Deeksha Lala
180104020

Assignment 3

1) Hebb's Learning Rule-

→ The Hebbian Rule was the first learning rule developed by Donald Hebb in 1949 as a learning algorithm of the unsupervised neural network.

→ It is used to identify how to improve the weights of nodes of a network.

→ The Hebb's learning rule assumes that - If two neighbour neurons are activated and deactivated at the same time, then the weight connecting these neurons should increase. For neurons operating in the opposite phase, the weight between them should decrease. If there is no signal correlation, the weight should not change.

→ When inputs of both the nodes are either positive or negative, then a strong positive weight exists between the nodes. If the input of a node is positive and negative for other, a strong negative weight exists between the nodes.

→ At the start, values of all weights are set to 0. This learning ~~[illegible]~~ rule can be used for both soft and hard activation functions.

(contd)

$$W_{ij} = x_i * x_j$$

Mathematical formula of Hebb Learning

2) Self Organizing Map -

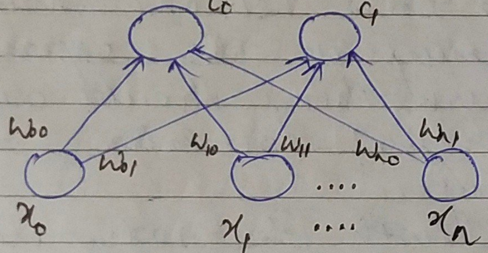
→ Self Organizing Map or Kohonen Map is a type of Artificial Neural Network inspired by the biological models of neural systems

→ It follows an unsupervised learning algorithm and trains its network through a competitive learning algorithm.

→ SOM is used for clustering and mapping techniques to map multidimensional data onto lower-dimensions which allows people to reduce complex problems for easy interpretation.

→ SOM has 2 layers - One is the input layer and other one is the Output layer.

→ The architecture of the SOM with two clusters and n input features of any sample is like-



Working -

→ Let's say an input data of size (m, n) where m is the number of training examples and n is the number of features in each example.

→ First, it initializes the weights of size (n, C) where C is the number of clusters. Then iterating over the input data, for each training example, it updates the winning vector. ~~Weight~~.

→ Weight updation rule is given by-

$$W_{ij} = W_{ij}(old) + alpha(t) * (x_i^k - W_{ij}(old))$$

where alpha is a learning rate at time t, j denotes the winning vector, i denotes the

18010402

i^{th} feature of training example and k denotes the k^{th} training example from the input data.

→ After training the SOM network, trained weights are used for clustering new examples. A new example falls in the cluster of winning neurons.

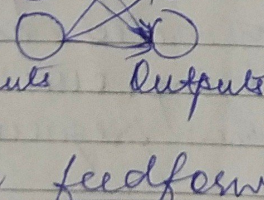
3) Network Topology -

→ A network topology is the arrangement of a network along its nodes and connecting lines.

→ According to the topology, Artificial Neural Network can be classified as the following -

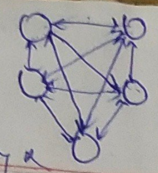
(1) FeedForward Network - It is a non-recurrent network having processing units / nodes in layers and all the nodes in a layer are connected with the nodes of the previous layers. There is no feedback loop means the signal can only flow in one direction, from input to output. It is of 2 types -

1.a) Single layer feedforward network - The input layer is fully connected to the output layer.



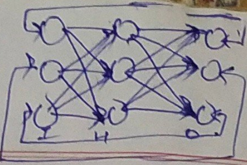
1.b) Multi layer feedforward network - As this network has one or more layers between the input and the output layers, it is called hidden layers.

2 b)

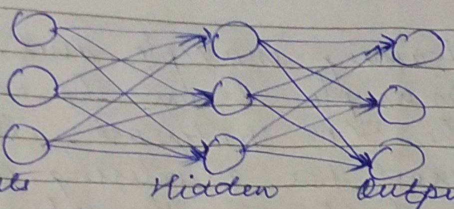


Fully R

2 c)



Jordan



(2) Feedback Network - A feedback network has feedback paths, which means the signal can flow in both directions using loops. It is divided into -

2 a) Recurrent Networks :- Feedback networks with closed loops.

