Soft Computing

PAGE NO.: 30
DATE: 29/07/22

1980.

⇒ Soft computing :- It is use of approximate calculations to provide imprecise but usable soln to complex computational problems.

computing (पूर्वपद/पूर्वगामी)

Antecedent (x) → [y = f(x)] → (y) consequent. (फलस्वरूप/परिणाम)

f :- formal methods/ Algorithm/ Mapping Function

* Features :- of computing.

(i) Precise (सटीक/सही) Solution
(ii) Unambiguous Accurate
(iii) Mathematical Model.

⇒

Soft computing	Hard computing.
• Soft computing is liberal of inexactness, uncertainty, partial truth & approximat.	• Hard computing needs a exactly state analytic Model.
• It relies on formal logic & probablistic Reasoning.	• Hard computing relies on binary logic of crisp system.
• It has features of approximation & dispositionality.	• It has the features of exactitude & categoricity
• It is stochastic in Nature.	• Hard computing is Deterministic in Nature.
• It produces approximate Results.	• Hard computing Produces precise results.
• Soft computing works on ambiguous & noisy Data	• Hard computing performs on exact data.
• It can perform parallel computations	• It performs on sequential computation

(M

I

Hard/Soft

* relies :- निर्भर करता है।
* brisk :- तेज/फुर्तीला

PAGE NO.:
DATE : 23/07/22

* **Hard Computing**:- It relies on binary logic & predefined instructions like a numerical analysis & brisk software & uses two-valued logic. Eg:- Quick sort, binary dep. greedy dy etc

⇒ Feature :-
- Precise Result
- Control Actions
  - → Unambiguous
  - → Formally defined

Numerical Problem
- Searching & Sorting
- Computational Geometry prob

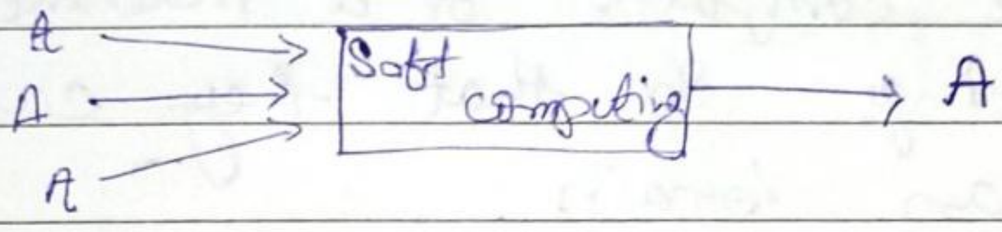
Jump to Art CC.

Soft Computing

PAGE NO.: 34
DATE: / /

Features :- ▫ Impercision (अशुद्धता) ▫ Dynamic/Adaptive ↗ Have to Handle
▫ Uncertainty ▫ Low Soln cost → Due to Uncertain output/Ans
▫ Do Not require Mathematical Model.
(Soln may be Uncertain)

Tools :- Fuzzy Logic • Neutral Network
Techinequs → • Evolutionary / Generic computing
Types



⇒ Applications :- • In games (checker/poker) • Microwave & Rice cooker • Washing Machine, AC, Heater
• Writing Recognige • Robotic Work • Image Processing.
• Automotive System.

① Fuzzy Logic :- The fuzzy logic algo is used to solve the models which are based on logical reasoning like imprecise & Vague. It introduced by Lotzi A. Zadeh in 1965. It provide stipulated truth value with the closed interval [0,1] where 0 = false 1 = True.

* What is Soft computing :- It is a set of Algorithms, including neural Network, fuzzy logic, and generic algo. These algo are tolerant of imprecision, Uncertainty, Partial truth and approximation.

* It is contrasted with hard computing.

S.C

PAGE NO.:
DATE: / /

(II) * Neural Network (ANN) - 1950s which helped Soft. computing to solve real-world problems, which a computer cannot do itself.

An (ANN) emulates a network of neurons that makes a human brain (machine act like hum. think) Thereby the computer or a machine can learn things so that they can take decision like human brain.

Artificial (NN) are mutually connected with brain cells & created using regular computing progra. It is like as the Human Neural Syste

(III) * Genetic Algorithms (GA):- It is almost based on Nature and take all permission प्रेरणा → inspirations from it. There is no genetic algo. that is based on search-based algo, which find its roots in natural selection & the concept of genetics.

⇒ Genetic Algo is a subset of a large branch of computat

* Key points & Meaning *

* stipulated :- तय करना
* contrasted :- विपरीत
* tolerent :- सहनशील / * emulate - अनुकरण करना
* Genetic :- आनुवंशिक
* Vague :- अस्पष्ट

M/3/S

Kohonen Self-Organizing N/W

(13M)

(30M) (KSON/M)

PAGE NO.:
DATE: / /

▷ It follows an Unsupervised learning approach and trained its N/W through a competitive learning algo.

Algorithm:-

(S-1) Initialize the weights Wij, Random values may be assumed. Initialize the learning rate 'α'

(S-2) Calculate Square of the euclidean distance, i.e for each j = 1 to m.

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

(S-3) find winning Unit index J, so that D(j) is minimum.

(S-4) for all Units j within a specific neighbourhood of 'J' and for all i, calculate new weight :-

