**Applied Machine Learning**

**Spring 2025**

**Assignment # 2**

**Max Marks: 100 Due Date: March 23, 2025**

**Q1.** **Forward Propagation (20 Marks)**

Write Python code and implement the following from scratch:

Consider a neural network with the following architecture:

* **Input layer:** 2 neurons
* **Hidden layer:** 2 neurons (ReLU activation)
* **Output layer:** 1 neuron (Sigmoid activation)

You are given the following initial values:

**Inputs:**

X= [x1, x2] = [0.5, 0.8]

**Weights:**

|  |  |  |
| --- | --- | --- |
| **From → To** | **Weight** | **Value** |
| x1→h1 | w11 | 0.2 |
| x1→h2 | w12 | -0.4 |
| x2→h1 | w21 | 0.7 |
| x2→h2 | w22 | 0.1 |
| h1→o | w31 | 0.5 |
| h2→o | w32 | -0.6 |

**Biases:**

|  |  |  |
| --- | --- | --- |
| **Node** | **Bias** | **Value** |
| h1 | b1 | 0.1 |
| h2 | b2 | -0.3 |
| o | b3 | 0.2 |

### ****Activation Functions:****

* Hidden layer: **ReLU** → ReLU(x)=max(0, x)
* Output layer: **Sigmoid** → σ(x)=1 / (1+e^-x)

#### ****Tasks:****

1. Compute the activations of the hidden layer (h1, h2).
2. Compute the final output of the network.
3. Show all steps and calculations.

**Q2.** **Back Propagation (30 marks)**

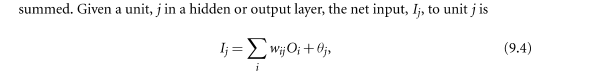
Now write Python code and implement back propagation on the previous problem from scratch. Use learning rate = 0.01. Target output Y = 1.

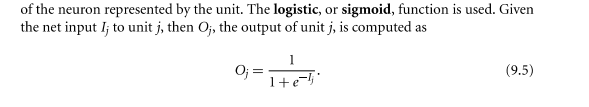
**1.** Compute Net Input and Output Calculations

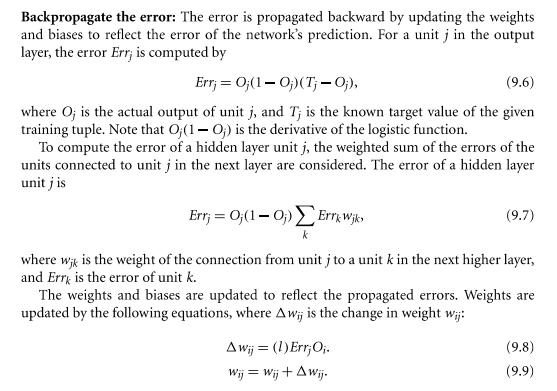
**2.** Calculate error at each node

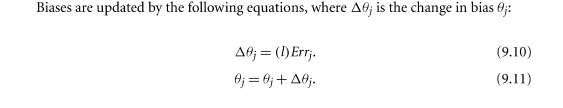
**3.** Compute the updated weight and bias values.

Use the following formulas as studied in class:









**Q3.** Implement ANN based solution for the detection and classification of Crop Weed. Then, generate your own data and test the trained system. Submit complete running code and documentation. **(50 marks)**