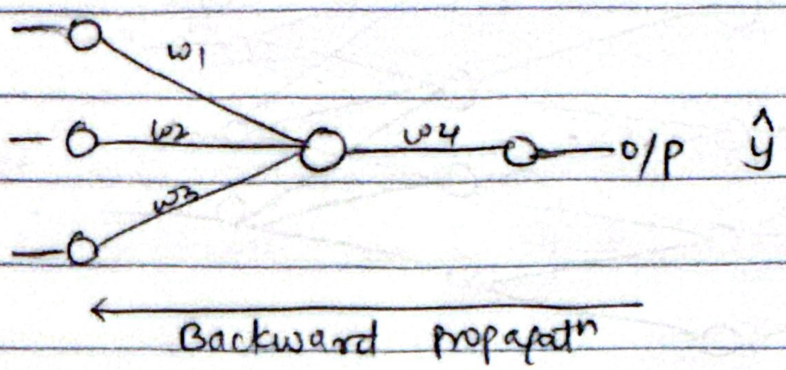
② Backpropagation

• Weight updation formula

↳ Chain rule of differentiation.



→ **Weight updation formula**

derivative of loss

$$W_{new} = W_{old} - \eta \frac{\partial L}{\partial W_{old}}$$

previous weight

learning rate

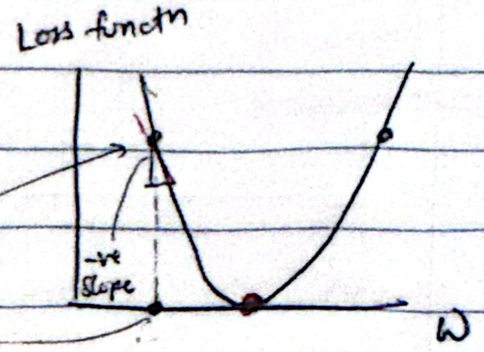
derivative of old

Important

basically calc the slope

$$\boxed{\frac{\partial L}{\partial W_{old}}}$$

when you apply this here it creates a slope



• 1st thing we need to notice is whether it is +ve slope or -ve

• So here inorder to reach global minima you have to increase the weight

- Now the formula would become

$$\boxed{\text{-ve slope}}$$

$$w_{new} = w_{old} - \eta\,(\text{-ve})$$

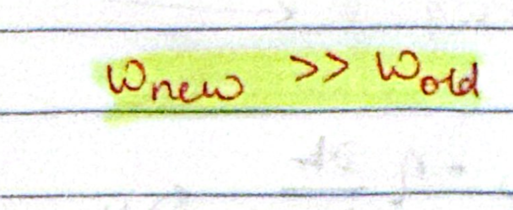
So for -ve slope you have to increase the weights to come close to global minima

$$w_{new} = w_{old} + \eta\,(\text{ve})$$

So, $w_{new} >> w_{old}$

So now if negative slope we are able to increase the weights as we got $w_{new} >> w_{old}$

→ Now in other case



Now Here, we have to decrease the weight

$$w_{new} = w_{old} - \eta\,(\text{+ve})$$
$$= w_{old} - \eta\,(\text{+ve})$$

so $\therefore$ $\boxed{w_{new} << w_{old}}$

$h = L \rightarrow$ loss

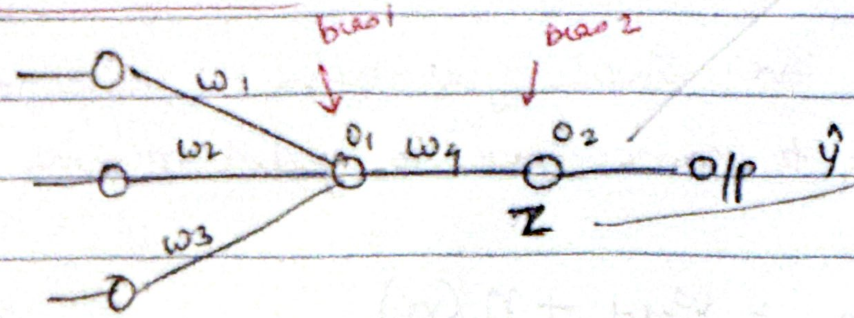
$\boxed{\text{learning rate should be a smaller number so we slowing converge into global minima}}$

$\eta = 0.01$ or $\eta = 0.001$

② Chain Rule of Differentiation

Acti funct^n

$Z = \sigma(O_1 W_4 + b)$



$W_{new} = W_{old} - \eta \boxed{\frac{\partial L}{\partial W_{old}}}$

now lets try for w4

$$W_{4\,new} = W_{4\,new} - \eta \frac{\partial L}{\partial W_{4\,old}}$$

How to get this value?

for this we use chain rule of derivative

W4 is depended on O2

$$\frac{\partial L}{\partial W_{4\,old}} = \frac{\partial L}{\partial O_2} * \frac{\partial O_2}{\partial W_{4\,old}}$$

Bias formula

$$b_{2\,new} = b_{2\,old} - \eta \frac{\partial L}{\partial b_{2\,old}}$$

O2 is dependent on O1 & O1 is dependent on w_1

→ w_1

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

$$\frac{\partial L}{\partial w_{1\,old}} = \frac{\partial L}{\partial O_2} * \frac{\partial O_2}{\partial O_1}$$

L is dependent on O2 & O2 is dependent on O1

$$\boxed{\frac{\partial L}{\partial w_{1\,old}} = \frac{\partial L}{\partial O_2} * \frac{\partial O_2}{\partial O_1} * \frac{\partial O_1}{\partial w_{1\,old}}}$$

