NEURAL NETWORK (PCC-CSE-401G)

UNIT 1

Overview of biological neurons: Structure of biological neuron, neurobiological analogy, Biological neuron equivalencies to artificial neuron model, Evolution of neural network.

Activation Functions: Threshold functions, Signum function, Sigmoid function, Tan-hyperbolic function, Stochastic function, Ramp function, , Linear function, Identity function.

ANN Architecture: Feed forward network, Feed backward network, single and multilayer network, fully recurrent network,

Course Objectives:

1. To understand the different issues involved in the design and implementation of a Neural Networks.
2. To study the basic of neural network and its activation functions.
3. To understand and use of perceptron and its application in real world
4. To develop an understanding of essential NN concepts such as: learning, feed forward and feed backward
5. To design and build a simple NN model to solve a problem

UNIT 2

McCulloch and Pits Neural Network (MCP Model): Architecture, Solution of AND, OR function using MCP model, Hebb Model: Architecture, training and testing, Hebb network for AND function.

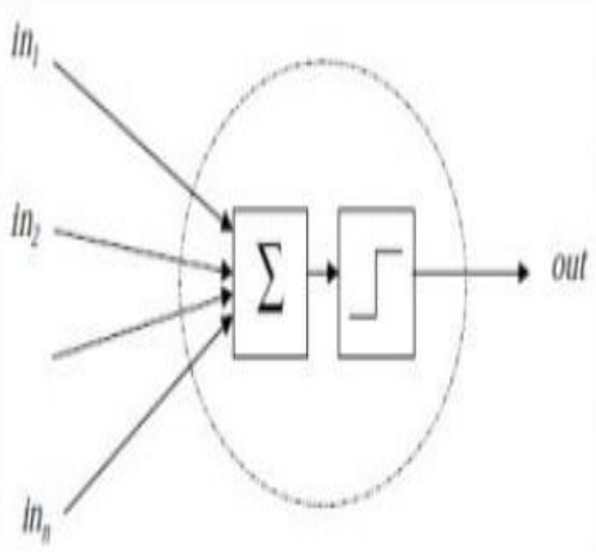
Perceptron Network: Architecture, training, Testing, single and multi-output model, Perceptron for AND function

Linear function, application of linear model, linear seperatablity, solution of OR function using liner seperatablity model.

- 1943 McCulloch and Pitts proposed the McCulloch-Pitts neuron model.
- 1949 Hebb published his book *The Organization of Behavior, in which the Hebbian* learning rule was proposed.
- 1958 Rosenblatt introduced the simple single layer networks now called Perceptrons.
- 1969 Minsky and Papert's book *Perceptrons demonstrated the limitation of single* layer perceptrons, and almost the whole field went into hibernation.
- 1982 Hopfield published a series of papers on Hopfield networks.
- 1982 Kohonen developed the Self-Organising Maps that now bear his name
- 1986 The Back-Propagation learning algorithm for Multi-Layer Perceptrons was rediscovered and the whole field took off again.
- 1990s The sub-field of Radial Basis Function Networks was developed.
- 2000s The power of Ensembles of Neural Networks and Support Vector Machiness becomes apparent.

The McCulloch-Pitts Neuron

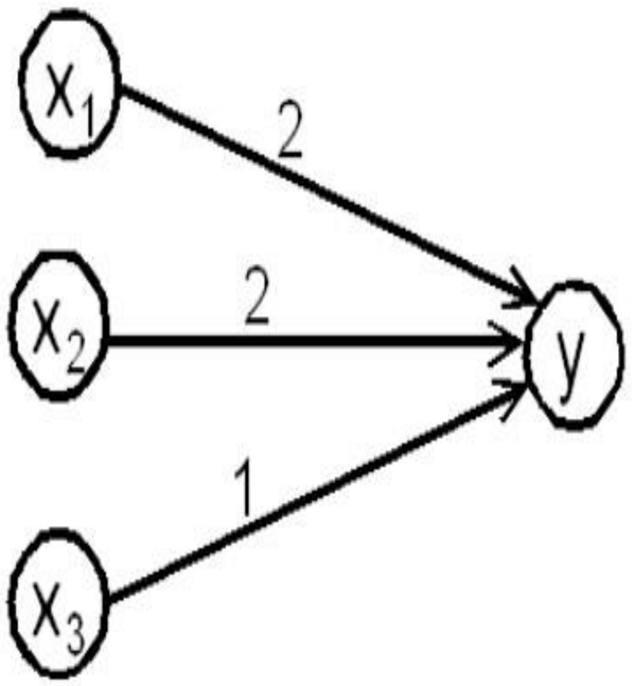
- The McCullock-Pitts Neurons
  - Vastly simplified model of real neurons (Threshold Logic Unit)



Characteristics

- The activation of a McCulloch Pitts neuron is binary.
- Neurons are connected by directed weighted paths.
- A connection path is excitatory if the weight on the path is positive else its inhibitory.
- All excitatory connections to a neuron have the same weights.
- Each neuron has a fixed threshold:
  - f(n) = 1 if n >= θ
    0 if n < θ
- The threshold is set so that inhibition is absolute.

