Deep Learning Assignment 1

Q1: Implement a Neural Network to Solve the XOR Problem

Design and train a simple feedforward neural network from scratch to solve the XOR classification problem.

- Input: Two binary inputs
- Output: Single binary output (0 or 1) indicating the XOR of the inputs
- Constraints:
  - Do not use any deep learning or machine learning libraries
  - You must implement the forward pass, backward pass (backpropagation), and weight update manually
  - Visualize the training loss

Hint:
- Use a 2-2-1 architecture (2 input neurons, 2 hidden neurons, 1 output neuron)
- Use sigmoid activation function

- When using sigmoid, remember its derivative:

```
sigmoid'(x) = sigmoid(x) * (1 – sigmoid(x))
```

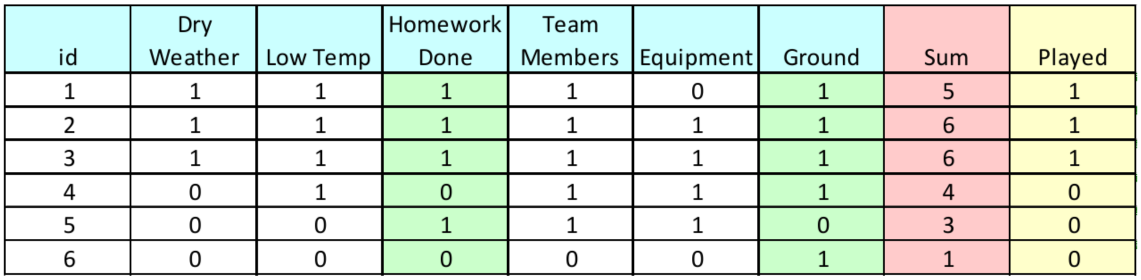
- Use binary cross-entropy loss

- Choose a small learning rate (e.g., 0.1). Too large will diverge; too small will be slow.

- Train for at least 10,000 iterations or until the loss plateaus.

- Initialize weights randomly and train using gradient descent

Q2: Predict If the Game Was Played Using a Neural Network (From Scratch)



id	Dry Weather	Low Temp	Homework Done	Team Members	Equipment	Ground	Sum	Played
1	1	1	1	1	0	1	5	1
2	1	1	1	1	1	1	6	1
3	1	1	1	1	1	1	6	1
4	0	1	0	1	1	1	4	0
5	0	0	1	1	1	0	3	0
6	0	0	0	0	0	1	1	0

NOTE: (Don’t consider Sum as a feature)

You are given a dataset with binary input conditions (e.g., weather, preparation, equipment) and a label indicating whether the game was played.

Task:

- Design a neural network from scratch to predict the 'Played' output using the input conditions
- Do not use any deep learning libraries
- Implement forward propagation, loss calculation, backpropagation, and parameter updates manually
- Report accuracy and training loss over iterations

Hint:

- Input Layer: 6 neurons (for each binary feature)
- Hidden Layer: Start with 3 to 5 neurons and use sigmoid/ReLU activation
- Output Layer: 1 neuron with sigmoid activation (for binary classification)
- Loss Function: Binary Cross-Entropy