Computer Science & Information Systems

Machine Learning - Lab sheet - Module 7

Exercise 1 - Perceptron

1 Objective

The objective is to

- implement a Perceptron.
- train the Perceptron for OR and XOR gates.

2 Steps to be performed

Tool Python3

Libraries required numpy, matplotlib

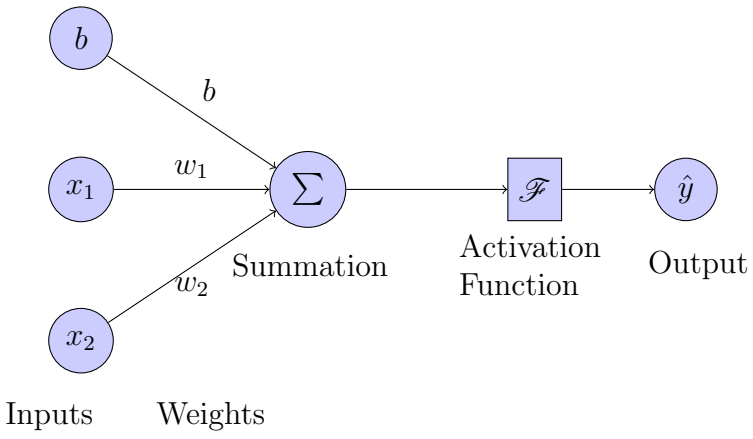
Input OR and XOR gate boolean data

Machine Learning Model Neural Network - Perceptron

Implementation ML_Lab 14 Perceptron.ipynb

Steps .

- Import required Python libraries.
- Create the dataset as numpy arrays.
- Visualize the dataset.
- Define the neural network architecture as shown below.



- Initialize the parameters, weights and bias, of the network.
- Implement forward propagation.
  For one example $x^{(i)}$

– Compute the hypothesis.

$$z^{(i)} = Wx^{(i)} + b \tag{1}$$

– Compute the activation

$$a^{(i)} = \begin{cases} 1 & \text{if } z^{(i)} \geq 0 \\ -1 & \text{if } z^{(i)} < 0 \end{cases} \tag{2}$$

- Compute cost of the network.

$$J = \frac{1}{m} \sum_{i=1}^{m} \max\left(0, -y^{(i)} x^{(i)} W\right) \tag{3}$$

- Update the parameters of the network using the learning rule.
  If $x^{(i)}$ is misclassified

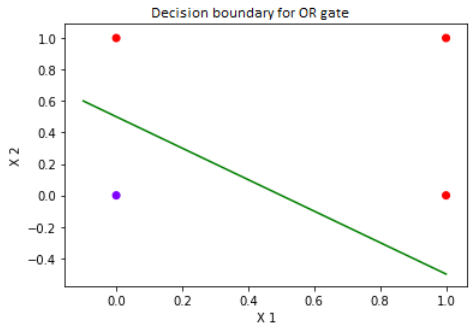
$$W = W - \frac{\alpha}{2} \sum_{i=1}^{m} \left(a^{(i)} - y^{(i)}\right) x^{(i)} \tag{4}$$

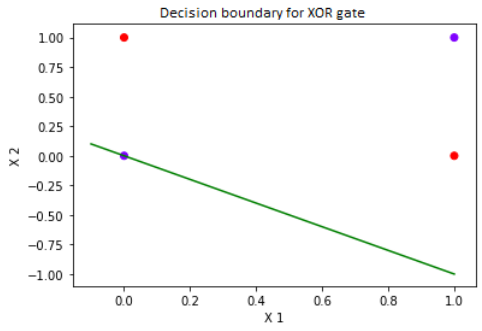
$$b = b - \frac{\alpha}{2} \sum_{i=1}^{m} \left(a^{(i)} - y^{(i)}\right) \tag{5}$$

- Visualize the decision boundary and cost function.
- Measure the performance of the model.

3 Results

- The Perceptron was implemented.
- The parameters that will predict the desired output for OR and XOR gates were learned.
- The decision boundary and cost were plotted for OR and XOR gates.





4 Observation

- The trained Perceptron predicted the OR gate outputs with 100% accuracy.
- The trained Perceptron predicted the XOR gate outputs with 50% accuracy. The Perceptron could not converge for the XOR gate data.