A McCulloch-Pitts Neuron



Examples

- Train a McCulloch-Pitts neural network to perform the OR function.
- Train a McCulloch-Pitts neural network to perform the AND function.
- Train a McCulloch-Pitts neural network to perform the AND NOT function.
- Train a McCulloch-Pitts neural network to perform the XOR function.

- McCulloch-Pitts Neuron Equation in terms of input and output can be written as:

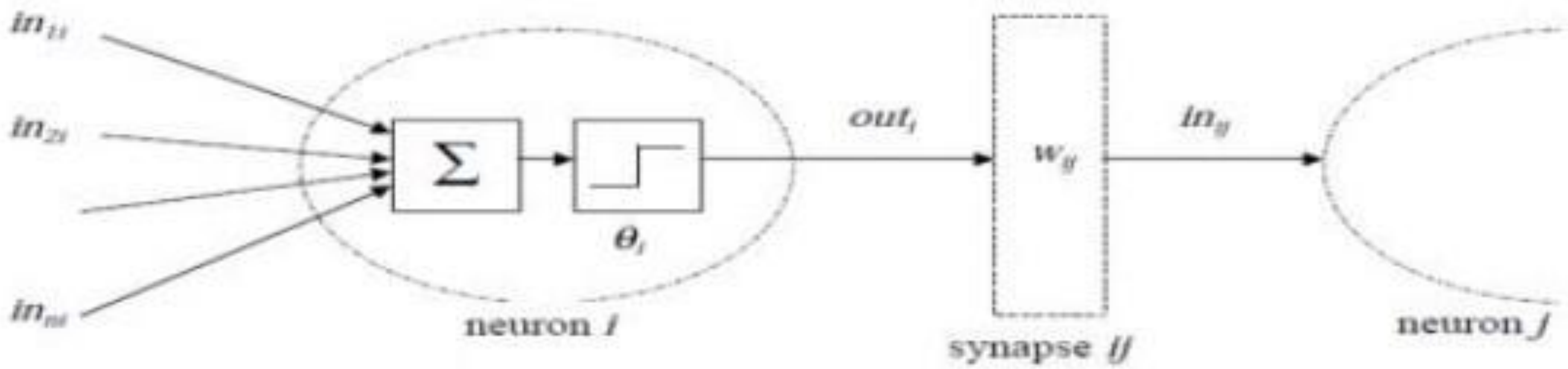
$$out = \operatorname{sgn}\left(\sum_{i=1}^{n} in_i - \theta\right)$$

- Θ is threshold used to squash neuron's output

$$out = 1 \quad \text{if} \sum_{k=1}^{n} in_k \geq \theta \qquad out = 0 \quad \text{if} \sum_{k=1}^{n} in_k < \theta$$

Networks of McCulloch-Pitts Neuron

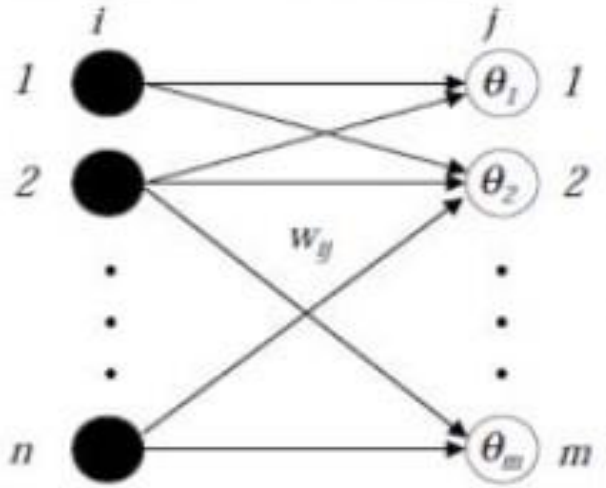
- We may have many neurons labeled by indices *k, i, j* and activation flows between them via synapses with strengths w_{ki} , w_{ij} :



$$in_{ki} = out_k w_{ki} \qquad out_i = \text{sgn}\left(\sum_{k=1}^{n} in_{ki} - \theta_i\right) \qquad in_{ij} = out_i w_{ij}$$

