1} \, \delta_k^{l+1} \right) \odot \sigma'(z_j^{l})$$

Results are same for output and hidden layer, only the value of δ_j^{l} will change.

Matrix Representation:

While writing code we won't use for loop to iterate over all neuron, instead we will take matrix and vector and do dot product. Like this we can do parallel computation

δ_j^{l} = loss associated with one unit



These neuron have some associated loss which they will propagate back

This entire vector is called $\delta^{L}_{l \times 1}$ and has dimension of $l \times 1$

L^{th} layer and l units

l = no. of hidden layer

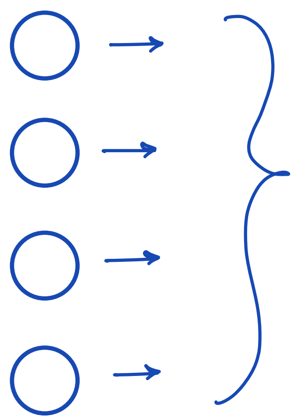
Earlier we calculated $\frac{\partial L}{\partial w_{ij}}$, now we want to calculate $\frac{\partial L}{\partial w^L}$?

$$\underbrace{\frac{\partial L}{\partial w^L}}_{(l-1,\,l)} = \underbrace{\underbrace{a^{L-1}}_{(l-1,\,1)} \cdot \underbrace{\big(\underbrace{\delta^L}_{(l,\,1)}\big)^T}_{(1,\,l)}}_{(l-1,\,l)}$$

$$w = \begin{bmatrix} \quad & \quad & \quad \end{bmatrix}_{(l-1,\,l)}$$

L-1 layer

for activation dimension:



l-1 activations vector of (l-1, 1)

why the above formula is correct ?

activations of previous layer →

$$\begin{bmatrix} a_1^{l-1} \\ a_2^{l-1} \\ a_3^{l-1} \\ \vdots \end{bmatrix} \begin{bmatrix} \delta_1^l & \delta_2^l & \delta_3^l & \cdots & \delta_l^l \end{bmatrix}$$

$$\begin{bmatrix} a_1^{l-1}\delta_1^l & a_1^{l-1}\delta_2^l & a_3^{l-1}\delta_3^l \\ a_2^{l-1}\delta_1^l & a_2^{l-1}\delta_2^l & \cdots \\ \vdots & & \end{bmatrix}$$

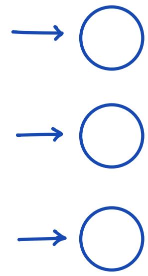
Each element is $\frac{\partial L}{\partial w_{ij}}$

$$\frac{\partial L}{\partial w^L} = a^{L-1} \cdot (\delta^L)^T$$

This formula is true both for input layer and output layer

For biases, $\frac{\partial L}{\partial b_j^l} = \delta_j^l$

$$\underbrace{\frac{\partial L}{\partial b^l}}_{\text{for entire bias matrix}} = \delta^L$$



$$b = b - \eta \frac{\partial L}{\partial b}$$

$$\underbrace{\begin{bmatrix} b_1^l \\ b_2^l \\ b_3^l \end{bmatrix} \qquad \frac{\partial L}{\partial b} = \begin{bmatrix} \delta_1^L \\ \delta_2^L \\ \delta_3^L \end{bmatrix}}_{\text{subtract}}$$

We have seen how to compute $\frac{\partial L}{\partial w^L}$ and $\frac{\partial L}{\partial b^l}$

vector / matrix notation

$$\frac{\partial L}{\partial w^L} = \underbrace{a^{l-1} (\delta^l)^T}_{\text{matrix}}$$

vector

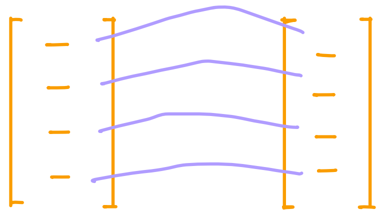
$$\frac{\partial L}{\partial b^l} = \delta^L$$

vector

true for both output and hidden layer.

Element wise Product (it scales up $a^L - y^L$ vector)

$$\delta^L = \underbrace{(a^L - y^L)}_{\text{vector}} \odot \underbrace{\sigma'(z^l)}_{\text{vector}}$$



You find out the difference between prediction and actual value and you just scale that difference by multiplying with derivative of <u>activation $f x^n$</u> for a given input.

↳ z^l

$$\left.\delta^l = \left(w^{l+1}\,\delta^{(l+1)}\right) \odot \sigma'(z^l)\ \right\} \text{ for hidden layer}$$