1. Feedforward Computation

Given a neural network with:

- 2 input neurons
- 1 hidden layer with 3 neurons (ReLU activation)
- 1 output neuron (Sigmoid activation)

Task:

Given input vector $x = [1, 2]$, weights, and biases:

- Compute the output step by step.

Let's define the parameters:

Input vector:

$$x = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

Weights and biases:

Hidden layer weights $W^{[1]}$:

(3 neurons × 2 inputs)

$$W^{[1]} = \begin{bmatrix} 0.1 & 0.2 \\ 0.3 & 0.4 \\ 0.5 & 0.6 \end{bmatrix}$$

Hidden layer biases $b^{[1]}$:

$$b^{[1]} = \begin{bmatrix} 0.1 \\ 0.1 \\ 0.1 \end{bmatrix}$$

1. Hidden Layer Linear Transformation

$$z^{[1]} = W^{[1]}x + b^{[1]}$$

$$= \begin{bmatrix} 0.1 & 0.2 \\ 0.3 & 0.4 \\ 0.5 & 0.6 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \end{bmatrix} + \begin{bmatrix} 0.1 \\ 0.1 \\ 0.1 \end{bmatrix} = \begin{bmatrix} 0.1(1) + 0.2(2) \\ 0.3(1) + 0.4(2) \\ 0.5(1) + 0.6(2) \end{bmatrix} + \begin{bmatrix} 0.1 \\ 0.1 \\ 0.1 \end{bmatrix}$$

$$= \begin{bmatrix} 0.1 + 0.4 \\ 0.3 + 0.8 \\ 0.5 + 1.2 \end{bmatrix} + \begin{bmatrix} 0.1 \\ 0.1 \\ 0.1 \end{bmatrix} = \begin{bmatrix} 0.6 \\ 1.2 \\ 1.8 \end{bmatrix}$$

2. Hidden Layer Activation (ReLU)

$$a^{[1]} = \text{ReLU}(z^{[1]}) = \max(0, z^{[1]})$$

$$a^{[1]} = \begin{bmatrix} \max(0, 0.6) \\ \max(0, 1.2) \\ \max(0, 1.8) \end{bmatrix} = \begin{bmatrix} 0.6 \\ 1.2 \\ 1.8 \end{bmatrix}$$

3. Output Layer Linear Transformation

$$z^{[2]} = W^{[2]}a^{[1]} + b^{[2]}$$

$$= \begin{bmatrix} 0.7 & 0.8 & 0.9 \end{bmatrix} \begin{bmatrix} 0.6 \\ 1.2 \\ 1.8 \end{bmatrix} + 0.1$$

$$= (0.7)(0.6) + (0.8)(1.2) + (0.9)(1.8) + 0.1$$

$$= 0.42 + 0.96 + 1.62 + 0.1 = 3.10$$

4. Output Activation (Sigmoid)

$$a^{[2]} = \sigma(z^{[2]}) = \frac{1}{1 + e^{-3.10}} \approx \frac{1}{1 + 0.045} \approx 0.957$$

Train a small network on **one XOR example**:

- **Input:** $x = [1, 1]$
- **Target:** $y = 0$

Network architecture:

- Hidden layer: 3 neurons (ReLU)
- Output layer: 1 neuron (Sigmoid)

🔢 Step 1: Initialize Weights & Biases

Let:

Hidden Layer Weights & Biases:

$$W_1 = \begin{bmatrix} 0.1 & -0.2 & 0.3 \\ 0.4 & 0.5 & -0.6 \end{bmatrix} \quad b_1 = [0.0, 0.0, 0.0]$$

Output Layer Weights & Bias:

$$W_2 = \begin{bmatrix} 0.7 \\ -0.8 \\ 0.9 \end{bmatrix} \quad b_2 = 0.0$$

▶️ Step 2: Forward Pass

Hidden Layer:

$$z_1 = x \cdot W_1 + b_1 = [1, 1] \cdot \begin{bmatrix} 0.1 & -0.2 & 0.3 \\ 0.4 & 0.5 & -0.6 \end{bmatrix} = [0.5, 0.3, -0.3]$$

ReLU activation:

$$a_1 = \mathrm{ReLU}(z_1) = [0.5, 0.3, 0.0]$$

Output Layer:

$z_2 = a_1 \cdot W_2 + b_2 = [0.5, 0.3, 0.0] \cdot [0.7, -0.8, 0.9] = 0.5 * 0.7 + 0.3 * (-0.8) + 0 = 0.35 - 0.24 = 0.11$

Sigmoid activation:

$$\hat{y} = \sigma(z_2) = \frac{1}{1 + e^{-0.11}} \approx 0.5275$$

📉 Step 3: Compute Loss

Using binary cross-entropy:

$$L = -[y \log(\hat{y}) + (1-y)\log(1-\hat{y})] = -\log(1-0.5275) \approx 0.747$$

Error:

$$\text{Error} = \hat{y} - y = 0.5275 - 0 = 0.5275$$

Let's **adjust output weights** opposite to error, scaled by activation:

Update output weights:

(learning rate $\eta = 0.1$)

$$W_2 = \begin{bmatrix} 0.7 \\ -0.8 \\ 0.9 \end{bmatrix} - 0.1 \cdot 0.5275 \cdot \begin{bmatrix} 0.5 \\ 0.3 \\ 0 \end{bmatrix} = \begin{bmatrix} 0.673625 \\ -0.815825 \\ 0.9 \end{bmatrix}$$

Bias:

$$b_2 = b_2 - 0.1 \cdot 0.5275 = -0.05275$$