The Perceptron

- Any number of McCulloch-Pitts neurons can be connected together in any way.
- An arrangement of one input layer of McCulloch-Pitts neurons feeding forward to one output layer of McCulloch-Pitts neurons is known as a ***Perceptron*.**



$$out_j = \operatorname{sgn}\left(\sum_{i=1}^{n} out_i w_{ij} - \theta_j\right)$$

Implementing Logic Gates using McCulloch-Pitts Neurons

- Logical operators AND, OR and NOT can be implemented using MP neurons. We can easily find it with inception.

NOT

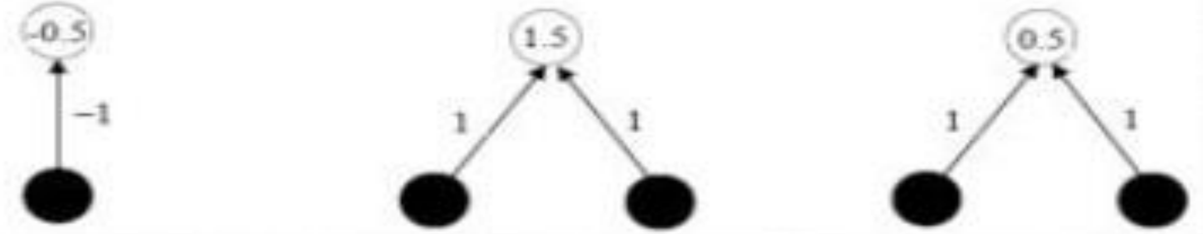
in	out
0	1
1	0

AND

in_1	in_2	out
0	0	0
0	1	0
1	0	0
1	1	1

OR

in_1	in_2	out
0	0	0
0	1	1
1	0	1
1	1	1

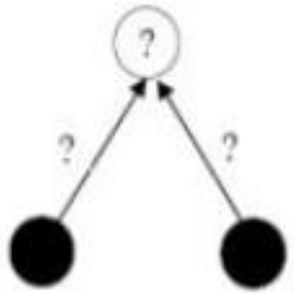


Need to find Weights Analytically

- XOR function

XOR

in_1	in_2	out
0	0	0
0	1	1
1	0	1
1	1	0



- Required to calculate the suitable parameters instead of finding solution by trial and error.
- It is required to calculate the weights and thresholds.

Finding Weights Analytical for the AND Network

- We have two weights w_1 and w_2 and the threshold Θ, and for each training pattern we need to satisfy

$$out = \mathrm{sgn}(w_1 in_1 + w_2 in_2 - \theta)$$

  - We have four inequalities

in_1	in_2	out
0	0	0
0	1	0
1	0	0
1	1	1

$$\Rightarrow \quad \begin{array}{l} w_1 0 + w_2 0 - \theta < 0 \\ w_1 0 + w_2 1 - \theta < 0 \\ w_1 1 + w_2 0 - \theta < 0 \\ w_1 1 + w_2 1 - \theta \geq 0 \end{array} \quad \Rightarrow \quad \begin{array}{c} \theta > 0 \\ w_2 < \theta \\ w_1 < \theta \\ w_1 + w_2 \geq \theta \end{array}$$

  - There are infinite number of solutions for AND, OR and NOT networks.

Perceptron's Limitations

- XOR function

in_1	in_2	out
0	0	0
0	1	1
1	0	1
1	1	0

$\Rightarrow$

$$\begin{aligned} w_1 0 + w_2 0 - \theta &< 0 \\ w_1 0 + w_2 1 - \theta &\geq 0 \\ w_1 1 + w_2 0 - \theta &\geq 0 \\ w_1 1 + w_2 1 - \theta &< 0 \end{aligned}$$

$\Rightarrow$

$$\begin{aligned} \theta &> 0 \\ w_2 &\geq \theta \\ w_1 &\geq \theta \\ w_1 + w_2 &< \theta \end{aligned}$$