2 b) Fully Recurrent Networks :- All nodes are connected to all other nodes and each node works as both input and output.

2 c) Jordan Network :- Closed loop network in which the output will go to the input again as feedback.

4) Applications of SOM -

Some of the applications of SOM are -

(1) <u>WEBSOM</u>: <u>Organising</u> a <u>Massive Document Collection</u>

WEBSOM is a SOM that organises massive document collections. The project successfully created a SOM that acted as a graphical search engine, classifying over 7,00,000 patent abstracts based on the frequency of occurrence of a set of words.

(2) <u>Phonetic Typewriter</u> - The Phonetic Typewriter is a SOM that breaks recorded speech to phonemes. It is set in the field of speech

Deeksha Lala
180104020

recognition and the problem is to classify phenomes in real time so that they could be used to drive as typewriter from dictation.

③ Classifying World Poverty-

The World Poverty Classifier is a SOM that maps countries based on 39 "quality of life factors" including health, nutrition, Education etc.

5) Hopfield Network and its Topology-

→ Hopfield Network is a special kind of neural network whose response is different from other neural networks.

→ It is calculated by converging iterative process. It has just one layer of neurons relating to the size of the input and output, which must be the same.

→ A Hopfield network is a single-layered and Recurrent network in which the neurons are entirely connected, i.e each neuron is associated with other neurons.

→ If there are two neurons i and j, then there is a connectivity weight W_{ij} lies between them which is symmetric $W_{ij} = W_{ji}$.

→ Discrete Hopfield Network:- It is a fully interconnected neural network where each unit is connected to every other unit. It behaves in a discrete manner, it it gives finite distinct output, generally of 2 types-

① Binary (0/1) ② Bipolar (-1/1)

→ Each neuron has an inverting and a non-inverting output.

→ Being fully connected, the output of each neuron is an input to all other neurons but not self.

6) Boltzmann Machines -

1 → These are stochastic learning processes having recurrent structure and are the basis of the early optimization techniques used in ANN.

2 → The main purpose of Boltzmann Machine is to optimize the solution of a problem. It is the work of Boltzmann Machine to optimize the weights and quantity related to that particular problem.

3 → Boltzmann Machines use recurrent structure.

4 → They consist of stochastic neurons, which have one of the two possible states, either 1 or 0.

5 → Some of the neurons in this are adaptive freestate and some are clamped frozen state.

6 → If we apply simulated annealing on discrete Hopfield network, then it would become Boltzmann Machine.

7 → Boltzmann Machines can be strung together to make more sophisticated systems such as deep belief networks.

Contd

7) Adaptive Resonance Theory (ART) -

1 → Adaptive Resonance Theory is a type of neural network technique that uses unsupervised learning technique.

2 → The term "adaptive" and "resonance" used in this suggests that they are open to new learning (i.e adaptive) without discarding the previous or the old

information (ie resonance).
3→ The ART networks are known to solve the stability-plasticity dilemma, ie stability refers to their nature of memorizing the learning and plasticity refers to the fact that they are flexible to gain new information.
4→ Due to this, the nature of ART they are always able to learn new input patterns without forgetting the past.
5→ ART networks implement a clustering algorithm.
6→ Input is presented to the network and the algorithm checks whether it fits into one of the already stored cluster.
7→ If it fits then the input is added to the cluster that matches the most, else a new cluster is formed.

→ The ARTs can be classified as follows-

1. ART 1 - It is the simplest and the basic ART architecture. It is capable of clustering binary input values.
2. ART2 - It is extension of ART1 that is capable of clustering continuous valued input data.
3. Fuzzy ART - It is the augmentation of fuzzy logic and ART.
4. ARTMAP - It is a supervised form of ART learning where one ART learns based on the previous ART module. It is also known as predictive ART.
5. FARTMAP - This is a supervised ART architecture with fuzzy logic included.

4/5

Hebb's Learning Rule

→ "If neuron i is near enough to excite neuron j, and repeatedly participate in its activation, the synaptic connection b/w these neurons is strengthened and neuron j becomes more sensitive to stimuli from neuron i."

