MRI

Class: 01

27/3/22

Neural Networks & Fuzzy Systems

Books:

① Neural Computing: An Introduction
- T. Jackson

Scanned with CamScanner

MRT

Class:- 02 30/3/22

Chapter - 01

Introduction

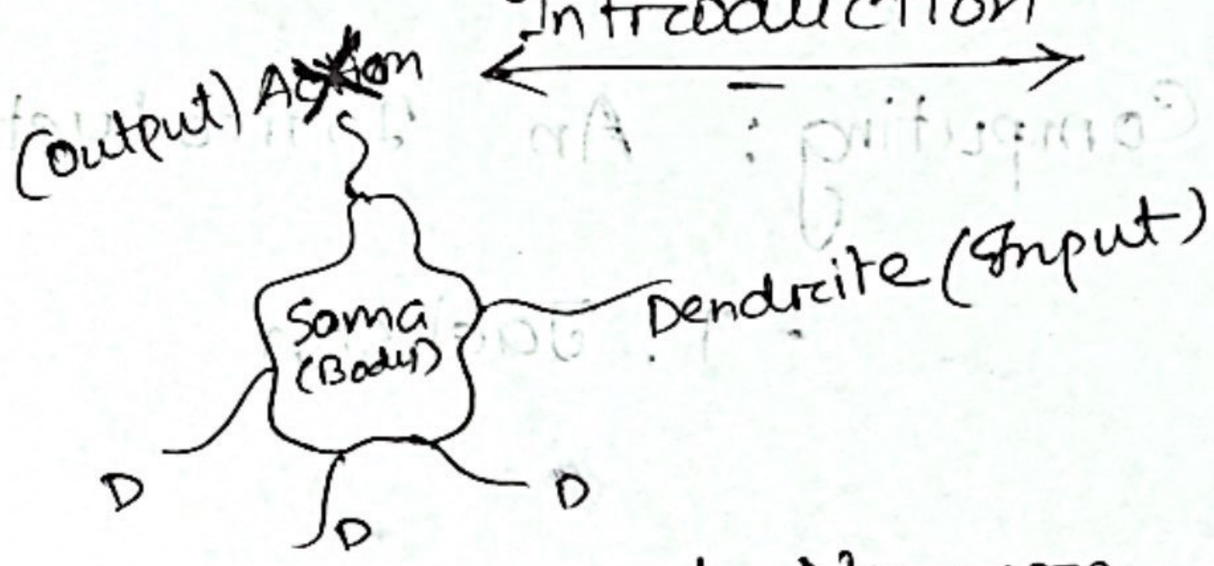


Fig: Biological Neuron

- 10^{10} Neurons In total
- each Neuron is connected with 10^4 other neurons

* Machine thinks serial
  Human thinks parallel

Topics!

- Human Computer Interaction
- structure of the brain

Scanned with CamScanner

MRI

Class - 03 + 04 3/4/22

1.2 The Structure of the brain

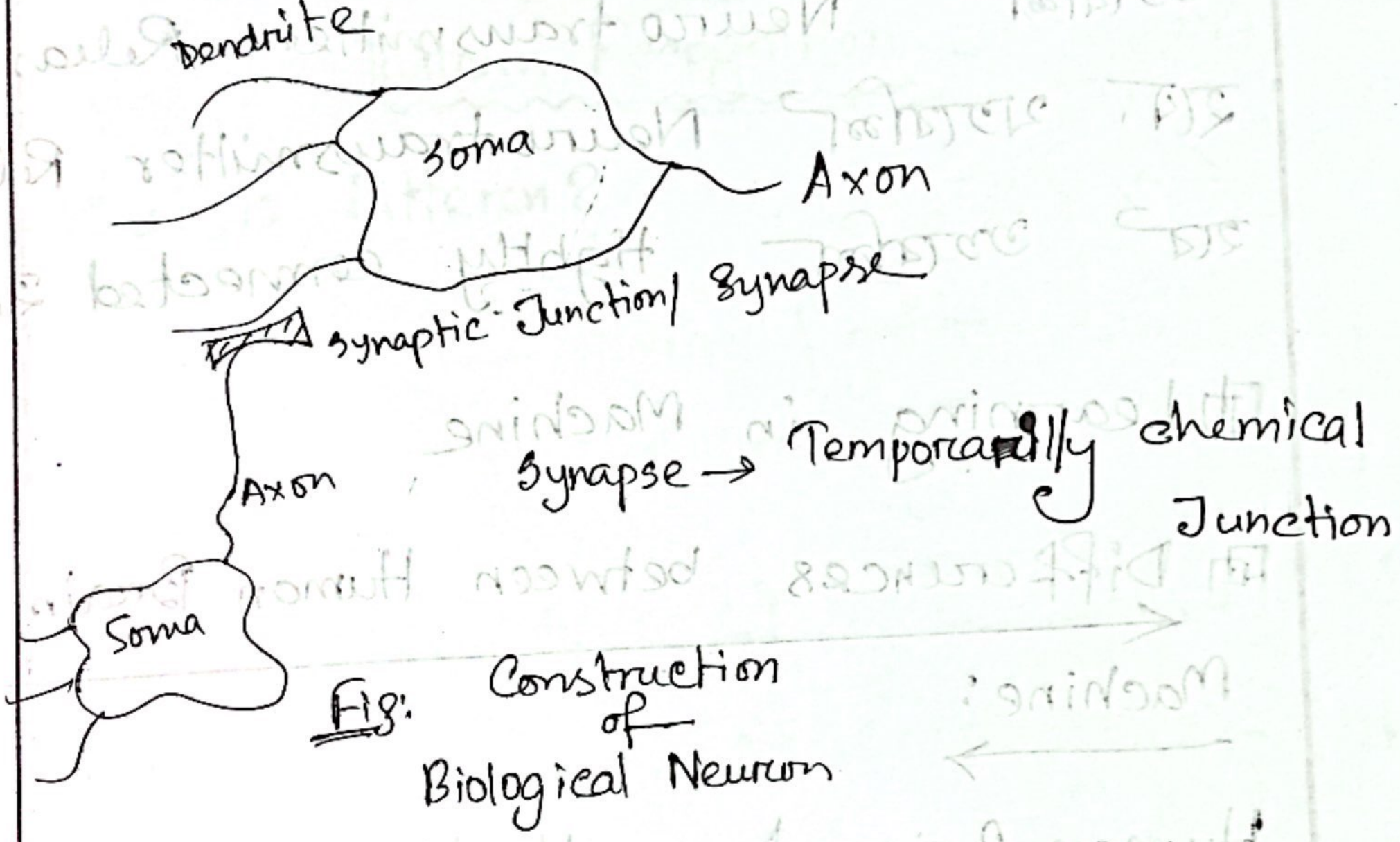
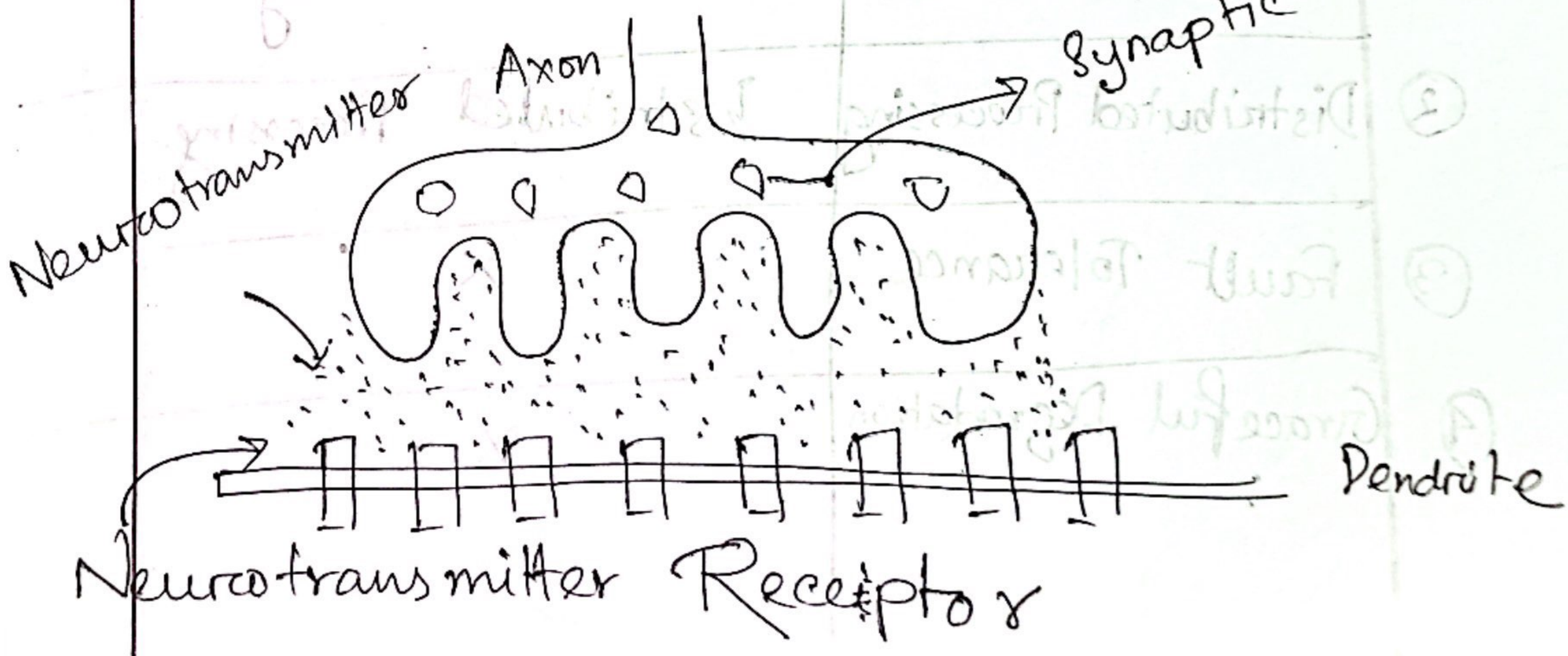


Fig: Construction of Biological Neuron

Learning in Biological System:



⊛ যতবেশি Output Axon দিয়ে আসবে তত বেশি Neuro transmitter Released হবে. যতবেশি Neuro transmitter Released হবে ততবেশি tightly connected হবে.

Learning in Machine

Differences between Human Brain & Machine:

	Human Brain	Machine
①	Parallel Processing	Serial Processing
②	Distributed Processing	Distributed Processing
③	Fault Tolerance	×
④	Graceful Degradation	×

Scanned with CamScanner

Chapter - 2

Pattern Recognition

- What is Pattern?

Scanned with CamScanner

MRI

Class - 5 6/4/22

Chapter - 02

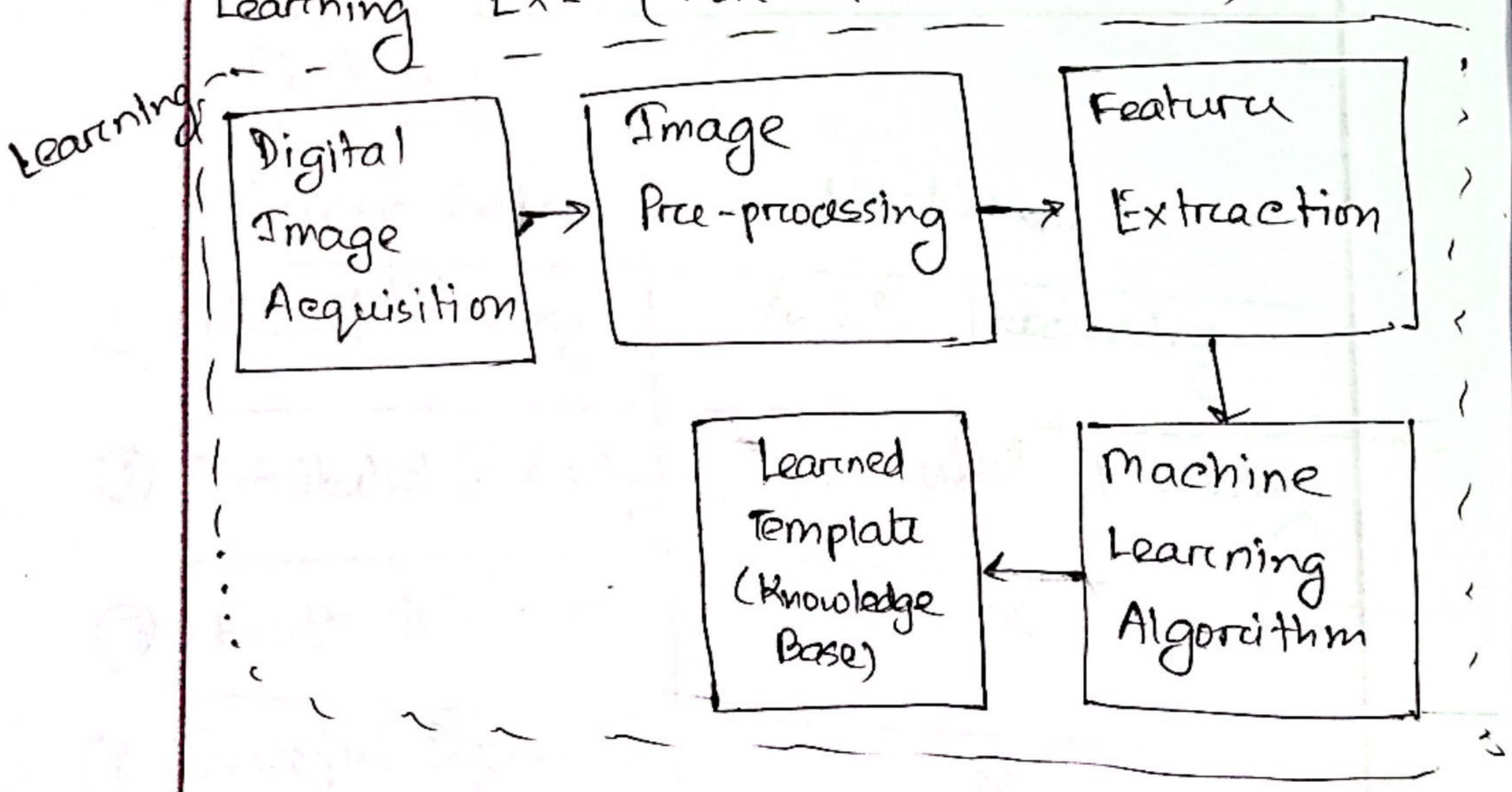
Pattern Recognition

→ 2 stages -

① Learning / Training

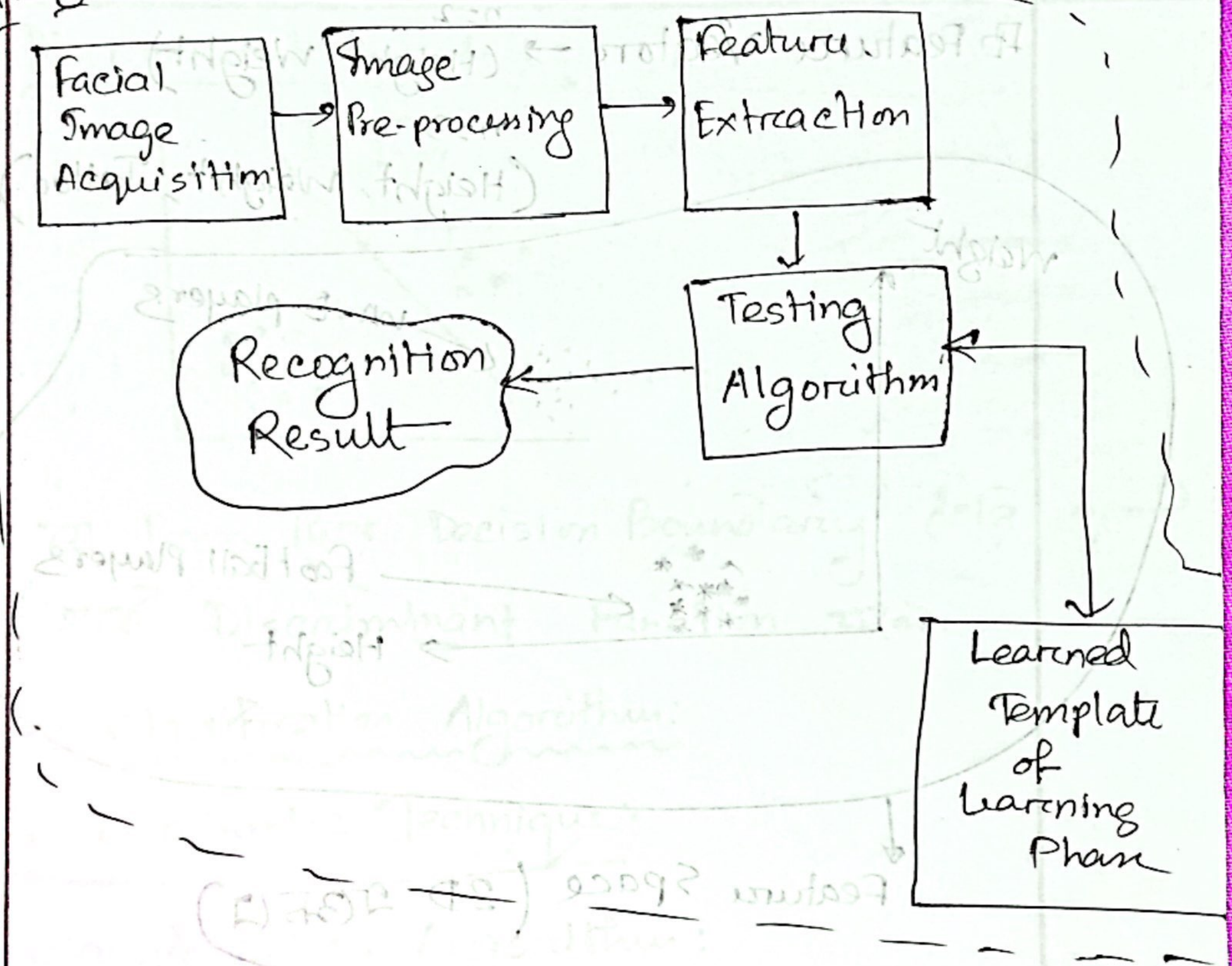
② ~~Recognition~~ Testing

Learning Ex - (For face detection)



Scanned with CamScanner

Testing



Recognition:

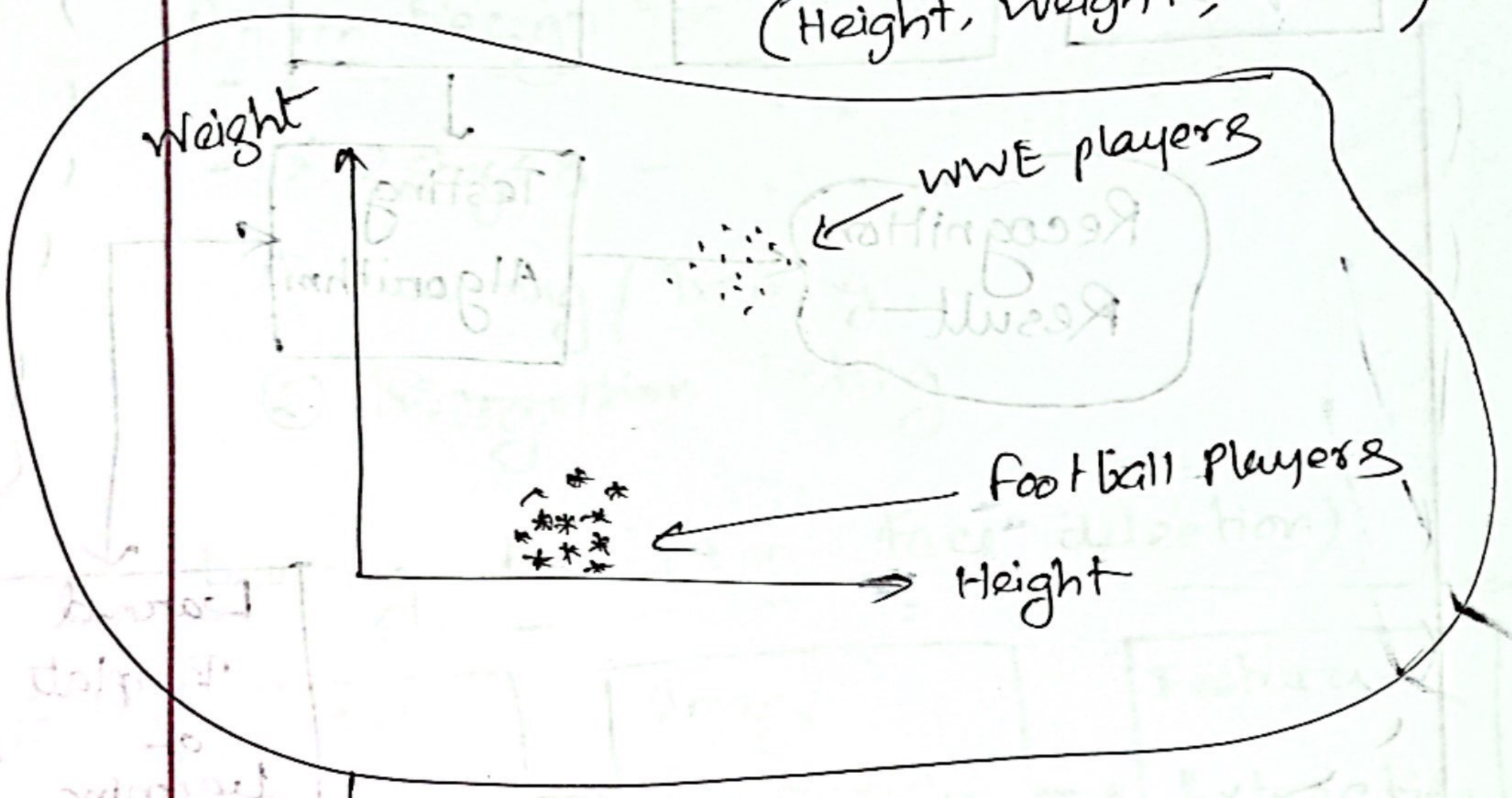
Identification / Verification

(1 to N) Y/N o

WWE vs Football

Feature Vector → $n=2$ (Height, Weight)

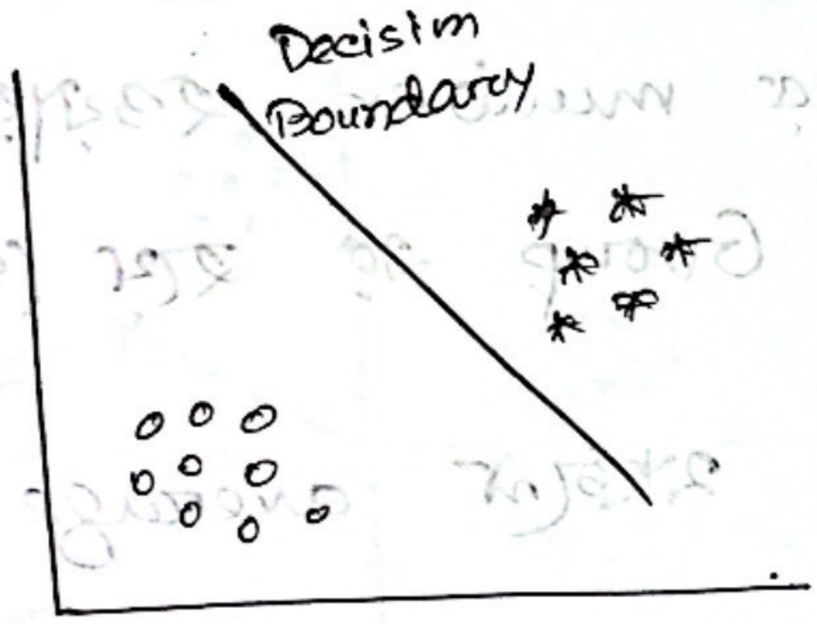
$n=3$ (Height, Weight, Tattoo)



Feature Space (2D ক্ষেত্র)

Scanned with CamScanner

Discriminant Function:



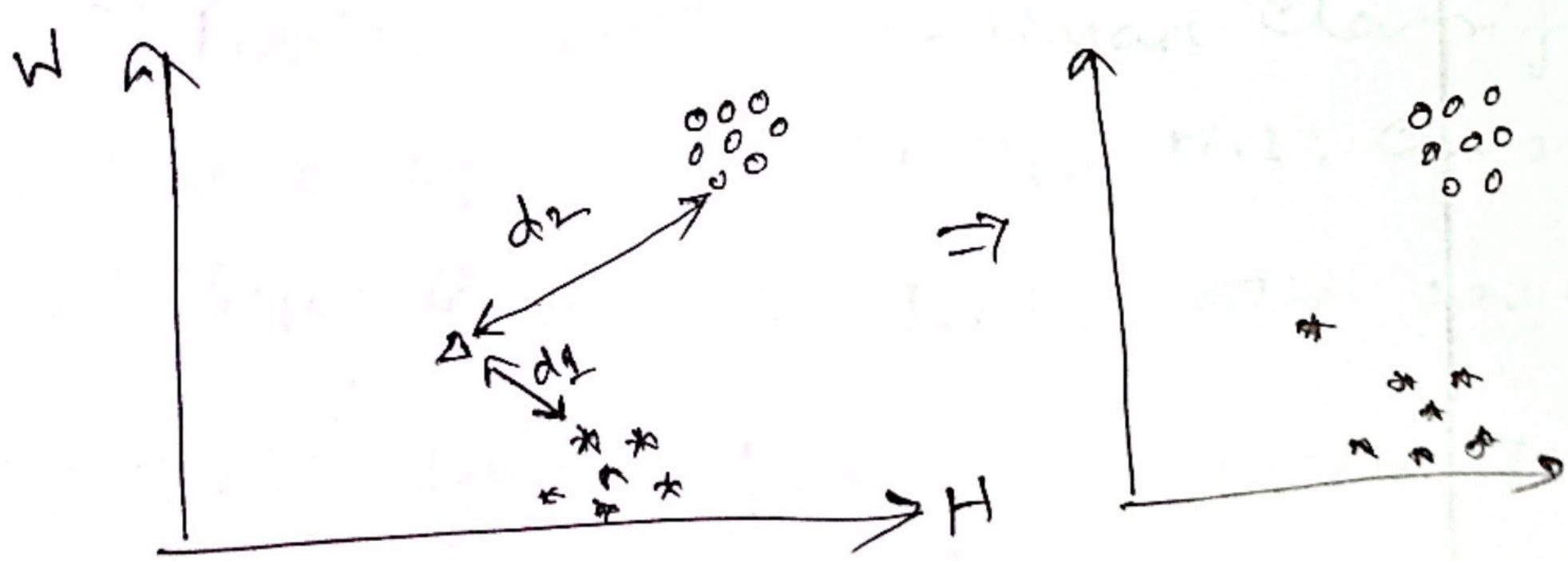
যে Process দিয়ে Decision Boundary তৈরি হয় তাকেই Discriminant Function বলে।

Classification Algorithm:

Deterministic Technique:

Classification Algorithm:

Nearest Neighbour Classification:



Scanned with CamScanner

Rogue Pattern

→ এক Group এর member হওয়ার কথা ছিল। But অন্য Group এর হয়ে গেছে।

(Rogue Pattern থাকলে average নিয়ে কাজ করতে হবে)

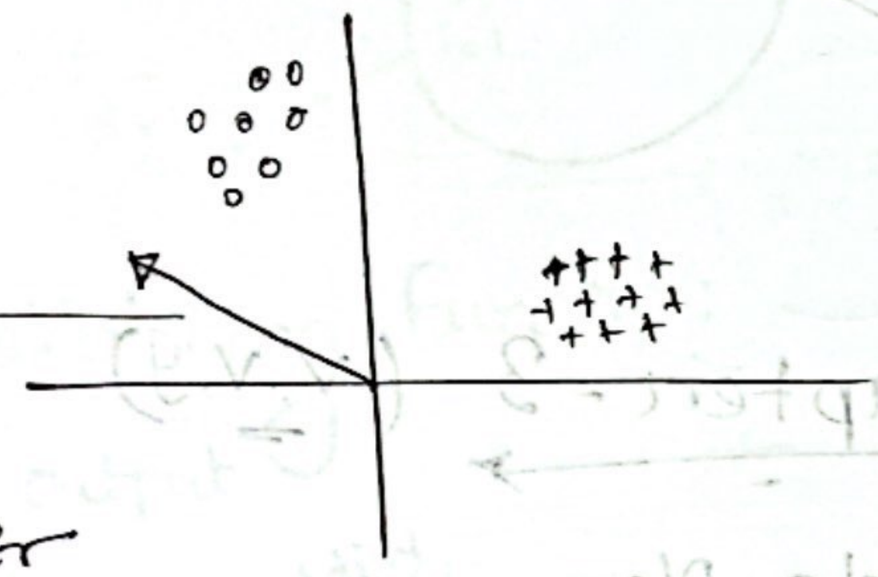
Scanned with CamScanner

MRI

Class-06

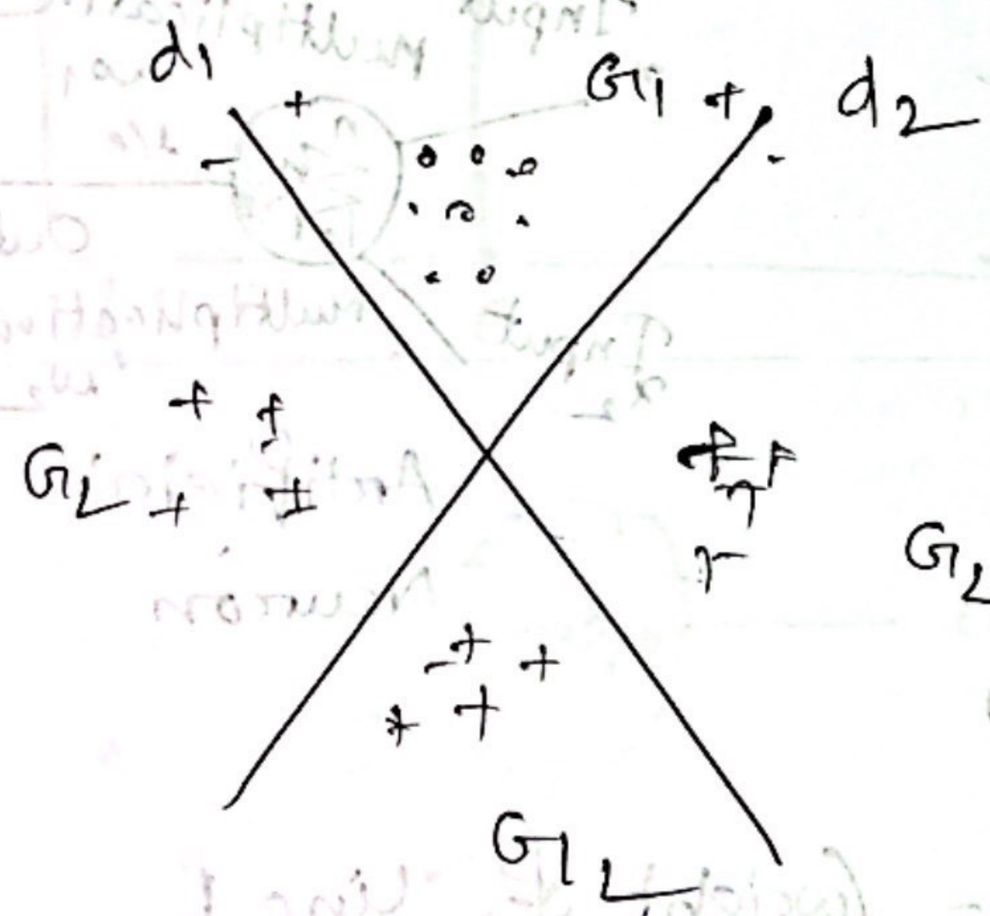
10/4/22

Linear Classifier:



weight vector initially যেকোনো এক জায়গায় থাকতে পারে. Then, algo. অনুযায়ী Move করে Cluster গুলোকে Divide

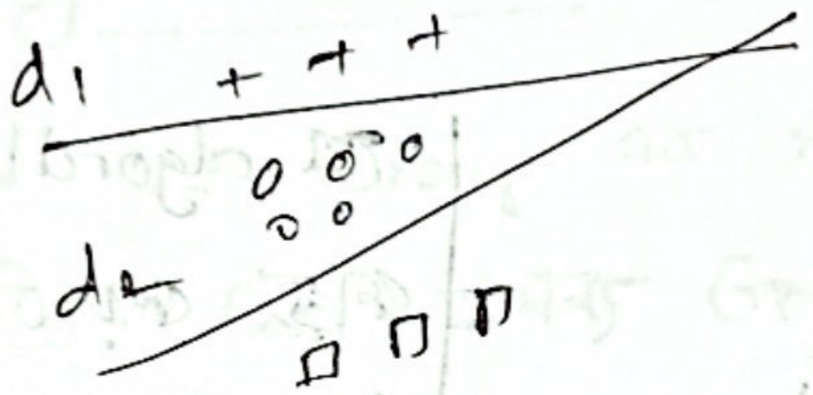
এই algorithm এই কাজ করতে যেটা Discriminant function বা M.L. algo.



d_1	d_2	
+	+	G1
−	+	G2
+	−	G2
−	−	G2

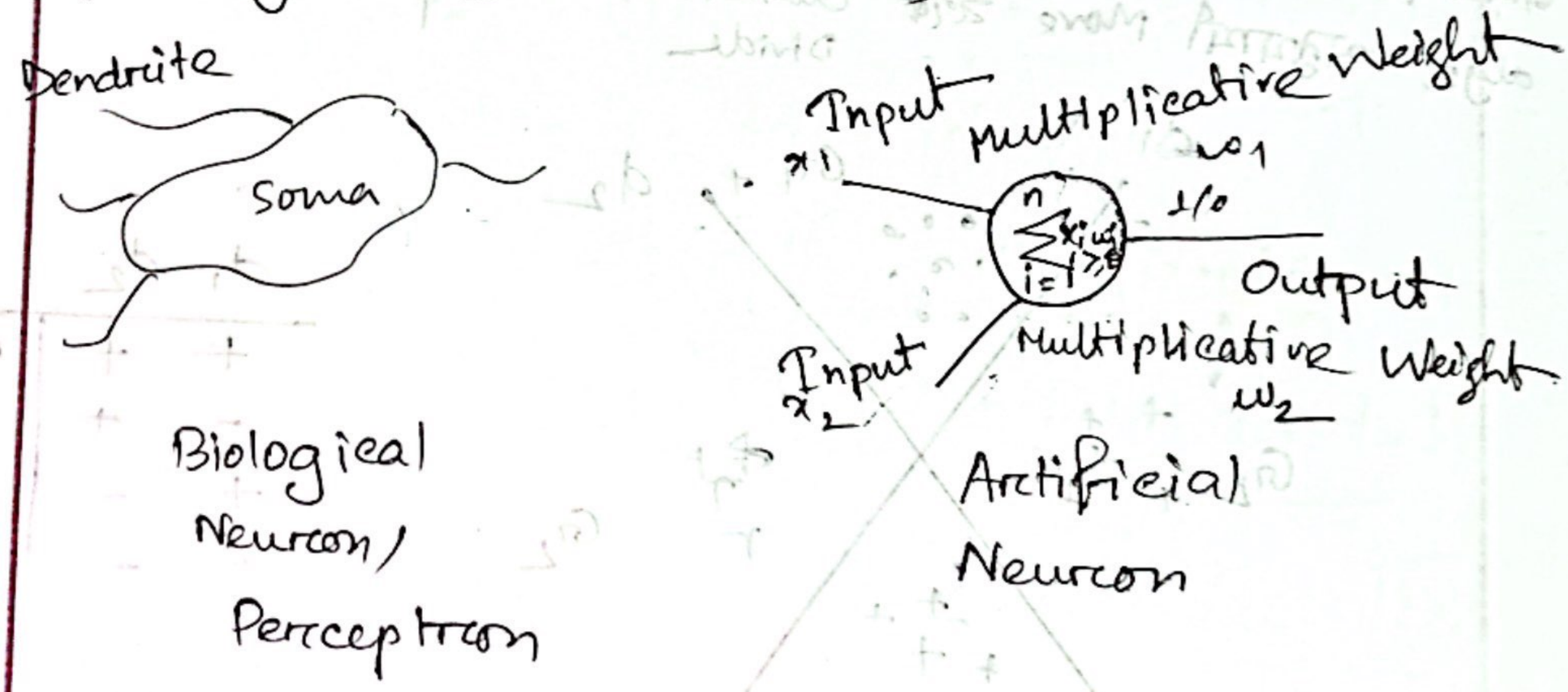
=> যে Problem গুলো Non-linear Classifier দিয়ে Solve করা যায় তাদেরকে N.L. Classifier দিয়েই Solve করতে হবে. But, অনেক সময় ইনপুট ডাটা Linear Decision Boundary দিয়ে অনেকসময় Classify করা সম্ভব.

Scanned with CamScanner



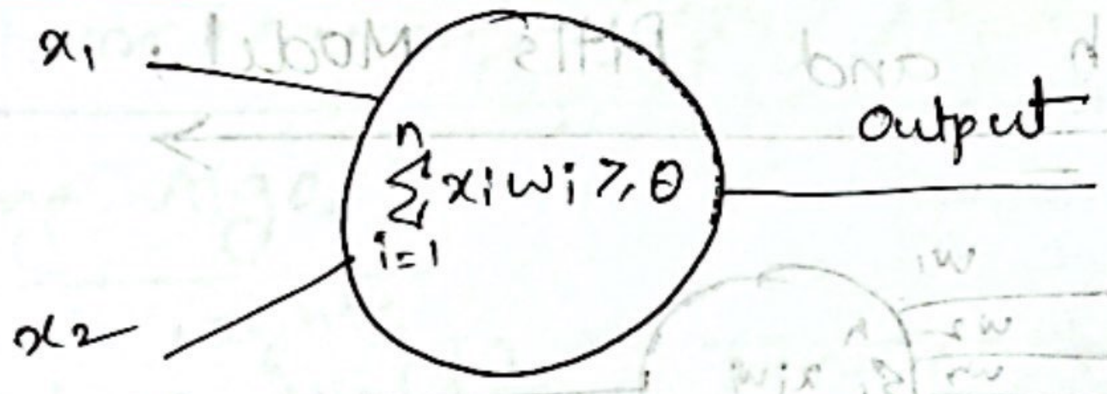
Chapter - 3 (V.V.I)

Modeling the single Neuron:

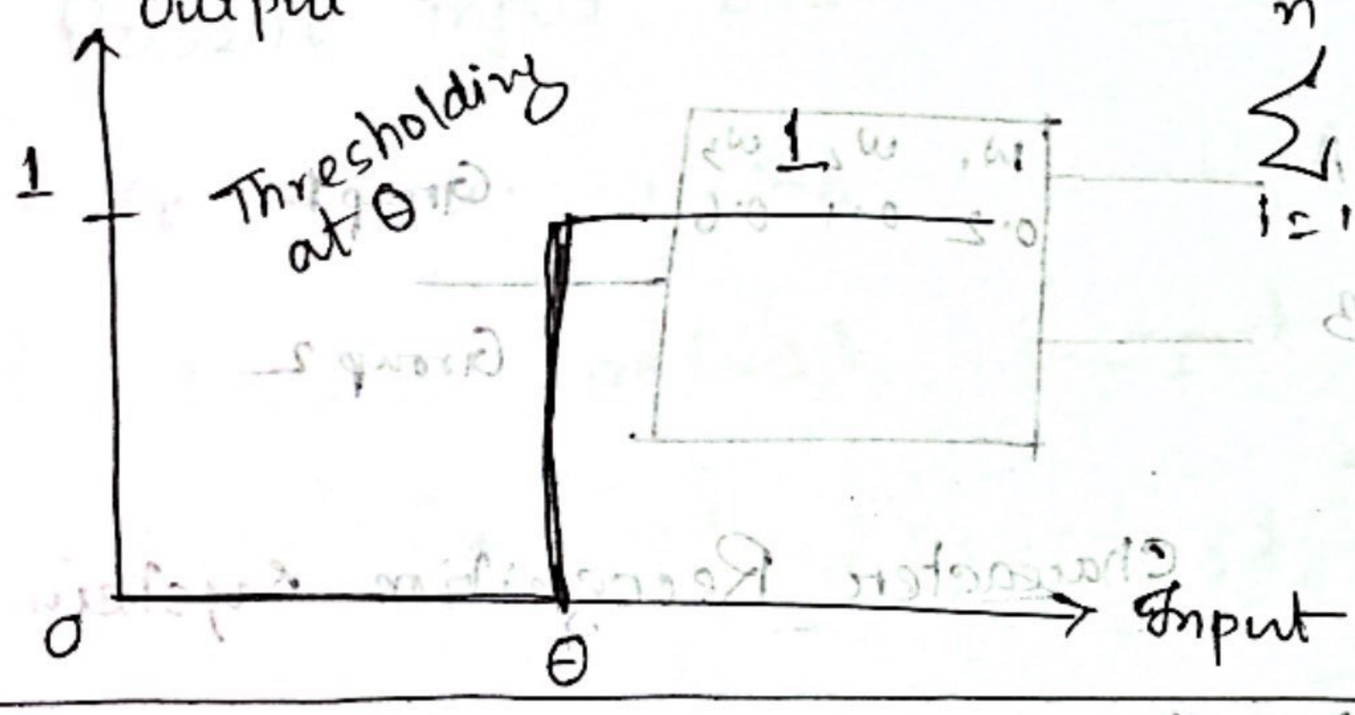


Total Input = (weight of line 1 × line 1 input)
+ (weight of line 2 × line 2 input) .... + (weight of line n × line n input)

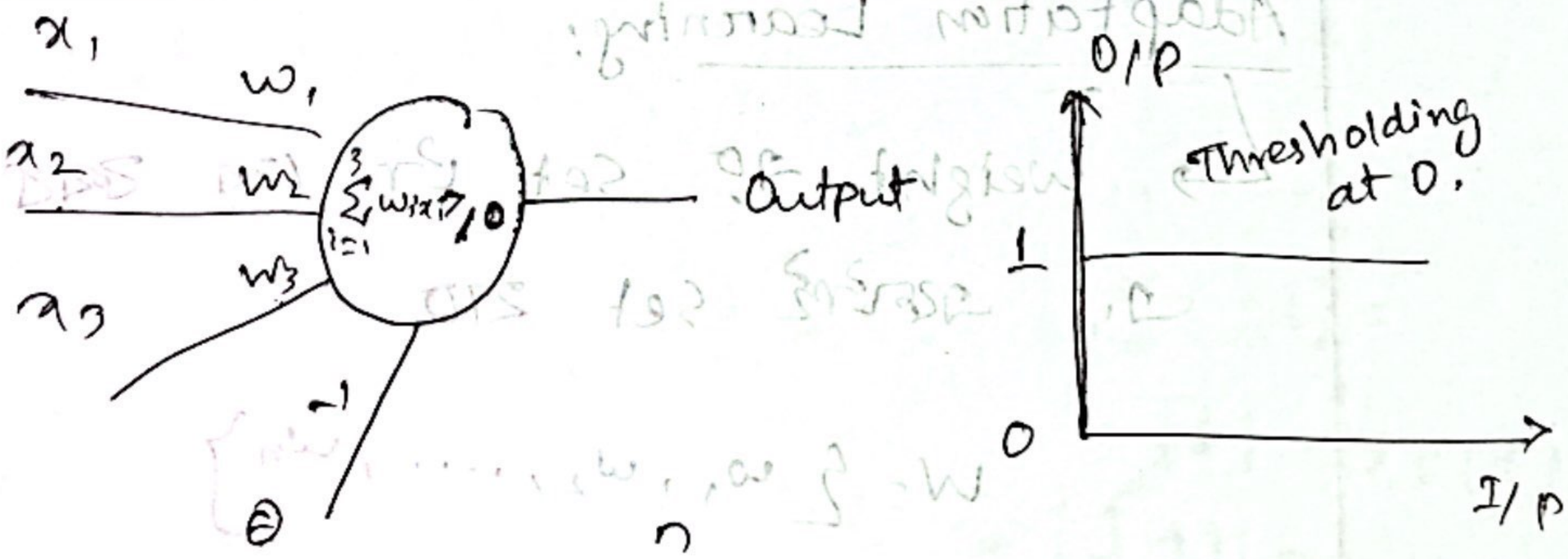
$= w_1 x_1 + w_2 x_2 + w \ldots + w_n x_n$



Thrushholding Function:



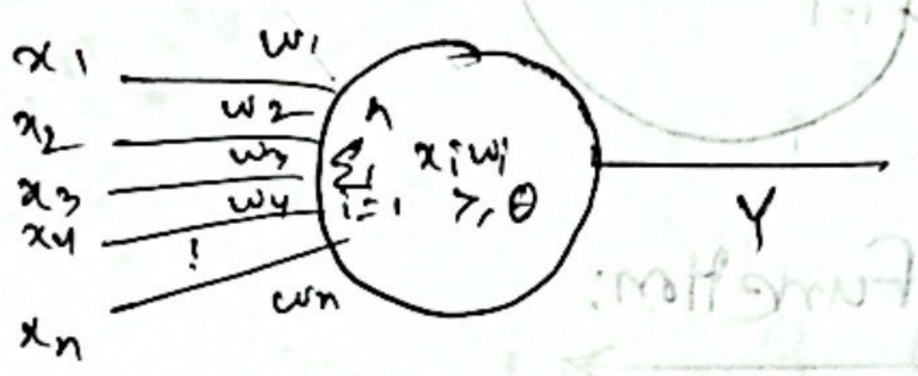
$$\sum_{i=1}^{n} w_i x_i \geq \theta$$

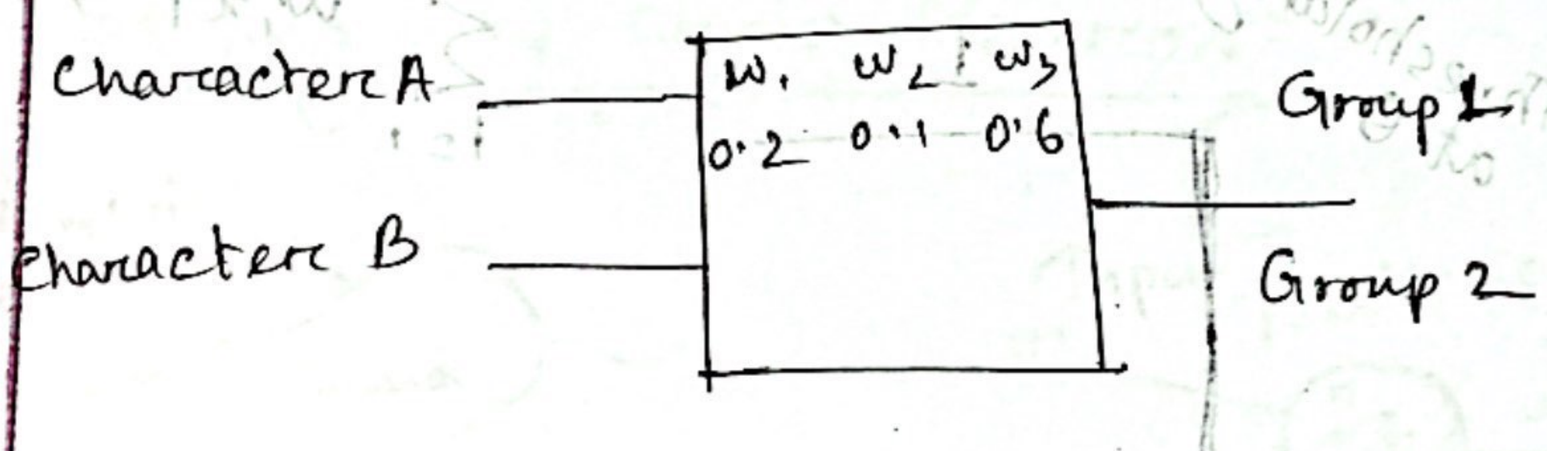


$$\sum_{i=1}^{n} w_i x_i - \theta \geq 0$$

Fig: Biasing the Neuron

McCulloch and Pitts Model:





Character Recognition System

Adaptation Learning:

→ weight এর set টা বের করার Learning এ, একটাই set হবে,

$$W = \{w_1, w_2, \ldots, w_n\}$$

Perceptron Learning Algo. / Single Layer Perceptron

Learning Algo. :

1. Define weights $w_i(t)$, where $0 < w_i(t) < 1$ and set threshold θ.

2. Present input and desired output $d(t)$

   $x_0, x_1, x_2, \ldots, x_n$

3. Calculate actual output

   $$y(t) = f_h\left[\sum_{i=0}^{n} w_i(t)\, x_i(t)\right]$$

4. Adapt weights

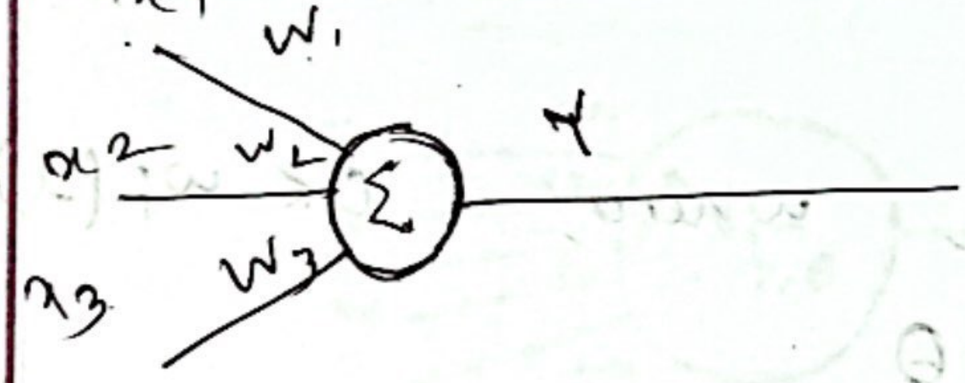
   if correct $w_i(t+1) = w_i(t)$

   if output 0, should be 1, $w_i(t+1) = w_i(t) + x_i(t)$

   if output 1, should be 0, $w_i(t+1) = w_i(t) - x_i(t)$

Scanned with CamScanner

Architecture of the Neuron



$W = \{ w_1 \quad w_2 \quad w_3 \}$
$1.2 \quad 0.4 \quad -0.4$ → এটার আসাগে algo অনুযায়ী calculate করে।

Problem Statement:

	x_3	x_2	x_1
Group 0	0	0	0
	0	0	1
	0	1	0
	0	1	1
Group 1	1	0	0
	1	0	1
	1	1	0
	1	1	1

Scanned with CamScanner

MKI

Class - 07

13/4/22

Step 4 of Perceptron learning Algorithm:

Adapt weights:

if correct $w_i(t+1) = w_i(t)$

if output 0, should be 1, $w_i(t+1) = w_i(t) + x_i(t)$

if output 1, should be 0, $w_i(t+1) = w_i(t) - x_i(t)$

Modification 01:

$\eta \rightarrow$ Gain term, $\eta = [0, 1]$

বাড়াতে হবে ধীরে ধীরে

So, η এর value টা এইভাবে set করব,

$$0 < \eta < 1.$$

Problem এর Complexity এর উপর depend করবে η এর value.

