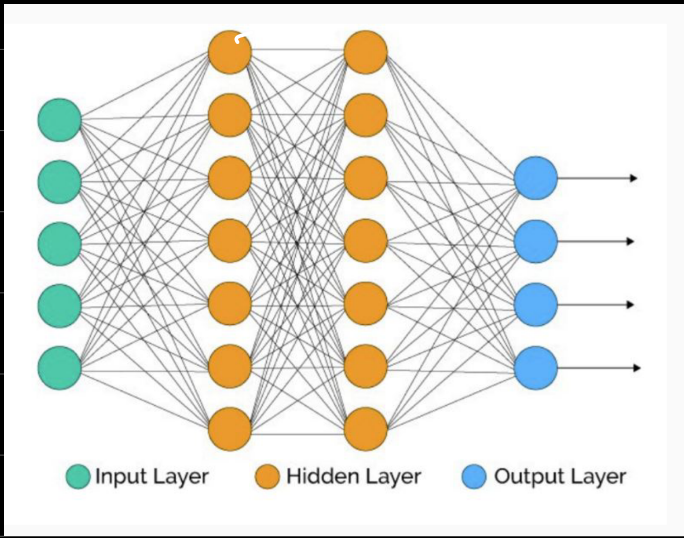
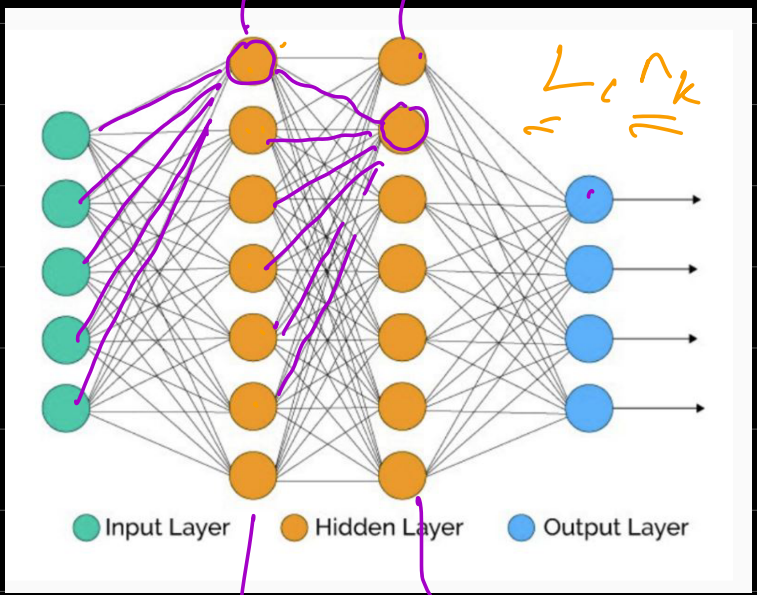


MLP → Multi Layer perceptrons

ANN → Artificial Neural Network

↳ Feed forward Neural Network

↳ Fully Connected Feed Forward Neural Network



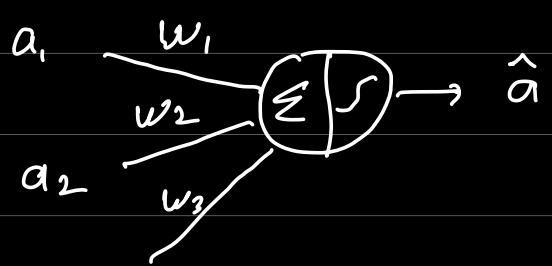
INPUT → OUTPUT SIDE

First hidden layer

Second hidden layer

Neuron - Computations

①



→ activations

$z = \sum w_i a_i$

$\hat{a} = g(z)$

Sigmoid → $\sigma(z)$

Relu → Relu(z)

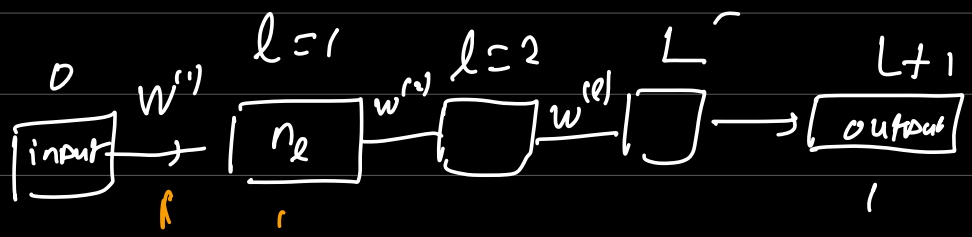
Forward pass

General case → Arbitrary input vector (multiple rows in input)

→ Arbitrary no of layers

→ Arbitrary no of neurons in each layer

→ No of output based on problem



n_l → number of neurons in each layer

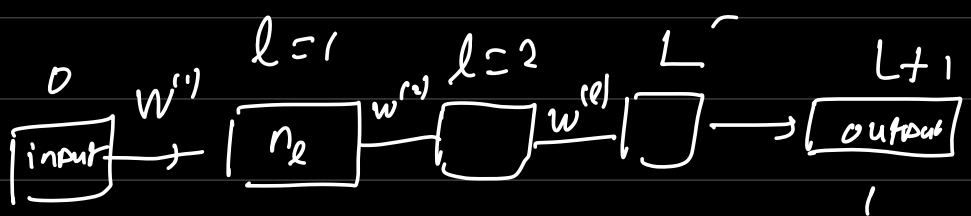
n → is the number of input features

m → Total number of data points

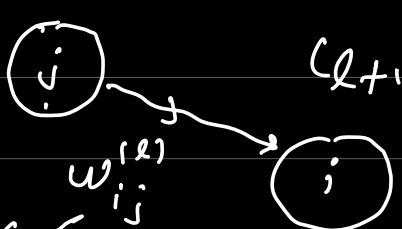
input → $(x_1, x_2) \dots (x_1 \dots x_n)$

$$X = \rightarrow \begin{pmatrix} x_1^{(1)} & x_2^{(1)} & \dots & x_n^{(1)} \\ x_1^{(2)} & x_2^{(2)} & \dots & x_n^{(2)} \\ \vdots & & & \\ x_1^{(m)} & x_2^{(m)} & \dots & x_n^{(m)} \end{pmatrix}$$