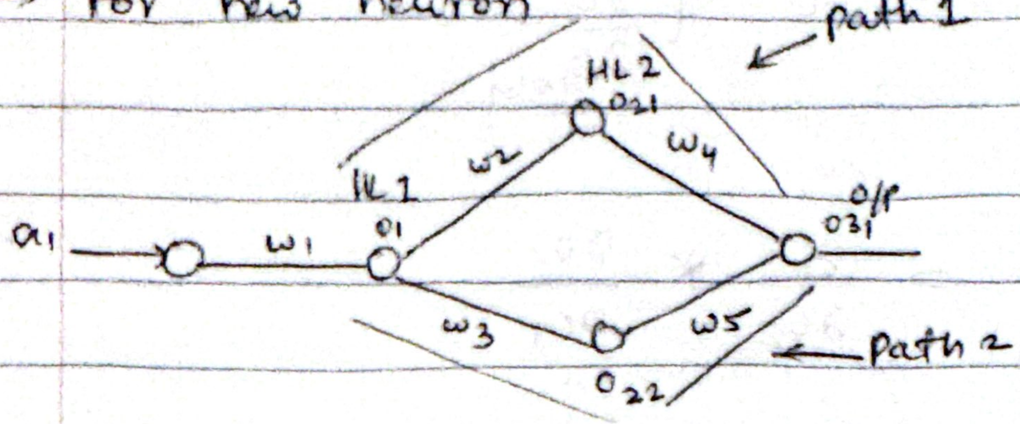
to update w_1 chain rule of defferentiation

→ w_2

$$w_{2\,new} = w_{2\,old} - \eta \frac{\partial L}{\partial w_{2\,old}}$$

$$\frac{\partial L}{\partial w_{2\,old}} =$$

→ For new neuron



Chain rule of derivatives

$$W_{1_{\cancel{old}\,new}} = W_{1_{old}} - \eta \frac{\partial L}{\partial W_{1_{old}}}$$

$$\frac{\partial L}{\partial W_{1old}} = \left[\frac{\partial L}{\partial O_{31}} * \frac{\partial O_{31}}{\partial O_{21}} * \frac{\partial O_{21}}{\partial O_1} * \frac{\partial O_1}{\partial W_{1old}}\right]$$

$$+$$

$$\left[\frac{\partial L}{\partial O_{31}} * \frac{\partial O_{31}}{\partial O_{22}} * \frac{\partial O_{22}}{\partial O_1} * \frac{\partial O_1}{\partial W_{1old}}\right]$$

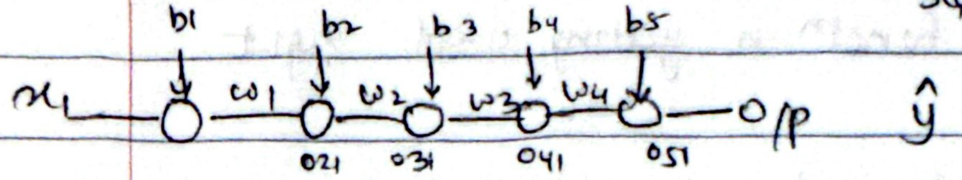
This task is done by optimizers.

Super Imp Interview Problem

↓

→ Vanishing Gradient Problem.

Sigmoid Act funct^n



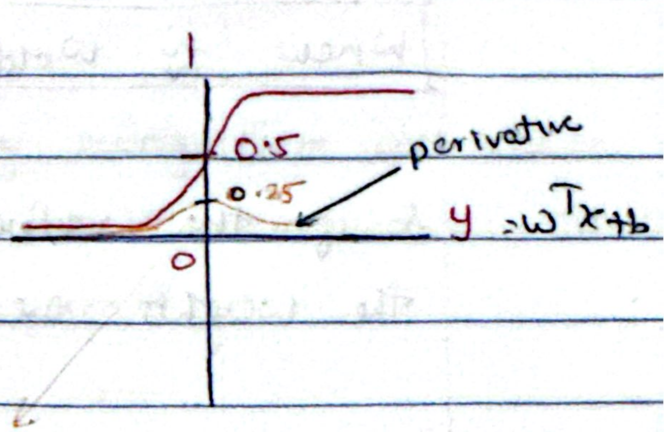
MSE = Loss Function $= \frac{1}{2}(y - \hat{y})^2$

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

$$\frac{\partial L}{\partial w_{1new}} = \frac{\partial L}{\partial O_{51}} * \frac{\partial O_{51}}{\partial O_{41}} * \frac{\partial O_{41}}{\partial O_{31}} * \frac{\partial O_{31}}{\partial O_{21}} * \frac{\partial O_{21}}{\partial w_1}$$

Sigmoid Act^n formula.

$$Z = \frac{1}{1+e^{-x}}$$



Imp → whenever we try to find out derivative of sigmoid it will be ranging bet^n 0 to 0.25

Derivative Condition

$$\boxed{0 \le \sigma(y) \le 0.25}$$

Now, in neuron dia behind while finding O_1, O_{21}, O_{31}, O_{41}, O_{51} everywhere sigmoid activation funct^n is getting used right

So now the eq^n of $w_1$new will be

$= 0.25 * 0.15 * 0.10 * 0.05 * 0.02$

So as the values are getting decreased now what will happen is

$$w_{new} = w_{old} - \underset{\text{small value}}{\eta} \text{ (small number)}$$

$$\boxed{w_{new} \approx w_{old}}$$

So if the values are hardly changing then the weights are not getting updated.

So this problem where the weights are not getting changed or are changed by small value this problem is called Vanishing Gradient problem