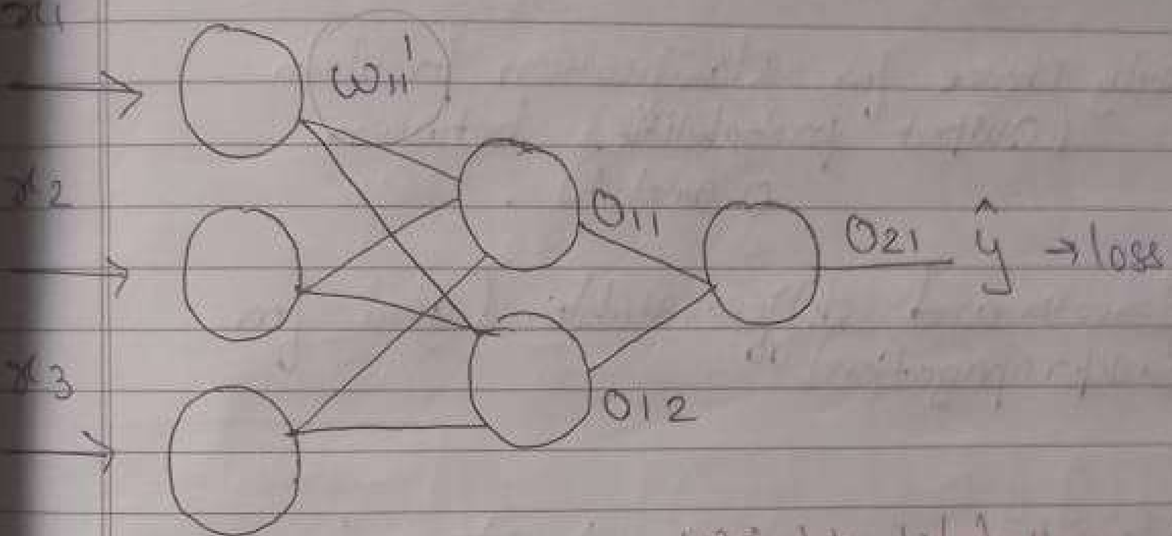
Vanishing Gradient Problem :-

previously we were using Sigmoid at that time ReLU was not invented.
the problem faced was Vanishing Gradient problem.



Weight Updation :-

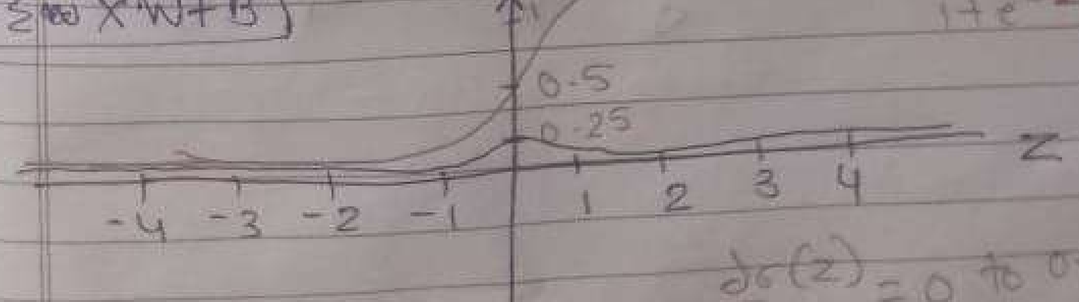
$$w_{11}^{1}\,new = w_{11}^{1}\,old - \eta \frac{\partial L}{\partial w_{11}^{1}\,old}$$

$$\frac{\partial L}{\partial w_{11}} = \frac{\partial O_{21}}{\partial O_{11}} \cdot \frac{\partial O_{11}}{\partial w_{11}^{1}} \rightarrow \text{Chain Rule}$$

Note :- Derivative of Sigmoid will Range between 0 to 0.25

$\boxed{Z = \sum X*W + B}$

$\frac{1}{1+e^{-z}}$



$$\frac{\partial \sigma(z)}{\partial z} = 0 \text{ to } 0.25$$

5/Aug/2025

Date ___/___/_____

$0 \le \sigma'(z) \le 0.25$

Sigmoid Activation →

$$f(x) = \frac{1}{1+e^{-x}}$$

- Early choice for classification problem. (Output probabilities between 0 and 1).

- Smooth and differentiable (good for Backpropagation)

- Vanishing gradient problem: Gradients become tiny, slowing learning in deep networks.

for large/small inputs gradients are almost zero.

Not zero-centred (all outputs are +ve, causing inefficient optimization)

Date ___/___/___

ReLU Activation →

$$f(x) = \max(0, x)$$

- Common in deep networks to avoid vanishing gradient.

- No vanishing gradient for +ve inputs (faster learning)

- Simple & efficient to compute

- Sparse activation :- Most neurons are inactive (output 0)

- Dying ReLU problem → Neurons can get stuck outputting 0 and stop learning.