$m \times n \quad . \quad X_{m \times n}$



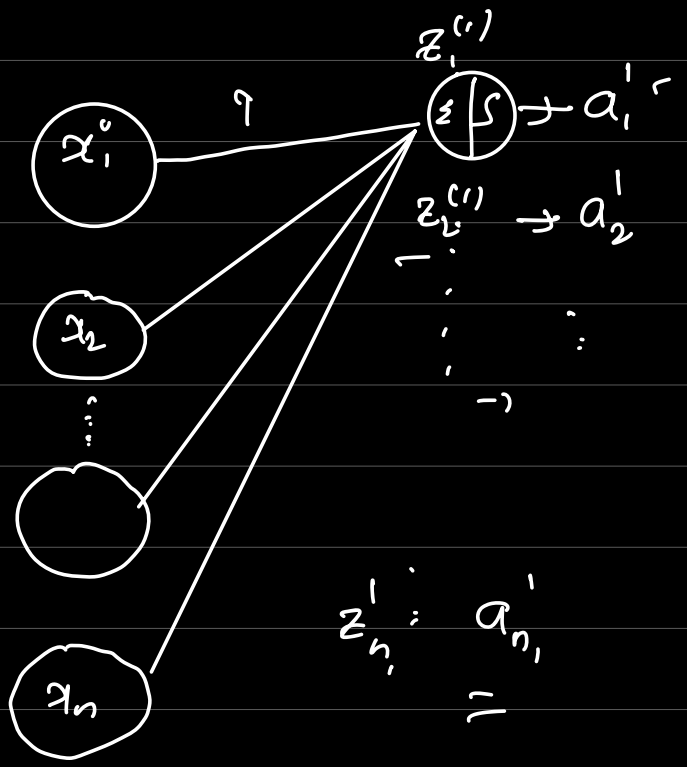
l^{th} layer $\quad 1 \dots n_l$



from l l^{th} layer

$W_{ij}^{(l)}$

TO i^{th} neuron in $(l+1)$ $\quad$ FROM j^{th} n in l



$a_1 = x_1, \quad a_2 = x_2 \cdots \quad a_n = x_n$

~~For~~ For one row of X

$$w_{11} x_1 + w_{12} x_2 + w_{13} x_3 \cdots + w_{1n} x_n = z_1^{(1)}$$

—— 9

$$w_{11}^{(1)} a_1 + w_{12}^{(1)} a_2 + w_{13}^{(1)} a_3 + \cdots + w_{1n}^{(1)} a_n = z_1^{(1)}$$

$$w_{21}^{(1)} a_1 + w_{22}^{(1)} a_2 + w_{23}^{(1)} a_3 + \cdots + w_{2n}^{(1)} a_n = z_2^{(1)}$$

$$\vdots$$

$$w_{n_1,1}^{(1)} a_1 + w_{n_1,2}^{(1)} a_2 + w_{n_1,3}^{(1)} a_3 + \cdots + w_{n_1,n}^{(1)} a_n = z_{n_1}^{(1)}$$

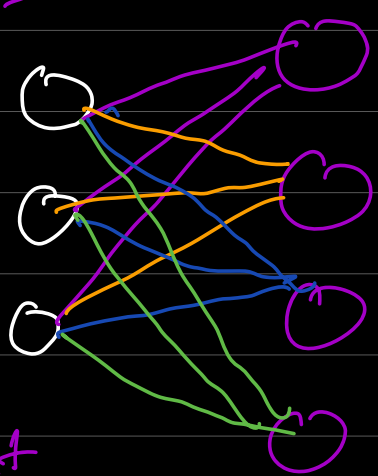
$$g\begin{bmatrix} z_1^{(1)} \\ z_2^{(1)} \\ \vdots \\ z_{n_1}^{(1)} \end{bmatrix} \rightarrow \begin{bmatrix} a_1^{(1)} \\ a_2^{(1)} \\ \vdots \\ a_{n_1}^{(1)} \end{bmatrix}$$

$$\begin{pmatrix} x_1^{(1)} & x_2^{(1)} & x_3^{(1)} \cdots & x_n^{(1)} \end{pmatrix} \begin{pmatrix} w_{11}^{(1)} & w_{12}^{(1)} & w_{13}^{(1)} & \cdots & w_{1n}^{(1)} \\ \vdots & & & & \vdots \\ w_{n_1,1}^{(1)} & & & & w_{n_1,n}^{(1)} \end{pmatrix}$$

