AI Unit-3 - Connectionist Models

Path:

- Bio-Inspired Computing
- Artificial Neural Network (ANN)
- Architectures of ANN
- Hopfield model of ANN, Parallel relaxation
- Linearly separable problems, Single Perceptron
- Non linearly separable problems, Fixed increment perceptron learning
- Multi layer perceptron, Back propagation in multi layer perceptron
- General Learning Paradigms: Supervised and Unsupervised Learning
- Applications of ANN and Cases.

→ Intro to ANN:

- Artificial Neural Network
- An artificial neural network is an attempt to simulate of neurons (Nodes) that make a human brain so that the computer will be able to learn things and make decisions in a humanike manner.
- ANNs are created by programming regular computers to behave as though they are interconnected brain cells.

→ Inspired Computing:

- Bio-inspired computing, short for biologically inspired computing, is a field of study which seeks to solve computer science problems using model of biology.
- It relates to connectionism, social behavior, and emergence.
- Includes models such as
  - Artificial Neural Network (ANN)
  - Genetic Algorithm (GA)
  - Swarm Intelligence (SI)

- Characterized by:
  - A large no. of very simple neuron like processing elements.
  - A large no. of weighted connection between the elements. The weights encode the knowledge of a network.
  - Highly parallel and distributed control.
  - Emphasis on learning internal representation automatically.

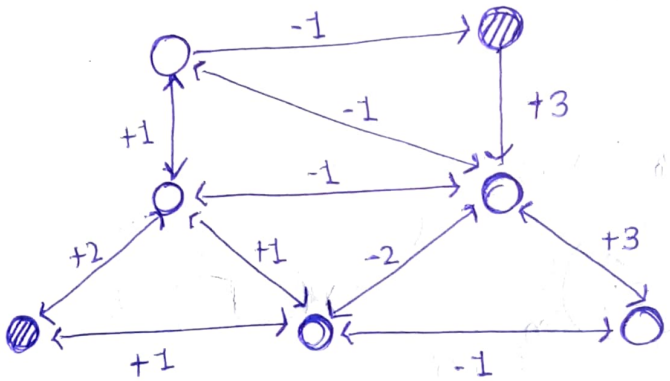
→ Architectures of ANN:

- Hopfield network
- Perceptron
- Multi-layer perceptron
- Self Organizing Network

- Hopfield network

In a Hopfield network, all processing units/elements are in two states either active or inactive.

- units are connected to each other with weighted connections.
- Two parts : 1. Training Algo
  2. Testing Algo
- Two states : 1. Active 2. Inactive



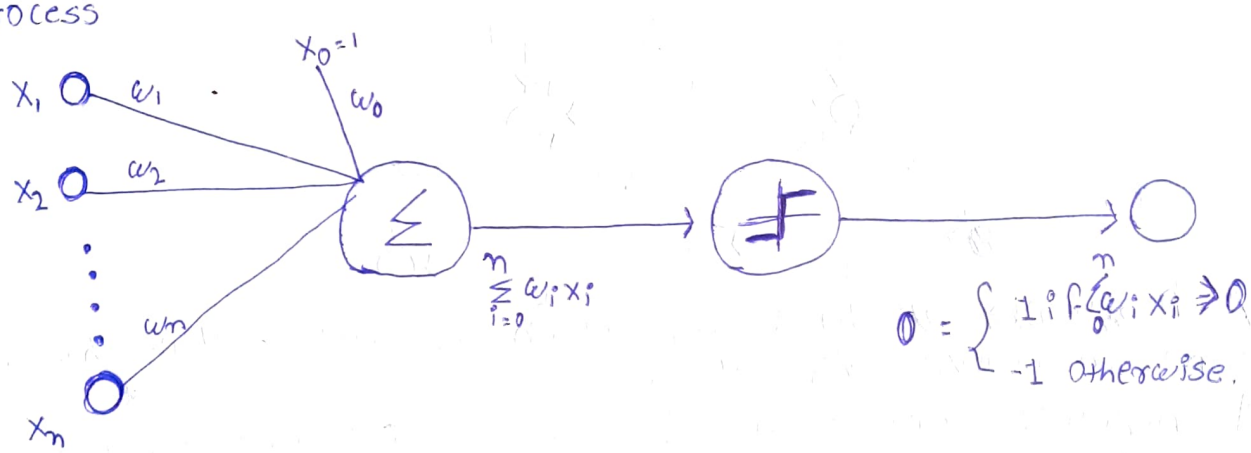
- A positively weighted connection indicates that the units tend to active each other.
- A negetive connection allows an active unit to deactivate a neighboring unit.
- A random unit is chosen.
- If any of its neighbors are active, the unit computes the sum of weights on the connections to those active neighbors.
- If sum is positive, the unit becoms active else new random unit is chosen.
- This process will continue till the network become stable That is no unit can change its status. This process is known as parallel relaxation.

