Calculate and Update Weights in a Neural Network

1. Initialization of Weights

Weights are initialized randomly or using specific strategies (e.g., Xavier initialization). For this example, let's start with the following initialized weights:

$$w_1 = 0.5, \quad w_2 = 0.4, \quad w_3 = 0.3$$

Inputs:

$$x_1 = 0.6, \quad x_2 = 0.2, \quad x_3 = 0.1$$

Bias:

$$b = 0.2$$

2. Calculate the Weighted Sum (Linear Combination)

The weighted sum z is calculated using:

$$z = w_1 \cdot x_1 + w_2 \cdot x_2 + w_3 \cdot x_3 + b$$

Substituting the values:

$$z = 0.5 \cdot 0.6 + 0.4 \cdot 0.2 + 0.3 \cdot 0.1 + 0.2 = 0.61$$

3. Activation Function

Using the **Sigmoid function** for non-linearity:

$$f(z) = \frac{1}{1 + e^{-z}}$$

Substituting $z = 0.61$:

$$f(0.61) \approx 0.648$$

The output of the neuron y is:

$$y = 0.648$$

4. Calculate the Loss

Assuming the **target output** is $y_{\text{true}} = 1$, using the **Mean Squared Error (MSE)** loss function:

$$L = \frac{1}{2}(y - y_{\text{true}})^2$$

Substituting the values:

$$L = \frac{1}{2}(0.648 - 1)^2 \approx 0.062$$

5. Backpropagation (Calculate the Gradient)

To update weights, we need the **gradient of the loss function with respect to each weight**.

Calculate the Gradient for w_1:

The gradient of the loss L with respect to weight w_1 is:

$$\frac{\partial L}{\partial w_1} = \frac{\partial L}{\partial y} \cdot \frac{\partial y}{\partial z} \cdot \frac{\partial z}{\partial w_1}$$

Step-by-Step Calculation:

- **Calculate $\frac{\partial L}{\partial y}$:**

$$\frac{\partial L}{\partial y} = y - y_{\text{true}} = 0.648 - 1 = -0.352$$

- **Calculate $\frac{\partial y}{\partial z}$ (Sigmoid derivative):**

$$\frac{\partial y}{\partial z} = f(z) \cdot (1 - f(z)) = 0.648 \cdot (1 - 0.648) \approx 0.228$$

- **Calculate $\frac{\partial z}{\partial w_1}$:**

$$\frac{\partial z}{\partial w_1} = x_1 = 0.6$$

- **Total Gradient for w_1:**

$$\frac{\partial L}{\partial w_1} = -0.352 \cdot 0.228 \cdot 0.6 \approx -0.048$$

6. Update the Weight (Using Gradient Descent)

Using the gradient descent rule:

$$w_1^{\text{new}} = w_1 - \eta \cdot \frac{\partial L}{\partial w_1}$$

Assume a learning rate $\eta = 0.1$. Substituting the values:

$$w_1^{\text{new}} = 0.5 - 0.1 \cdot (-0.048) = 0.5048$$

7. Repeat for Other Weights

- For w_2:

$$\frac{\partial L}{\partial w_2} \approx -0.016, \quad w_2^{\text{new}} \approx 0.4016$$

- For w_3:

$$\frac{\partial L}{\partial w_3} \approx -0.008, \quad w_3^{\text{new}} \approx 0.3008$$

8. Final Updated Weights

After one iteration, the new weights are:

$$w_1^{\text{new}} = 0.5048, \quad w_2^{\text{new}} = 0.4016, \quad w_3^{\text{new}} = 0.3008$$