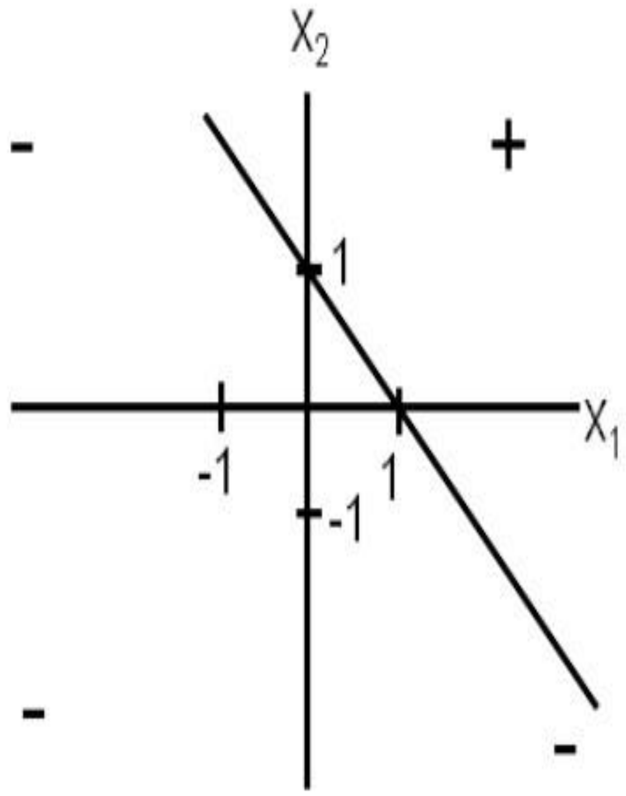
- What is problem??
  - We need more complex networks
  - Or we need different activation/threshold/transfer functions
  - We need to learn parameters

Linear Separability

- A single layer neural network can only learn linear separable problems.
- Multilayer nets using a linear activation function have the same problem.
- In linear separable problems the region where y is positive, i.e. the neuron fires, is separated from the region where y is negative, i.e where the neuron does not fire, by the line :

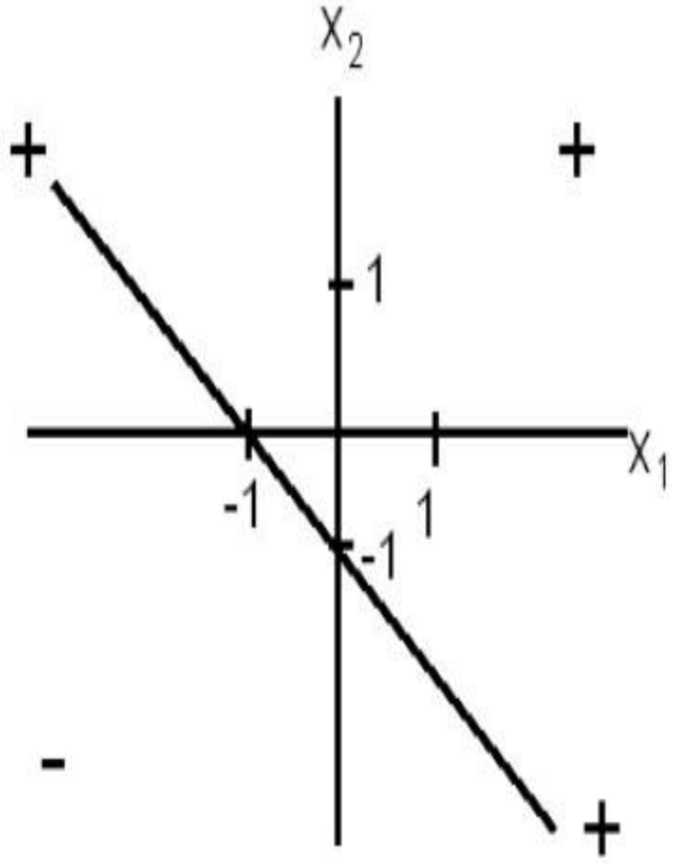
$$x_2 = -\frac{w_1}{w_2}x_1 - \frac{b}{w_2}$$