Perceptron - ANN

- A perceptron unit is used to build the ANN system.
- A perceptron takes a vector of real-valued inputs, calculates a linear combination of these inputs, then outputs a 1 if the result is greater then some threshold and -1 otherwise.
- More precisely, given input x1 through x_n, the o/p $O(x_1, \ldots, x_n)$ computed by perceptron is

$$O(x_1 \ldots, x_n) = \begin{cases} 1 \text{ if } w_0 + w_1 x_1 + w_2 x_2 + \ldots + w_n x_n > 0 \\ -1 \text{ otherwise} \end{cases}$$

- where each w_i is a real-valued constant, or weight, that determines the contribution of input x_i to the perceptron output.
- Process



- This process is repeated, iteration through the training examples as many times as needed until the perceptron classifies all training example correctly.
- Weights are modified at each step according to the perceptron training rule, which revises the weight w_i associated with input x_i according to rule

$$w_i \leftarrow w_i + \Delta w_i$$

where,

$$\Delta w_i \leftarrow \eta (t - o) x_i$$

w_i = weight
x_i = I/P
o = Actual o/p
t = target o/p

• Logical 'AND' gate perception training rule

$w_1 = 1.2$, $w_2 = 0.6$ Threshold = 1 and Learning Rate $\eta = 0.5$

A	B	A ^ B
0	0	0
0	1	0
1	0	0
1	1	1

1. A = 0, B = 0 & Target = 0
   - $w_i x_i = 0 * 1.2 + 0 * 0.6 = 0$
   - This is not greater than the threshold of 1, so O/P = 0.

2. A = 0, B = 1 & Target = 0
   - $w_i x_i = 0 * 1.2 + 1 * 0.6 = 0.6$
   - Not greater than the threshold of 1, so O/P = 0.

3. A = 1, B = 0 & Target = 0
   - $w_i x_i = 1 * 1.2 + 0 * 0.6 = 1.2$
   - greater than the threshold of 1, so O/P = 1.

So, $w_i = w_i + \eta (t - O) x_i$

$w_1 = 1.2 + 0.5 (0 - 1) 1$
$= 0.7$

$w_2 = 0.6 + 0.5 (0 - 1) 0$
$= 0.6$

here, $w_i = 1.2$ or 0.6 (weight), $X_i = 1$ or 0 (A, B)