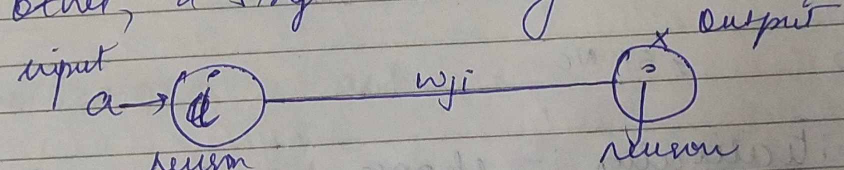
→ Hebb's Law can be represented by two rules-

(1) If 2 neurons on either side of connection are activated synchronously, then the weight of that connection is increased.

(2) If 2 neurons on either side of connection are activated asynchronously, then the weight of that connection is decreased.

→ If two neurons have no relationship, then weight will not change.

→ If inputs of both nodes are either +ve or -ve, then a strong +ve weight exists b/w them.
If inputs of a node is either +ve or -ve for other, a strong -ve weight exists b/w them.



Hebb's Formula

$$\boxed{\begin{aligned} W_{ji}^{k+1} &= W_{ji}^{k} + \Delta W_{ji}^{k} && \text{— (1)} \\ \text{where } \Delta W_{ji}^{k} &= \alpha a_i^{k} x_j^{k} && \text{— (2)} \end{aligned}}$$

W_{ji}^{k} = weight of connection at time k

W_{ji}^{k+1} = " " " at time k+1

ΔW_{ji}^{k+1} = increment by which weight of the connection is strengthened.

$\alpha \Rightarrow$ the constant coefficient which determines learning rate.

$a_i^k \Rightarrow$ input value from presynaptic neuron i at time k.

$x_j^k \Rightarrow$ Output value at post synaptic neuron j at time k.

Self organising map

→ works on unsupervised competitive learning model,
∴ self-organising ⇒ learn on their own.

→ def - SOMs are neural networks that uses unsupervised learning approach and train its their network through a competitive learning algo to map multidimensional data into lower-dimensional which allows easy interpretation of complex real problems.

2 layers → I/P & O/P
n units, m units

SOM Training Algo — ISMNC

① Initialisation :- choose random values for the initial weights W_{ij}.

② Sampling - Take a sample training input vector $x(x_1, x_2 .. x_n)$ from input layer.

③ Matching - find the winning neuron from the output layer that has weight vector ie closest to the input vector.

It can be calculated by taking the square of Euclidean distance for each output unit and find the output unit that has minimum Euclidean distance from the input vector.

$$D(j) = \sum_{i=1}^{n} \sum_{j=1}^{m} (x_i - w_{ij})^2$$

(4) New Weight Calculation – Find new weight b/w input vector example and winning output unit (neuron).

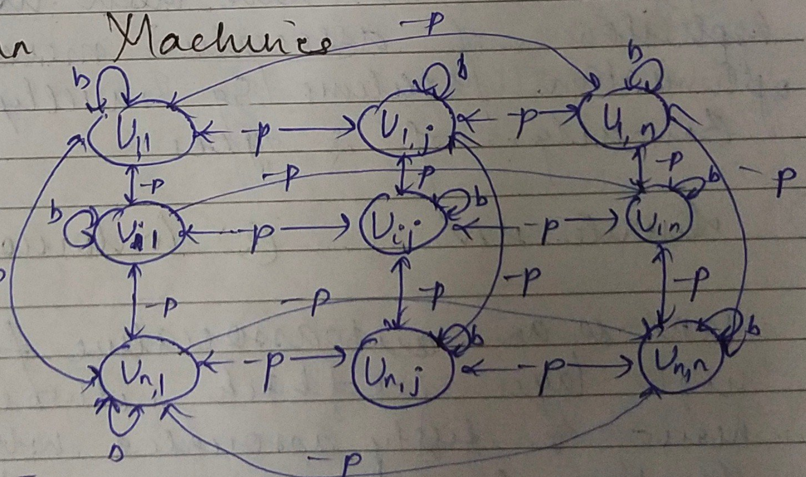
$$W_{ij}(new) = W_{ij}(old) + \underbrace{\alpha(t)}_{\text{learning rate}} * (\underbrace{x_i^k}_{k^{th}\text{ training example}} - W_{ij}(old))$$

where ————

(5) Continuation: – Repeat step 2–4 until weight updation is negligible (ie new weights are similar to old weights) or feature map stops changing.

