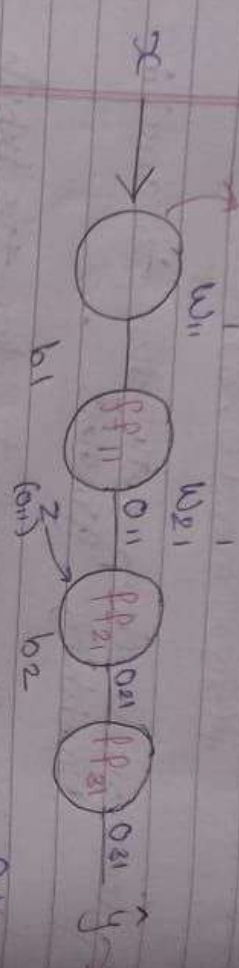


# EXPLAINING GRADIENT Problem →



$$w'_{11 \text{ new}} = w_{11 \text{ old}} - \eta \left( \frac{\partial L}{\partial w_{11}} \right)$$

Learning Rate

$$\frac{\partial L}{\partial w_{11}} = \frac{\partial O_{31}}{\partial O_{21}} \cdot \frac{\partial O_{21}}{\partial O_{11}} \cdot \frac{\partial O_{11}}{\partial w_{11}}$$

$$Z = w_{21} \cdot O_{11} + b_2 \quad \left[ O_{21} = \phi(Z) \right] \rightarrow \frac{1}{1+e^{-z}} \quad \left[ Z = w_{21} \cdot O_{11} + b_2 \right]$$

$$\frac{\partial O_{21}}{\partial O_{11}} = \frac{\partial \phi(Z)}{\partial Z} \times \frac{\partial Z}{\partial O_{11}}$$

$$= 0 \leq \phi(Z) \leq 0.25 \quad * \quad \frac{\partial (w_{21} \cdot O_{11} + b_2)}{\partial O_{11}}$$

$$\text{Suppose} = 0.25 \times 500 = 125$$

$$\text{mean}, \frac{\partial O_{21}}{\partial O_{11}} = 125$$

$$\text{Now consider } \frac{\partial O_{11}}{\partial w_{11}} \quad \text{If multiplied by 125}$$

If weights are higher, they reach near converge to global minimum point.

Exploding gradient problem happens when the numbers (gradients) used to update the weights in a neural network become very large, especially during training.

→ Weights get updated with huge values.

→ The model becomes unstable.

→ The model fails to learn.

In training neural networks, we update weights:

$$\text{New weight} = \text{Old weight} - \text{Learning Rate} \times \text{Gradient (loss)}$$

Gradients are calculated using chain Rule in Backpropagation.

$$\frac{dL}{dw} = \frac{dL}{da_n} \cdot \frac{da_n}{da_{n-1}} \cdots \frac{da_2}{da_1} \cdot \frac{da_1}{dw}$$

This is product of many derivatives (like 10, or even 100 terms for deep networks).

If in case the number is slightly bigger than 1 then multiplying many times :-

Gradient  $\approx 1.5^n$

If  $n = 16$

$1.5^{16} \approx 57$

If  $n = 50$ ,

$1.5^{50} \approx 888,000$

(Huge)

New weight = Old weight  $\times n \times 888,000$

Even if  $n = 0.01$

this is a massive change

If you multiply a number greater than 1 many times, it explodes.