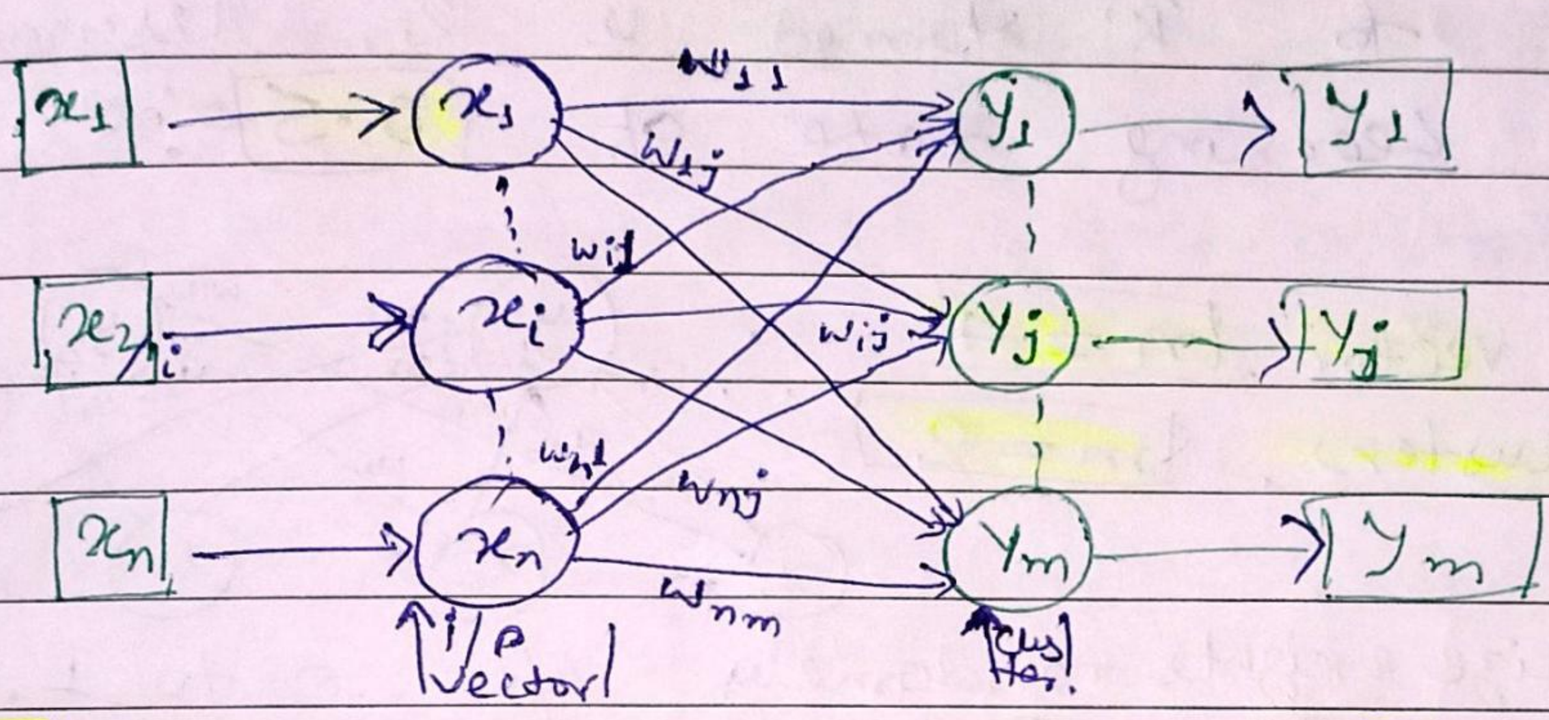
$$w_{ij}(new) = w_{ij}(old) + \alpha[(x_i - w_{ij}(old)$$

(S-5) Update Learning Rate 'α' using the formula :-

$$\alpha(t+1) = 0.5\ \alpha t$$

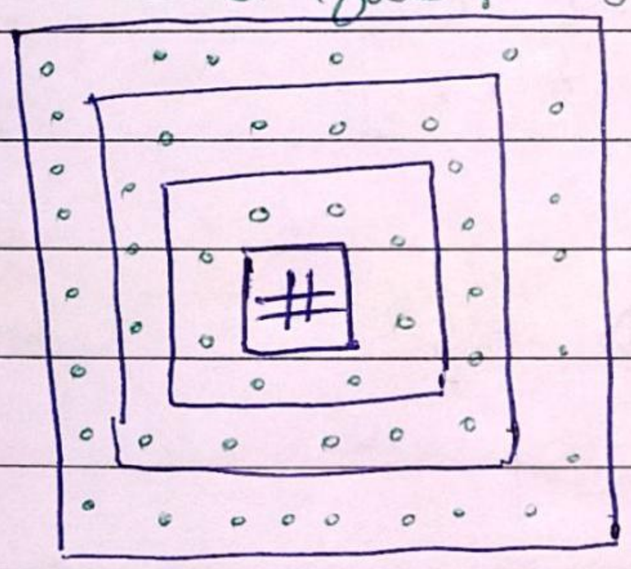
* The Architecture of KSOM

It similar to that of the competitive N/w.

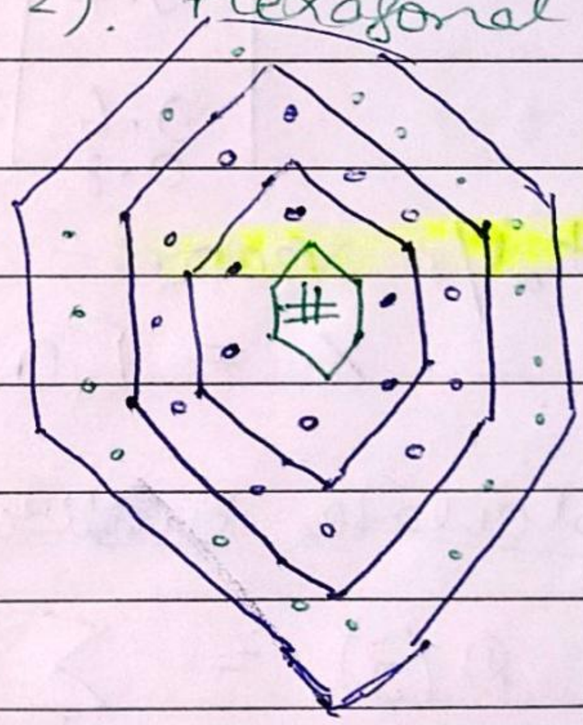


* Most Used topologies in KSOM :-

1. Rectangular Grid (24 nodes) 16, 8



2). Hexagonal Grid Topo. (12 nodes) 12, 6.



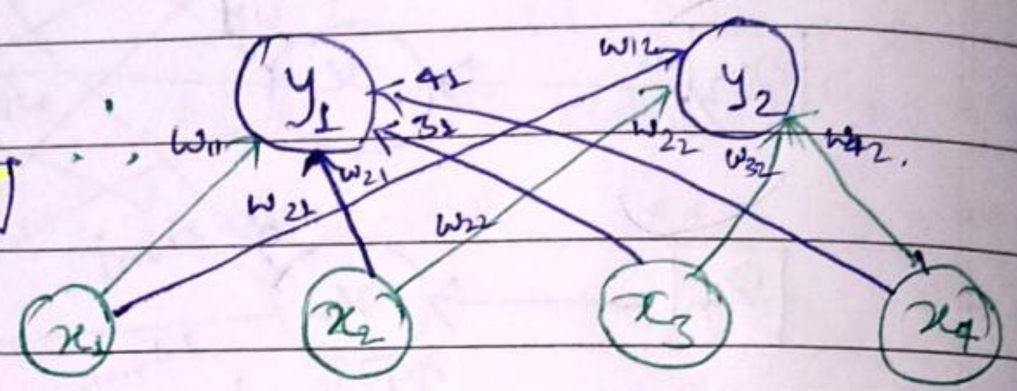
KSOM
Kohonen

PAGE NO.:
DATE: / /

Ques:- Construct KSOFM to cluster four given vectors [0 0 1 1], [1 0 0 0], [0 1 1 0] and [0 0 0 1] No. of clusters to 'k' formed is 2. Assume an initial Learning rate of 0.5 = 'α'

Soln

* No. of i/p vector, n = 4
* No. of clusters, m = 2



(1st) Initialize weights randomely b/w 0 to 1.

$$w_{ij} = \begin{bmatrix} 0.2 & 0.9 \\ 0.4 & 0.7 \\ 0.6 & 0.5 \\ 0.8 & 0.3 \end{bmatrix} \quad (y_1, y_2) \quad \text{or} \quad \begin{bmatrix} 0.2 & 0.9 \\ 0.4 & 0.7 \\ 0.6 & 0.5 \\ 0.8 & 0.3 \end{bmatrix}$$