$z_1 \quad z_2 \quad z_{n_1}$

$[n \times n_1]$

$n = 3$



Input layer

First layer $n_1 = 4$

$$\begin{pmatrix} x_1^{(1)} & \cdots & x_2^{(1)} & \cdots & x_n^{(1)} \\ \vdots & & & & \end{pmatrix} \begin{bmatrix} W^{(1)} \end{bmatrix} = \begin{bmatrix} z_1^{(1)} & \cdots & z_{n_1}^{(1)} \end{bmatrix}$$

$$[1 \times n] \quad [n \times n_1] \rightarrow [1 \times n_1]$$

$$(m \times n) \quad [n \times n_1] \rightarrow m \times n_1$$

$=$ W Z-Matrix

$$\rightarrow \begin{pmatrix} x_1^{(1)} & \cdots & x_n^{(1)} \\ \vdots & & \\ x_1^{(m)} & \cdots & x_n^{(m)} \end{pmatrix} \begin{bmatrix} \quad \end{bmatrix} = \begin{bmatrix} \quad \end{bmatrix}$$

X

$$X \cdot W = Z$$

$m \times n \quad n \times n_1 \rightarrow m \times n_1$

No of data points

No of neurons in the hidden layer

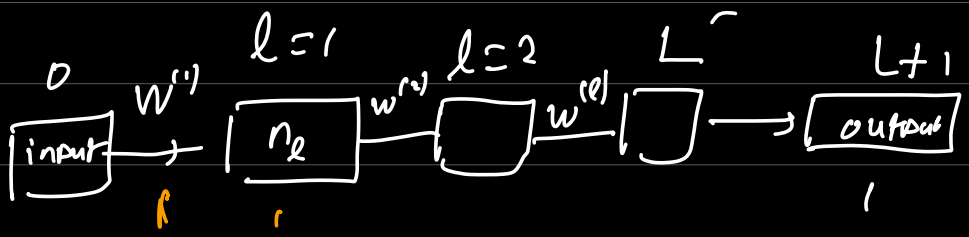
$X \quad W = Z \cdot \quad g(Z)$

↓ pointwise

$A = g(Z) \qquad 1 \times n_1$

↓

$m \times n_1$



$X \longrightarrow XW^{(1)} \xrightarrow{g} A^{(1)} \longrightarrow A^{(1)}W^{(2)}$

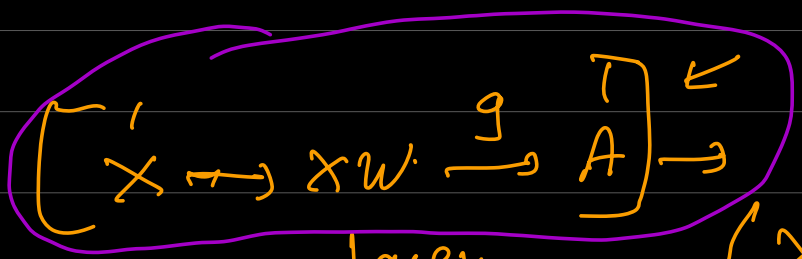
$\cdots \longleftarrow A^{(2)} \longleftarrow g$

$\longrightarrow A^{(L)} \cdot W^{(L+1)} \longrightarrow \text{output}$

$$X \to (W^{(1)})^T X \xrightarrow{g} A^{(1)} \to (W^{(m)})^T A^{(1)} \to \cdots \to g$$

$$\to (W^{(L+1)})^T A^{(L)} \to \text{output}$$

Assuming you know the weights



one layer calculation

(X, A^1)

