#Vanishing & Exploding Gradients.

In a deep neural network, the gradient of the loss with respect to early-layer weights is computed using the chain rule.

$$\boxed{\frac{\partial L}{\partial W_0} = \frac{\partial L}{\partial z_L} \prod_{l=1}^{L} \frac{\partial z_l}{\partial z_{l-1}}}$$

Each term in the product typically contains weights and activation derivatives.

Vanishing Gradient

If

$$\left| \frac{\partial z_l}{\partial z_{l-1}} \right| < 1$$

then repeated multiplication causes

$$\prod_{l=1}^{L} \frac{\partial z_l}{\partial z_{l-1}} \longrightarrow 0$$

$$\Rightarrow \frac{\partial L}{\partial W_0} \approx 0$$

Early layers learn very slowly or stop learning.

Exploding Gradient

If

$$\left| \frac{\partial z_l}{\partial z_{l-1}} \right| > 1$$

then

$$\prod_{l=1}^{L} \frac{\partial z_l}{\partial z_{l-1}} \longrightarrow \infty$$

$\Rightarrow \frac{\partial L}{\partial W_0}$ becomes very large.

This causes unstable updates and numerical overflow.

Key Mathematical Insight

$$\text{Gradient magnitude} \propto \prod (\text{weights} \times \text{activation derivatives})$$

Conclusion

- Small weights → vanishing gradients
- Large weights → exploding gradients
- Proper weight initialization is required to keep gradients stable.