* **First i/p vector:-**

$$X = [0\ 0\ 1\ 1] \quad \text{(Given)}$$

* **Calculate Euclidean distance.**

$$D(j) = \sum_i (w_{ij} - x_i)^2$$

$$D(1) = (0.2 - 0)^2 + (0.4 - 0)^2 + (0.6 - 1)^2 + (0.8 - 1)^2$$
$$= 0.04 + 0.16 + 0.16 + 0.04$$
$$= 0.4$$

* $$D(2) = \sum (w_{i2} - x_i)^2$$
$$= (0.9 - 0)^2 + (0.7 - 0)^2 + (0.5 - 1)^2 + (0.3 - 1)^2$$
$$= 0.81 + 0.49 + 0.25 + 0.49 = 2.04$$

Note:- The winning cluster D(1) < D(j) Therefore winning cluster is j = 1, i.e y_1

* Update weights on winning cluster unit (J=1),

$w_{ij(new)} = w_{ij(old)} + \alpha [x_i - w_{ij(old)}]$

$w_{i1(new)} = w_{i1(old)} + \alpha [x_i - w_{i1(old)}]$

α

$w_{11(new)} = w_{11(old)} + 0.5 [x_1 - w_{11(old)}]$

$= 0.2 + 0.5 [0 - 0.2] = \boxed{0.1}$

$w_{21(new)} = w_{21(old)} + 0.5 [x_2 - w_{21(old)}]$

$= 0.4 + 0.5 [0 - 0.4] = \boxed{0.2}$

$w_{31(new)} = 0.6 + 0.5 [0 - 0.6] = \boxed{0.8}$

$w_{41(new)} = 0.8 + 0.5 [0 - 0.8] = \boxed{0.9}$

$\begin{bmatrix} 0.2 & 0.9 \\ 0.4 & 0.7 \\ 0.6 & 0.5 \\ 0.8 & 0.3 \end{bmatrix}$

⇒ Updated weight Matrix :-

$w_{ij} = \begin{bmatrix} 0.1 & 0.9 \\ 0.2 & 0.7 \\ 0.8 & 0.5 \\ 0.9 & 0.3 \end{bmatrix}$

* Second i/p vector :- (D1)

$[X = 1000]$. (Given)

* Calculate Euclidean Distance :-

$D(2) = (0.9 - 1)^2 + (0.7 - 0)^2 + (0.5 - 0)^2 + (0.3 - 0)$

$= 0.01 + 0.09 + 0.25 + 0.49$

$= \boxed{0.84}$

$D(1) = (0.1 - 1)^2 + (0.2 - 0)^2 + (0.8 - 0)^2 + (0.9 - 0)$

$= 0.81 + 0.04 + 0.64 + 0.01 = \boxed{2.3}$

Here, The winning cluster is i.e $D(2)$ $\{ D(2) < D(1) \}$.

$\boxed{J = 2}$

*Remaining (334) at End (For 334 Set) Answer

31/07/22
M = 2

Soft Computing

Overview

PAGE NO.: 12
DATE: / /

* **Neural Network :-** It Designed to work just like the human brain does. In the case of recognizing handwriting or facial recognition, The brain very quickly makes some decisions.

Ex:- In facial "Brain detect quickly that face looks like → woman or man ?

[Neural Network] ⟹ [Artificial Neural N/W]

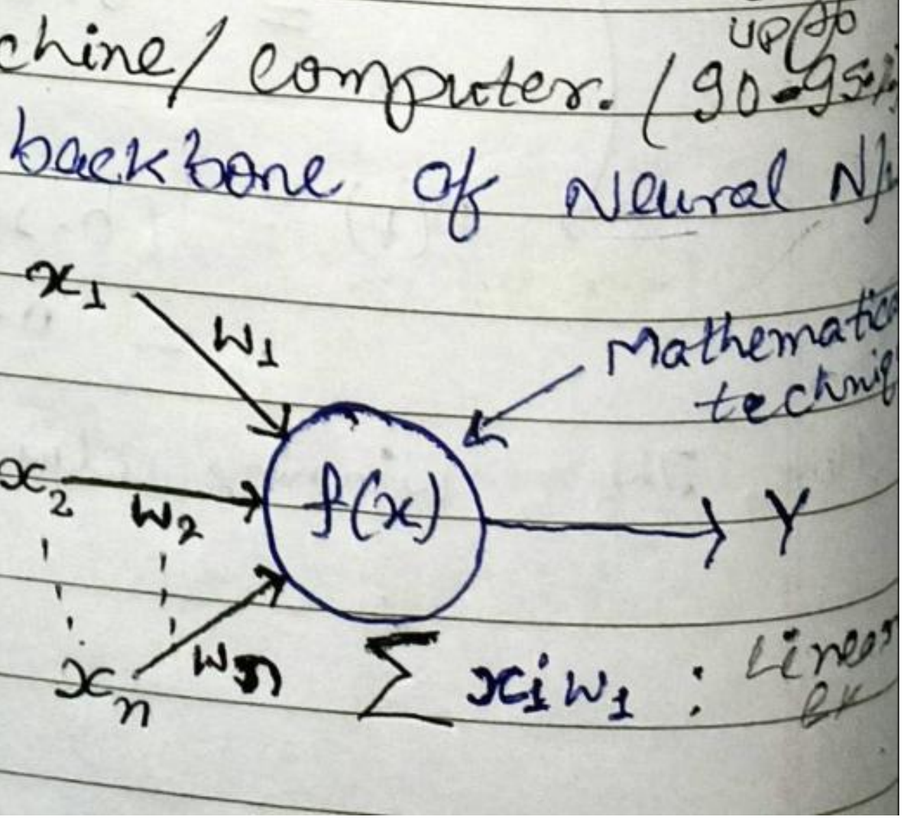
Brain → Machine

Note ⇒ To get a machine which act like human brain, it means implementation of Neural Network System in the computer System to works like brain is process. to get Artificial Neural Network.

[Input Data] ⟹ (∴ Algorithm) [Meaningful value] ⟹ (∴ Neural Networks) [Learning and Improvement.]

(1@) :- Neural Networks improve the Accuracy of Machine/computer. (90-95% upto

* "Learning Algorithm" is backbone of Neural N/



M
2.

History of Neural N/W

PAGE NO.: 38
DATE: 31/07/

1944 By Warren Mcculloch & Walter pitts.

* Two major concept that are precursers to Neural Network are :-

- 'Threshold Logic' - converting continious ~~ip~~ input to discrete output.
- Hebbian Learning - a model of learning Based on Neural plasticity

(9) 1943 Warren Mcculloch & walter pitts developed a mathematical Model of an artificial neural N/W using threshold logic.

1958 (0)

Frank Rosenblatt created the "Perceptron" Model which was the first of its kind to perform pattern recognition.

(5) 1975 paul werbos developed Back propagation to solve the XOR problem in the perceptron model.

1992 (0)

Max - pooling introduced 3D object recognition as it helped with least Shift in-variance and tolerance to deformation.

