SNS COLLEGE OF TECHNOLOGY

Coimbatore-35
An Autonomous Institution
Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A++' Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

DEPARTMENT OF COMPUTER APPLICATIONS

23CAT702 – MACHINE LEARNING

II YEAR III SEM

UNIT II – LINEAR MODELS

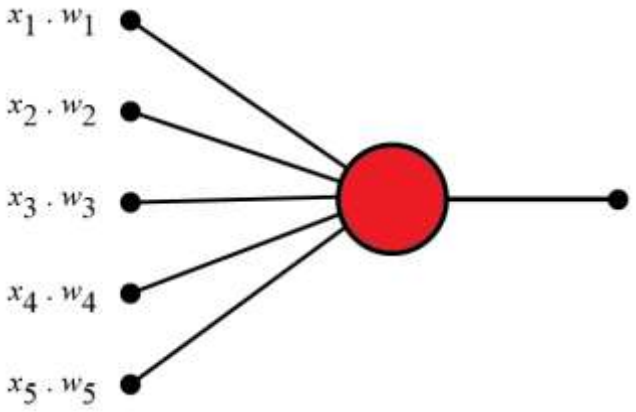
TOPIC 14 – Perceptrons-Multilayer Neural Networks

Perceptron

1. **Frank Rosenblatt** (1928 – 1971) was an American psychologist notable in the field of Artificial Intelligence.
2. In **1957** he started something really big. He "invented" a **Perceptron** program, on an IBM 704 computer at Cornell Aeronautical Laboratory.
3. Scientists had discovered that brain cells (**Neurons**) receive input from our senses by electrical signals.
4. The Neurons, then again, use electrical signals to store information, and to make decisions based on previous input.
5. Frank had the idea that **Perceptrons** could simulate brain principles, with the ability to learn and make decisions.

Perceptron

1. The original **Perceptron** was designed to take a number of **binary** inputs, and produce one **binary** output (0 or 1).
2. The idea was to use different **weights** to represent the importance of each **input**, and that the sum of the values should be greater than a **threshold** value before making a decision like **true** or **false** (0 or 1).



Perceptron Example

1. Imagine a perceptron (in your brain).
2. The perceptron tries to decide if you should go to a concert.
3. Is the artist good? Is the weather good?
4. What weights should these facts have?

Criteria	Input	Weight
Artists is Good	**x1** = 0 or 1	**w1** = 0.7
Weather is Good	**x2** = 0 or 1	**w2** = 0.6
Friend will Come	**x3** = 0 or 1	**w3** = 0.5
Food is Served	**x4** = 0 or 1	**w4** = 0.3
Alcohol is Served	**x5** = 0 or 1	**w5** = 0.4

The Perceptron Algorithm

1. Set a threshold value
2. Multiply all inputs with its weights
3. Sum all the results
4. Activate the output

1.Set a threshold value: Threshold = 1.5

2. Multiply all inputs with its weights:

x1 * w1 = 1 * 0.7 = 0.7

x2 * w2 = 0 * 0.6 = 0

x3 * w3 = 1 * 0.5 = 0.5

x4 * w4 = 0 * 0.3 = 0

x5 * w5 = 1 * 0.4 = 0.4

3. Sum all the results: 0.7 + 0 + 0.5 + 0 + 0.4 = 1.6 (The Weighted Sum)

4. Activate the Output: Return true if the sum > 1.5 ("Yes I will go to the Concert")

EXAMPLE

```
const threshold = 1.5;
const inputs = [1, 0, 1, 0, 1];
const weights = [0.7, 0.6, 0.5, 0.3, 0.4];

let sum = 0;
for (let i = 0; i < inputs.length; i++) {
  sum += inputs[i] * weights[i];
}

const activate = (sum > 1.5);
```

Reference

1. https://w3schools.com
2. https://www.analyticsvidhya.com/
3. https://www.geeksforgeeks.org/
4. https://www.section.io/engineering-education/using-a-hard-margin-vs-soft-margin-in-support-vector-machines/#support-vector-machines-with-a-soft-margin