t = target O/P, O = Actual O/P

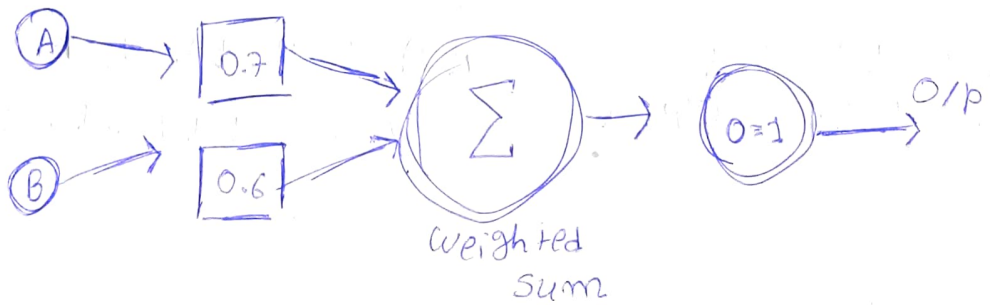
So now $w1 = 0.6$ & $w2 = 0.7$

1. A = 0, B = 0, T = 0
   - $w_i x_i = 0$ & O/P = 0

2. A = 0, B = 1, T = 0
   - $w_i x_i = 0.7 < 1$ & O/P 0

3. A = 1, B = 0, T = 0
   - $w_i x_i = 0.6 < 1$ & O/P = 0

4. A = 1, B = 1, T = 1
   - $w_i x_i = 0.6 + 0.7 = 1.3 > 1$ ✓ = O/P = 1



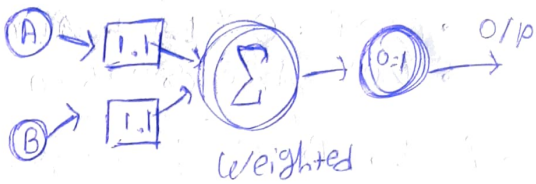
Logical or gate perceptron training rule.

$w1 = 0.6$, $w2 = 0.6$ Threshold = 1 & Learning rate $n = 0.5$

A	B	A+B
0	0	0
0	1	1
1	0	1
1	1	1

1. A=0, B=0 & Target = 0
   - $w_i x_i = 0 * 0.6 + 0 * 0.6 = 0$
   - This is not greater than the threshold of 1, so O/P = 0.

2. A = 0, B = 1 & Target 1.
   - $w_i x_i = 0 * 0.6 + 1 * 0.6 = 0.6$
   - O/P = 0 so update.



$w_i = w_i + n(t-o)x_i$

$w_1 = 0.6 + 0.5(1-0)0 = 0.6$

$w_2 = 0.6 + 0.5(1-0)1 = 1.1$

So, now

$w1 = 0.6$, $w2 = 1.1$ & Learning rate $n = 0.5$

1. A = 0, B = 0 & T = 0
   - $w_i x_i = 0 * 0.6 + 0 * 1.1 = 0$
   - O/P = 0
2. A = 0, B = 1 & T = 1
   - $w_i x_i = 0 * 0.6 + 1 * 1.1 = 1.1 > 1$
   - O/P = 1
3. A = 1, B = 0 & T = 1
   - $w_i x_i = 1 * 0.6 + 0 * 1.1 = 0.6 < 1$
   - O/P = 0

$w_i = w_i + n(t-o)x_i$

$w_1 = 0.6 + 0.5(1-0)1 = 1.1$

$w_2 = 1.1 + 0.5(1-0)0 = 1.1$

So, now

$w1 = 1.1$, $w_2 = 1.1$ & $n = 0.5$

1. A = 0, B = 0 & T = 0
   - $w_i x_i = 0$ & O/P = 0
2. A = 0 & B = 1 & T = 1
   - $w_i x_i = 1.1 > 1$ & O/P = 1
3. A = 1, B = 0 & T = 1
   - $w_i x_i = 1.1 > 1$ & O/P 1
4. A = 1, B = 1 & T = 1

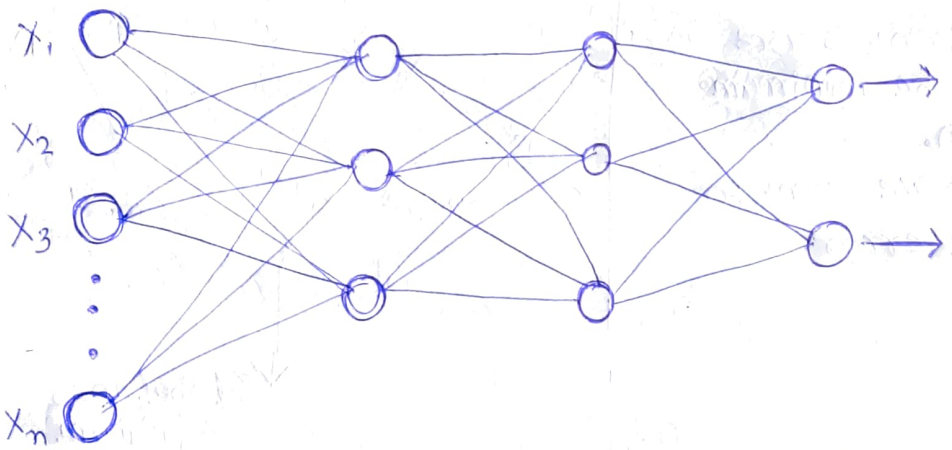
   $w_i x_i = 1.1 + 1.1 = 2.2 > 1$

   O/P = 1

> Multi layer perceptron:

A multilayer perceptron is a perceptron with multi layer It has: Input Layer, Output layer & Hidden layer. Each neuron compute an activation Function.