(6) 2009-12

Re-current N. N/W and deep feed forward Neural Networks were created by 'Jurgen - Schmidhuber's research group excelled at pattern recognition and Machine learning.

PAGE NO.:
DATE: 31/07/

Advantage:- (ANN)

- Parallel Processing capability.
- Ability to learn irrespective of the type of Data (Linear or Non-)
- ANN is highly ~~architectures~~ ~~ma~~ Volatile and Servers best in financial time series forecasting.

⇒ DisAdvantage:- • The simple architecture makes it difficult to explain the behavior of the N/W.
• This Network is dependent on hardware.

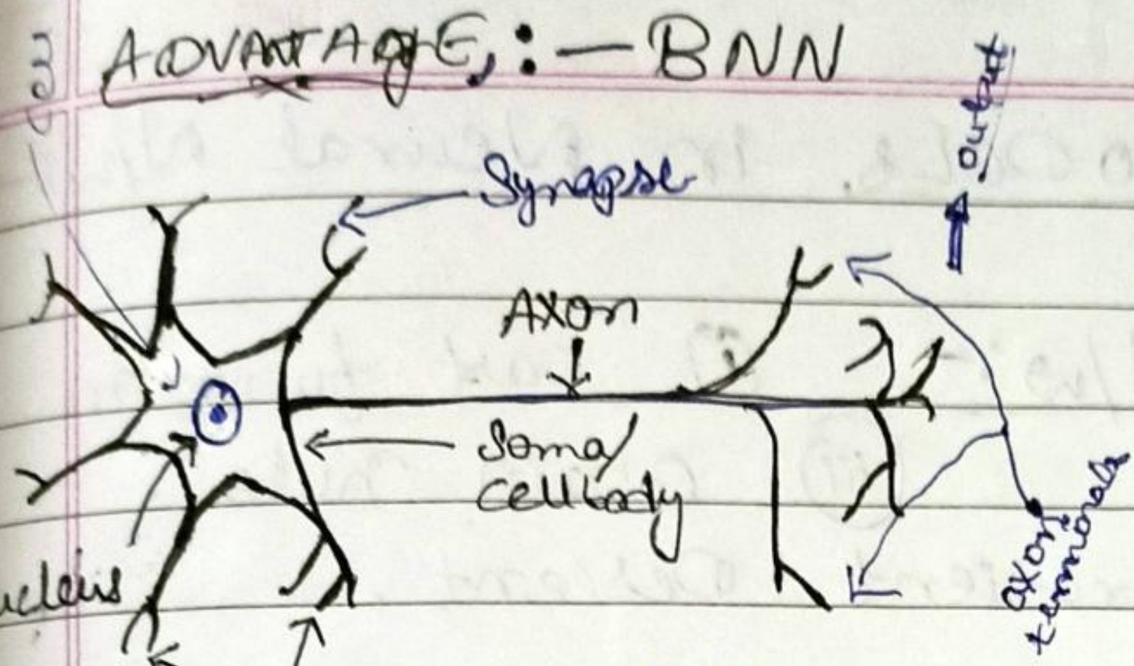
Biological Neural Network

⇒ It is a structure that consists of Synapse, dendrites, cell body, and axon. In this N. N/W, the processing is carried out by neurons, soma sums all the incoming signals and axon transmits the signals to other cells.

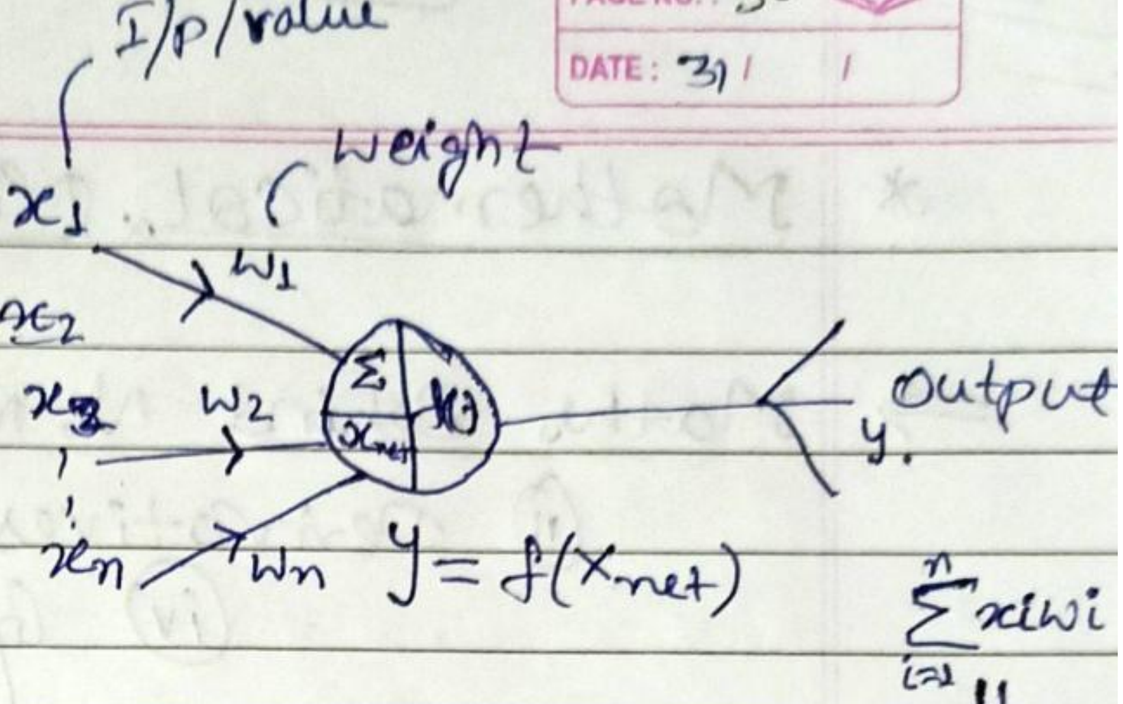
* Precursors :- अग्रगामी/दूत * Key Point * Threshold :- सीमा
* Propogation - प्रचार • deformation :- विकृति • Excelled - उत्कृष्ट प्रदर्शन

Biological Neuro-N/W

PAGE NO.: 39
DATE: 31 / /



Fig



$$x_{net} = x_1 w_1 + x_2 w_2 + w_3 x_3$$

ADVANTAGE:—BNN

- The Synapses are the i/p processing elem
- It is able to process highly complex parallel inputs.

- DisAdvantage :-
  - There is no controlling Machanism.
  - Speed of processing is slow being it complex.

(Input) ↗ used to receive the signals from neurons.

