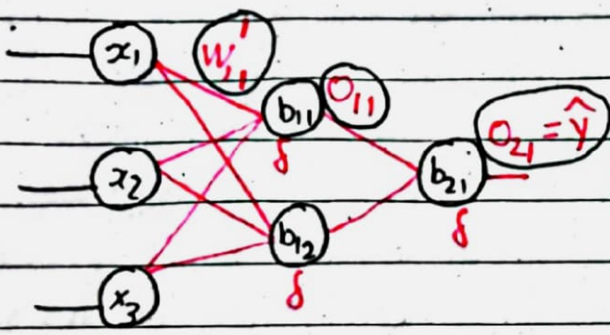
VAnishing Gradient problem

⇒ This is the problem which we face by using the $\boxed{\text{Sigmoid as activation function}}$ in deep neural network.

first of all we need to know this fact
ie multiplying small number gives more small number
⇒ $0.001 \times 0.1 \times 0.001 \times 0.003 = 0.0000000003$



δ = Sigmoid activation

We have send our row data to Neural Network By feed forwarding we have calculated $\hat{y}$. Now, We have calculated loss and Neural Network performs back propagation to update parameters.

lets update w'_{11} parameter

$$W'_{11}new = w'_{11}old - \alpha \frac{dloss}{dw'_{11}old}$$

(lets suppose α as 0.01 and $w'_{11}old$ as 0.5 which we can set during inilization)

$$w_{11}^{1}\,new = 0.5 - \alpha \underbrace{\boxed{\frac{d\,loss}{dw_{11}^{1}\,old}}}_{A}$$

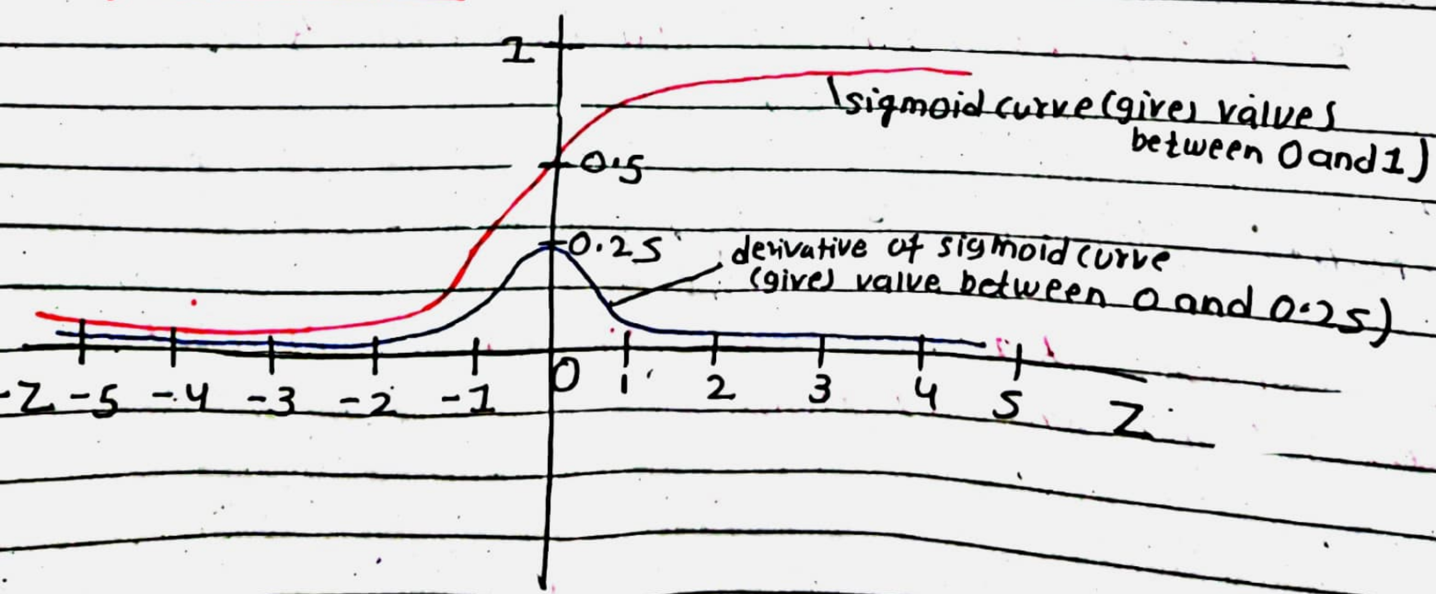
$$\therefore A = \frac{d\,loss}{dw_{11\,old}^{1}}$$

$$[w_{11}^{1} \rightarrow O_{11} \rightarrow O_{21}(\hat{y}) \rightarrow loss]$$

$$\Rightarrow \frac{d\,loss}{dw_{11\,old}^{1}} = \frac{d\,loss}{dO_{21}} \times \frac{dO_{21}}{dO_{11}} \times \frac{dO_{11}}{dw_{11}^{1}}$$

[O_{21} and O_{11} are output given by sigmoid function and dO_{21} and dO_{11} are differentation of sigmoid function whose value ranges between 0 – 0.25] [$\because O_{21} = \frac{1}{1+e^{-z}}$ $\therefore z = w_{11}^{2} \cdot O_{11} + w_{21}^{2} \cdot O_{12} + b_{21}$]

from graph



$$\frac{d\,loss}{dO_{21}} \times \frac{dO_{21}}{dO_{11}} \times \frac{dO_{11}}{dw'_{11}}$$

[0-0.25] · [0-0.25] [It's value lies over here]

let's say we got $\frac{d\,loss}{dO_{21}} = 25$, $\frac{dO_{21}}{dO_{11}} = 0.25$, $\frac{dO_{11}}{dw'_{11}} = 0.20$

$$\therefore \frac{d\,loss}{dO_{21}} \times \frac{dO_{21}}{dO_{11}} \times \frac{dO_{11}}{dw'_{11}} = 25 \times 0.25 \times 0.20 = 1.25$$

Now,

$$w'_{11new} = 0.25 - \alpha \frac{d\,loss}{dw'_{11old}}$$

$$w'_{11new} = 0.25 - 0.01 \times (1.25)$$

$$w'_{11}new = 0.25 - 0.0125$$

∴ New w'_{11} will be =

$$\Rightarrow 0.25 - 0.0125 = 0.2375 \text{ (very less change)}$$

If we have very big deep Neural Network then

$$\frac{d\,loss}{dw'_{11}} = \frac{d\,loss}{dO_{51}} \times \frac{dO_{51}}{dO_{41}} \times \frac{dO_{41}}{dO_{31}} \times \frac{dO_{31}}{dO_{21}} \times \frac{dO_{21}}{dO_{11}} \times \frac{dO_{11}}{dw'_{11}}$$

$$= 25 \times 0.25 \times 0.22 \times 0.20 \times 0.15 \times 0.10$$

$$\frac{d\,loss}{dw'_{11}} = 0.004125$$

Now,

$$w'_{11}new = 0.25 - \underset{\alpha}{(0.01)} \times \underset{\frac{d\,loss}{dw'_{11}old}}{(0.004125)}$$

$$= 0.25 - 0.00004125$$

$$= 0.249999 \sim 0.25 \text{ (weight never get updated)}$$

due to this problem we have to use other activation functions like Relu, tanh etc.