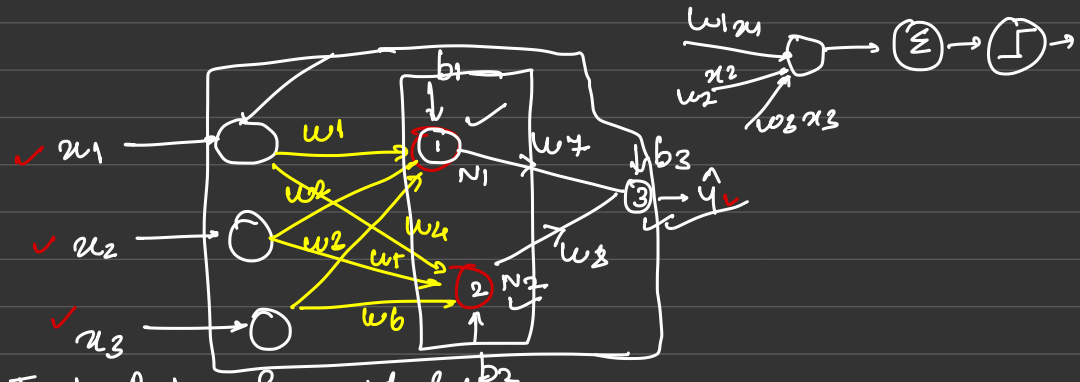
Forward and Backward propagation



Tanh, Relu, Sigmoid, Softmax

$$N_1 = (w_1x_1 + w_2x_2 + w_3x_3 + b_1)$$

$$N_2 = (w_4x_1 + w_5x_2 + w_6x_3 + b_2)$$

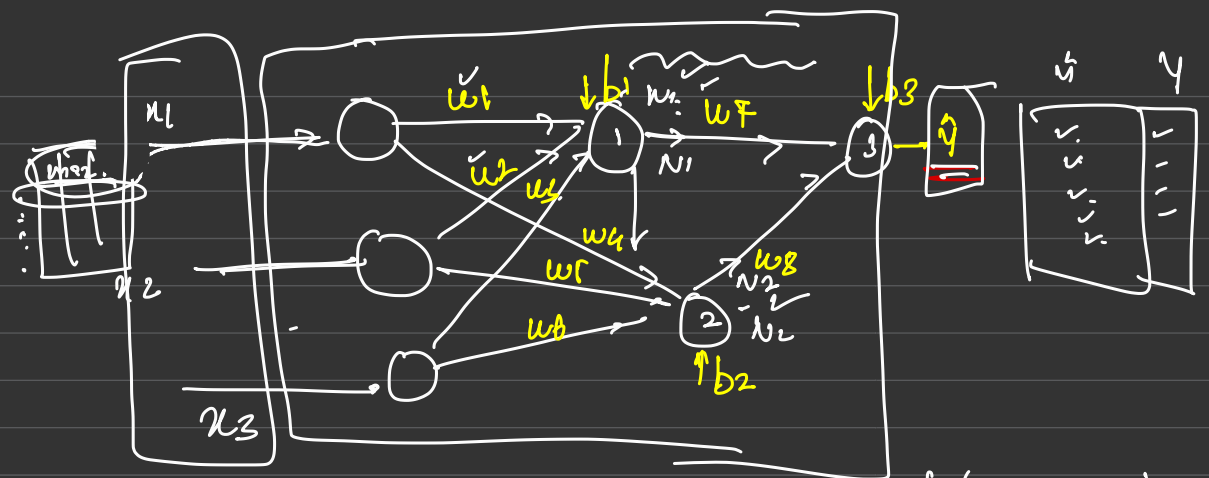
$$\hat{y} = (w_7N_1 + w_8N_2 + b_3)$$

x_1	x_2	x_3	y

$y = f(x_1, x_2, x_3)$

$\hat{y} \leftarrow$ MLP (x_1, x_2, x_3)

f will achieve (my $\hat{y}$ becomes closer to y) by manipulating the weights and biases



loss of my neural network (MLP)

$$\text{loss} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

Objective: To manipulate my weights and biases to reduce the loss.

1) We are going to initialise my weights and biases randomly.

Forward Propagation

2) Based on the weights and biases initialised the network will calculate $\hat{y}$.

