MLPR - Lab 10

INSTRUCTIONS

**Design perceptron from scratch to classify two distributions (Classes).**

**Parameters:**

* **Number of total samples = 500**
* **Distribution 1:**
  + **mu1 = [2 2]**

**sigma1 = [0.9 -0.0255; -0.0255 0.9]**

* **Distribution 2:**
  + **mu2 = [5 5]**
  + **sigma2 = [0.5 0; 0 0.3]**
* **You can use any other distribution of your choice.**
* **Learning rate=0.1 (Hyperparameter tuneable)**
* **Bias=1**
* **Max\_Epochs/Iterations= 1000**

Submit Code, scatter plot with two classes and decision boundary and its confusion matrix as shown in reference output images.

**Step1**: Import libraries

* Numpy
* Matplotlib.pyplot/Seaborne
* Train test split
* Confusion matrix

**Step2**: Generate synthetic data for two classes. Use above given parameters for samples distribution or of your choice. Also seed(0) for reproducibility.

**Step3**: Combine both the distribution (classes) and their labels to form a dataset. Use np.vstack(), np.hstack().

**Step4**: Include bias term by adding a column of ones to feature matrix.

**Step5**: Split the dataset into train and test.

**Step6**: Write a function to design perceptron that will take data, labels, learning rate and max\_epochs as parameters.

**Step7**: Define a step activation function where it will return 1 if activation >= 0 else 0.

**Step8**: train the perceptron on training set.

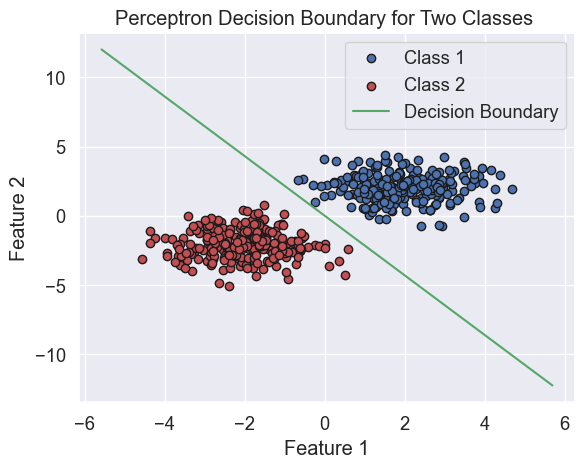
**Step9**: Make predictions using trained perceptron on test set. Tune the hyperparameters like learning rate, test size and find the optimal accurate perceptron model.

**Step10**: Plot the decision boundary between two classified class.

**Step11**: Plot the confusion matrix.

**Output Reference**

A diagram of a confusion matrix

Description automatically generated

QUESTIONS

1. What are the parallels between Biological NNs and Artificial NNs? List any three.
2. What was the McCulloch-Pitts Neuron model? List any two of its limitations.
3. How does a perceptron learn?
4. List any three activation functions and their respective output ranges.
5. Why was the classical perceptron not able to solve the XOR problem? Can you suggest a way to solve the XOR problem using neural networks but without going into a higher dimensional space?