Each row corresponds to a data point

$X W^{(1)} W^{(2)} \cdots W^{(L)})$

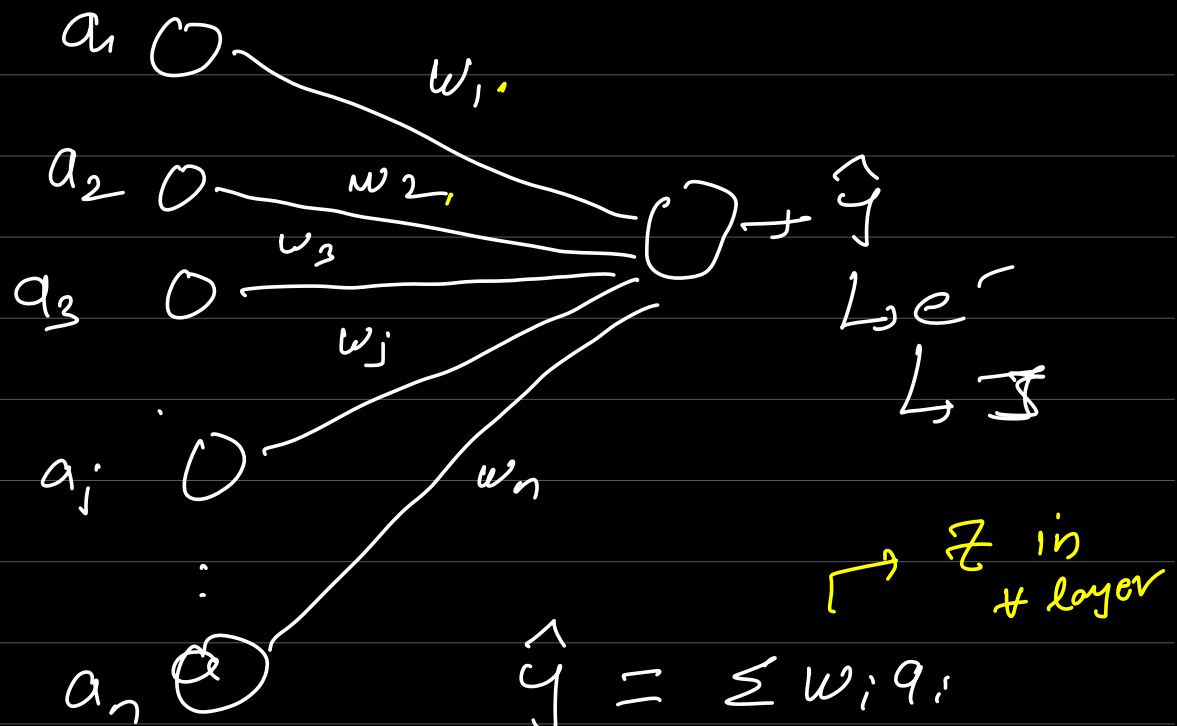
↳ Another linear regression model

Back propagation Algorithm

No hidden layer Model

With and without non-linearities

THE DELTA RULE



$$\hat{y} = \sum w_i a_i$$

$$= w_1 a_1 + w_2 a_2 + \cdots + w_n a_n$$

e = error term $= y - \hat{y}$, y → ground truth label

$$J(y, \hat{y}) = \frac{e^2}{2} = J(e)$$

$$\frac{\partial J}{\partial w_1} = \frac{\partial J}{\partial e} \frac{\partial e}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial w_1}$$

$$= e(-1) a_1 = \boxed{-a_1 e}$$

$$\boxed{\Delta w_1 = -\frac{\partial J}{\partial w_1} = a_1 e}$$

$$w_j \leftarrow w_j + \alpha \Delta w_j$$

$$\leftarrow w_j - \alpha \frac{\partial J}{\partial w_j}$$

update term $= \Delta w_j = a_j e$

$a_j \rightarrow$ input to the weight

$e \rightarrow$ error in the output

This is for a linear neuron

$$z = \sum w_i a_i \qquad \hat{y} = g(z)$$

$$-\frac{\partial J}{\partial w_j} = \pm \frac{\partial J}{\partial e} \cdot \frac{\partial e}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial z} \cdot \frac{\partial z}{\partial w_j}$$

$$\frac{\partial J}{\partial e} = e \qquad \frac{\partial e}{\partial \hat{y}} = 1 \qquad \frac{\partial \hat{y}}{\partial z} = g'(z) \qquad \frac{\partial z}{\partial w_j} = a_j$$

$$= e \cdot g'(z) \, a_j$$

$g'(z) \rightarrow$ Derivative of g wrt z, evaluated at particular z

$$\Delta w_j = g'(z) \, a_j \, e$$

$g'(z)$ → Sigmoid is bad

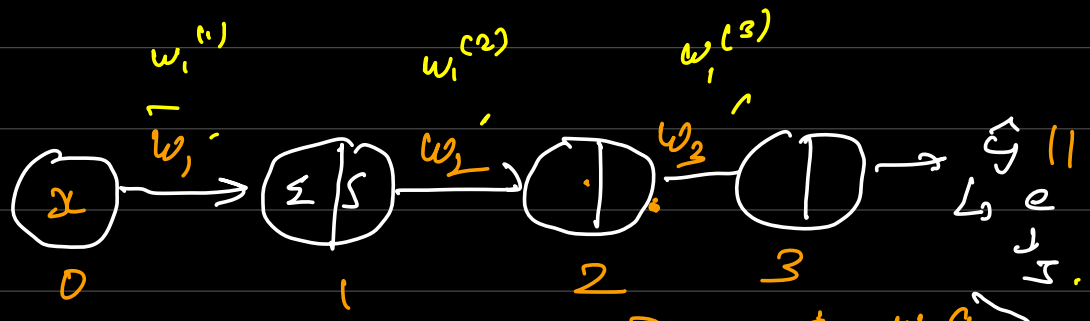
non-linearity

Calculate $g'(z)$ at $z \approx 0$

for $g(z) = \sigma(z)$

$X = A^0$

Multilayer Single Neuron Case



$a_0 = x$ ..... $z_1 = w_1 a_0$ ... $z_2 = w_2 a_1$ ... $z_3 = w_3 a_2$

Input $a_1 = g(z_1)$ ... $a_2 = g(z_2)$ ... $a_3 = g(z_3)$

$e_3 = (y - \hat{y})$

$\hat{y} = a_3$

what is the objective?

$$-\frac{\partial J}{\partial w_1}, \quad -\frac{\partial J}{\partial w_2}, \quad -\frac{\partial J}{\partial w_3} \; ?$$

$$\Delta w_1, \quad \Delta w_2, \quad \Delta w_3$$

$$w_1^{new} \leftarrow w_1^{old} + \Delta w_1 \quad (a)$$

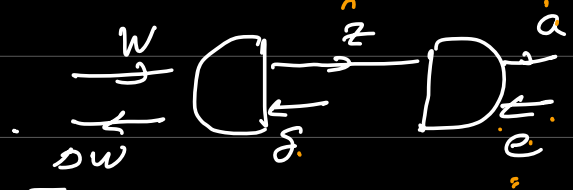
$$w_2^{new} \leftarrow w_2^{old} + \Delta w_2 \quad (a)$$

