Gradient Problems

1. Vanishing gradient problem.
2. Exploding gradient problem.

1. Vanishing Gradient Problem:

Vanishing gradient problem is a phenomenon that occurs during the training of deep neural networks, where the gradients that are used to update the network become extremely small or "vanish" as they are backpropagated from the output layers to the earlier layers.

- We should use chain rule for finding $\frac{\partial L}{\partial \omega_{old}}$ if we use gradient descent to reduced loss function.
- If we observe the child role poll differentiation terms are multiplying each other. In mathematics the differentiation term is always a very less value those values may exist like 0 .1 or 0 .001 or 0 .00025 etc.
- When we multiply these small values in chain rule, then it will give very small value like 10^{-10} or 10^{-15} etc. The problem is when we try to find the local minima of loss function somewhere the point will be vanished because it treated $\frac{\partial L}{\partial \omega_{old}}$ as 0 ($\frac{\partial L}{\partial \omega_{old}} \approx 0$).
- Actually, the differentiation loss function ($\frac{\partial L}{\partial \omega_{old}}$) is not zero. This problem is called vanishing gradient problem.

1.1. Vanishing Gradient Mathematical Proof

- Let's understand this vanishing gradient problem mathematically.

$$\omega_{new} = \omega_{old} - \alpha \frac{\partial L}{\partial \omega_{old}} \dots\dots\dots\dots \quad (1)$$

As per our previous explanation the derivative term is very less value and again, we multiply with Learning rate α.

Consider $\alpha = 0.001$ and $\frac{\partial L}{\partial \omega_{old}} = 2 \times 10^{-10}$

$$\alpha \frac{\partial L}{\partial \omega_{old}} = 0.001 \times 0.00000000002$$

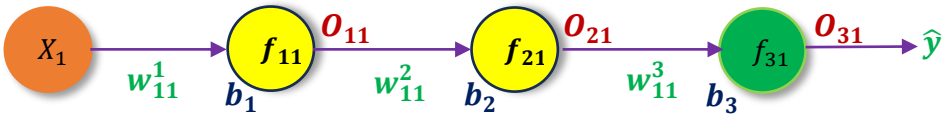
$$\alpha \frac{\partial L}{\partial \omega_{old}} \approx 0$$

- The $\alpha \frac{\partial L}{\partial \omega_{old}}$ is approximately zero and if we subtract this value from ω_{old} then there is no change in weights (ω_{new}), if weighs not much changed then predicted $\hat{y}$ value is almost same, and loss function also will not change, finally the back propagation takes it as minimum point of loss function but actually it is not.

2. Exploding Gradient Problem:

If the gradients are large, the multiplication of these gradients will become huge over time.

- In this case the weights of neural network are very high. These higher weights are not good to reach global minima point of loss function in GD.
- Let's understand this problem mathematically for that consider single line from neural network as shown in below:



Let's take f_{21} neuron hypothesis function $Z = O_{11}\, w_{11}^2 + b_2$

The sigmoid of Z is $\oint(Z) = O_{21}$

Now let's try to update the weight w_{11}^1 :

$$w_{11new}^1 = w_{11}^1 - \alpha \frac{\partial L}{\partial w_{11}^1}$$

$$\frac{\partial L}{\partial w_{11}^1} = \left[\frac{\partial L}{\partial O_{31}} \cdot \frac{\partial O_{31}}{\partial O_{21}} \cdot \frac{\partial O_{21}}{\partial O_{11}} \cdot \frac{\partial O_{11}}{\partial w_{11}^1}\right] \ldots\ldots\ldots\ldots (1)$$

- We know that sigmoid activation function lies between zero to one.

$$0 \leq \oint(Z) \leq 1$$

- And derivative of same sigmoid function lies between 0 to 0.25.

$$0 \leq \frac{\partial \oint(Z)}{\partial Z} \leq 0.25$$

- Now let's apply simple giant rule to $\frac{\partial O_{21}}{\partial O_{11}}$:

$$\frac{\partial O_{21}}{\partial O_{11}} = \frac{\partial O_{21}}{\partial Z} \cdot \frac{\partial Z}{\partial O_{11}}$$

$$= \frac{\partial \oint (Z)}{\partial Z} \cdot \frac{\partial Z}{\partial O_{11}}$$

$$= (0 \leq \frac{\partial \oint (Z)}{\partial Z} \leq 0.25) \cdot \left[\frac{\partial (O_{11} w_{11}^2 + b_2)}{\partial O_{11}}\right]$$

$$= (0 \leq \frac{\partial \oint (Z)}{\partial Z} \leq 0.25) \cdot [w_{11}^2]$$

- Now take care scenario $\frac{\partial \oint (Z)}{\partial Z} = 0.25$ and $w_{11}^2 = 500$

$$\frac{\partial O_{21}}{\partial O_{11}} = 0.25 \times 500$$

$$= 125$$

- Let's substitute all values derivative in equation one

$$\frac{\partial L}{\partial w_{11}^1} = 300 \times 200 \times 125 \times 1001$$

- The value of $\frac{\partial L}{\partial w_{11}^1}$ is very large. Whenever $\frac{\partial L}{\partial w_{11}^1}$ is large value then w_{11new}^1 will be *negative value*. If w_{11new}^1 is negative, then weights are jumping positive to negative side of the loss function graph. it will never reach to global minima point.

Note: What we did till now is for simple neural network, if we consider a large neural network and weights updating something like this, that will never take the global minimum point.