Modification 2 :

$$\Delta = d(t) - y(t)$$

desired → label

$$\boxed{w_i(t+1) = w_i(t) + \eta \cdot \Delta \cdot x_i(t)}$$

→ Widraw - Hoff delta rule

⊛ Apply to the widraw Hoff delta rule to single Neuron/ single layer অথবা, Perceptron learning এ এটাকে বলে,

Remember :

১টা Neuron নিয়ে কাজ হবে just.

Scanned with CamScanner

Limitation of Widrow-Hoff Delta rule:

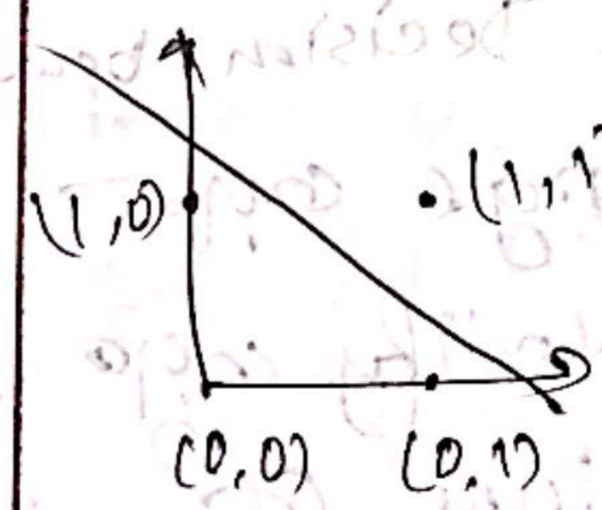
⇒ প্রথম 4টার জন্য 0 এবং বাকিগুলো 1 দিলে তো 000 এবং এর কখনোই label 1 আসবে না।

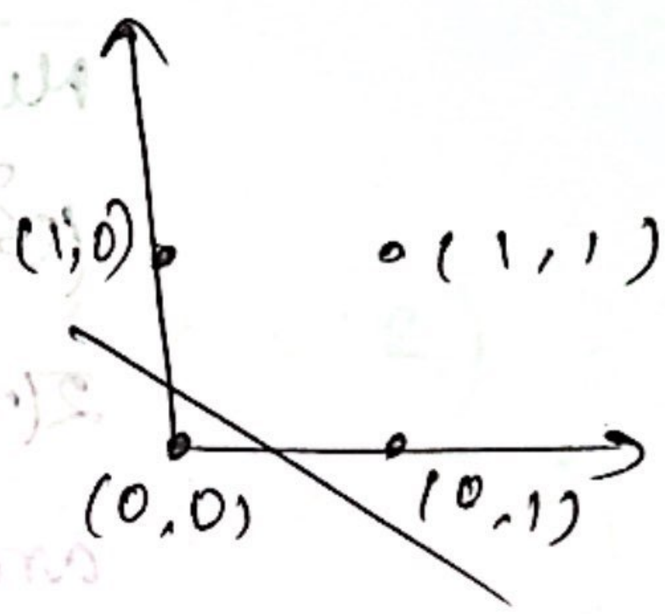
AND

0	0	0
0	1	0
1	0	0
1	1	1

OR

0	0	0
0	1	1
1	0	1
1	1	1





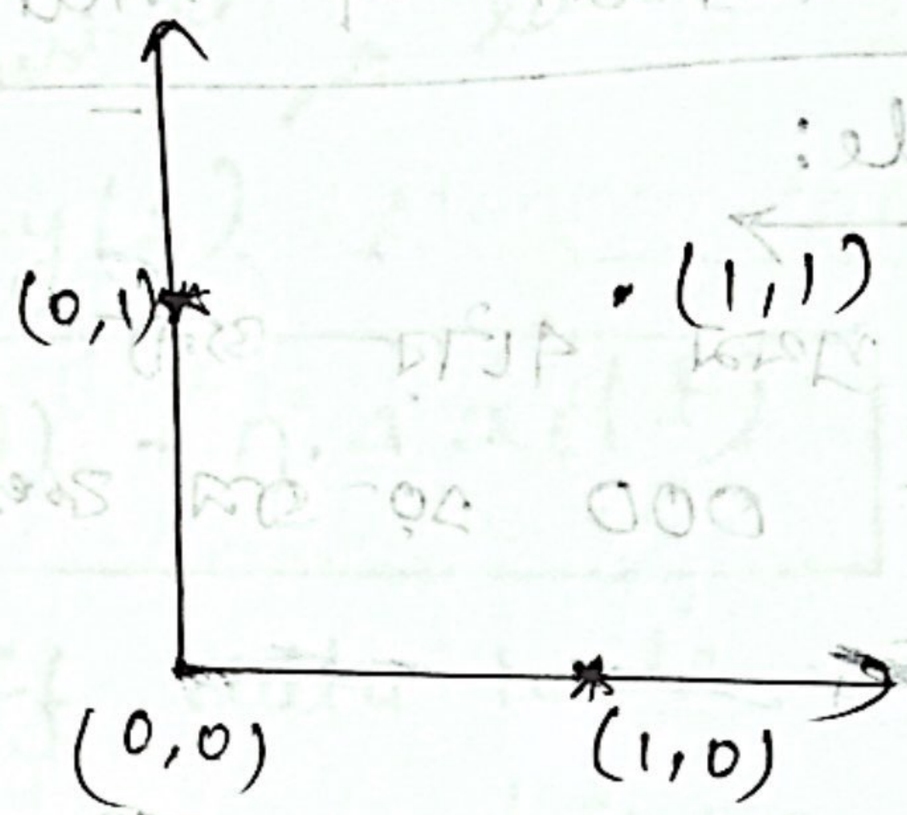
Scanned with CamScanner

Limitation:

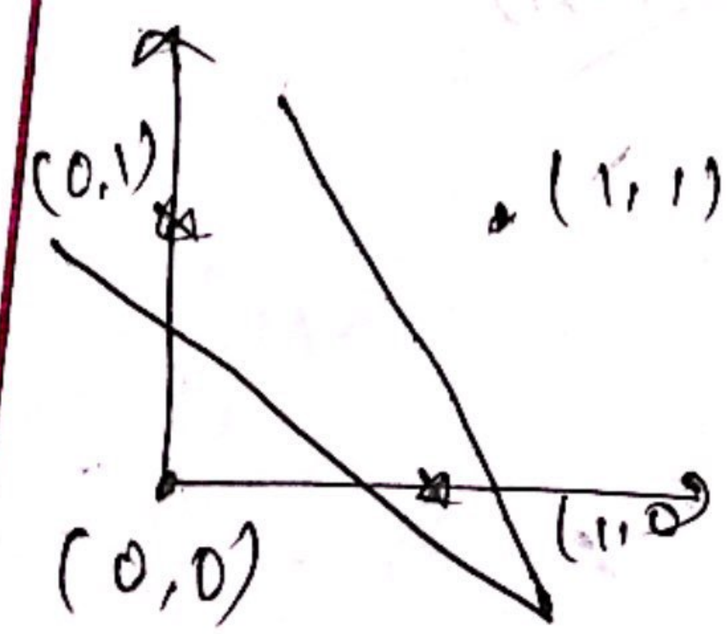
XOR এর Limitation

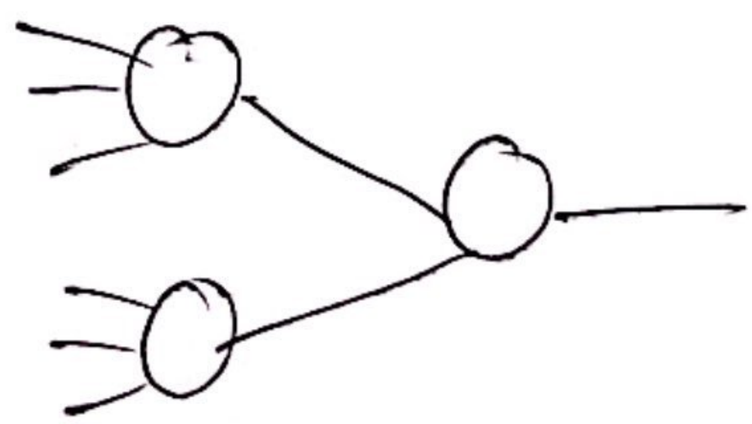
XOR

0	0	0
0	1	1
1	0	1
1	1	0



(Decision Boundary দিয়ে Separatable না)
কারণ, ২টা Neuron দিয়ে ১টা Decision boundary (Linear) তৈরি হয়,
Multiple Decision boundary তে Merge করতে হলে Modify করতে হয় architecture কে।





Scanned with CamScanner

Self - Study

Perceptron Algorithm (Exhibition):

Given,

$n = 2$ bits

$w = \{0.1, 0.2\}$

$th = 0.5$

So, table -

index	x_1	x_0	y
0	0	0	0 ✓
1	0	1	0
2	1	0	1
3	1	1	1

index = 0,

$y' = ?(0 \times 0.1 + 0 \times 0.2)$

$= ?(0)$ [is $0 < 0.5$]

$= 0$ ✓

is 0 matches y label? (Yes)

So, $y' = y = 0$

Scanned with CamScanner

index = 1,

$y' = ?\ (0 \times 0.1 + 1 \times 0.2)$

$= ?\ (0.2) < 0.5$

$y = y' = 0$

index = 2,

$y' = ?\ (1 \times 0.1 + 0 \times 0.2)$

$= ?\ (0.1) < 0.5$

$= 0 \rightarrow$ but it should be 1.

So,

$w = \{0.1 + 1,\ 0.2 + 0\}$

$= \{1.1,\ 0.2\}$

Now, $y' = ?\ (1 \times 1.1 + 0 \times 0.2)$

$= ?(1.1) < 0.5$

$= 1$

$\therefore y' = y = 1$

Scanned with CamScanner

index = 0,

Now,

$y^{\wedge} = ?\ (0 \times 1.1 + 0 \times 0.2)$

$= ?(0) < 0.5$

$= 0$

$\therefore y' = y = 0$

index = 1,

$y' = ?\ (0 \times 1.1 + 1 \times 0.2)$

$= ?\ (0.2) < 0.5$

$= 0$

$\therefore y' = y = 0$

index = 2,

$y' = ?\ (1 \times 1.1 + 0 \times 0.2)$

$= ?\ (1.1) < 0.5$

$= 1$

$\therefore y' = y = 1$

$\therefore$ final weight,

$w = \{1.1, 0.2\}$

index = 3,

$y' = ?\ (1 \times 1.1 + 1 \times 0.2)$

$= ?\ (1.3) < 0.5$

$= 1$

$\therefore y' = y = 1$

Scanned with CamScanner

MRI

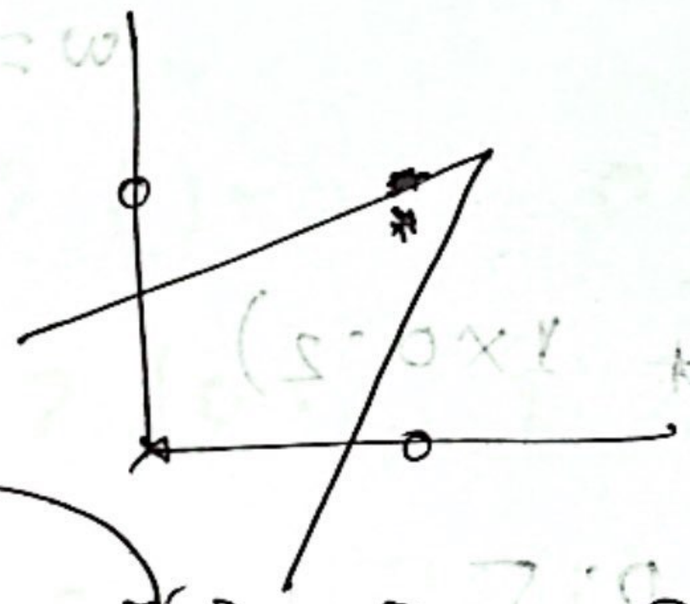
Class - 08+09

22/5/22

Chapter - 04

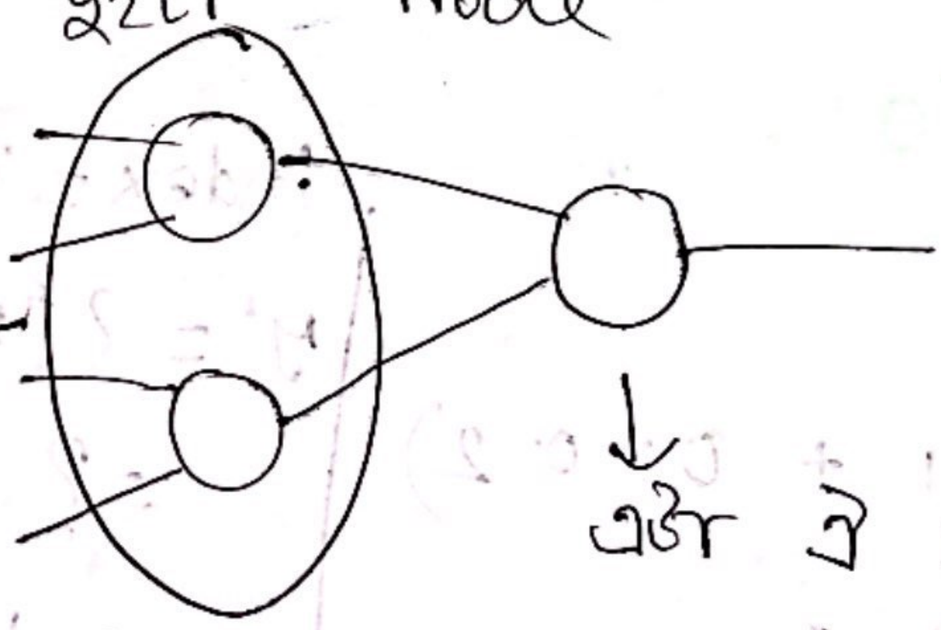
The Multilayer Perceptron:

For any case, 1টি node দিয়ে 1টি decision boundary possible

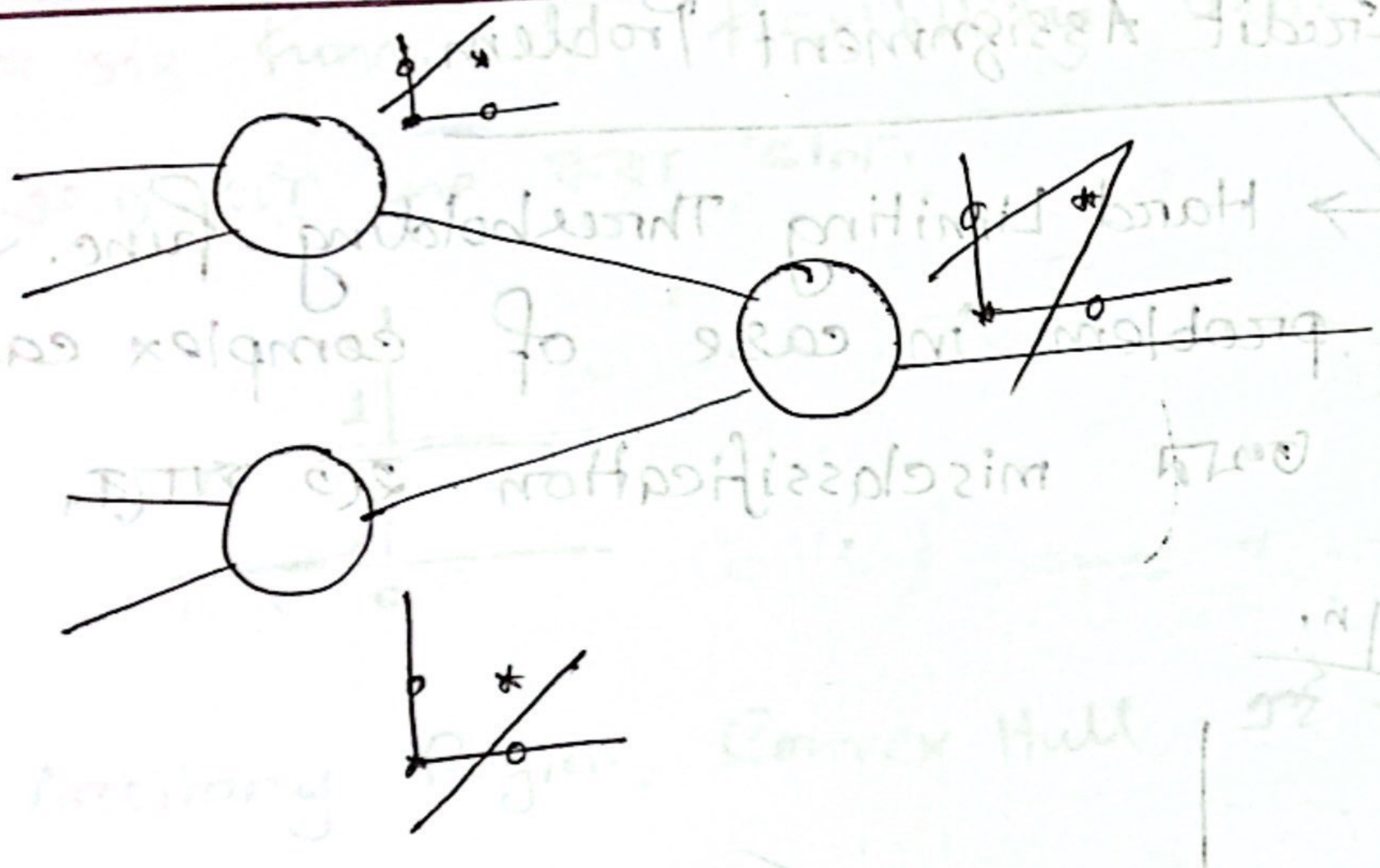


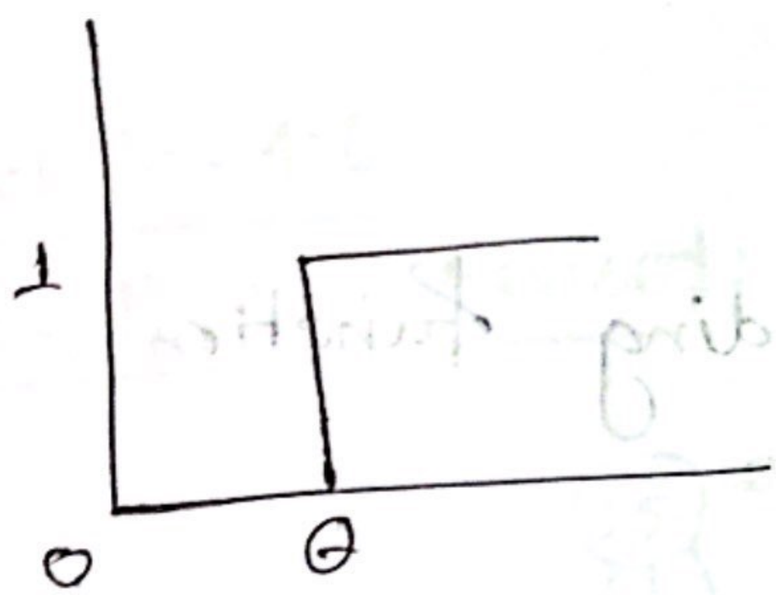
এক্ষেত্রে 2টি Dec. Boun. লাগবে.

