



Faculty of Computing and Information Technology

**University of the Punjab,
Lahore**

Artificial Intelligence Lab 13

Instructor: Qamar U Zaman

Perceptron and XOR Problem Using Neural Networks

Objective:

- Implement a Perceptron for basic binary classification tasks.
 - Train a neural network to solve the XOR problem using a simple architecture.
 - Visualize the results, including data points and decision boundaries.
-

Key Concepts:

1. **Perceptron:** A single-layer neural network used for linear classification tasks.
 2. **XOR Problem:** A classical example where single-layer perceptrons fail, requiring multi-layer networks.
 3. **Visualization:** Helps in understanding decision boundaries and network behavior.
-

Tasks for Students:

1. **Perceptron Implementation:**
 - Train a Perceptron using the dataset provided.
 - Visualize the decision boundary.
 2. **XOR Problem:**
 - Use a neural network with one hidden layer to solve XOR.
 - Visualize the decision boundary using a mesh grid.
 3. **Visualizations:**
 - Plot decision boundaries for both Perceptron and XOR tasks.
-

Code Template:

```
# Perceptron Implementation
def perceptron(X, y, learning_rate, epochs):
    """
    Train a Perceptron model.
    """
    pass

def plot_perceptron_boundary(X, y, weights, bias):
    """
    Visualize the decision boundary for the Perceptron.
    """
    pass

# XOR Neural Network Implementation
def train_xor_nn(X, y):
    """
    Train a neural network to solve the XOR problem.
    """
    pass

def visualize_xor_boundary(model, X, y):
    """
    Visualize the decision boundary for the XOR problem.
    """
```

pass

Instructions:

1. For Perceptron:

- Train the Perceptron using the given dataset.
- Visualize the decision boundary after training.

2. For XOR:

- Train the neural network using the XOR dataset.
- Plot the decision boundary after training.