- **Input layer**
  1. Intro. I/P values in to the network
  2. No activation fun. or other processing.
- **Hidden layer**
  1. Perform classification of features
  2. Two hidden layers are sufficent to slove any Prob
  3. Featurs imply more layers may be later.
- **Output layer**
  1. Functionality just like hidden layer
  2. O/P are passed onto the world.

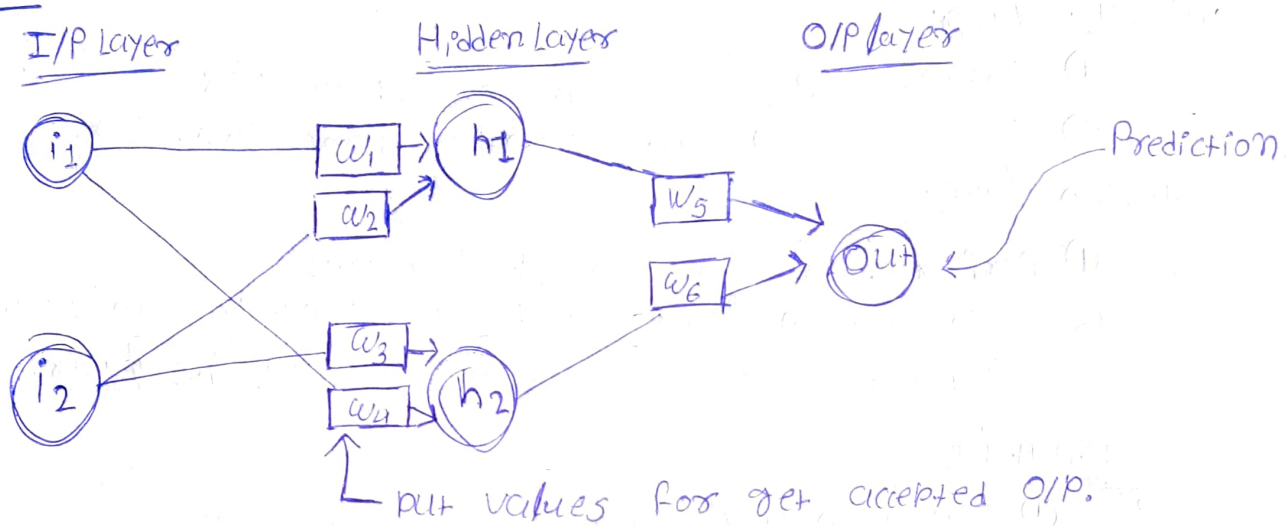


Input Layer Hidden Layer Output Layer

> Back Propagation:

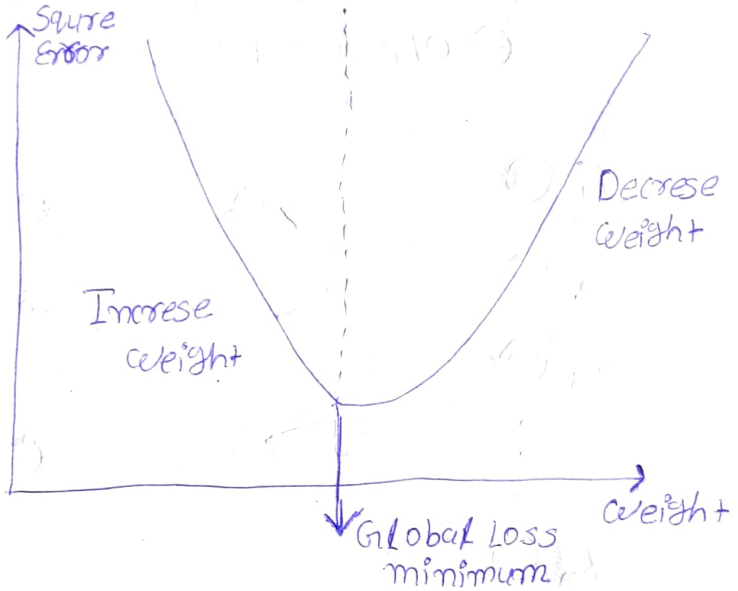
Back propagation is a supervised learning algo., for training Neural Network.