So, দুইটি Node



এটা ঐ 2টি Dec. Boun. কে merge করার জন্য



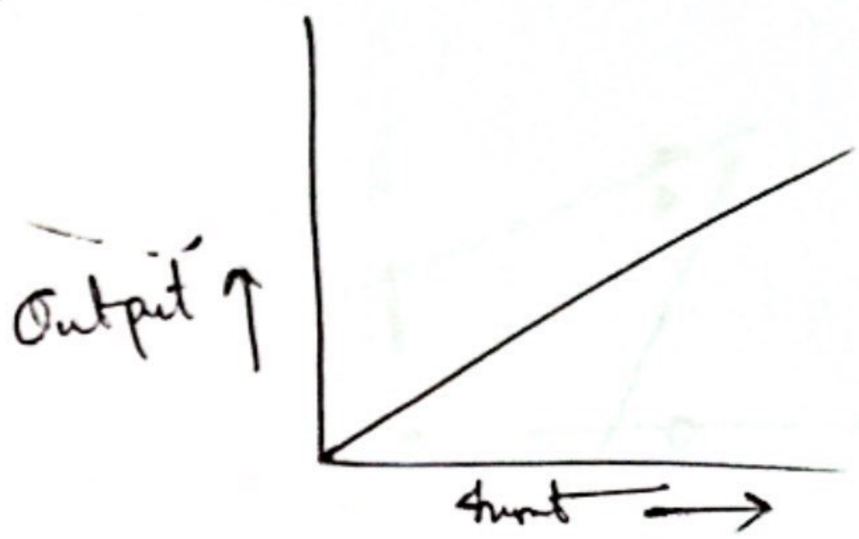


(step function / Hard-limiting thresholding function)

Credit Assignment Problem:

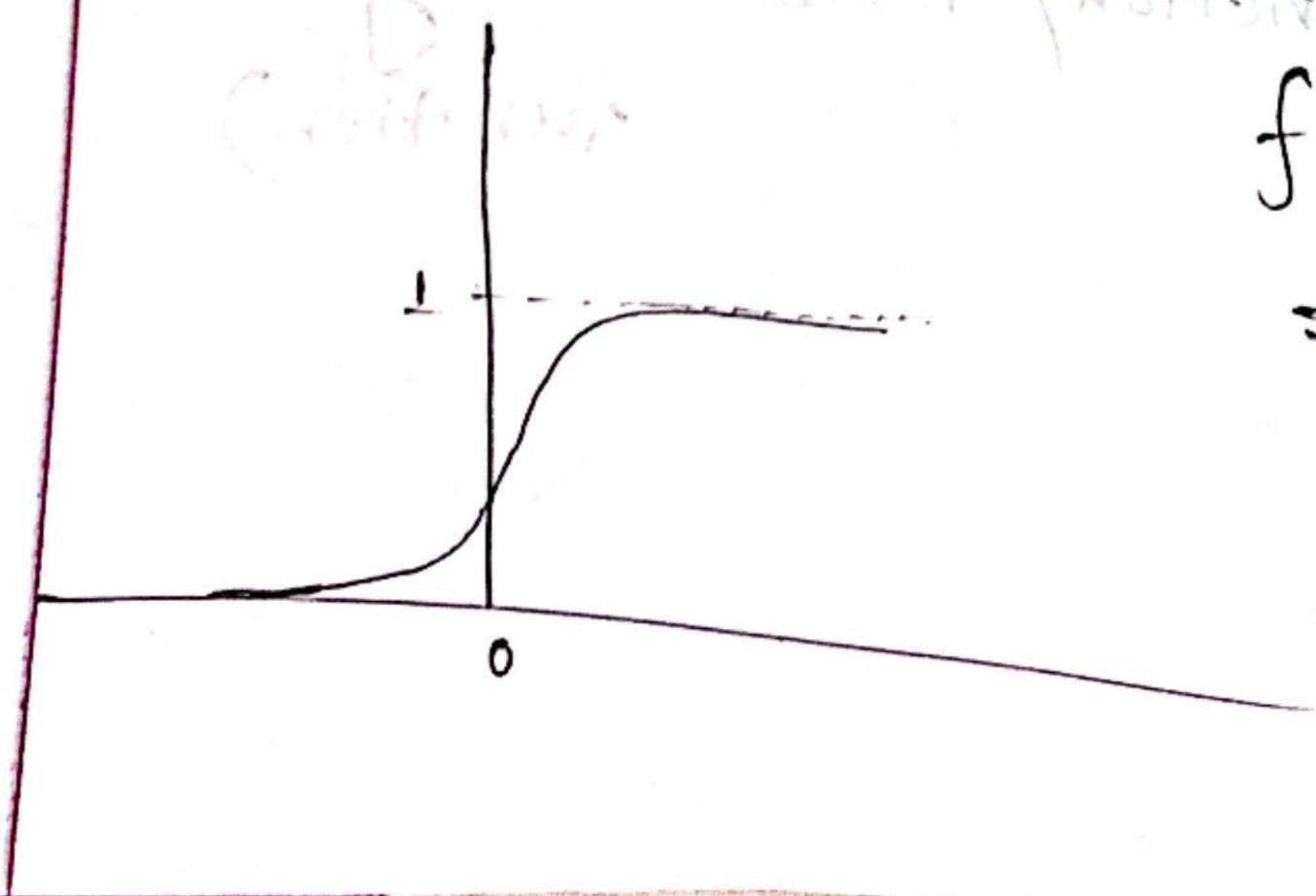
→ Hard Limiting Thresholding func. এর problem in case of complex cases. তখন misclassification হতে পারে।

Sol^n:



Linear Thresholding function

Sigmoidal Function

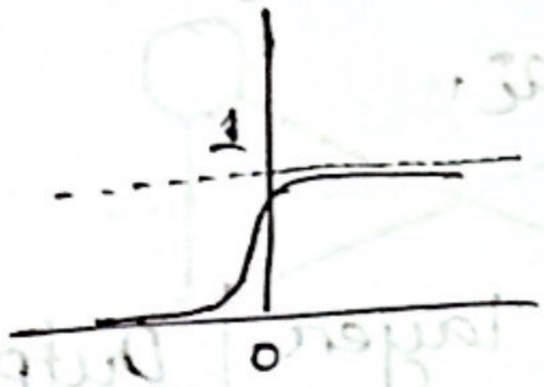


$$f(net) = \frac{1}{1 + e^{-k * net}}$$

net = weighted sum

k = speed factor

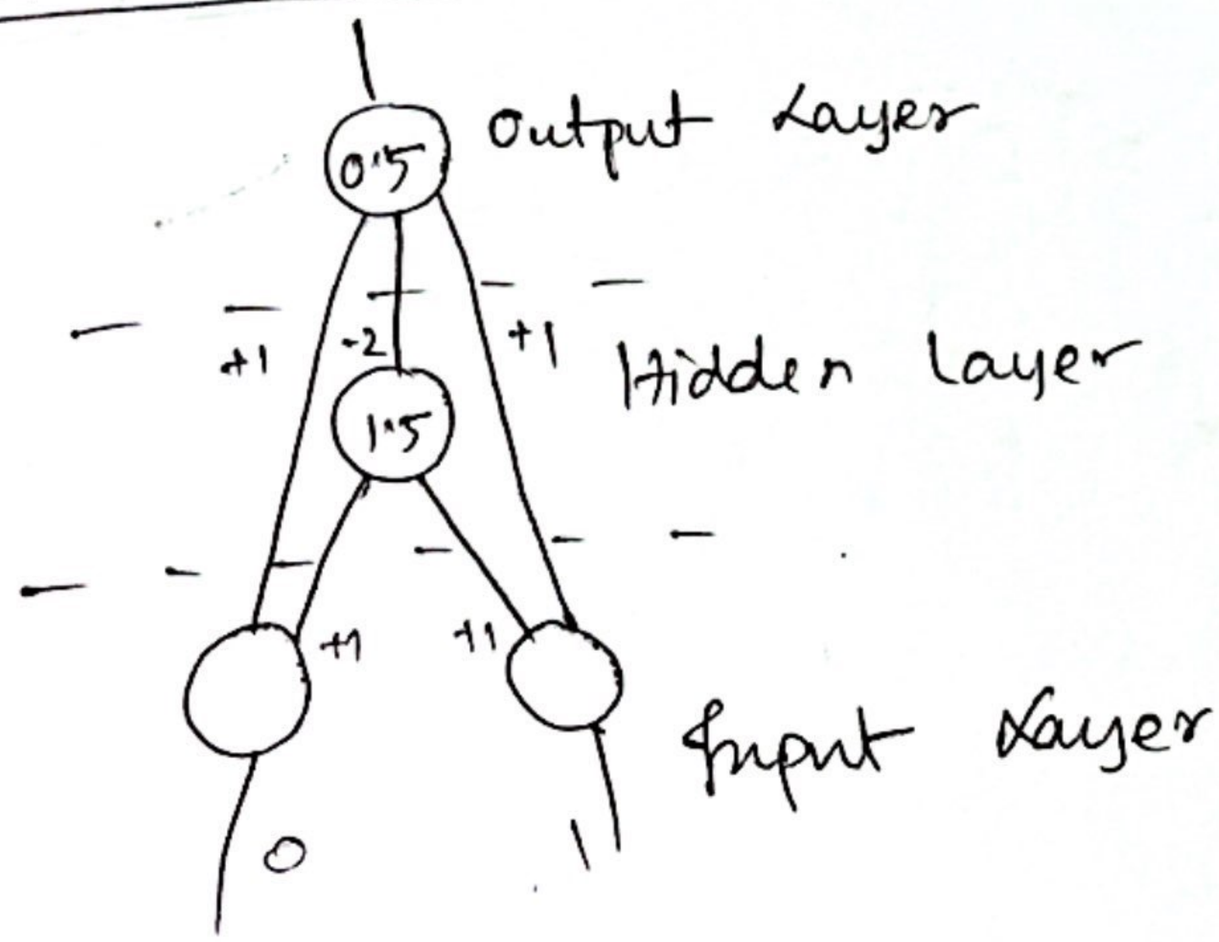
* sig. func. এর flexibility অনেক high এজন্য এটা use করা ভাল।



→ Arbitary Region, Convex Hull এই term গুলো দেখতে হবে।

Article - 4.6

XOR Problem Revised:



Layer এর unit calculate হলে active layer.

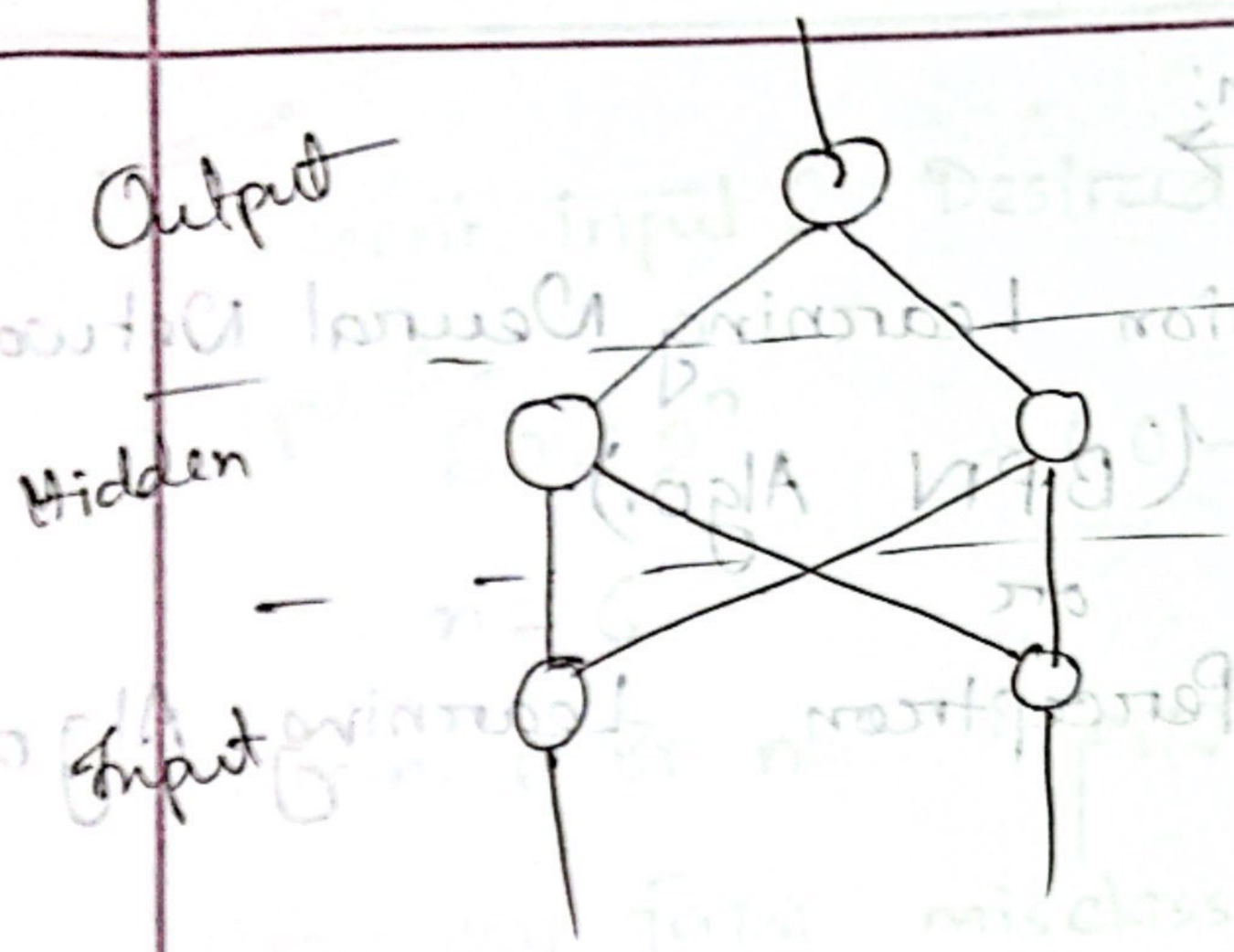
But এই বইয়ে হবে না সেটা

So, আমরা active layer হিসেবে as input layer এর parameter/weight মনে layer এ নাম্বার দিলাম।

Feature Detector

Input layer		Hidden layer	Output layer
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Scanned with CamScanner



আমরা এখানে দেখার চেষ্টা করলাম যে hidden layer এর কাজ কী.

MRJ

Class - 10

25/5/22

BPN Algorithm:

Backpropagation Learning Neural Network

(BPN Algo.)

or

Multilayer Perceptron Learning Algo.

Step: 1

$0 < W_{ij} = [3][2] < 1$

$0 < W_{jk} = [2][3] < 1$

} Initialize

$0 < V_{hj} = [2] < 1$] Initialize

$0 < V_{ok} = [3] < 1$] Initialize

Scanned with CamScanner

Step-2:

Present input & Desired Output

Output এ $60 < 2^n$ # 60 → ৬০টা class এ ভাগ করছি.

$n = 6$

∴ min. 6 টা node লাগবে.

6 এর কম দিলে misclassification হবে.

আর, input এ যতগুলো feature গুলো node নিব.

Like -

$x_{100} \cdots x_3 \; x_2 \; x_1$ হলে 100টা node নিব.

MRS

Class - 11+12

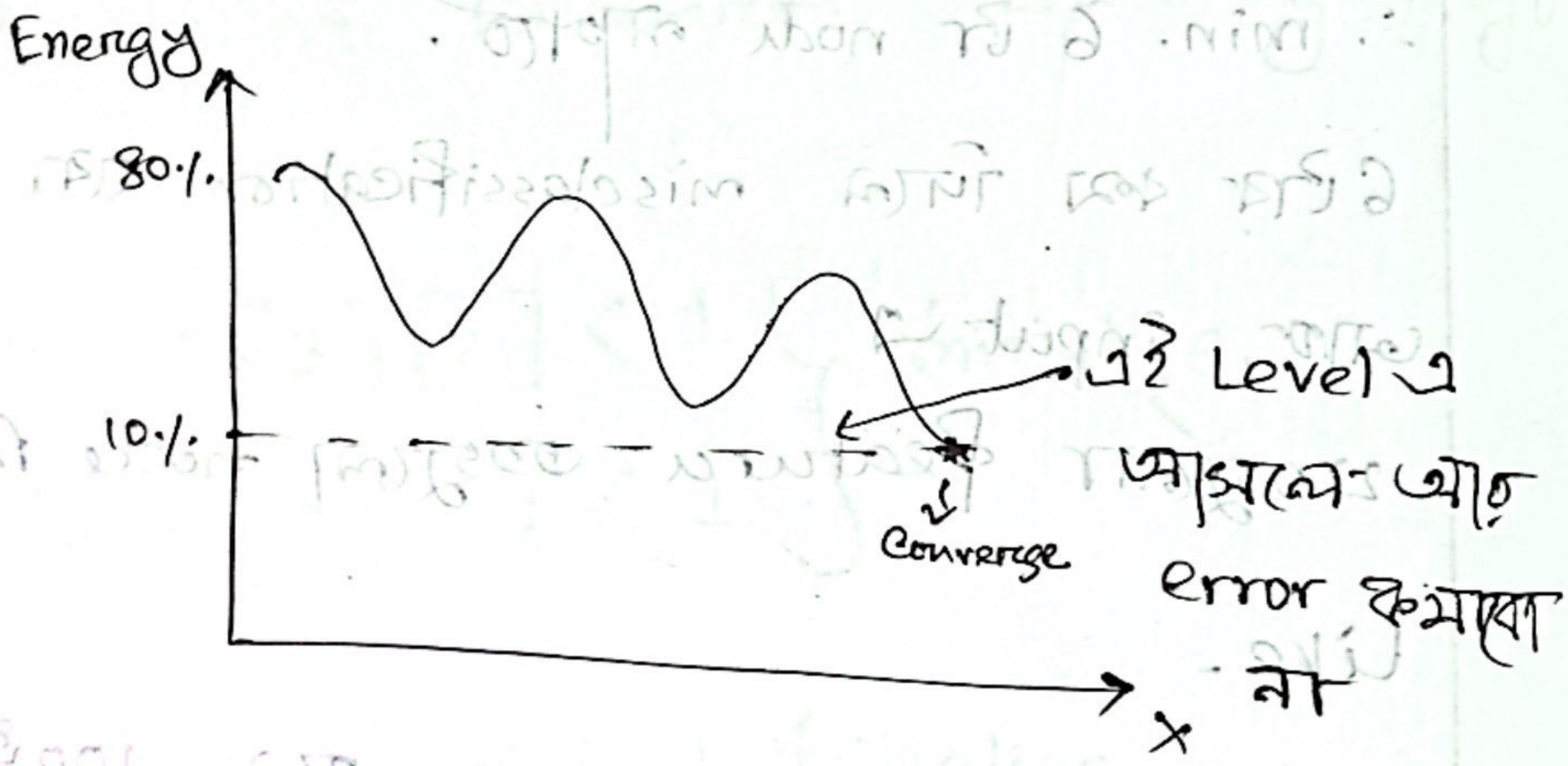
30/5/22

A.7 Visualizing Network Behaviour:

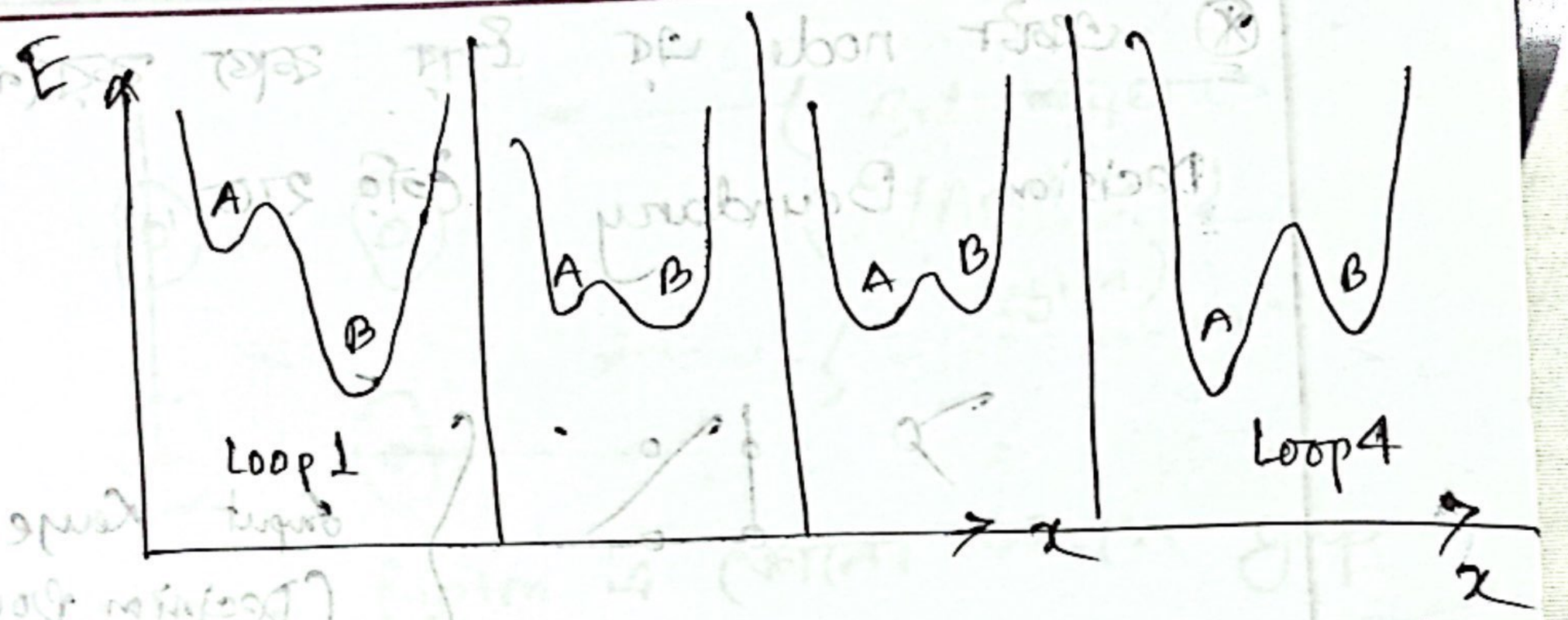
Energy/ error, $\delta = t - o$

↓

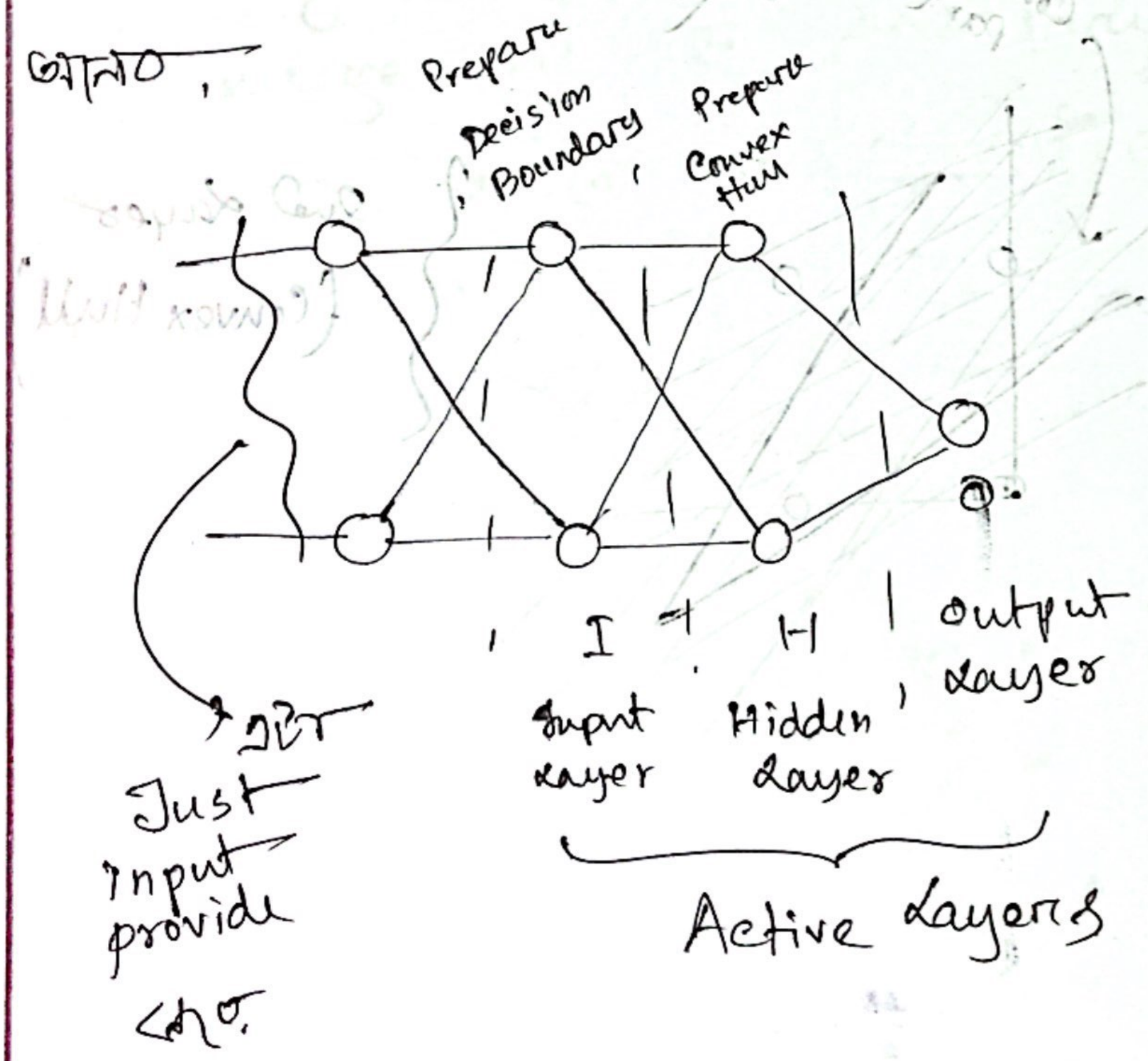
এটাকে reduce করতে হবে upto a satisfied value



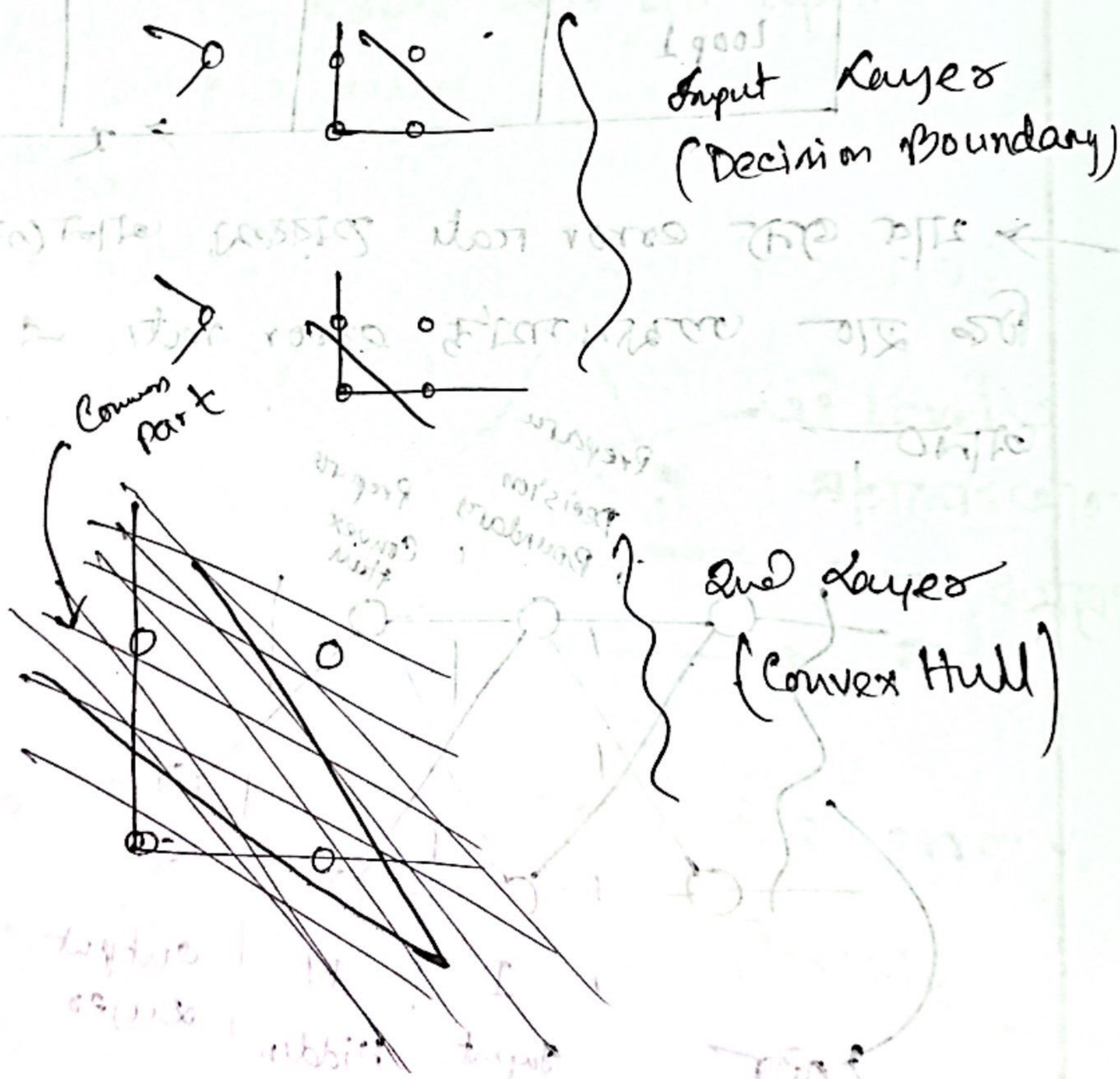
System বেশি secured হতে চাইলে error % অনেক lower করব, আর flexible রাখতে চাইলে বেশি না কমালেও চলবে.