Fig * Dendrites :- Used to carry signals, It connected to Soma or cell body

* Axon :- It is single, long connection extending (output) from the cell body. It carries signals from the neuron to

[Each Neuron has one axon]

BNN	ANN
• Dendrites	Input
• Cell Nucleus	Nodes
• Synapse	weights
• Axon	output

M
2

Soft computing (M=2)

PAGE NO.:
DATE: / /

* **Mathematical Models in Neural N/w:-**

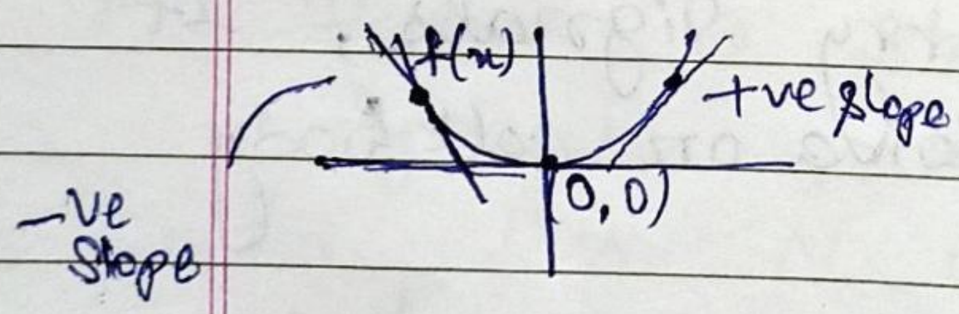
⇒ **Maths Behind N.N/w:-** (i) cost function
(ii) Derivatives (iii) Chain Rule
(iv) Gradient Descent.

(1) cost function :- $y = mx + c$ ∴ Straight line

$$J(m,c) = \sum_{i=1}^{n} \left[Y_{actual} - \underbrace{(mx+c)}_{\text{Predicted}} \right]^2$$: Cost function for linear Regression

Above Model, try to optimize the cost funct
of f bring as possible Zero (0) or close
to zero.

(2) $f(x) = x^2$



S.C

ANN Architecture

PAGE NO.:
DATE: / /

* Activation Function;—

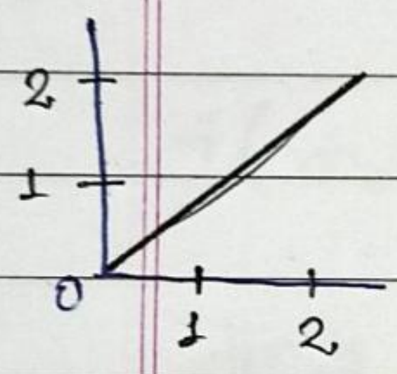
(i) Linear fun.

(ii) Heviside step fn.

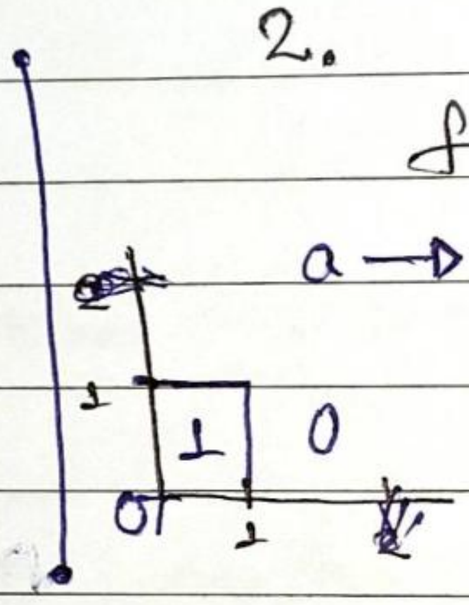
(iii) Sigmoid fun.

1. $f(v) = a + v$ (weighted sum)
$= a + \sum w_i \cdot x_i$

a → Bais



2.



weighted sum.

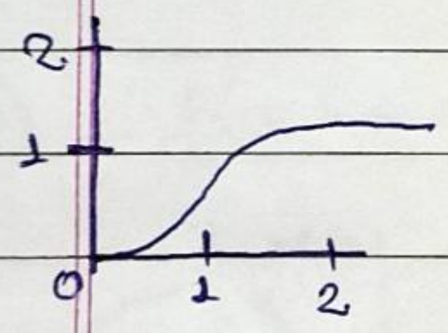
$$f(v) = \begin{cases} 1 & \text{if } v \geq a \\ 0 & \text{otherwise.} \end{cases}$$

a → Threshold

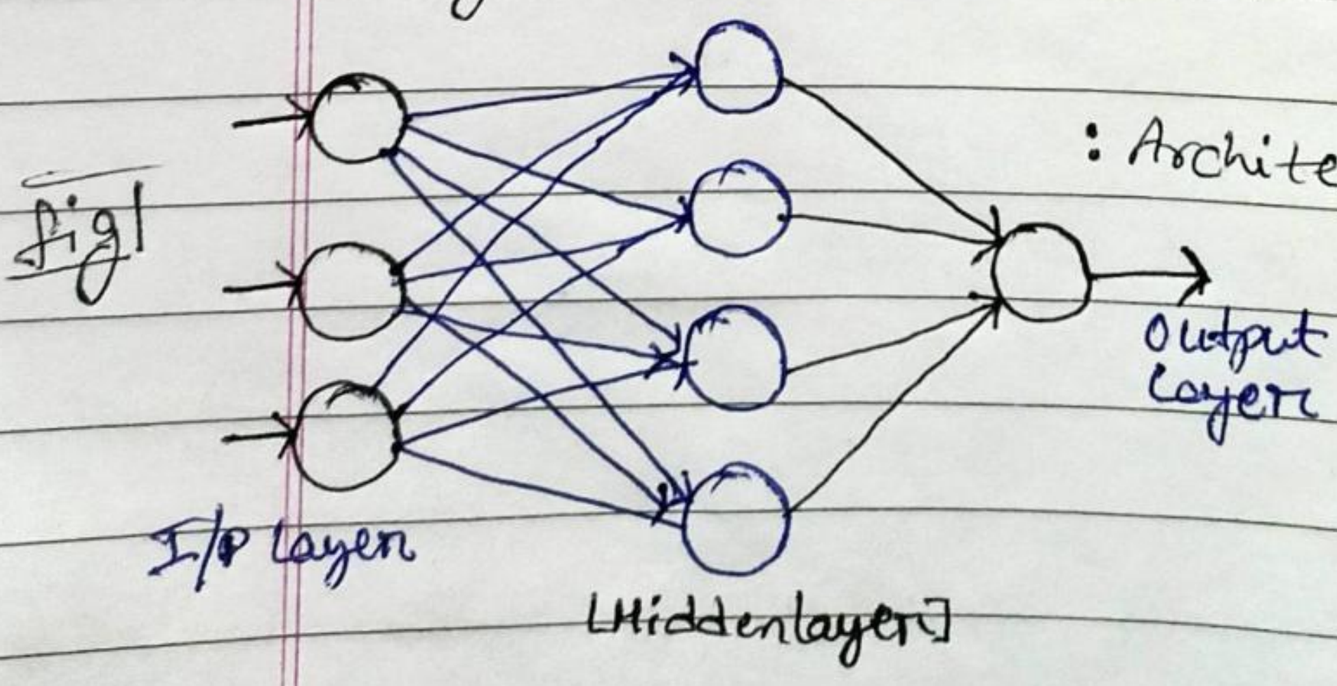
∴ Heviside fun only given two value (0, 1) zero or one.

for exm:— $V = 4$ $a = 3$ this type gives '1' as a output

3. $f(v) = \frac{1}{1+e^{-v}}$



\# ANN Architecture is based on the structure and function of the bio-Logical Neural Network.



