MCA572– Neural Networks and Deep Learning

V MCA

20-09-2024

Regular lab Question – I

Lab Exercise 1:

You have been tasked with building simple neural networks to simulate the behavior of logic gates using a Single Layer Perceptron. This task will involve constructing, training, and testing perceptrons for the following gates: AND, OR, AND-NOT and XOR.

1. AND Gate Classification

- Scenario:
  You are tasked with building a simple neural network to simulate an AND gate using a Single Layer Perceptron. The AND gate outputs 1 only if both inputs are 1; otherwise, it outputs 0.
- Lab Task: Implement a Single Layer Perceptron using Python in Google Colab to classify the output of an AND gate given two binary inputs (0 or 1). Follow these steps:
- Create a dataset representing the truth table of the AND gate.
- Define the perceptron model with one neuron, including the activation function and weights initialization(Try both random weights and defined weights).
- Train the perceptron using a suitable learning algorithm (e.g., gradient descent).
- Test the model with all possible input combinations and display the results.

Questions:

- How do the weights and bias values change during training for the AND gate?
- Can the perceptron successfully learn the AND logic with a linear decision boundary?

2. OR Gate Classification

- Scenario:
  Your next task is to design a perceptron that mimics the behavior of an OR gate. The OR gate outputs 1 if at least one of its inputs is 1.
- Lab Task: Using Google Colab, create a Single Layer Perceptron to classify the output of an OR gate. Perform the following steps:
- Prepare the dataset for the OR gate's truth table.
- Define and initialize a Single Layer Perceptron model.
- Implement the training process and adjust the perceptron's weights.
- Validate the perceptron's performance with the OR gate input combinations.

Questions:

- What changes in the perceptron's weights are necessary to represent the OR gate logic?
- How does the linear decision boundary look for the OR gate classification?

3. AND-NOT Gate Classification

- Scenario:
  You need to implement an AND-NOT gate, which outputs 1 only if the first input is 1 and the second input is 0.
- Lab Task: Design a Single Layer Perceptron in Google Colab to classify the output of an AND-NOT gate. Follow these steps:
- Create the truth table for the AND-NOT gate.
- Define a perceptron model with an appropriate activation function.
- Train the model on the AND-NOT gate dataset.
- Test the model and analyze its classification accuracy.

Questions:

- What is the perceptron's weight configuration after training for the AND-NOT gate?
- How does the perceptron handle cases where both inputs are 1 or 0?

4. XOR Gate Classification

- Scenario:
  The XOR gate is known for its complexity, as it outputs 1 only when the inputs are different. This is a challenge for a Single Layer Perceptron since XOR is not linearly separable.
- Lab Task: Attempt to implement a Single Layer Perceptron in Google Colab to classify the output of an XOR gate. Perform the following steps:
- Create the XOR gate's truth table dataset.
- Implement the perceptron model and train it using the XOR dataset.
- Observe and discuss the perceptron's performance in this scenario.

Questions:

- Why does the Single Layer Perceptron struggle to classify the XOR gate?
- What modifications can be made to the neural network model to handle the XOR gate correctly?

Instructions for Lab Work in Google Colab:Setup: Import necessary libraries (e.g., NumPy) and create datasets for each logic gate.Model Implementation: Write the code for the Single Layer Perceptron, including forward and backward passes.Training: Use a learning algorithm to adjust weights and biases.Evaluation: Test the perceptron's performance for each logic gate and plot the decision boundaries where applicable.Analysis: Document the results and answer the provided questions.

Program Evaluation Rubrics

Evaluation Criteria	
5 marks	C1-Implementation, Correctness and Complexity
2 marks	C2-Documentation and Visualization
3 marks	C3-Concept Clarity and Explanation

General Instructions

1. The file you have to save with your name, last 3 digits of register number and program number “Aaron_201_Lab1”.
2. The implemented code you have to upload in Github and in the Google Classroom in the given scheduled time.
3. Failure to upload within the allotted time will result in the loss of all marks for the corresponding lab exercise.