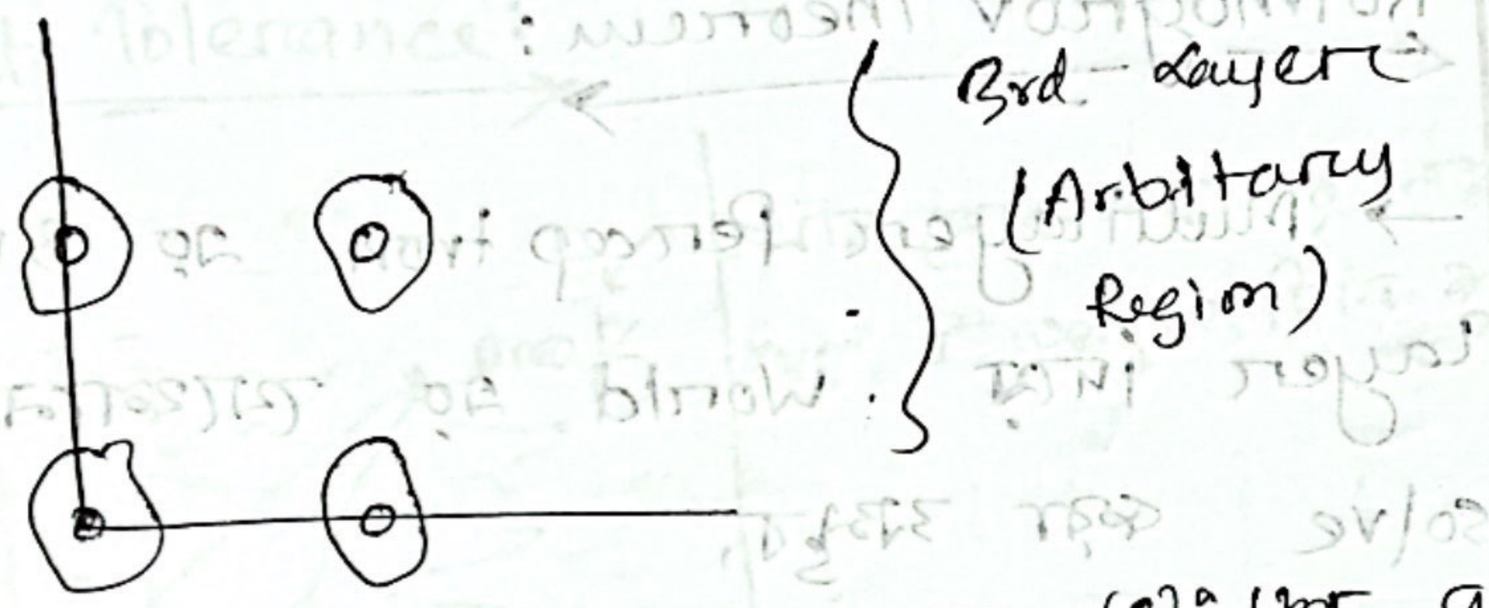
→ যার জন্য error rate বেরহলে আলাদা weight দিতে হাত তখনপর্যন্ত error rate এ modification আনাও,



* একটা node এর কাজ করলে ১টা Decision Boundary তৈরি হয়।



Scanned with CamScanner



* Arbitary Region এ কোনো রকম gap থাকবে না, Convex Hull merge করে Arbitary Region পাব; যদি পুরোটা convex hull merge না হয়ে Arbitary Region পাওয়া যায় তাহলে misclassification হতে পারে।

Kolmogrov Theorem:

→ Multilayer Perceptron এর 3টি Active layer দিয়ে World এর যেকোনো Problem solve করা সম্ভব।

Generalization:

→ Property to observe real world performance.

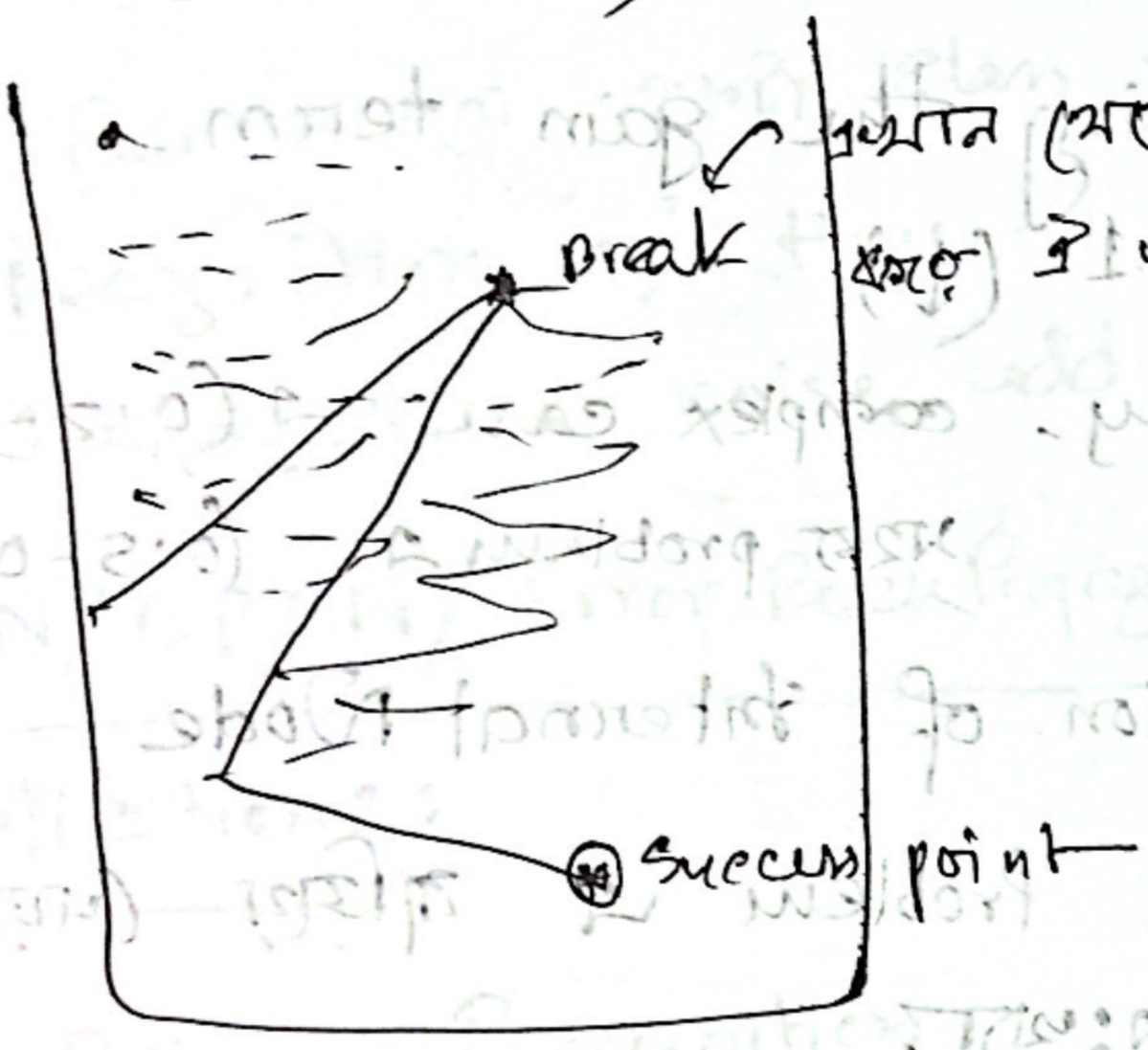
Interpolation → কিছু জানা আছে

Extrapolation → জানা নাই। তাও Predict করতে চাই

* Backpropagation এ Interpolation এ performance বাড়ে।

Scanned with CamScanner

Fault Tolerance:



এখান থেকে আবার নতুন করে weight দিয়ে শুরু

Learning Difficulties:

Local Minima:

→ একটা group এ যাওয়ার পর অন্য group এ চলে যাওয়া।

generally, 1% case এ BPN এ L.M (এখান থাকে।

Solⁿ

1) Lowering the gain term

$0 < \eta < 1$ (↓)

Naturally complex case → (0.2-0.4)

সহজ problem → (0.5-0.7)

2) Addition of Internal Node

→ Complex Problem এ বাড়িয়ে দেয়া হয় Node সংখ্যা।

3) Momentum Term Addition

$$\Delta w_{ij} = \cdots + \underbrace{\alpha \left(w_{ij}(t) - w_{ij}(t-1) \right)}_{\text{momentum term}}$$

$\alpha \rightarrow$ momentum factor

$0 < \alpha < 1$

Scanned with CamScanner

4) Addition of Noise

↳ Twin থাকলে একটা class এ অনেক face Recognition এর time এ testing এ সমস্যা এড়ানোর জন্য noise add করে দিতে।

BPN/BPLN/ Error Back Propagation Learning Algo.:

Applications:

→ Pattern Recognition

Scanned with CamScanner

MRJ

Class - 13

1/6/22

No. of. people = 100000 ← classify them

by using BPN → Exam → 10 marks

Problem

Prepare tasks -

1) Architectural diagram.

2) How to minimize learning difficulties.

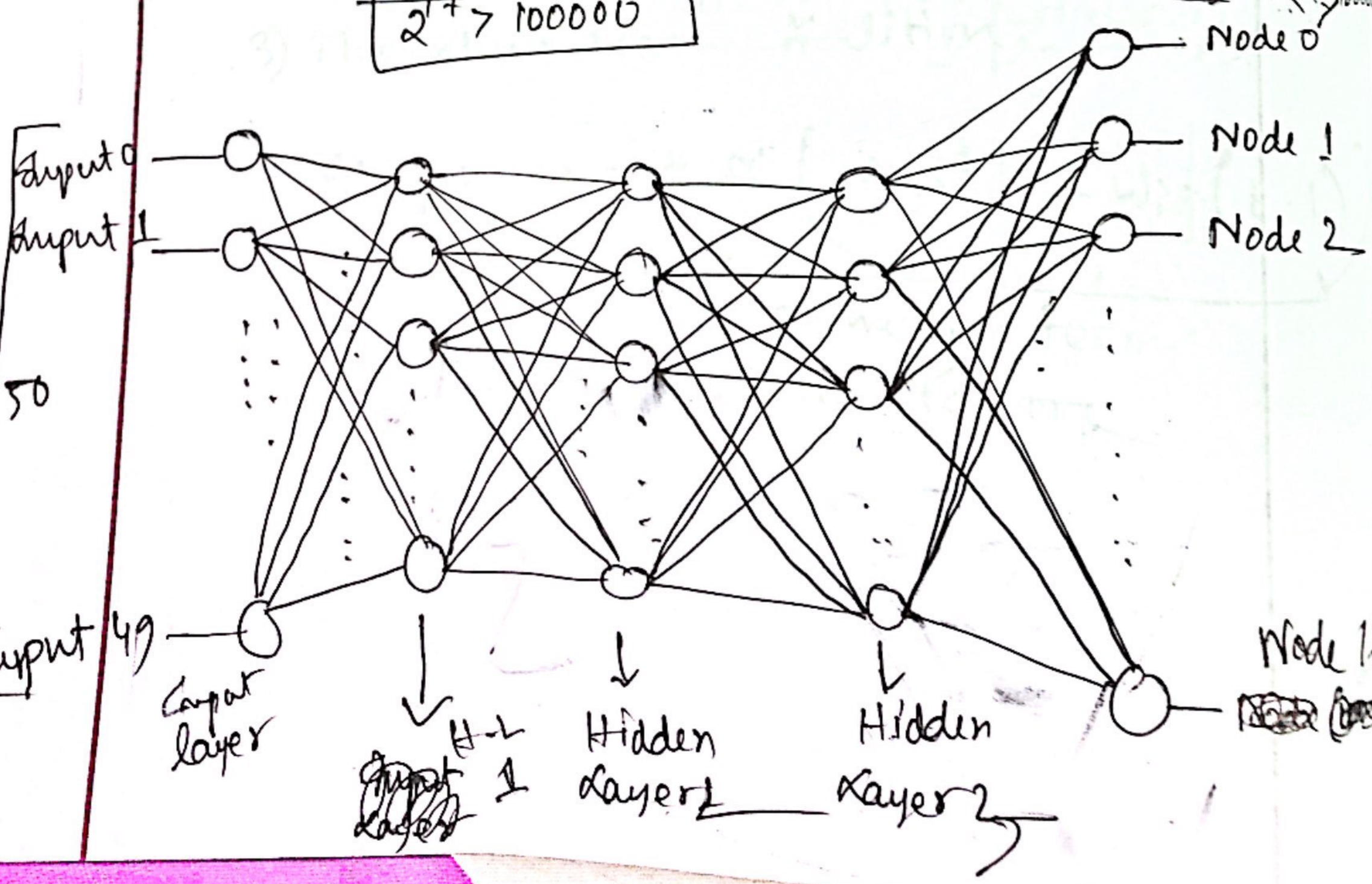
Sol^n:

①:

As, 100000 classes so, output nodes = N

$2^N > 100000$

$\boxed{2^{17} > 100000}$



Let,
there are 50 features. So, 50 nodes for denoting features. This's not active layer

no. of hidden layers = 2

total active layers = (1 + 2 + 1)
= 4

no. of nodes in each hidden layer = 10

" " " " input layer = 10

(2)

1) Local minima Problem
$0 < \eta < 1 \rightarrow \eta = 0.3$ সাধারণত নিয়ে

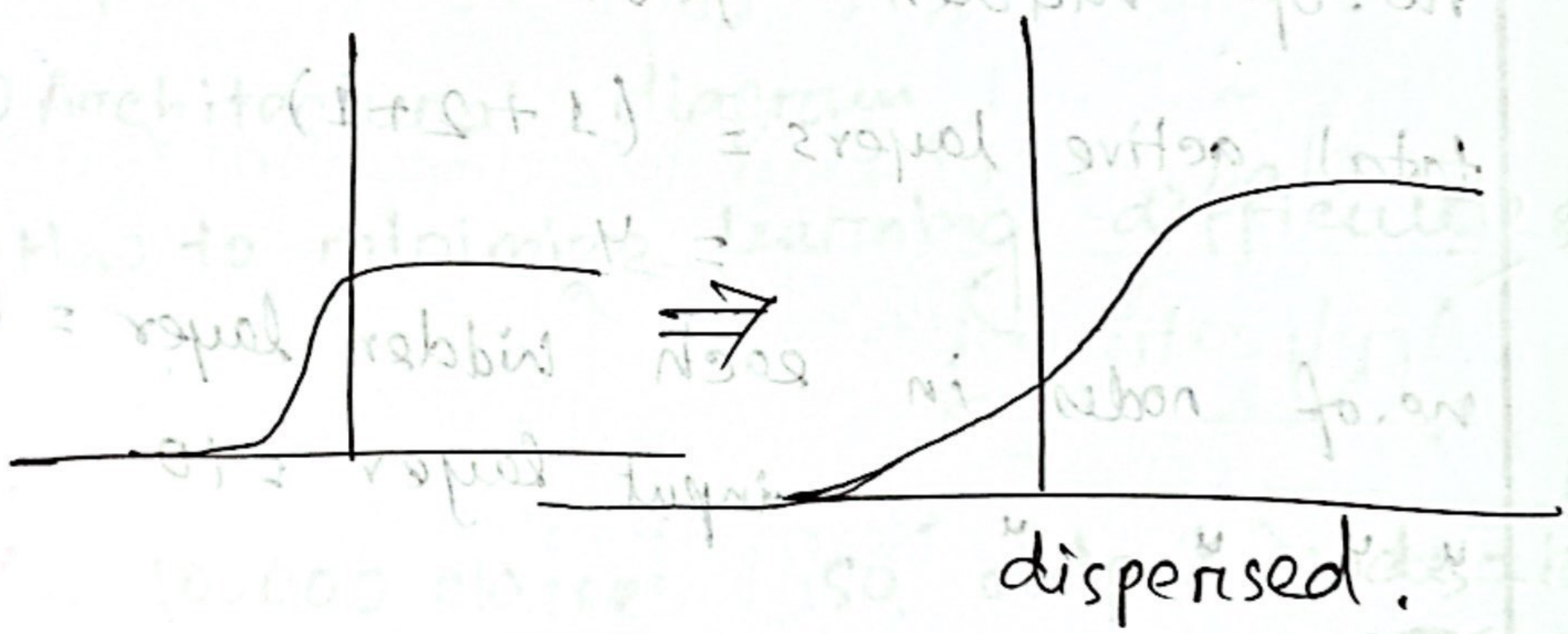
2) Momentum Factor Increment
$\alpha(w_{ij}(t) - w_{ij}(t-1))$

3) No. of nodes. Complex হলে বেশি
এবং, layer এর ক্ষেত্রেও same.

4) Relearning করাতে পারি।

Scanned with CamScanner

5) sigmoidal function এ 'K' (speed factor দিয়ে graph টাকে disperse করে দেয়া হত Complex problem এ.



6) Weight symmetric / similar হওয়া যাবে না, কোনকিছু layer গুলোর ভিতরে.

7) Error rate কত তে Converge করে?

→ Highly Secured এ E.R. কমিয়ে দেয়া হয়

→ কম " " E.R বাড়িয়ে দেয়া হয়,

Scanned with CamScanner

MRJ

Class - 14+15 5/6/22

Chapter-05

Kohonen Self-Organizing Neural Network Algo.:

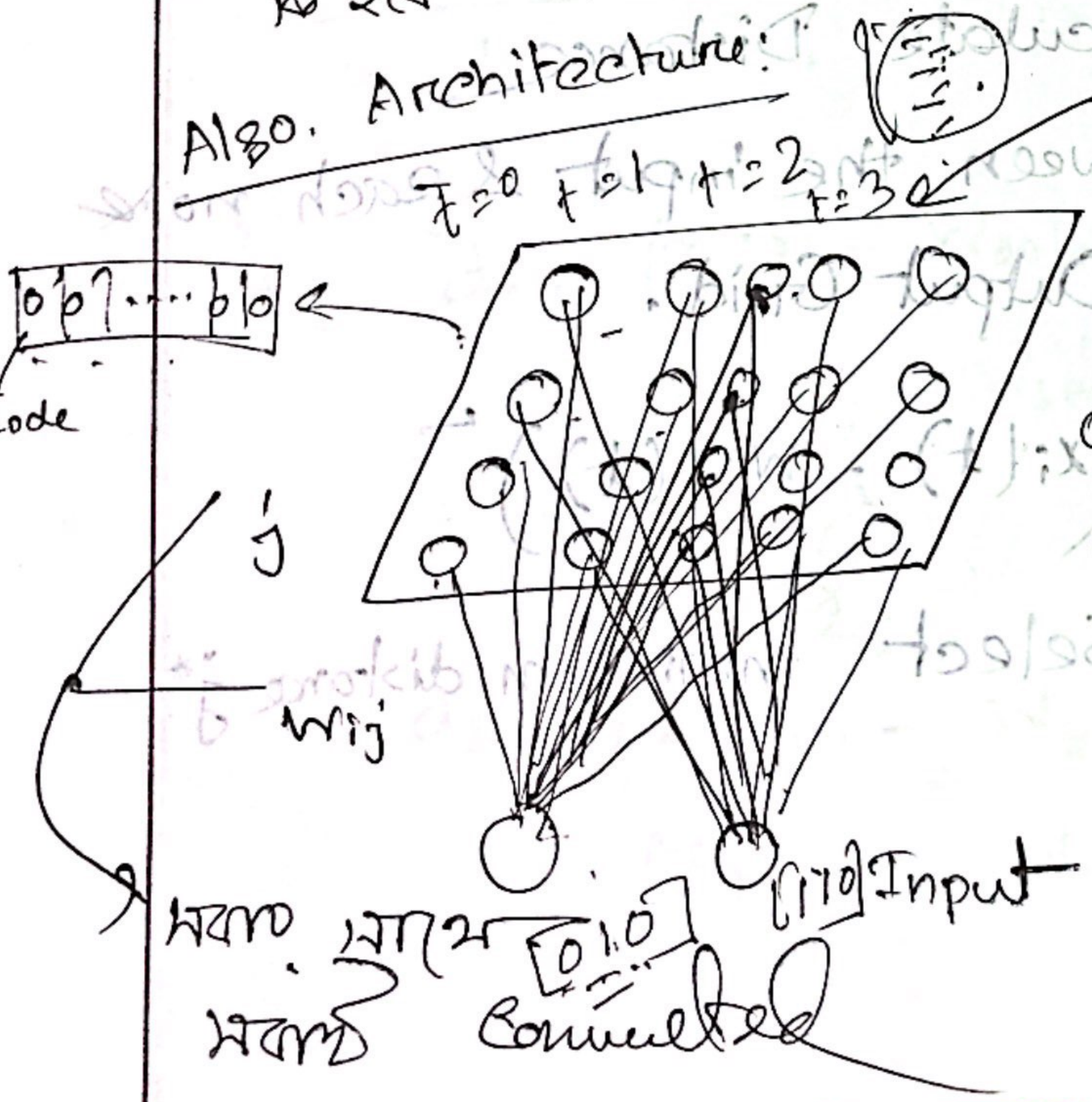
→ আগের algo. গুলো ছিল Supervised

→ এখন থেকে বাকিগুলো Unsupervised.