Fig] : Architecture OF ANN]

Soft Comp.

PAGE NO.: 41
DATE: 01/08/22

* Learning Rules

⇒ Learning in ANN :- Basically, learning means to do and adapt the change in itself as and when there is a change in environment. ANN is a complex system or more precisely we can say that it is a complex adaptive system, which can change its internal structure based on the information passing through it.

* The Learning Rule is a technique or a mathematical logic which encourages a neural N/w to gain from the existing condition and uplift its performance. It is an iterative procedure.

▷ Neural Network Learning Rule :- We know that, during ANN learning, to change the input/output behaviour, we need to adjust the weights. Hence, a method is required with the help of which the weights can be modified. These methods are called Learning rules, which are simply algorithms or equations.

* Learning Rules *

(1) Hebbian
(2) Perceptron
(3) Delta
(4) Correlation
(5) Out Star

Sol-1

M=2

Hebbian Rules

PAGE NO.:

DATE : 01/08/

* Objective of N.N/W :- (1) Learning / Training.
(2) Generalization
(3) Application.

⇒ Hebbian :- One of the oldest and simplest Rule, was introduced by Donald Hebb 1949. It is a kind of feed-forward, Unsupervised learning.

Important ANN Terminology

PAGE NO.:
DATE: / /

(1) Weights (2) Bias (3) Threshold (4) Learning Rate
(5) Momentum Factor.

⇒ In ANN Each neuron is connected to the other neurons through Connection links. These links carry a weight. The weight has information about the input signal to the neuron. The weights and input signal are used to get an output. The weights can be denoted in a matrix ~~from~~ form ~~Last~~ that is also called a connection Matrix

* Each neuron is connected to every other neuron of the next layer through connection Weights. Hence, if there are 'n' nodes and each node has 'm' weights, then the weight matrix will be :-

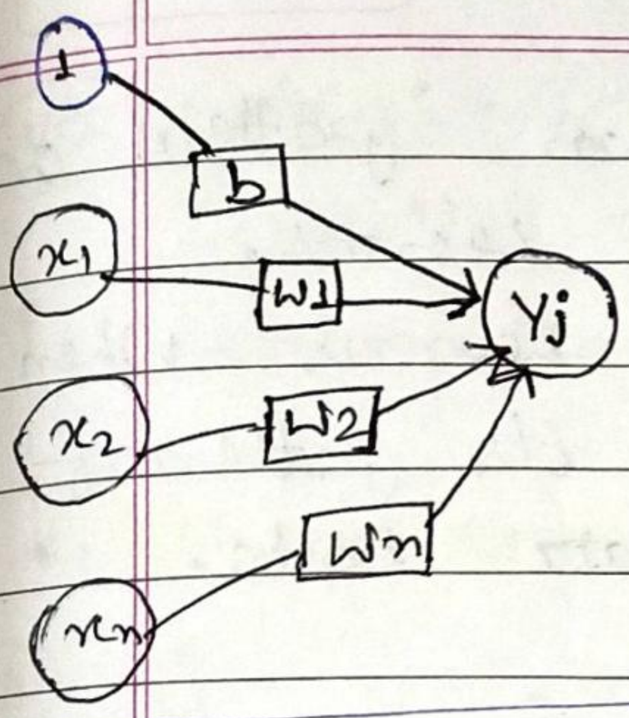
$$\text{weight matrix} = W = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ \vdots \\ w_n \end{bmatrix} = \begin{bmatrix} w_{11} & w_{12} & w_{1m} \\ w_{21} & w_{22} & w_{2m} \\ w_{31} & w_{32} & w_{3m} \\ \vdots & \vdots & \vdots \\ w_{n1} & w_{n2} & w_{nm} \end{bmatrix}.$$

⇒ **Bias** :- It is added to the N/W by adding an input element $x(b) = 1$ into the input vector. The bias also carries a weight denoted by $w(b)$.

▷ It plays imp. role in calculating the O/P of the neuron. The bias can be positive or negative. Positive bias increases and -ve bias decreases the net input weight

Soft computing

PAGE NO.: 43 RBP
DATE: 01/08/22



Threshold :- A threshold value is used in the activation function. The net input is compared with the threshold to get the output. In NN, the activation function is defined based on the threshold value and output is calculated.

value :

$$f(net) = \begin{cases} 1 & \text{if net} >= \text{threshold} \\ 0 & \text{if net} < \text{threshold} \end{cases}$$

⇒ Learning Rate :- It is denoted by alpha The learning rate ~~changes~~ ranges from 0 to 1. It is used for weight adjustment during the leaning process of NN.

⇒ Momentum factor :- It is added for faster convergence of results. The momentum factor is added to the weight and is generally used in back-propagation N/w.

notes!!

