04/10/2021

Machinelearning

Lab-5

Pranava
Raman
BMS
2019103555

Aim

To implement a single layer perceptron to solve
1. OR Problem
2. AND problem

and check if it can be modeled to solve XOR problem.

The perceptron algorithm

- Initialization :-
  - set all the weights w_{ij} to small (+ve and -ve) random numbers.

- Training :-
  - for T iterations or until all the outputs are correct :-
    - for each input vector :-
      - compute the activation of each neuron j using 'activation function' g :-

$$y_j = g\left(\sum_{i=0}^{m} w_{ij} x_i\right) = \begin{cases} 1, & \text{if } \sum_{i=0}^{m} w_{ij} x_i > 0 \\ 0 & \text{if } \sum_{i=0}^{m} w_{ij} x_i \leq 0 \end{cases}$$

- update each of the weights individually using :-

$$w_{ij} \leftarrow w_{ij} - \eta(y_j - t_j) x_i$$

* Recall :-

- compute activation of each neuron j using:-

$$y_j = g\left(\sum_{i=0}^{m} w_{ij} x_i\right) = \begin{cases} 1, & \text{if } \sum w_{ij} x_i > 0 \\ 0, & \text{if } \sum w_{ij} x_i \leq 0 \end{cases}$$

Result

* Given a large enough ~~of~~ number of iterations, the single layer perceptron was able to solve both the 'OR' and 'AND' problem.

* But since they can work only on linearly separable problems, XOR could not be solved using ~~a~~ single layer perceptron.