$N_1 = (w_1x_1 + w_2x_2 + w_3x_3 + b_1)$

$N_2 = (w_4x_1 + w_5x_2 + w_6x_3 + b_2)$

$\hat{y} = (w_7N_1 + w_8N_2 + b_3)$

3) We will calculate the loss of the network

$$\sum_{i=1}^{n}\frac{1}{n}(y_i - \hat{y}_i)^2$$

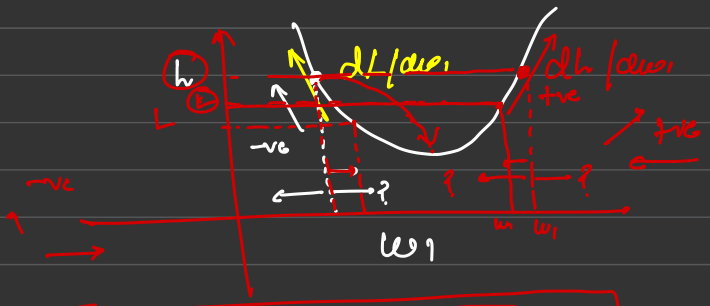
Backward Propagation:

① To minimize the loss by manipulating the weights and biases

$$L = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

$$L = \frac{1}{n}\sum_{i=1}^{n}(y_i - (w_1 x_{1i} + w_2 x_{2i} + b_3))^2$$

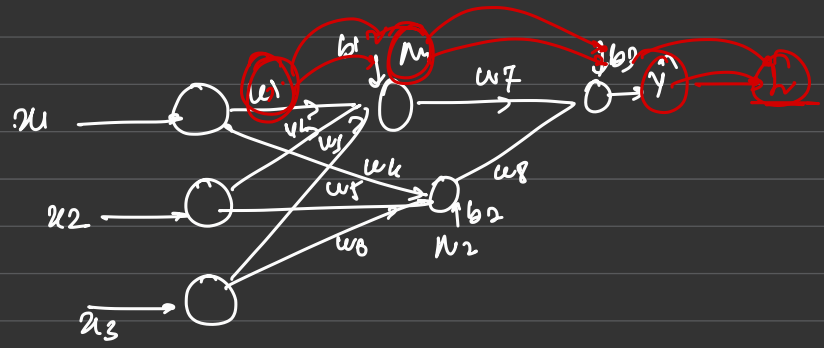
Gradient Descent:



$$w_1(t+1) = w_1(t) - \alpha \, dL/dw_1(w_1)$$

$\frac{dL}{dw_1}, \frac{dL}{dw_2}, \frac{dL}{dw_3} \cdots \frac{dL}{dw_p}$

$\frac{dL}{db_1}, \frac{dL}{db_2} \cdots \frac{dL}{db_p}$



$$L = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

$$L = \frac{1}{n}\sum_{i=1}^{n}(y_i - (w_7 N_{1i} + w_8 N_{2i} + b_3))^2$$

$$\frac{dL}{dw_7} = \frac{1}{n}\sum_{i=1}^{n} -2\,(y_i - (w_7 N_{1i} + w_8 N_{2i} + b_3))\,N_{1i}$$

$\frac{dL}{dw_8}$

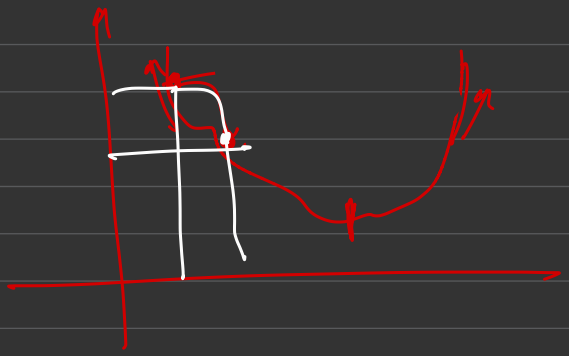
$\frac{dL}{dw_1}$

$\frac{dL}{dw_1}$ = Chain Rule

$$\frac{dL}{dw_1} = \frac{dL}{dN_1} \times \frac{dN_1}{dw_1}$$

$\frac{dL}{dw_2} = \frac{dL}{dN_1} \times \frac{dN_2}{dw_2}$

$N_1 = (w_1 x_1 + w_2 x_2 + w_3 x_3 + b_1)$



We initialise the weights and biases randomly.

given these weights and biases we calculate the loss by propagating through network → Forward Propagation

given the loss we calculate the derivative wrt to all the weights and biases and do the gradient descent to get new set of weights and biases

↓ Backward Propagation