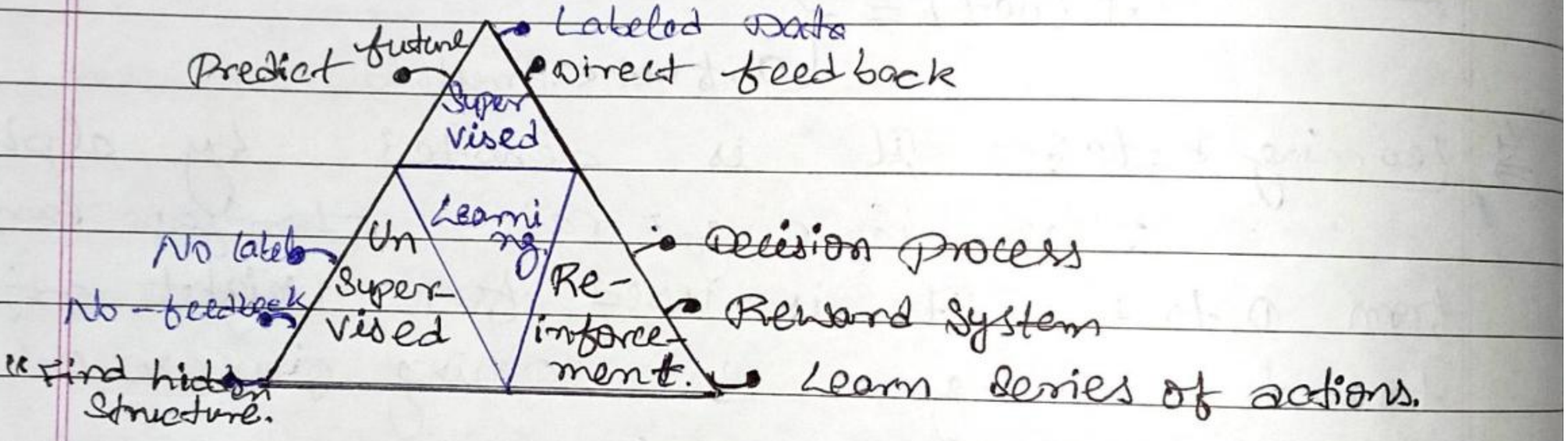
Learning Paradigm

PAGE NO.:
DATE: 01/08/

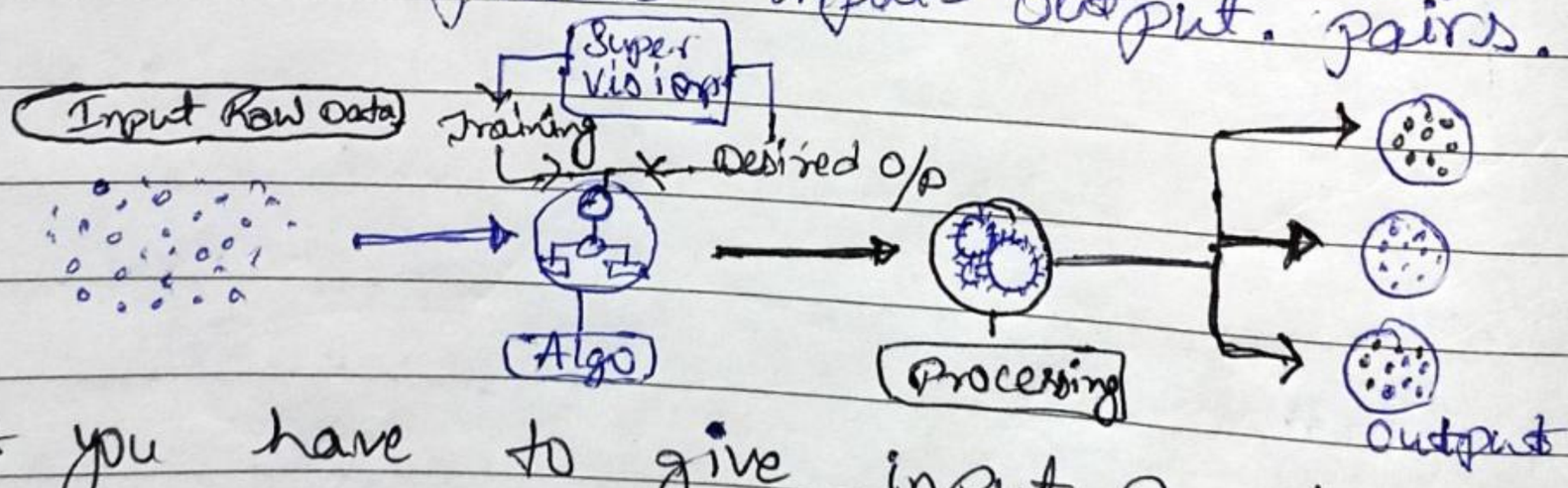
⇨ It Basically states a particular pattern on which something or someone learns.

It Says how a machine learns when some data is given to it, its pattern of approach for some particular Data.

* Three basic types :- (1) Supervised Learning
(2) Un-Supervised Learning.
(3) Reinforcement Learning.



* 1) Supervised :— It is a machine learning task in which a function maps the input to output data using the provided input-output pairs.



⇒ * you have to give input & output Both. (outp usually in the form of labels) to the comput for it to learn how from it.

↳ This is more basic type of learning Paradigm, & most algo we learn today are based on this type of learning Pattern.

#1 TWO

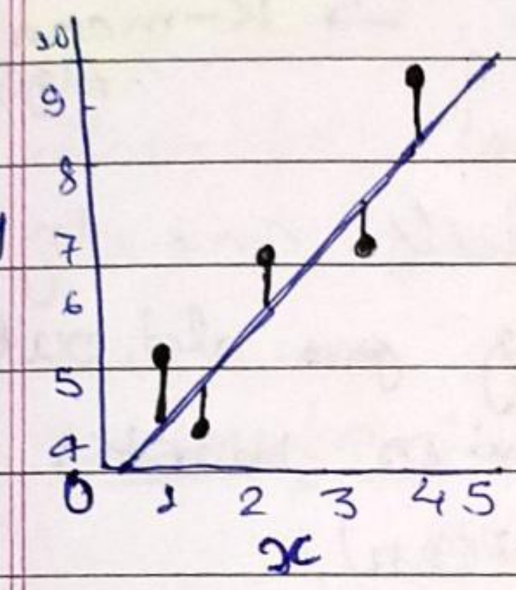
Supervised Learning

PAGE NO.: 44
DATE: 01 /

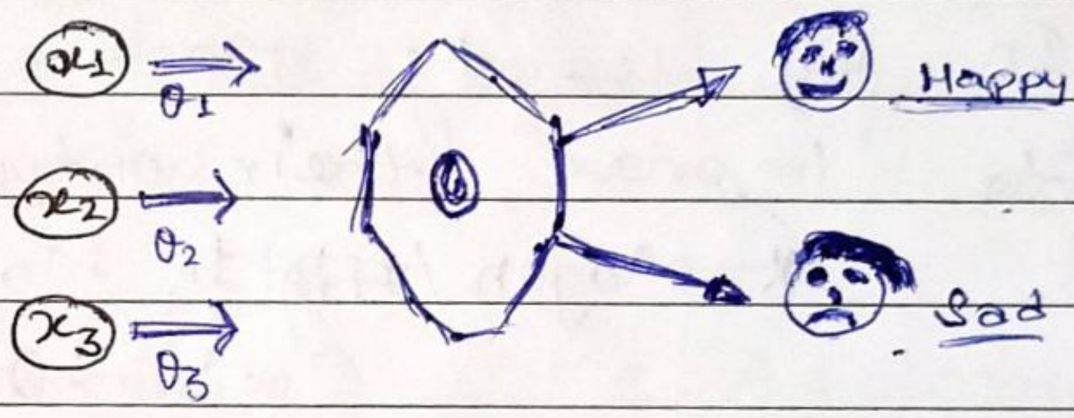
(V.I) Explain :- It deals with or learns with "labelled Data". This implies that some Data is already 'tagged' with the correct answer

1. Linear Regression (the simple Line fun.)

(will)



2. Logistic Regression (0 or 1) Logic.



⇒ Advantage :- • It allow collecting Data and produces Data output from previous experiences.

• Helps to optimize performance criteria with the help of experience

• It helps to solve various types of real-world computation problem.

⇒ DisAdvantages :-

• classifying big data can be challenging

• It trainging / learning needs a lot of computation time.

2). Un-Supervised :- It is the traing of a machine using information that is neither classified nor 'labeled' & allowing the algo. to act on that information without guidance.

PTO

Soft computing

Un-Supervised

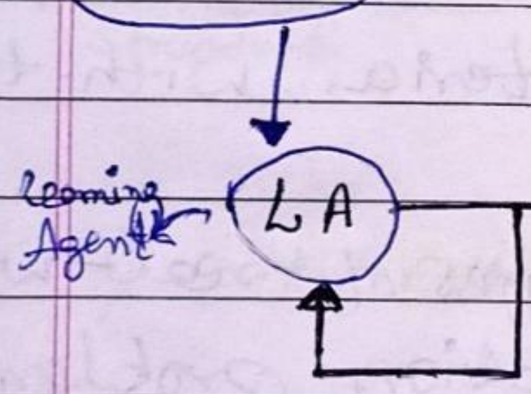
PAGE NO.:
DATE: / /

⇒ The task of the machine is to group unsorted information according to similarities, patterns, and differences without any prior training of data.