Boltzmann Machines

: Architecture
- 2D array of units
- weights on interconnections b/w units are $-p$ where $p > 0$.
- Weights of self-connections are given by b, $b > 0$



→ BM has a set of units U_i and U_j and has bi-directional connections on them.
- Let fixed weight be W_{ij}.
- $W_{ij} \neq 0$ if U_i & U_j are connected.
- There also exists a symmetry in weighted interconnection, i.e $W_{ij} = W_{ji}$
- W_{ij} also exists i.e there would be self-connection ~~co~~ b/w units
- For any unit U_i, its state u_i is either 0 or 1.

→ The main ~~sol~~ objective of BM is to maximize the Consensus Function (CF) =)

$$CF = \sum_i \sum_{j \le i} W_{ij} U_i U_j$$

Hopfield Network

1→ John J. Hopfield proposed a ~~method~~ network in 1982 which consists of a set of neurons with output of each neuron as feedback to all other neurons.

2→ Hopfield networks have been used in many applications of associative memory and many optimization problems (eg Travelling Salesman)

3→ Basically of 2 types

① Discrete HN ② Continuous HN

4→ HN is an autoassociative fully interconnected single layer feedback network, i.e each neuron is fully associated with other neurons.

5→ If there are two neurons i & j, then there is a connectivity weight W_{ij} b/w them which is symmetric $W_{ij} = W_{ji}$. =) symmetrically weighted network

Discrete Hopfield Network

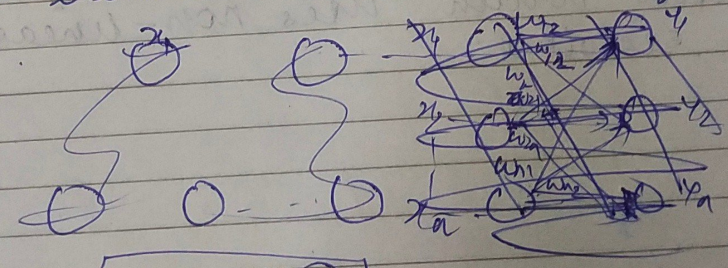
NN when operated in
1) discrete line fashion ⇒ DHN
2) Fully interconnected NN where each unit is connected to every other unit.
3) Behaves in a discrete manner as it gives finite distinct outputs, from 2 types of input
a) Binary (0/1) (2) Bipolar (-1/1)

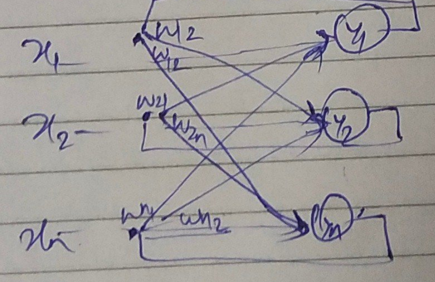
4) Network has symmetrical weights and no self connection $\boxed{W_{ij} = W_{ji}}$ $\boxed{W_{ii} = 0}$

5) The architecture of DHN consists of 2 outputs, one inverting & the other non-inverting.

6) The outputs from each processing element are to feedback to the input of other processing elements but not to itself

$W_{11} = 0$
$W_{22} = 0$
$W_{nn} = 0$





$S(P) = [0\ 1\ 1\ 0\ 1]$ Binary pattern
$S(P) = [-1\ 1\ 1\ -1\ 1]$ Bipolar pattern

⇒ For storing a set of binary pattern, where p = 1 to P,
the weight matrix is

$$W_{ij} = \sum_{p=1}^{P} [2S_i(p) - 1][2S_j(p) - 1], \text{ where } i \neq j$$

⇒ For storing a set of bipolar pattern
the weight matrix is

$$W_{ij} = \sum_{p=1}^{P} \{S_i(p) \cdot S_j(p)\} \quad \text{where } i \neq j$$

and weight for self connection $W_{ij} = 0$

Continuous HN

→ A DHN can be modified to continuous HN
→ The nodes of this network have a continuous output rather than a two state output (b/w 0,1)
→ The CHN can be realised as an electronic circuit which uses non-linear amplifiers & resistors.