Unsupervised:

→ input এর corresponding এ Output কি হবে সেটা বলা থাকবে না।

Algo. Architecture:



এটা হচ্ছে 1-D array এর মত।

Output Grid (এখানে যতগুলো class লাগবে ততগুলো বা তার বেশি Node নিব)

Algo. Steps:

STEP 01: Initialize Network

Define weights $W_{ij}(t)$, $[0 < W_{ij}(t) < 1]$

set neighbourhood $N_j(0)$ [Generally 3 হয়]

STEP02: Present Input

$x_0(t), x_1(t), x_2(t), \ldots\ldots, x_{n-1}(t)$

STEP03: Calculate Distance

d_j between the input & each node of Output Grid.

$$d_j = \sum_{i=0}^{n-1} \left(x_i(t) - W_{ij}(t) \right)^2$$

STEP04: Select minimum distance j^*

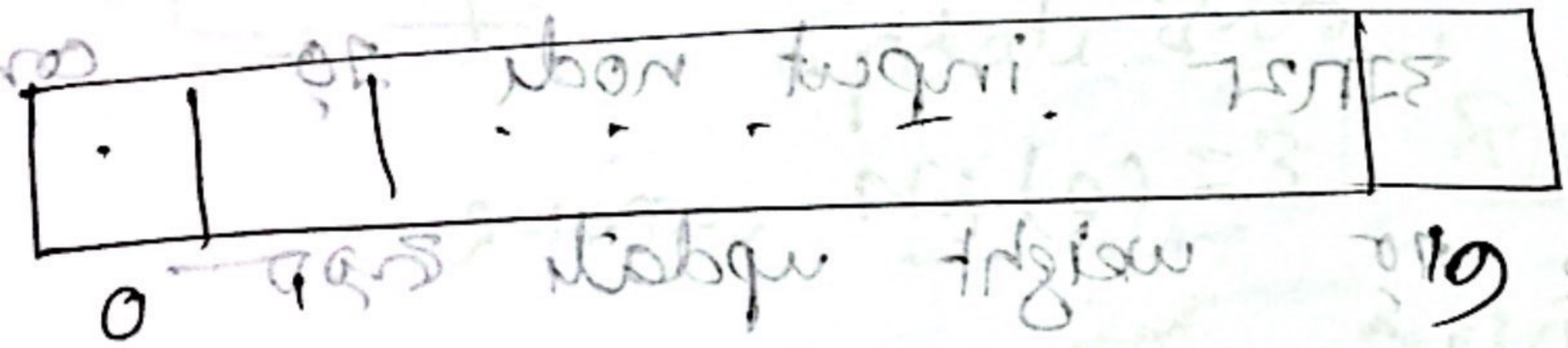
Scanned with CamScanner

STEP 05: Update weights for node j* and its neighbours

$$W_{ij}(t+1) = W_{ij}(t) + \eta(t)\,(x_i(t) - W_{ij}(t))$$

$$\boxed{N_j(t+1) = N_j(t) - \eta\, N_j(t)}$$

Neighbour Update, [neighbour এর size আরও আরও ছোট হয়ে finally একজনই থাকবে]



Firstly,

$$d_0 = (x_0(t) - W_{00}(t)) + (x_1(t) - W_{10}(t))$$

$$d_1 = (x_0(t) - W_{01}(t)) + (x_1(t) - W_{11}(t))$$

$\vdots$

এভাবে $i = 0, 1$ as 2টি node input 1

$$d_{19} = \left(x_0(t) - w_{0\,19}(t)\right) + \left(x_1(t) - w_{1\,19}(t)\right)$$

Then, sort করে d গুলো then

$$\boxed{d_{j(min)} = j^*}$$

Then, j^* যে node এর সাথে পেয়েছি তার সাথে input node এর connected line এর weight update হবে এবং তার neighbour গুলো update হবে, এখানে, neighbour এর হিসাব হবে distance array এর value থেকে.

$$\left(x_0(t) - w_{0\,j^*}(t)\right) + \left(x_1(t) - w_{1\,j^*}(t)\right) + \cdots$$

Such as:

	d
0	0.3
1	0.8
2	0.9
3	1.2
4	1.3
⋮	⋮
19	1.9

if $N_j(0) = 1$ then এখানে 1 বা তার থেকে কমপক্ষে value আছে তবে সেই no value এর node এর connected line এর weight কে update করব।

এখানে, $N_j(0) = 3$ কিন্তু so, দুরত্ব (0-19) পর্যন্ত Neighbour এর 19 এ গিয়ে value পাচ্ছি

$1.9 \leq 3$

Stopping Parameter:

① কোনো Neighbour থাকলে এবং Neighbour size ছোট হতে হবে,

এবং,

② এমনটা min. distance set করা পর্যন্ত শেষ করব।

Output grid এ

(*) এখানে সবার জন্য আলাদা weight set থাকবে।

Learning :

যেমন, 60 জন হলে তাদের weighted set এবং corresponding min. dis এর found node গুলো saved হবে

like.

$W_{ij} \rightarrow 7$

$W_{ij} \rightarrow 10$

$W_{ij} \rightarrow 12$

⋮

Testing :

কোনো এক Unknown person এর জন্য এই table এর সাথে comparison এবং min. distance calculate করে পেলাম, then, সেই weight এর জন্য

যে unknown feature এর minimum distance পাব সেই weight এর corresponding node কে then হবে দিব যে Unknown feature এটা কোথায় যাবে।

উদাহরণ -

distance

$W_{ij} \rightarrow 7 \rightarrow 0.2$

$W_{ij} \rightarrow 10 \rightarrow 0.3$

$W_{ij} \rightarrow 12 \rightarrow 0.1$

এর min. dis.

Thus যে Unknown কোনটার হচ্ছে সেই নাম্বার Node এর মধ্যে যাবে।

CT-01:

Chapter - 1, 2, 3

CT-02:

Chapter - 4

Scanned with CamScanner

MRI

8/6/22

Class - 16

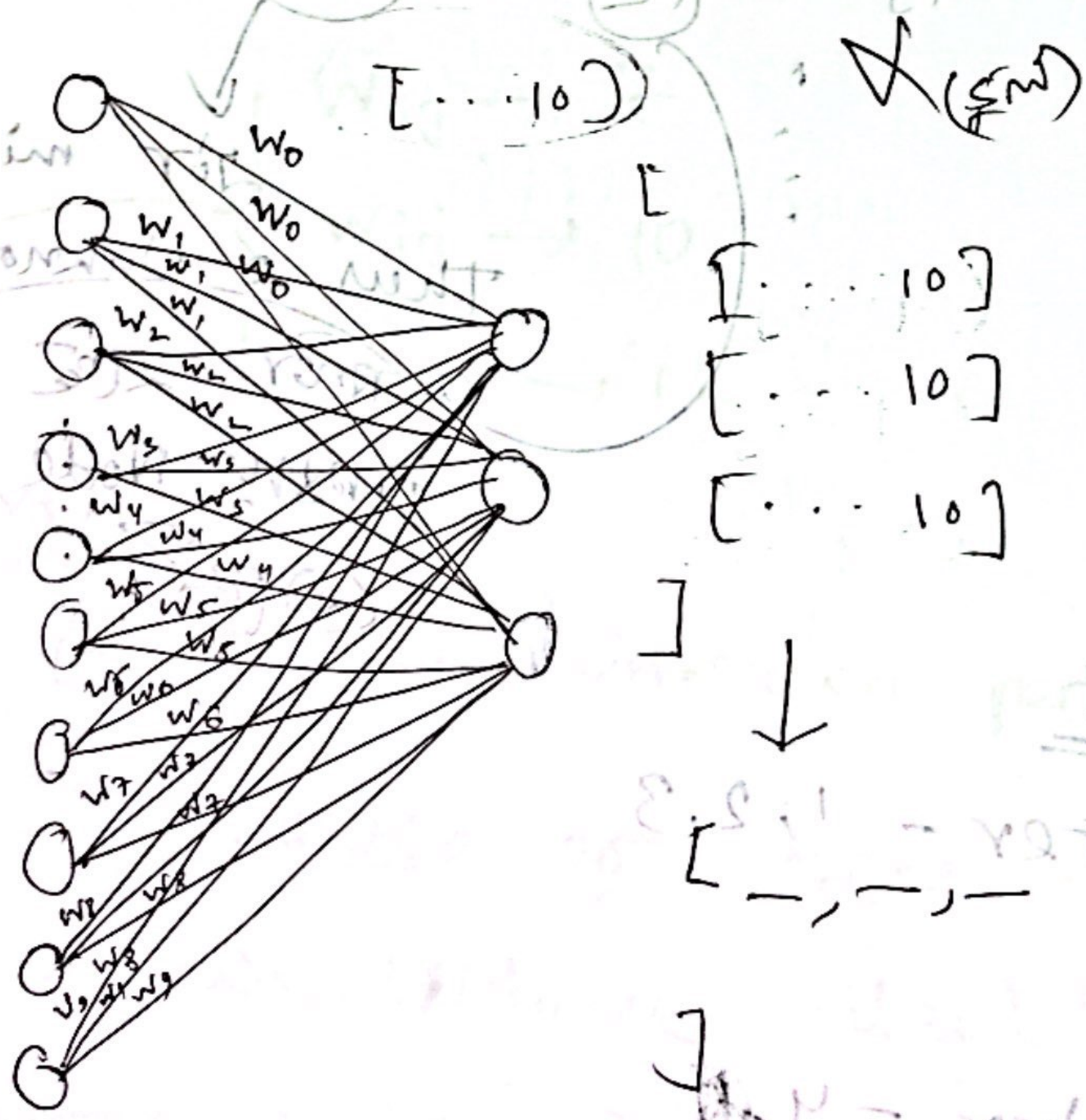
Vector Quantization:

Dimensionality Reduction → Most prominent dimension ও reduce করে নিয়ে আসা

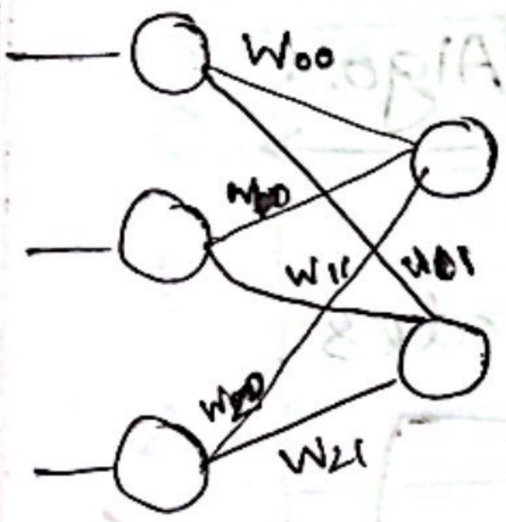
PCA, LDA, ICA → এগুলো D.R technique

LAB - 04

8/6/22



$$W_0 x_0 + W_0 x_1 + W_0 x_2$$
$$W_1 x_0 + W_1 x_1 + W_1 x_2$$
$$\cancel{W_2 x_0 + W_2 x_1 + W_2 x_2}$$



Scanned with CamScanner

MRS

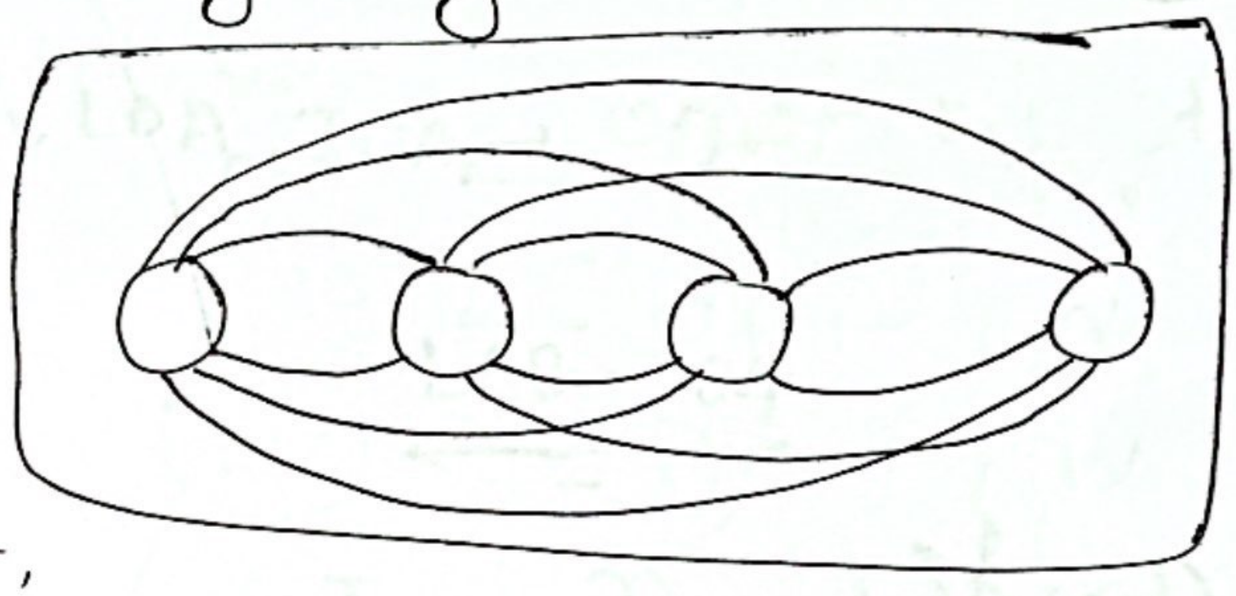
19/6/22

Class - 17 + 18

Chapter - 06

Hopfield Neural Networks - Algo. :

→ single layer multiple nodes.



একটাই layer

But,

এরকম হবে না,

সব Node থেকে সব Node এ যাবে

→ symmetrically weighted (means W_{01} যা হবে W_{10} তা হবে)

W_{ij}

যে দুটো Node নিয়ে কাজ করব তাদেরকে i, j বলছি,

Scanned with CamScanner

	0	1	2	3
0	0			
1		0		
2			0	
3				0

W_{00}

$W_{01} = W_{10}$

$W_{02} = W_{20}$

$W_{03} = W_{30}$

~~W_{04}~~

$\vdots$

$W_{32} = W_{23}$

↳ Bipolar input $(+1, -1)$

Algo:

Step-01: Assign connection weights

$$W_{ij} = \begin{cases} \sum_{s=0}^{M-1} x_i^s x_j^s, & i \neq j \\ 0, & i = j \end{cases}$$

where, W_{ij} is the connection weights

$s \rightarrow$ pattern

Scanned with CamScanner

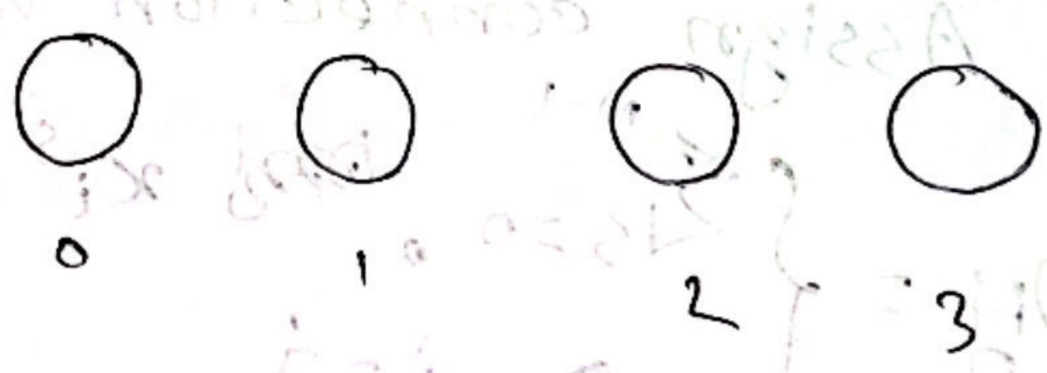
Step-02: Initialize unknown pattern

$$\mu_i(0) = x_i \qquad 0 \le i \le N-1$$

where, $\mu_i(t)$ is the output of node i at time t.

Step-03: Iterate until converge

$$\mu_i(t+1) = f_h\left[\sum_{j=0}^{N-1} w_{ij}\, \mu_j(t)\right] \quad 0 \le j \le N-1$$



Scanned with CamScanner

Let,

patterns be like -

$S0 = \{1, -1, -1, -1\}$

$S1 = \{1, 1, 1, -1\}$

$S2 = \{1, -1, -1, 1\}$

$S3 = \{-1, -1, 1, 1\}$

So,

$W_{01} = (1 \times -1) + (1 \times 1) + (1 \times -1) + (-1 \times -1)$

$= -1 + 1 - 1 + 1$

$= 0$

$W_{02} = (1 \times -1) + (1 \times 1) + (1 \times -1) + (-1 \times 1) = -2$

Weighted Matrix -

	0	1	2	3
0	0	0	-2	
1	0	0		
2	-2		0	
3				0

Scanned with CamScanner

step-02

Now,

a unknown pattern of $\mu_i(0) = \{-1, 1, 1, \ldots$ comes. এখন S_0, S_1, S_2, S_3 এর কার সাথে সবচেয়ে বেশি match করে এটা দেখব.

সব Node থেকে ঐ সময়ে যে Output বের হচ্ছে ঐটাই $\mu_i(t)$

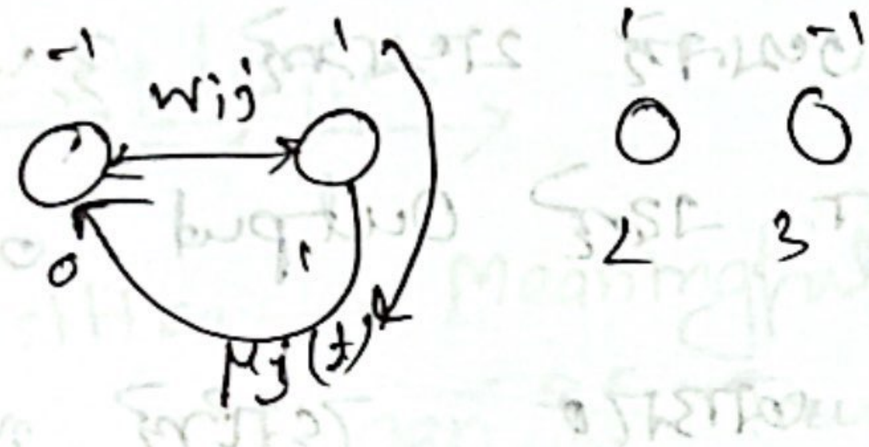
$$\overset{-1}{\bigcirc}_{0} \quad \overset{1}{\bigcirc}_{1} \quad \overset{1}{\bigcirc}_{2} \quad \overset{-1}{\bigcirc}_{3}$$

step-03:

$$\mu_i(t+1) = f_h\left[\sum_{j=0}^{N-1} w_{ij}\,\mu_j(t)\right], 0 \le j \le N-1$$

j হচ্ছে বই যে i আছে,

i-ই হচ্ছে মনে হয়,



Let,

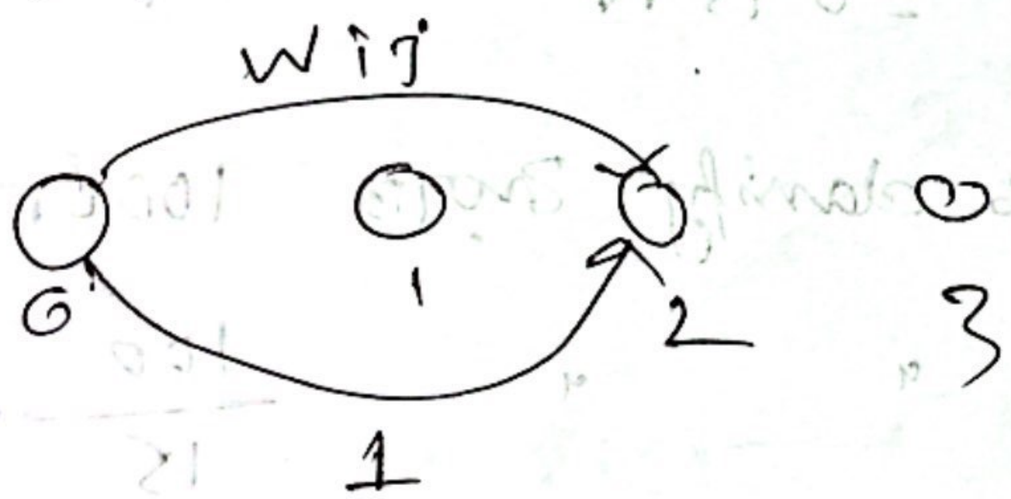
i যখন 0 j যখন 1,

তখন, W_{ij} $\mu_j(t)$ হবে,

j থেকে যে output আসে that means 1

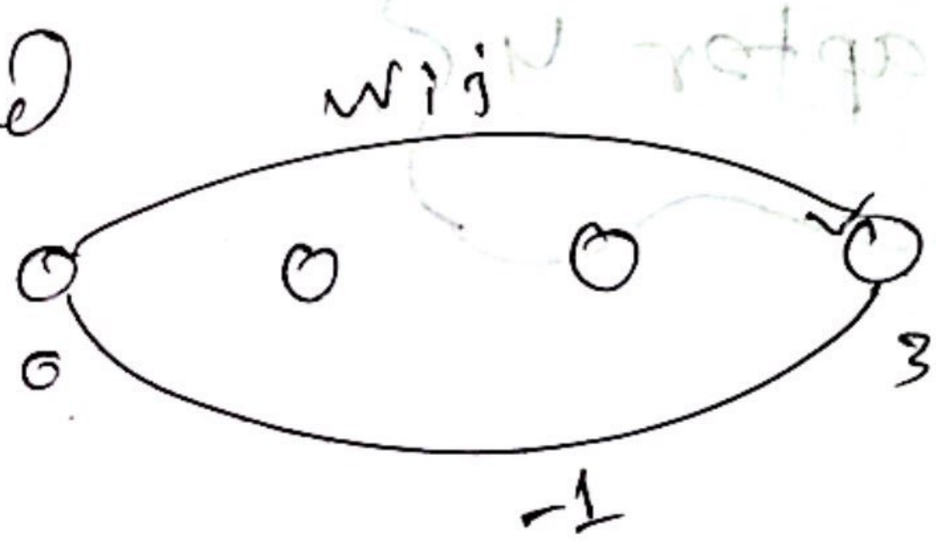
তাহলে W_{01} এর মান, That means $W_{01} \times 1$

এভাবে, চিত্রে অনুযায়ী পাবো like



$W_{02} \times 1$

and



$W_{03} \times -1$

So, $W_{01} \times 1$
$+ W_{02} \times 1$
$+ W_{03} \times -1$

যেহেতু i=0 হওয়ায়

Converge করতে সাধারণত 3, 4 টি iterate এ এসে এলে output আসবে,

Finally, যেটা আসবে সেটাই match করবে Bitwise সবগুলো given pattern এর সাথে, যার সাথে Percentage সবচেয়ে বেশি match করবে সেটাই হবে আমাদের Unknown pattern এর pattern.

No. of nodes $= 0.15\,N$

15টি pattern কে classify করতে 100টি Node লাগবে

$\therefore$ 1 " " " " $\frac{100}{15}$ " " "

CTH02 - Chapter 4

MRI

Class - 19 + 20 26/6/22

Fuzzy Logic:

Proposition → Meaningful statements that can be True or False.

→ 0 থেকে 1 এর মধ্যে vary করবে।

(today, 0.8)

today → Candidate for inclusion

0.8 → degree of membership

(today, 0.8) → Fuzzy set

Scanned with CamScanner

MRI

Class - 21 + 22 17/7/22

→ Fuzzy Logic PDF

Class 22 + 23 24/7/22

MRI

Fuzzy Logic Controller

Class - 24 27/7/22

MRI

→ Just Lab এর Code related আলোচনা হয়েছে।

Scanned with CamScanner

MRJ

NN

7/8/22

Genetic Algorithms:

- Normally Searching এর জন্য used হয়
- exact বা optimimum solⁿ দেয়
- Robust Condition এ কাজের Output দিতে,

*** Goldburg এর বই এর Chap-1 থেকে পড়তে হবে

Genetic Algorithm.