$$w^{new} = w^{old} + \alpha \Delta w$$

Requires Calculation of intermediate quantities

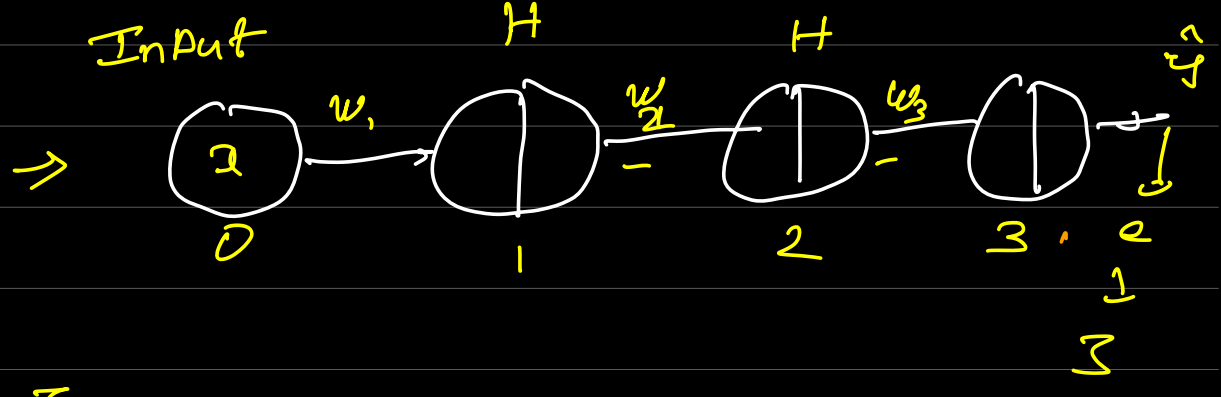
δ_1, δ_2, δ_3

$-\frac{\partial J}{\partial z_1}, \frac{\partial J}{\partial z_2}, \frac{\partial J}{\partial z_3} \rightarrow \delta_1, \delta_2, \delta_3$ → 'linear' terms



e_1, e_2, e_3

$\frac{\partial J}{\partial a_1}, \frac{\partial J}{\partial a_2}, \frac{\partial J}{\partial a_3} \rightarrow e_1, e_2, e_3$ 'non-linear' terms



$$\frac{\partial J}{\partial w_1} = \left| \frac{\partial J}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial a_3} \cdot \frac{\partial a_3}{\partial z_3} \cdot \frac{\partial z_3}{\partial a_2} \cdot \frac{\partial a_2}{\partial z_2} \cdot \frac{\partial z_2}{\partial a_1} \cdot \frac{\partial a_1}{\partial z_1} \cdot \frac{\partial z_1}{\partial w_1} \right|$$

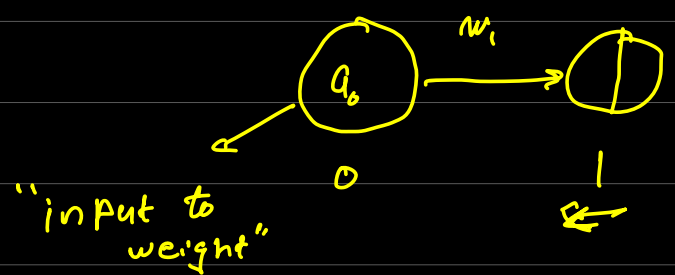
$[z_1 = a_0 w_1]$

$$= \frac{\partial J}{\partial z_1} \cdot a_0 = \delta_1 \cdot a_0$$

↳ input

$$\frac{\partial J}{\partial w_1} = \left\{ \frac{\partial J}{\partial z_1} \cdot \frac{\partial z_1}{\partial w_1} \right\}$$

$$= \delta_1 \cdot a_0$$



$$\frac{\partial J}{\partial w_1} = \left[\left\{ \frac{\partial J}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial a_3} \frac{\partial a_3}{\partial z_3} \frac{\partial z_3}{\partial a_2} \frac{\partial a_2}{\partial z_2} \frac{\partial z_2}{\partial a_1} \frac{\partial a_1}{\partial z_1} \right\} \frac{\partial z_1}{\partial w_1} \right]$$

$$\frac{\partial J}{\partial z_1} a_0 = \delta_1 a_0 \quad \text{"delta rule"}$$

We don't know δ_1

$$\delta_1 = \left\{ \left(\frac{\partial J}{\partial \hat{y}} \quad \frac{\partial \hat{y}}{\partial a_3} \quad \frac{\partial a_3}{\partial z_3} \quad \frac{\partial z_3}{\partial a_2} \quad \frac{\partial a_2}{\partial z_2} \quad \frac{\partial z_2}{\partial a_1} \right) \frac{\partial a_1}{\partial z_1} \right\} = \frac{\partial J}{\partial z_1}$$

$$\frac{\partial J}{\partial a_1} \cdot \frac{\partial a_1}{\partial z_1} \cdot$$

$$\frac{\partial a_1}{\partial z_1} = ? \qquad a_1 = g(z_1)$$

$$\frac{\partial a_1}{\partial z_1} = g'(z_1)$$

$$\delta_1 = g'(z_1)\, e_1$$

what is e_1 ?

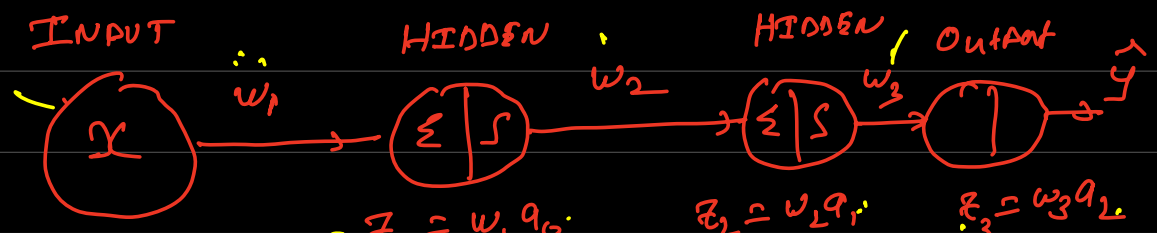
$$\left(\frac{\partial J}{\partial \hat{y}} \quad \frac{\partial \hat{y}}{\partial a_3} \quad \frac{\partial a_3}{\partial z_3} \quad \frac{\partial z_3}{\partial a_2} \quad \frac{\partial a_2}{\partial z_2} \right) \left(\frac{\partial z_2}{\partial a_1} \right) = \frac{\partial J}{\partial a_1}$$