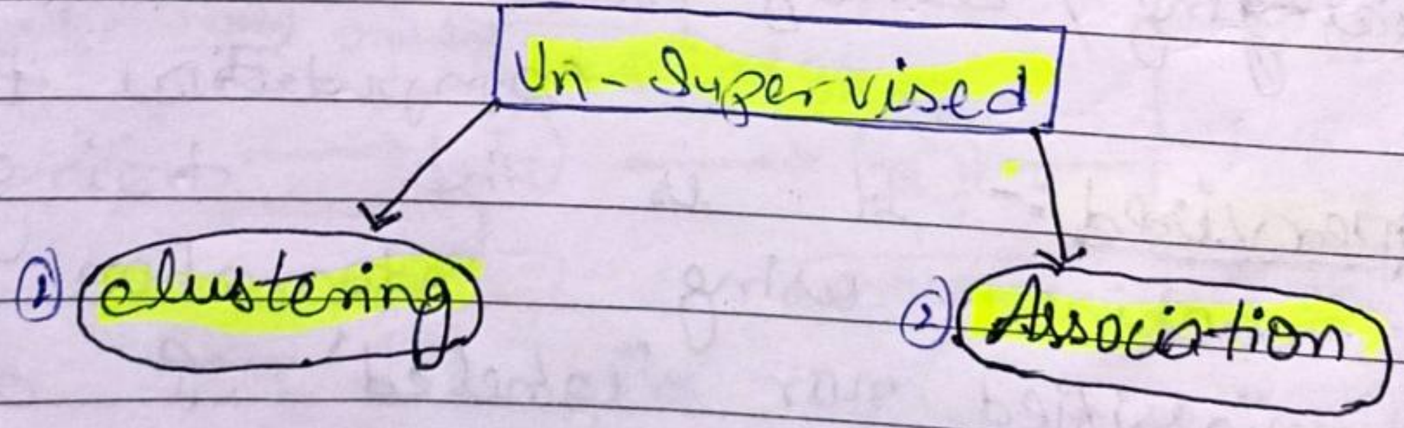
* → Only Inputs → clustering → K-mean [Ex Algo
→ Filtering

In this learning Machine learn itself and try to improve their output by using ~~pas~~ old. output
* Major/~~Highly~~ Most machines works on it (unsupervised).

Ex: A Machine trying to grouping of a crowd ~~in~~ according to their shape, size, identity, color (Man) or (women), (young) or (old) etc.



• Benifits of it :- It using at starts with 'Un-labelled' Data being easier and cheaper to obtain.
• No Needed of prior knowledge of the Data.



(1) A clustering problem is where you want to discover the.

inherent groupings in the data, such as grouping customers by purchasing behavior.

(2) Association: It learning problem is where you want to discover rules that describe large portions of your data, such as people buy 'x' also tend to by Y.
('x' → Bread; Y → Butter/Jam)

* clustering *

1. Exclusive(partitioning) 2. Agglomerative
3. Overlapping 4. probabilistic.

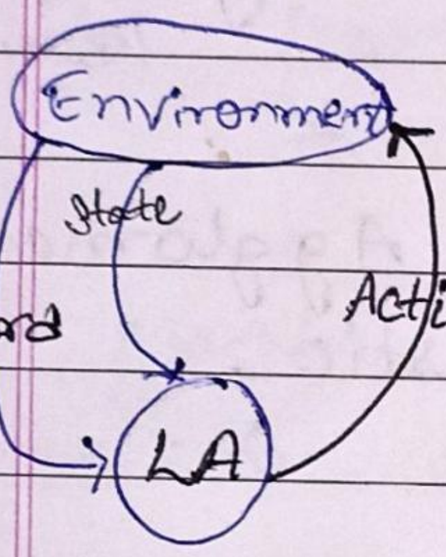
⇒ Un Supervised Algo :- • k-means clustering • Neural N/w
• Apriori Algo. • Hierarchy cl. • Anomaly detection.

ADVANTAGE :- • It used to solve complex task
• It is preferable as it is easy to get Unlabeled data in comparison to labeled Data.

DisAdvantage :-
• It is intrinsically mor difficult that Supervised learning, at it does not have corresponding output.
• The results might be less Accurate due to Un-labeled input.

* key - point *

3. * Re-Enforcement *



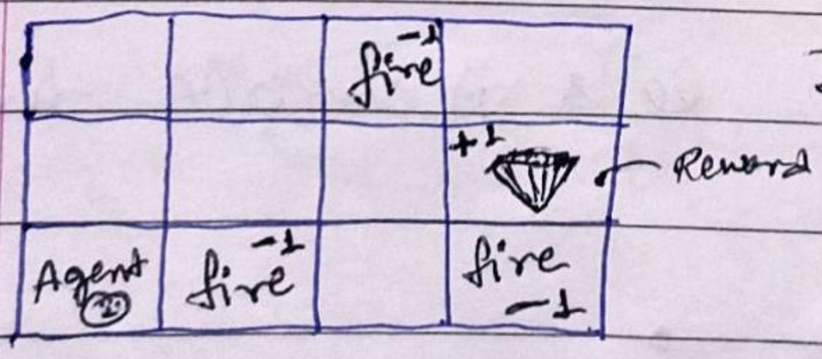
→ Reward/penalty

→ Q-Learning.

* It mainly used to Develop a game. where we get +ve/-ve pair

⇒ It is a feedback-based Machine-learning-techniques in which an agent learns to behave in an environment by performing the actions and seeing the results of actio
if (Action = Good) Reward; else (Bad) Penalty;.

Exmpl: follow↓: we have an agent & a reward, with many hurdles in b/w. The agent is supposed to find the best possible path to reach the reward.



main Points:- Input:- i/p should be at initial state

output:- Multiple possible o/p as there are a variety of soln.

DATE: / /

Training:- The training based upon the input, The model will return a state and the user will decide to 'reward' or 'punish' the model based on its o/p.

- The Model keeps continues to learn.
- The Best soln is decided as maximum 'Reward'

Example:- chess game, pacman, etc.

* **Re-Inforcement** (Types) *

1. positive
2. Negative

* Positive Re-inforcement is defined as when an event, occurs due to a particul. behavior, increases the Strength and the frequency of the behavior. it has a +ve effect on Behavior.

ADVN:-

- Maximum performance.
- Sustain change for a long period of time.
- Too much Reinforcement can lead to an overload of states which can dimish the results.

* Negative is defined as Strenghening of behavior because a negative condition is stopped or avoided.

ADVNT:-

- Increases Behavior
- provide defianance to a min. Standard of Performance.
- It only enough to meet up the min. behavior.