GA operations
- Reproduction
- Crossover
- Mutation
- Objective/ fitness Function ← এখন থেকে কিন্তু Qus. আসবে,

Crossover:

→ Single Point Crossover

→ Multi Point "

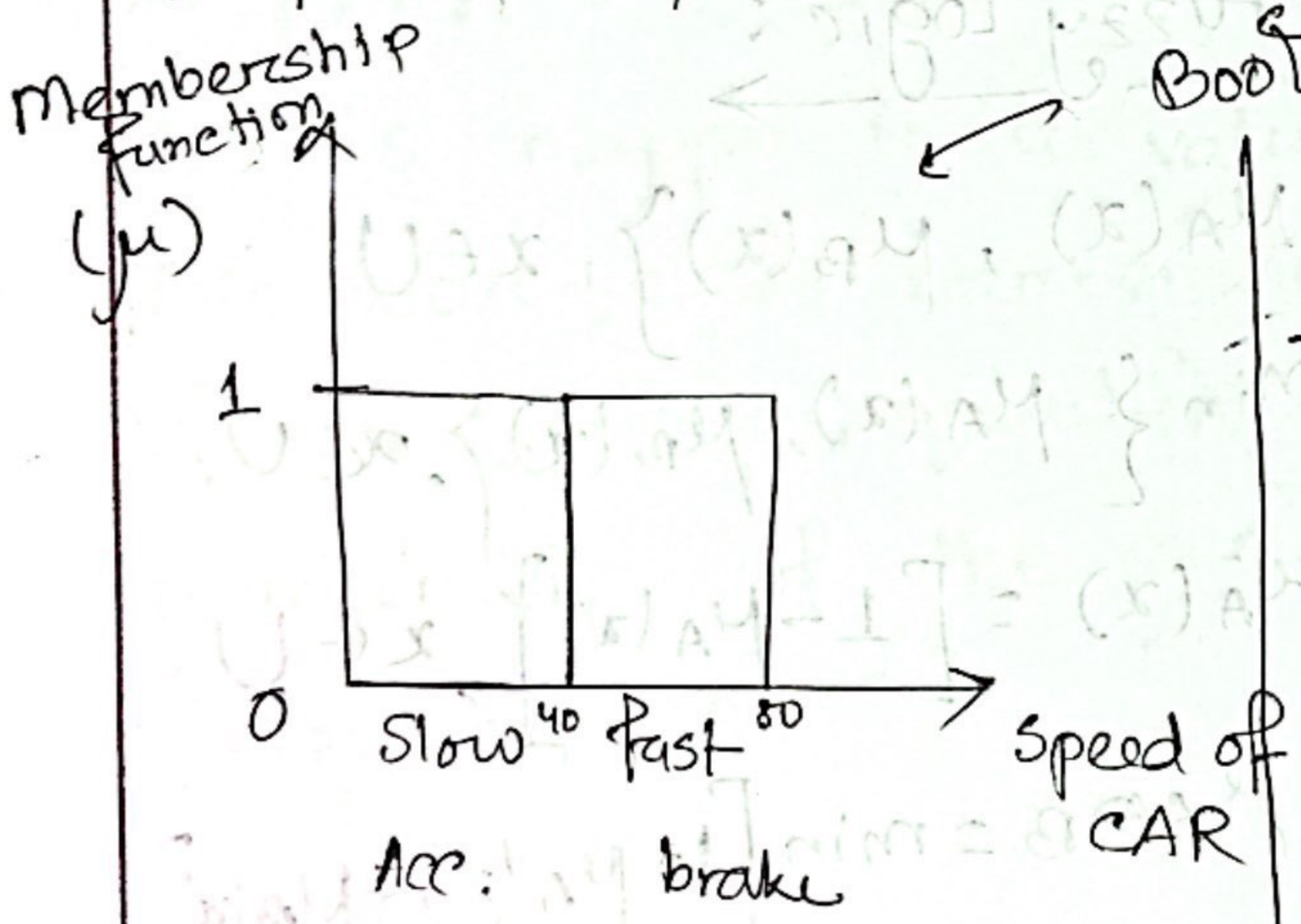
Scanned with CamScanner

Fuzzy Logic

F.L → Superset of Boolean Logic

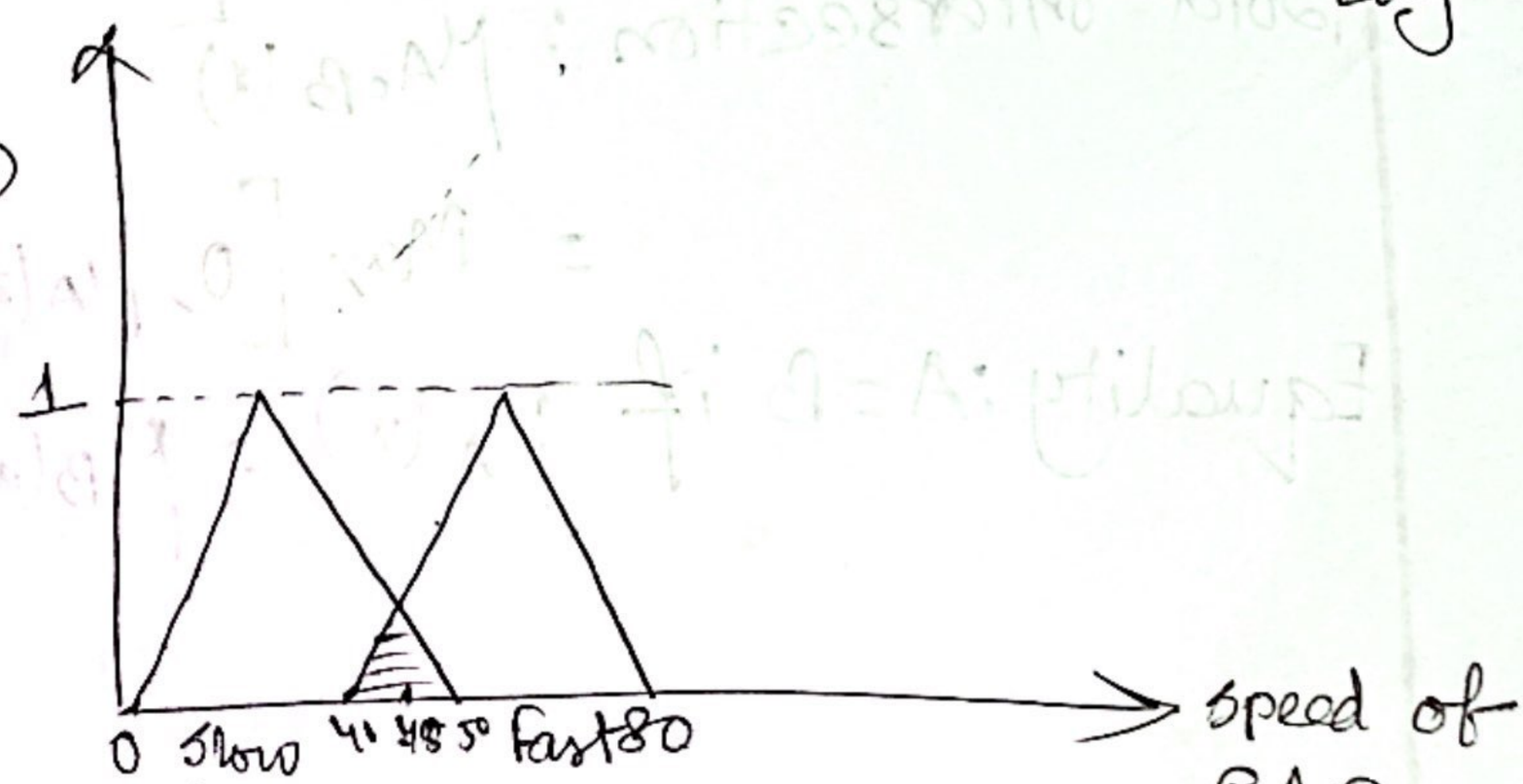
- → represent uncertainty [0, 1]
- → represent with degree
- → represent the belongingness of a member of a crisp set to fuzzy set.

Boolean Logic

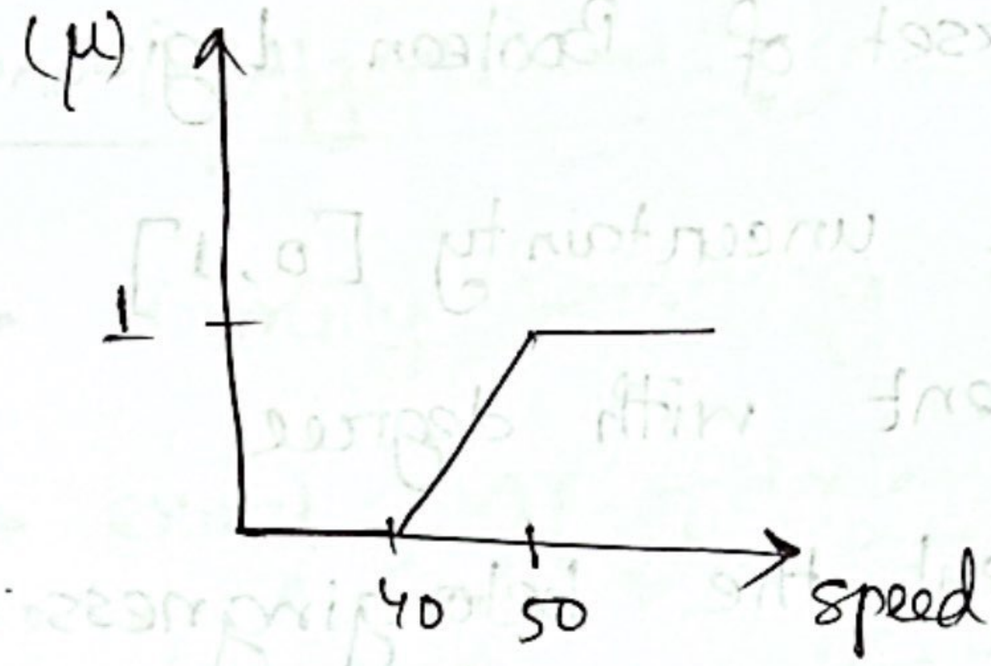


$$\begin{cases} 0; & \text{if speed} \le 40 \\ \dfrac{\text{speed}-40}{10}; & \text{if } 40 < \text{speed} < 50 \\ 1, & \text{if speed} > 50 \end{cases}$$

Fuzzy logic



Scanned with CamScanner



Operations in Fuzzy Logic :

Union : $\text{Max}\{\mu_A(x), \mu_B(x)\}, x \in U$

Intersection : $\text{Min}\{\mu_A(x), \mu_B(x)\}, x \in U$

Complement : $\mu_{\bar{A}}(x) = [1 - \mu_A(x)], x \in U$

Bold Union : $\mu_{A \oplus B} = \min[1, \mu_A(x) + \mu_B(x)]$

Bold Intersection : $\mu_{A \circ B}(x)$

$= \text{Max}[0, \mu_A(x) + \mu_B(x) - 1]$

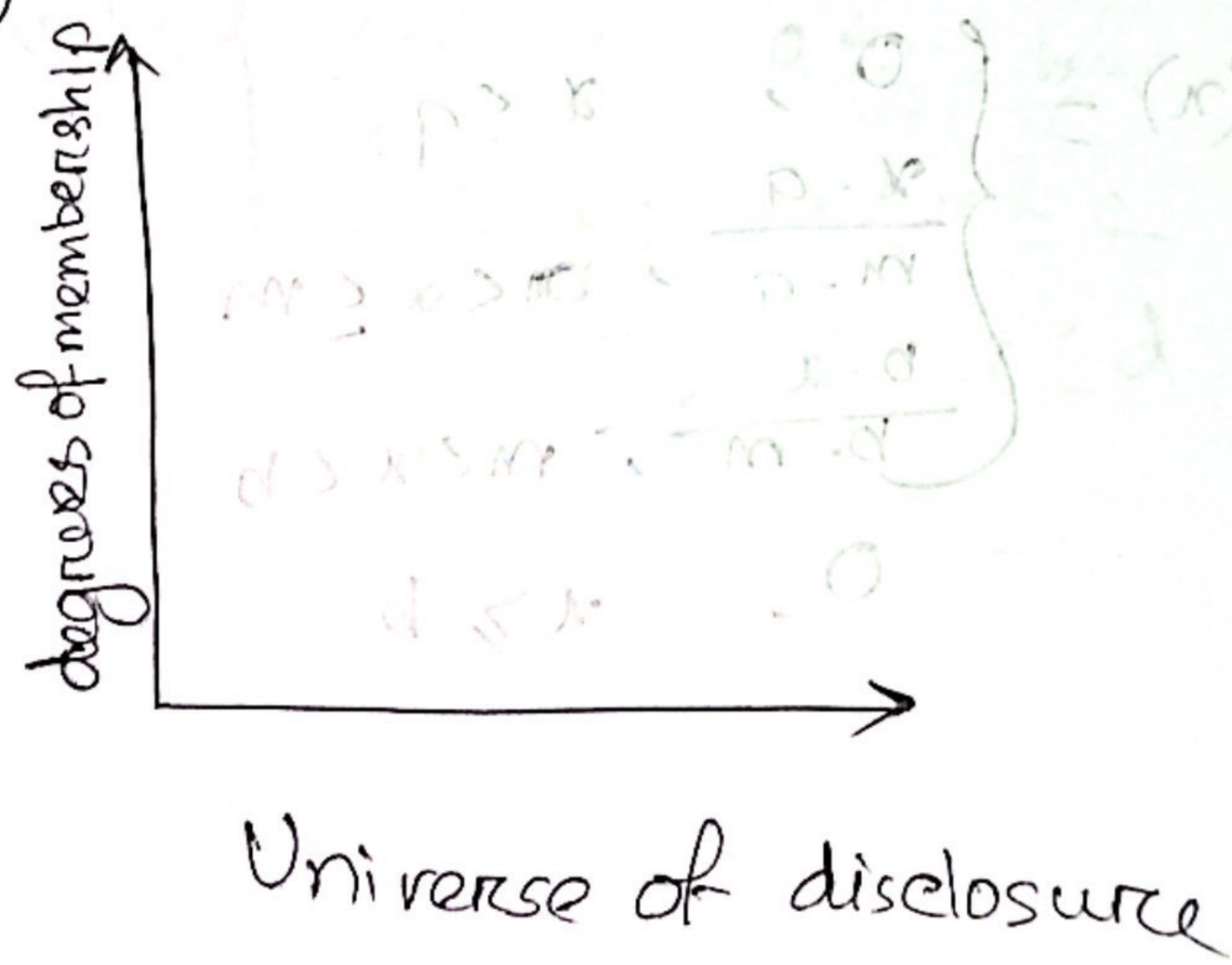
Equality : $A = B$ if $\mu_A(x) = \mu_B(x) \; \forall x \in S$

Scanned with CamScanner

Fuzzy Membership Functions:

$\mu_A : X \rightarrow [0, 1]$

A membership function for a fuzzy set A on the universe of disclosure X is defined as $\mu_A : X \rightarrow [0, 1]$, where each element of X is mapped to a value between 0 & 1. This value, called membership value or degree of membership, quantifies the grade of membership of the element in X to the fuzzy set A.



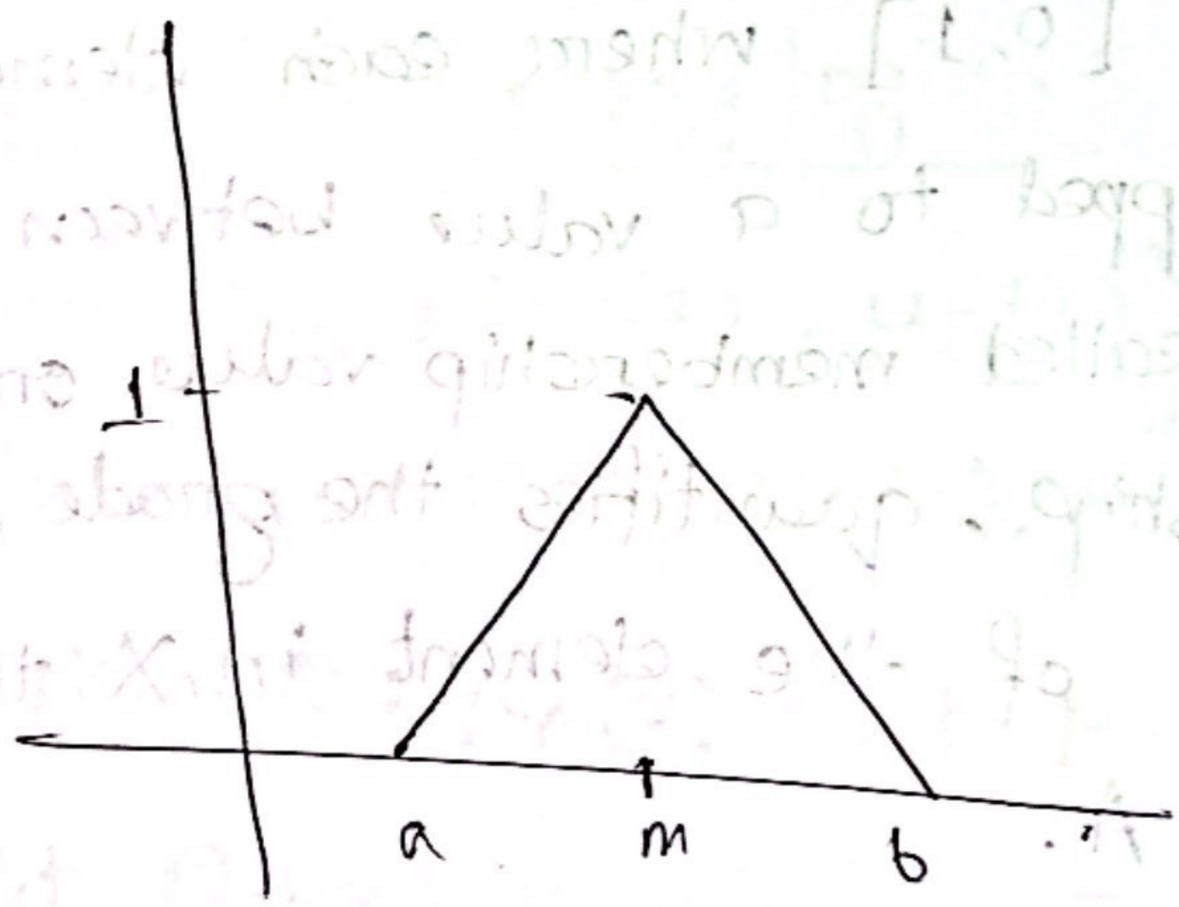
Scanned with CamScanner

Triangular Function:

$a \rightarrow$ lower limit

$b \rightarrow$ upper limit

$a < m < b$

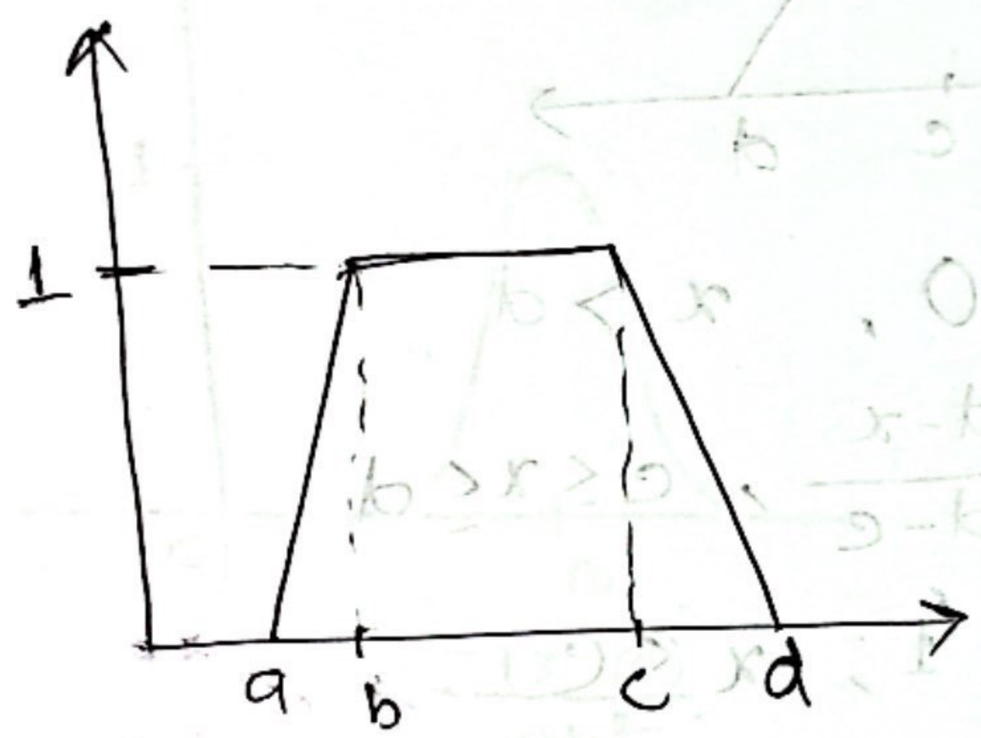


$$\mu_A(x) = \begin{cases} 0, & x \le a \\ \dfrac{x-a}{m-a}, & a < x \le m \\ \dfrac{b-x}{b-m}, & m < x < b \\ 0, & x \ge b \end{cases}$$

Trapezoidal Function:

$a < b < c < d$

- a → lower limit
- b → lower support limit
- c → upper support limit
- d → Upper limit

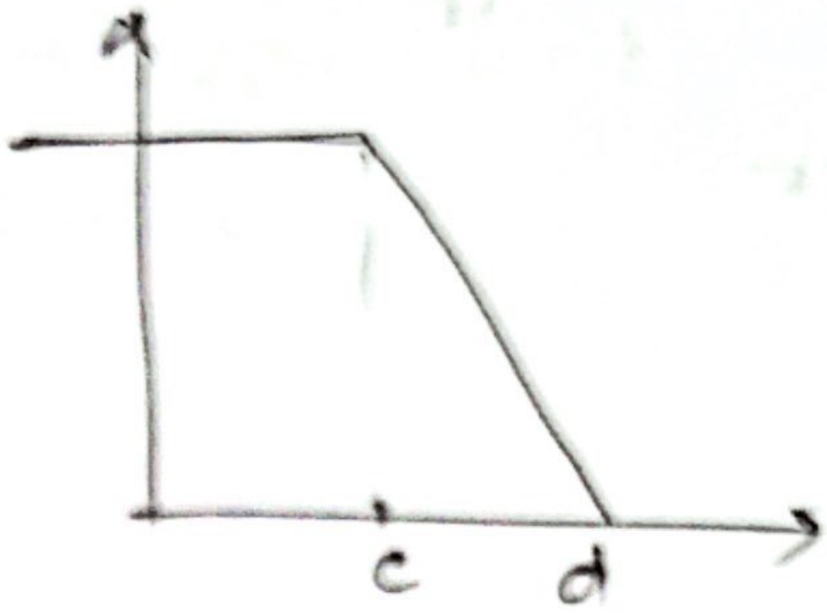


$$\mu_A(x) = \begin{cases} 0 \; ; & x < a \text{ or } x > d \\ \dfrac{x-a}{b-a} \; ; & a \le x \le b \\ 1 \; ; & b \le x \le c \\ \dfrac{d-x}{d-c} \; ; & c \le x \le d \end{cases}$$

Scanned with CamScanner

R- functions:

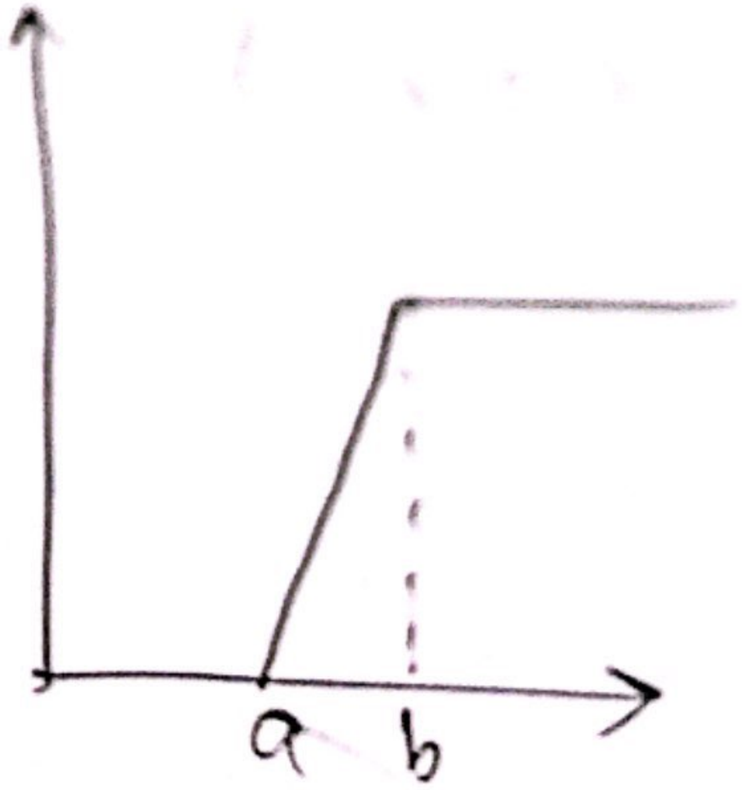
$a = b = -\infty$



$$\mu_A(x) = \begin{cases} 0, & x > d \\ \frac{d-x}{d-c}, & c \le x \le d \\ 1, & x < c \end{cases}$$

L- functions:

$c = d = +\infty$

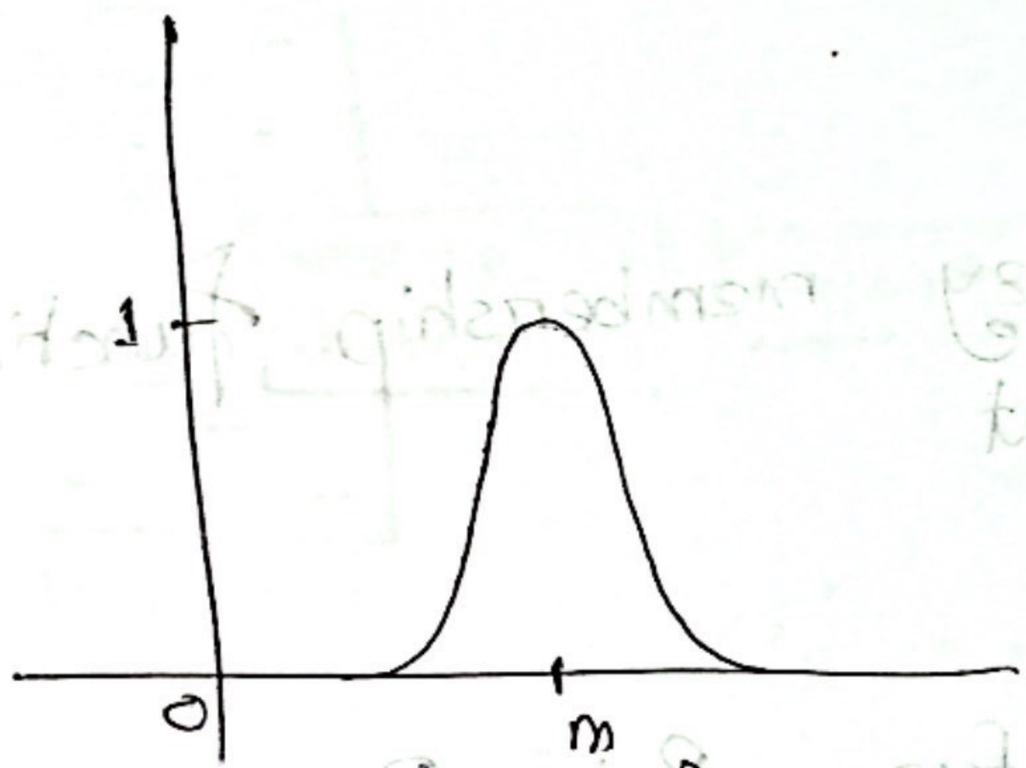


$$\mu_A(x) = \begin{cases} 0 ; & x < a \\ \frac{x-a}{b-a} & a \le x \le b \\ 1 ; & x > b \end{cases}$$

Scanned with CamScanner

Gaussian function:

→ defined by a central value m and a standard deviation $k>0$. The smaller k is, the narrower the "bell" is.



$$\mu_A(x) = e^{\frac{-(x-m)^2}{2k^2}} = e^{-k'(x-m)^2}$$

where, $k' = \frac{1}{2k^2}$

Design of FLC

STEP-01:

Define inputs and outputs for FLC

STEP-02:

Fuzzify the inputs

STEP-03:

Setup fuzzy membership functions for the output

STEP-04:

Create a fuzzy Rule Base.

STEP-05:

Defuzzify the Outputs.

Scanned with CamScanner

FLC design to Control Water Tank

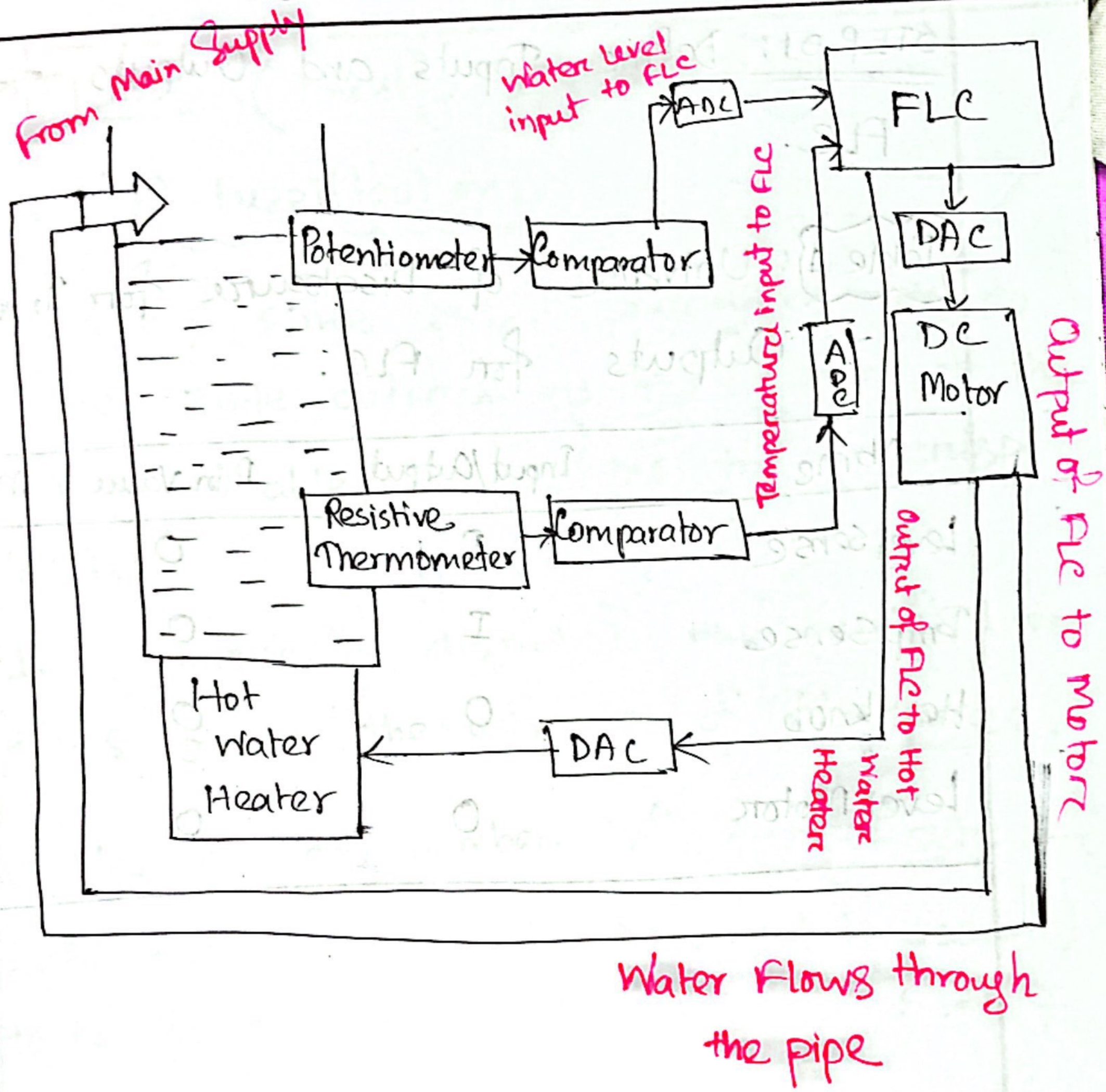


Fig: Circuit Diagram of Maintenance Constant Water Level & Temperature of an Overhead Tank

Scanned with CamScanner

STEP-01: Define Inputs and Outputs for the FLC.

Table 1: Universe of Disclosure for Inputs & Outputs for FLC.

Name	Input/Output	Min. Value	Max. Value
Level sense	I	0	10
Temp Sense	I	0	130
Heat knob	O	0	10
Level Motor	O	0	10

Scanned with CamScanner

STEP-02: (Fuzzify the Inputs)

Rules of Fuzzification:

There are some general guidelines ~~whi~~ to follow while determining the range of the fuzzy variables as related to the crisp inputs:

1. Symmetrically distribute the fuzzified ~~the~~ values across the universe of discbourse.

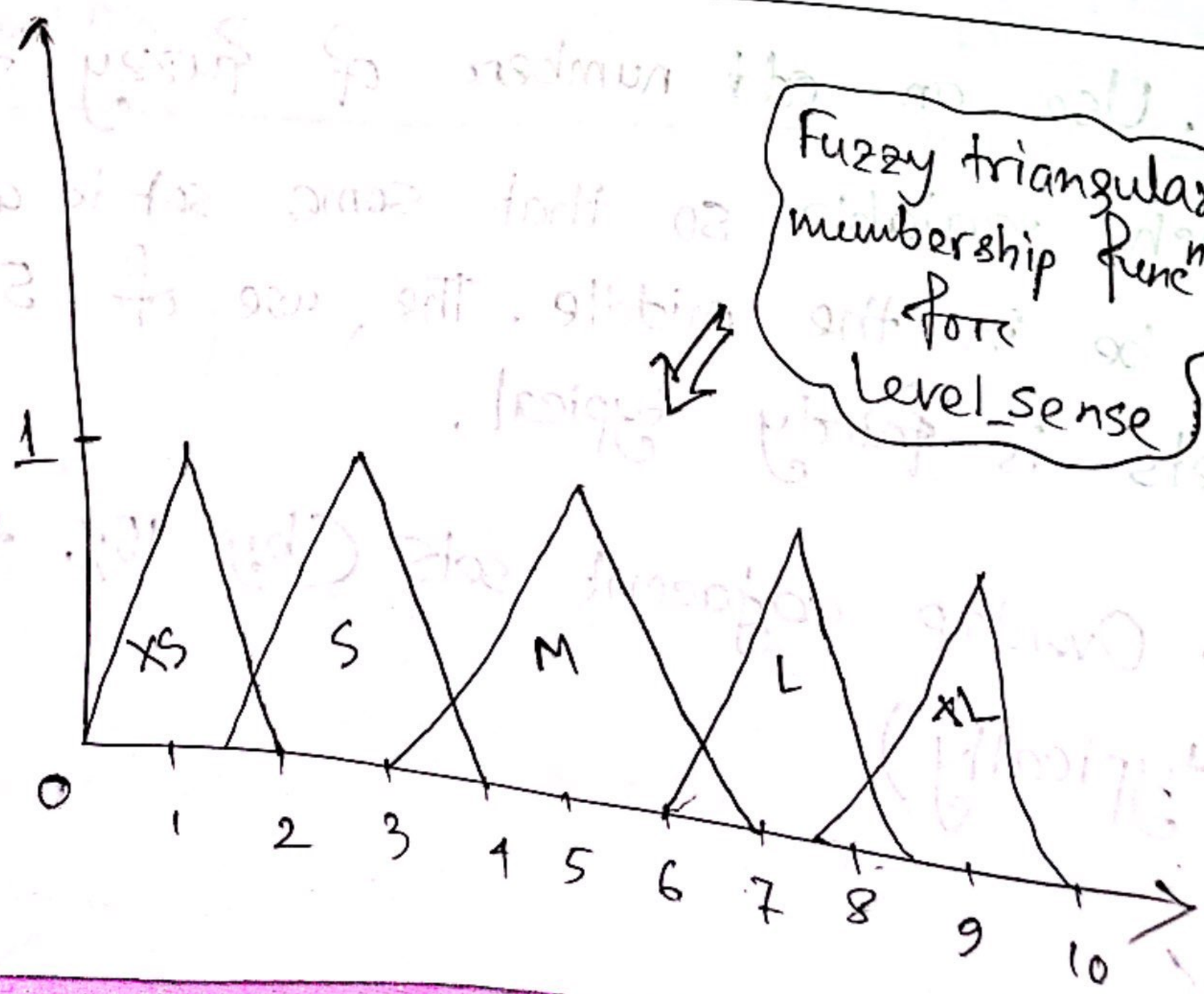
2. Use an odd number of fuzzy sets for each variable so that some set is assured to be in the middle. The use of 5 to 7 sets is fairly typical.

3. Overlap adjacent sets (by 15% to 25% typically)

Scanned with CamScanner

Table 2: Fuzzy variable ranges for Level-sense

Crisp input range	Fuzzy Variable
0-2	XSMALL
1.5 - 4	SMALL
3 - 7	MEDIUM
6 - 8.5	LARGE
7.5 - 10	XLARGE



Scanned with CamScanner

Table 03: (Fuzzy Variable range for Temp_Sense)

Crisp Input Range	Fuzzy Variable
0-20	XSMALL
10-35	SMALL
30-75	MEDIUM
60-95	LARGE
85-125	XLARGE

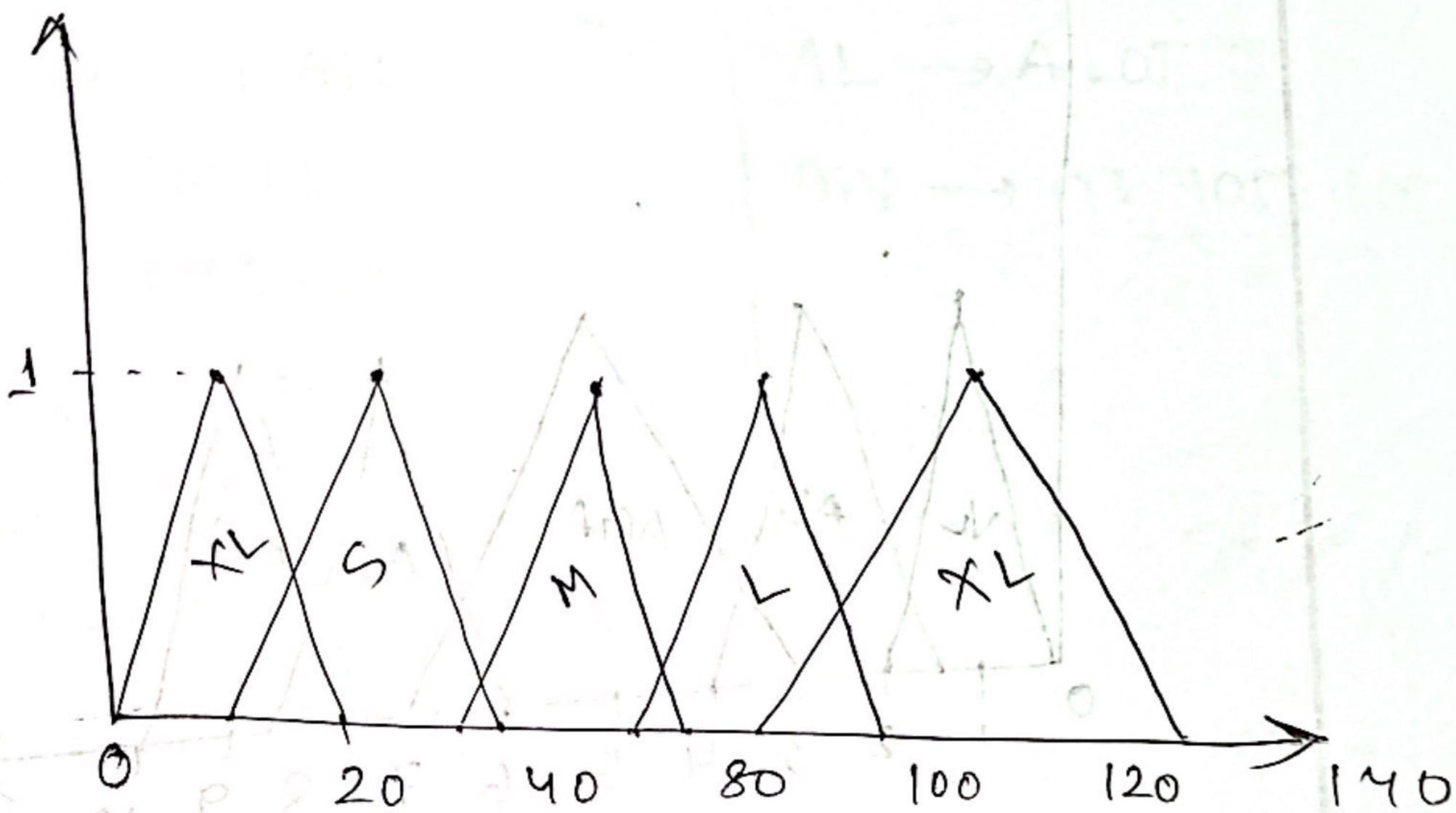
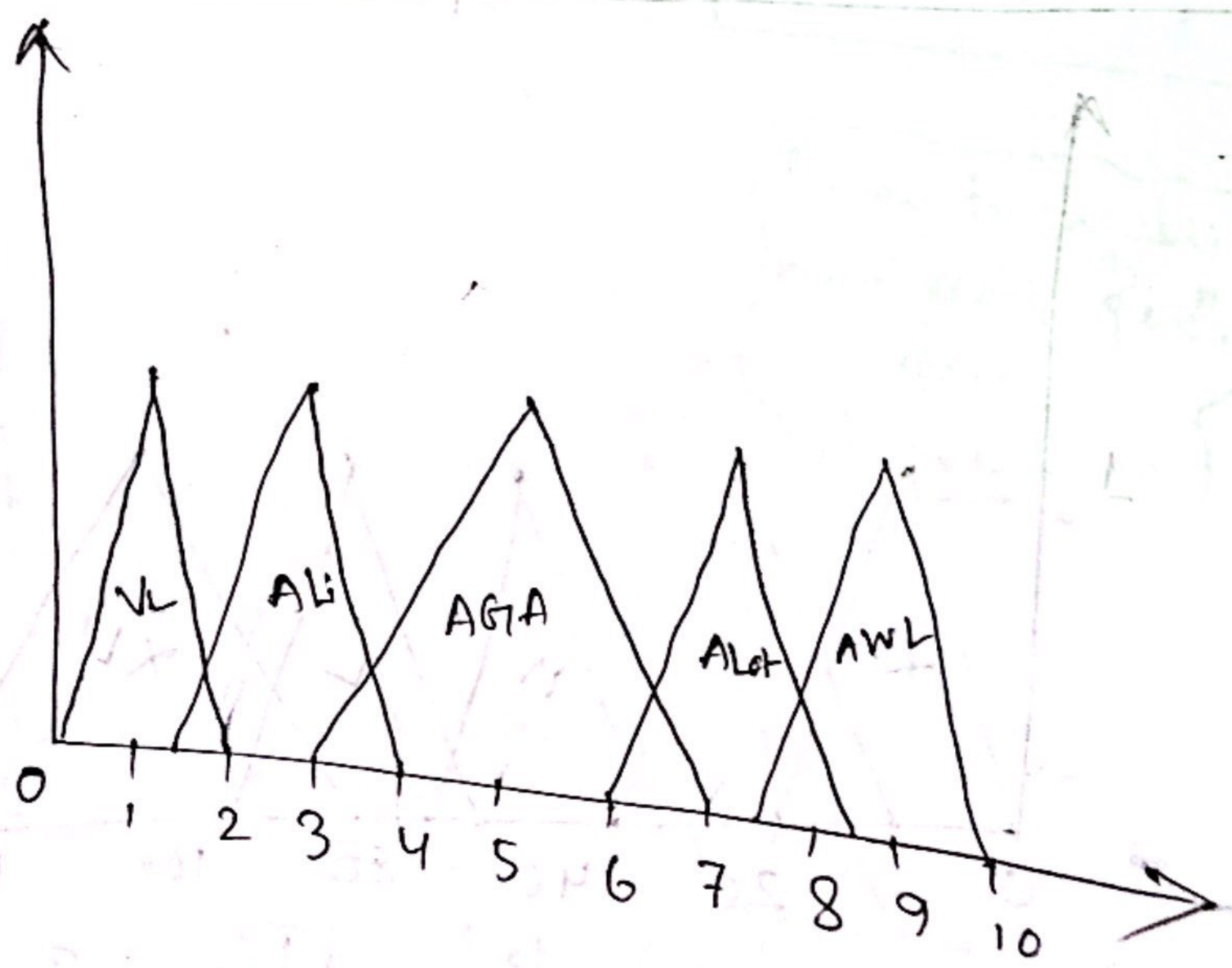


Fig: F.T.M.F for Temp_Sense

Scanned with CamScanner

Table 04: (Fuzzy variable range for Heat knob and Level motor)

Crisp Input Range	Fuzzy Variable
0-2	Very Little
1.5-4	A Little
3-7	A good amount
6-8.5	A lot
7.5-10	A Whole lot



Scanned with CamScanner

STEP-04: (Create a Fuzzy Rule Base)

Table-05: (Fuzzy Logic Base for the FLC)

		Temp Sense				
		XS	S	M	L	XL
Level Sense	XS	AG	ALL	VL	-	-
	S	AL	AG	VL	VL	-
	M	AW	AL	AG	VL	-
	L	AW	AL	AL	ALL	-
	XL	AW	AL	AL	AG	-

XS → XSMALL
S → SMALL
M → MEDIUM
L → LARGE
XL → XLARGE
AG → AGOODAMOUNT
ALL → A LITTLE
VL → VERY LITTLE

AL → A LOT
AW → A WHOLE LOT

Scanned with CamScanner

STEP-05:

Assume that at a particular point in time, LevelSense = 7·0 & TempSense = 65

Crisp input : LevelSense = 7·0

Fuzzy outputs with membership values :

Medium : 0·4

Large : 0·6

all others : 0·0

Crisp input : Tempsense = 65

Fuzzy outputs with membership values :

Medium : 0·75

Large : 0·25

all others : 0·0

Scanned with CamScanner

This results in four rules firing :

1. (Tempsense = Medium (0·75)) AND (L.S = Medium (0·4))

2. (T.S = Large (0·25)) AND (L.S = Medium (0·4))

3. (T.S = Medium (0·75)) AND (L.S = Large (0·6))

4. (T.S = Large (0·25)) AND (L.S = Large (0·6))

First ~~you~~ we must determine, for each of the AND clauses in the IF-THEN rules, what the output should be.

1. $(0{\cdot}75) \perp (0{\cdot}4) = 0{\cdot}4$

2. $(0{\cdot}25) \perp (0{\cdot}4) = 0{\cdot}25$

3. $(0{\cdot}75) \perp (0{\cdot}6) = 0{\cdot}6$

4. $(0{\cdot}25) \perp (0{\cdot}6) = 0{\cdot}25$

Scanned with CamScanner

By using the fuzzy rule base and the strengths assigned previously we find the rules recommend the following output values for Heat Kroft.

1. A Good Amount (0.4)
2. Very Little (0.25)
3. A Lot (0.6)
4. A Little (0.25)

Scanned with CamScanner

Now, we'll use defuzzification method to arrive at a single crisp value. Here, we use a disjunction or maximum operator to combine the rules. We obtain the following -

$$(0.4) \cdot (0.25) \cdot (0.6) \cdot (0.25)$$

$$\Downarrow$$

$$(0.6)$$

The crisp output value for HeatKnob would then be this membership value multiplied by the range of the output variable, or,

$$(0.6)(10-0) = 6.0$$

(Ans)

Scanned with CamScanner