$$= \frac{\partial J}{\partial z_2} \cdot \frac{\partial z_2}{\partial a_1}, \quad z_2 = w_2 a_1$$

$$\frac{\partial z_2}{\partial a_1} = w_2$$

$$\frac{\partial J}{\partial a_1} = e_1 = \frac{\partial J}{\partial z_2} \cdot w_2$$

$$= e_1 = \delta_2 w_2$$



$$\left[\frac{\partial J}{\partial w_1}\right]$$

$a_0 = x$

$\frac{\partial J}{\partial w_1} = a_0 \delta_1$

$= a_1 \delta_2$

$\delta_1 = g'(z_1) e_1$ $\quad \delta_2 = g'(z_2) e_2$ $\quad \delta_3 = g'(z_3) e_3$

$e_2 = w_2 \delta_2$ $\quad e_2 = w_3 \delta_3$

$(y - \hat{y})$

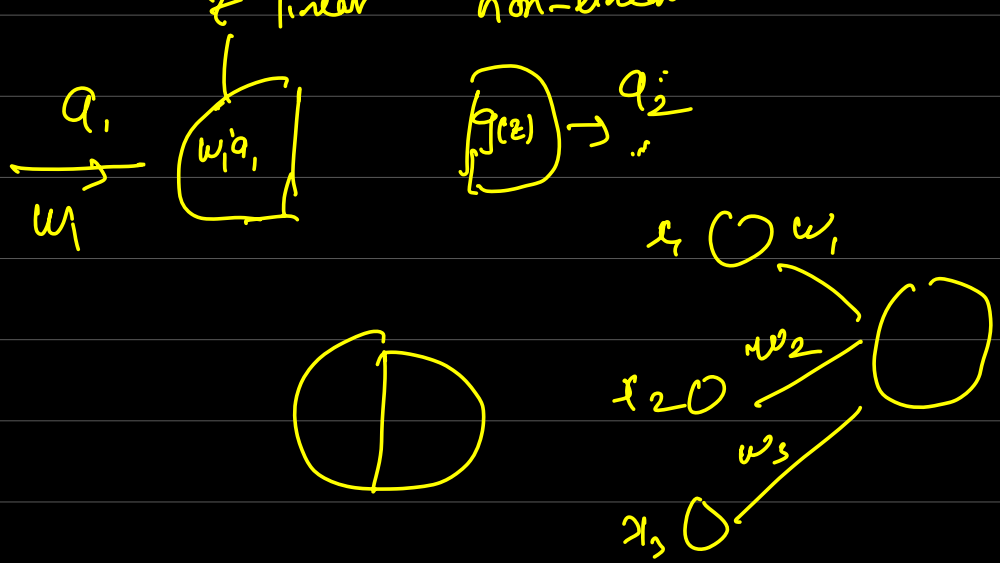
$$\frac{\partial J}{\partial w_2} = a_1 \delta_2$$

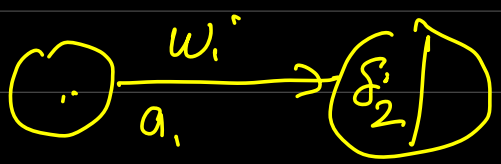
$$J = \frac{1}{2}(y - \hat{y})^2$$

$$J = \frac{1}{2}(y - a_3)^2$$

$$e_3 = \frac{\partial J}{\partial a_3} = (y - a_3) = (y - \hat{y})$$

$$\begin{bmatrix} \frac{\partial J}{\partial q_1} & \frac{\partial J}{\partial q_2} & \frac{\partial J}{\partial q_3} \\ \frac{\partial J}{\partial z_1} & \frac{\partial J}{\partial z_2} & \frac{\partial J}{\partial z_3} \end{bmatrix}$$





$$z = w_1 x_1 + w_2 q_1 + w_3 x_3$$

$$\frac{\partial J}{\partial w_2} = \delta x_3$$

Full algorithm (scalar rep)

1) Do the forward pass

$$a_0 = x_i \rightarrow z_1, a_1, \; z_2, a_2, \; z_3 a_3 \dots z_i a_i$$

$$\hat{y} \rightarrow e = (y - \hat{y}) = e_3$$

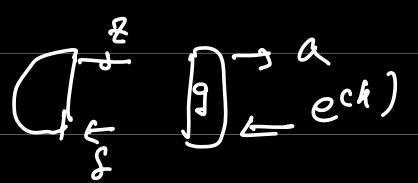
Back prop

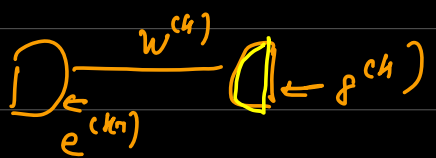
2) $$\underline{\delta}^{(k)} = g'(z^{(k)}) \; \underline{e}^{(k)}$$

linear error — non-linear error

$$\underline{e}^{(k-1)} = W_k \underline{\delta}_k$$

$$\rightarrow \Delta W = \underline{a}_i \underline{\delta}_i$$





3) Continue to calculate $\delta^{(k)}$ and $e^{(k)}$ till you reach the first layer

4) $\Delta w_i = \eta \cdot \delta_i$ $\qquad \frac{\partial J}{\partial w_1} =$

$\frac{\partial J}{\partial w_i}$ → required to update weights

why backpropagation?