Graph for the AND Function



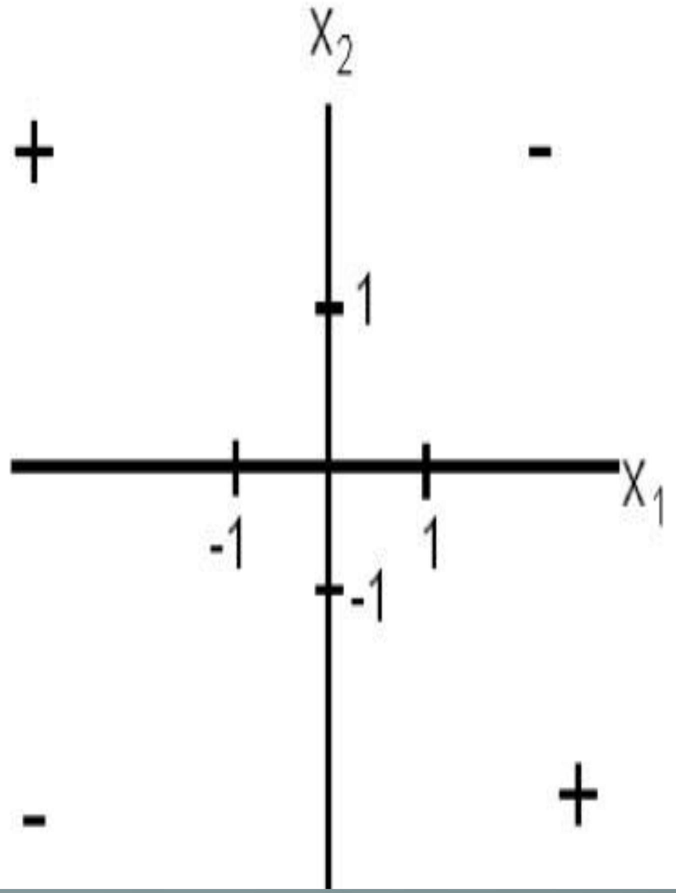
The AND function is linearly separable

Graph for the OR Function



The OR function
is linearly separable

Graphic for the XOR Function

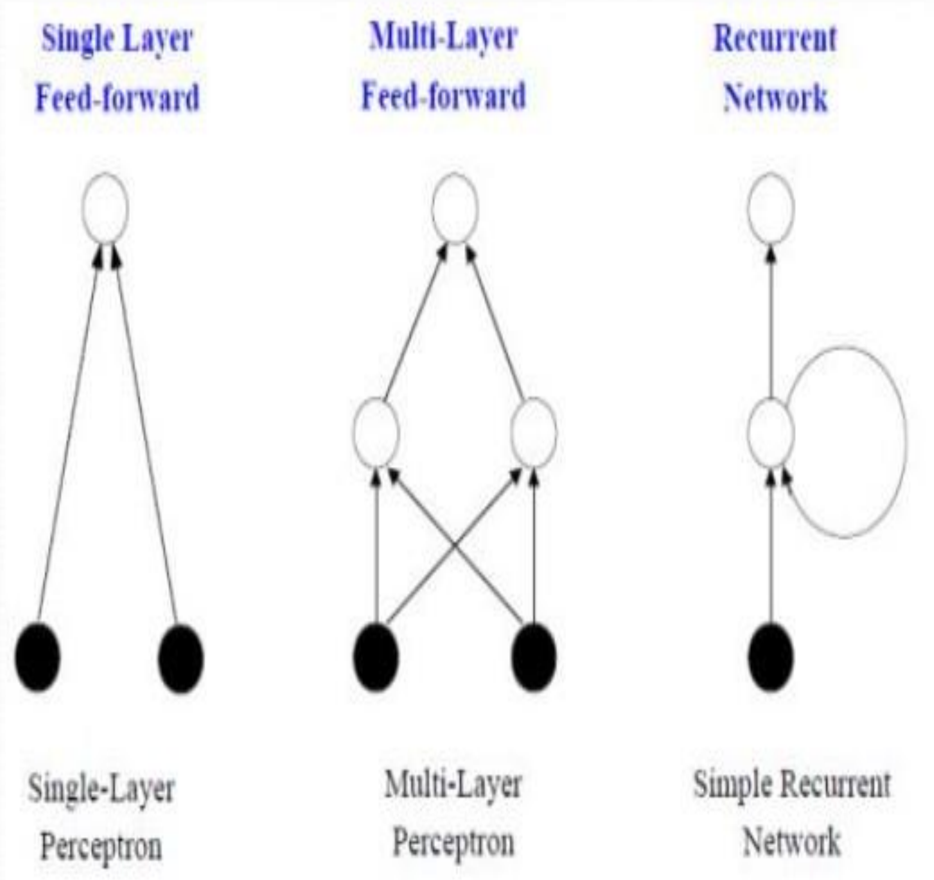


The XOR function
is not linearly separable

Structures of ANN

- ANN is a weighted directed graph considering the activation flowing between the processing units through one way connections.
- Three types of ANNs
  - Single-Layer Feed-forward NNs
    - One input layer, one output layer with no feedback connections (a simple perceptron)
  - Multi-Layer Feed-forward NNs
    - One input layer, one output layer, one or more hidden layers with no feedback connections (A multilayer perceptron)
  - Recurrent NNs
    - The network has atleast one feedback connection. May or may not have hidden units (A simple Recurrent Network)

Network's Structures Examples



THANKS