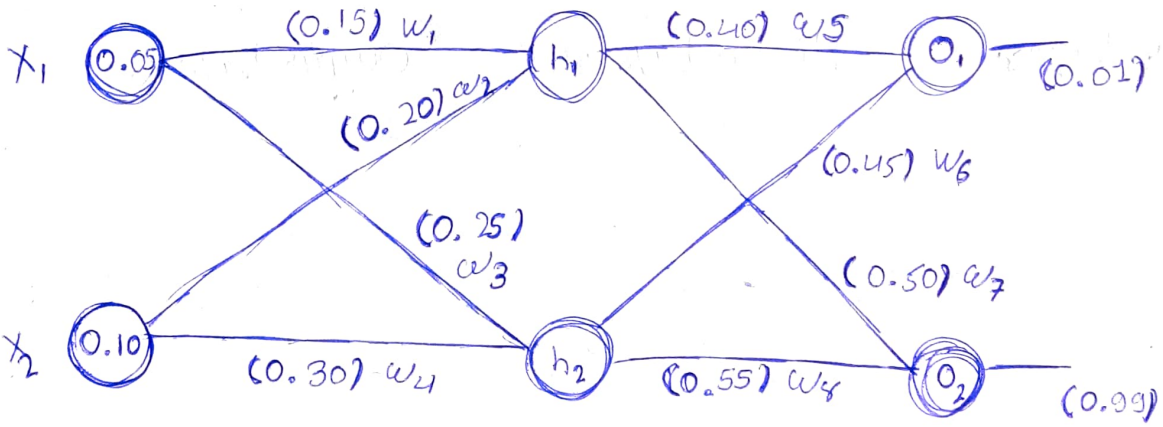
Why?



→ Gradient Descent

- Update the weight using gradient descent.
- Gradient descent is used for finding the minimum of a function.
- In our case we want to minimize the error function.





$b_1 = (0.35)$ $b_2 = (0.60)$

$$E_{total} = E_{O_1} + E_{O_2} = 0.2983$$

Supervised Learning	Unsupervised Learning
① Supervised Learning method involves the training of the system or machine where the training set along with the Target pattern (output pattern) is provided to the system for performing a task.	① Unsupervised learning method does not involve the target output which means no training is provided to the system. The system has to learn by its own through determining & adapting according to the structural characterstics in the I/P patterns.
② Supervised learning method deals with the labelled data where the output data patterns are known to the system.	② Unsupervised learning method work with unlabelled data which the output is just based on the collection of perception.
③ The outcome of Supervised learning is more accurate & reliable.	③ unsupervised learning generates moderate accurat but reliable result.
④ Supervised learning method is less complex.	④ Unsupervised learning method is more complex.
⑤ Classification, regression are the types of problems solved under the supervised learning method.	⑤ Clustering, Association rule mining are the problems sloved under the unsupervised learning.
⑥ Supervised learning is the technique of accomplishing a task by provide training, input & output data patterns to the system.	⑥ Unsupervised learning is a self-learning technique in which the system has to discover the fetch of the input population by its own.
⑦ Supervised learning Employees the offline analysis.	⑦ Unsupervised learning Employees the real-time Analysis.

⇒ Applications of ANN:

- Image Processing
  - Face recognition
  - Face detection - search for the face in given image
  - Image recognization & image reconstruction
  - Noise removed
- Speech processing
  - speech to text
  - Text to speech
  - Speech recognition.
- Healthcare / Medicine
  - Disease recognition
    eg. → Brain tumar, Heart disease
  - CAS (Computer Aided Surgery)
- Defence
  - UAV (unmanned aerial Vehicle)
  - Automated target recognition
  - Autonomous Soldier robot.
- Industry
  - Repeatative task
- Natural Lang. processing (NLP)
  - Script conversion
    eg. Eng to hindi
- Bank
  - Chat bot for Queries
- Automobile
  - Self drive car
- Sales / Marketing
  - predicting the price of commodities.