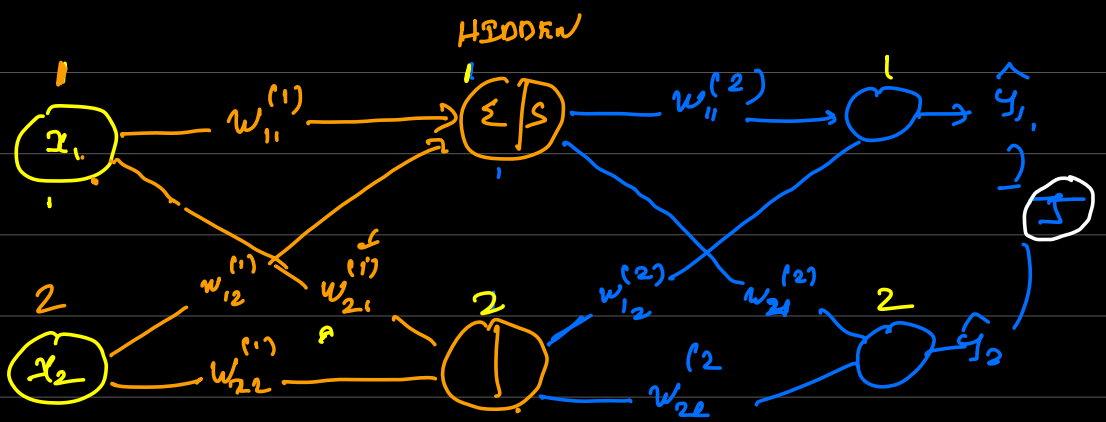
If you use finite difference

$$\frac{\partial J}{\partial w_1} = \frac{J(w_1 + \Delta w_1, w_2, w_3) - J(w_1, w_2, w_3)}{\Delta w_1}$$

(one forward; one forward)

4 passes forward- passes

MULTI-LAYER MULTI-NEURON CASE



(l) → layer

w_{ij} → TO / FROM

J → scalar

$(y_1 - \hat{y}_1)^2 + (y_2 - \hat{y}_2)^2$

FORWARD PASS: $z_1^{(1)}$, $z_2^{(1)}$

$$z_1^{(1)} = w_{11}^{(1)} x_1 + w_{12}^{(1)} x_2$$

$$z_2 = w_{21}^{(1)} x_1 + w_{22}^{(1)} x_2$$

$$a_1^{(1)} = g(z_1^{(1)})$$

$$a_2^{(1)} = g(z_2^{(1)})$$

$$a = g(z)$$

$$\begin{bmatrix} z_1^{(1)} \\ z_2^{(1)} \end{bmatrix} = \begin{bmatrix} w_{11}^{(1)} & w_{12}^{(1)} \\ w_{21}^{(1)} & w_{22}^{(1)} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

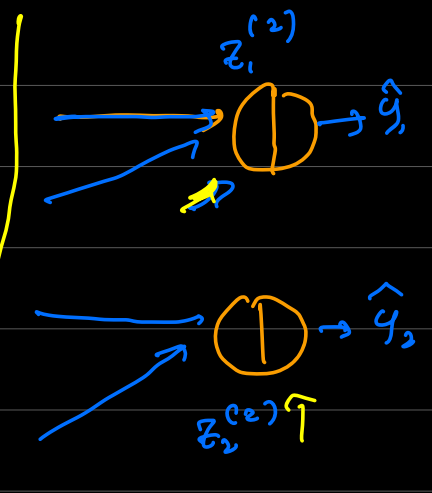
$$\begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = g\left(\begin{bmatrix} z_1 \\ z_2 \end{bmatrix}\right)$$

$x = a^{(0)}$

$$z^{(1)} = w^{(1)} x$$

$2 \times 1 \quad 2 \times 2 \quad 2 \times 1$

Final layer



$$\frac{\partial J}{\partial \hat{y}_1} = e_1^{(2)} = y - \hat{y}_1$$

$$\frac{\partial J}{\partial \hat{y}_2} = e_2^{(2)} = y - \hat{y}_2$$

$$\delta_1^{(2)} = \frac{\partial J}{\partial z_1^{(2)}} = \frac{\partial J}{\partial a_1} \frac{\partial a_1}{\partial z_1} = g'(z_1^{(2)}) \cdot e_1$$

$$\delta_2^{(2)} = \frac{\partial J}{\partial a_2} \frac{\partial a_2}{\partial z_2} = g'(z_2^{(2)}) \cdot e_2$$

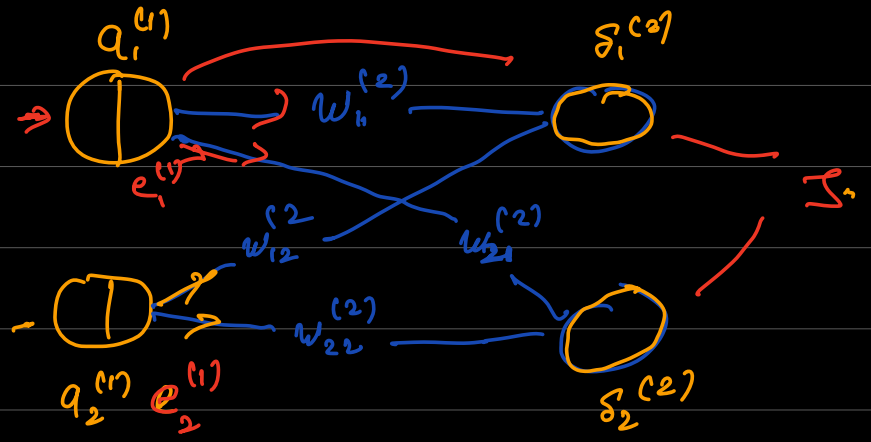
$$\delta^{(2)} = g'(z^{(2)}) \odot e$$

$\odot$ → hadamard product

$$\begin{pmatrix} \delta_1^{(2)} \\ \delta_2^{(2)} \end{pmatrix} = \begin{pmatrix} g'(z_1^{(2)}) \\ g'(z_2^{(2)}) \end{pmatrix} \cdot \begin{pmatrix} e_1^{(2)} \\ e_2^{(2)} \end{pmatrix}$$

$$c = a \odot b$$

$$c_i = a_i b_i$$



$$e_1^{(1)} = \frac{\partial J}{\partial a_1^{(1)}} = \frac{\partial J}{\partial z_1^{(2)}} \cdot \frac{\partial z_1^{(2)}}{\partial a_1^{(1)}} + \frac{\partial J}{\partial z_2^{(2)}} \cdot \frac{\partial z_2^{(2)}}{\partial a_1^{(1)}}$$

$$e_1^{(1)} = \delta_1^{(2)} w_{11}^{(2)} + \delta_2^{(2)} w_{21}^{(2)}$$

$$e_2^{(1)} = \frac{\partial J}{\partial a_2^{(1)}} = \frac{\partial J}{\partial z_1} \frac{\partial z_1}{\partial a_2} + \frac{\partial J}{\partial z_2} \frac{\partial z_2}{\partial a_2}$$

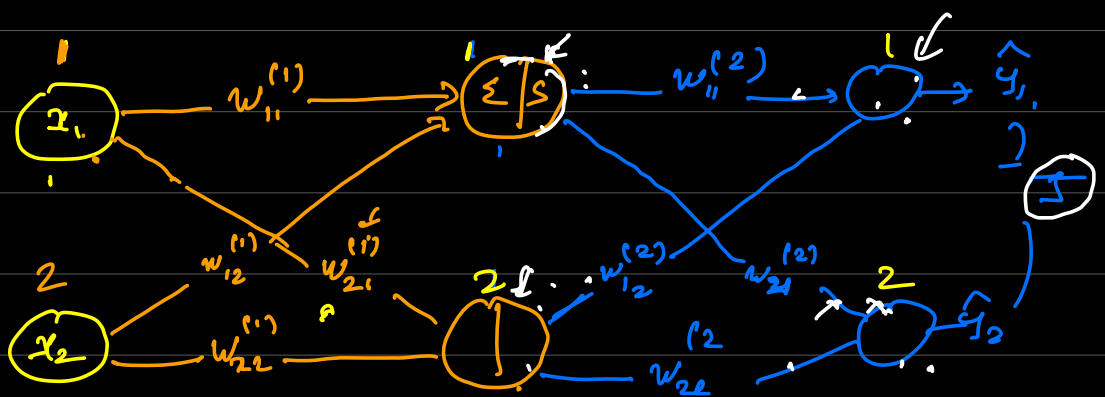
$$e_2^{(1)} = \delta_1^{(2)} w_{12}^{(2)} + \delta_2^{(2)} w_{22}^{(2)}$$

$$\begin{bmatrix} e_1^{(1)} \\ e_2^{(1)} \end{bmatrix} = \begin{bmatrix} w_{11}^{(2)} & w_{21}^{(2)} \\ w_{12}^{(2)} & w_{22}^{(21} \end{bmatrix} \begin{bmatrix} \delta_1^{(2)} \\ \delta_2^{(21} \end{bmatrix}$$

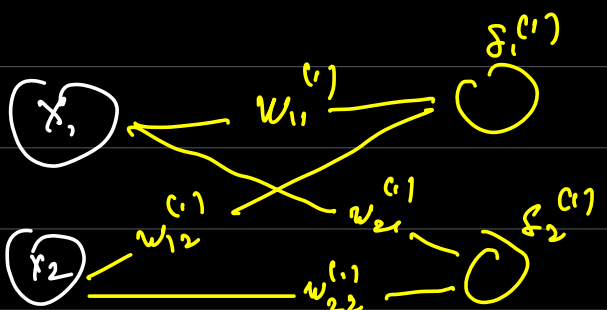
$$e^{(1)} \qquad W^T \left(W^{(21}\right)^T \cdot \quad \delta^{(2)}$$

$$e^{(1)} = \left(W^{(21}\right)^T \delta^{(2)}$$

$$\delta^{(2)} = g'(z^{(2)}) \odot e^{(2)}$$



From hidden to weight update



$$\Delta w_{11}^{(1)} = x_1 \delta_1^{(1)}$$

$$\Delta w_{12}^{(1)} = x_2 \delta_1^{(1)}$$

$$\Delta w_{ij}^{(l)} = \delta_i a_j$$

---

(1) Forward Pass $z^{k+1} = w^{(k)} a^{(k)}$

$$a^{(k+1)} = g(z^{(k+1)})$$

(2) Calculate error vector at final layer $\rightarrow e$

(3) $\delta^{(k)} = g'(z^{(k)}) \odot e^{(k)}$

$e^{k-1} = (w^{(k-1)})^T \delta^{(k)}$ $\rightarrow$

Continue to first layer

(4) update weights

$\Delta w_{ij} = \delta_i a_j$ (Automatic Differentiation)