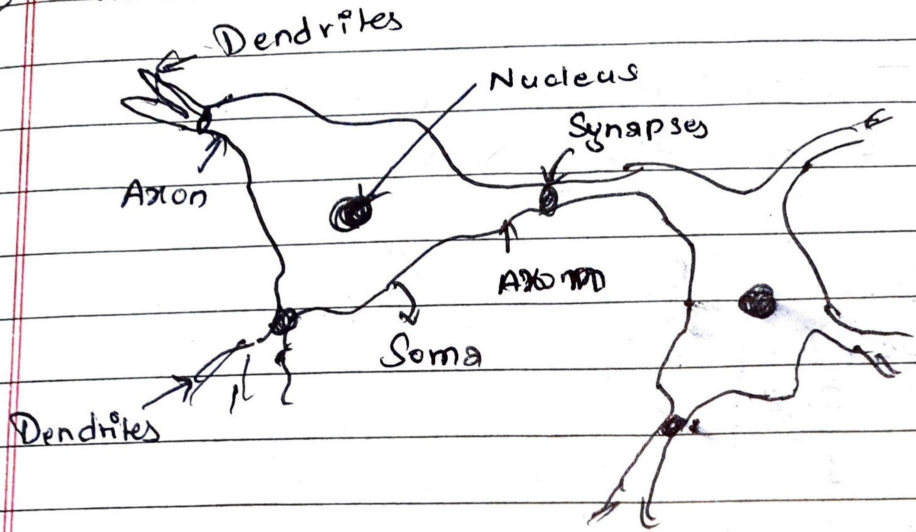
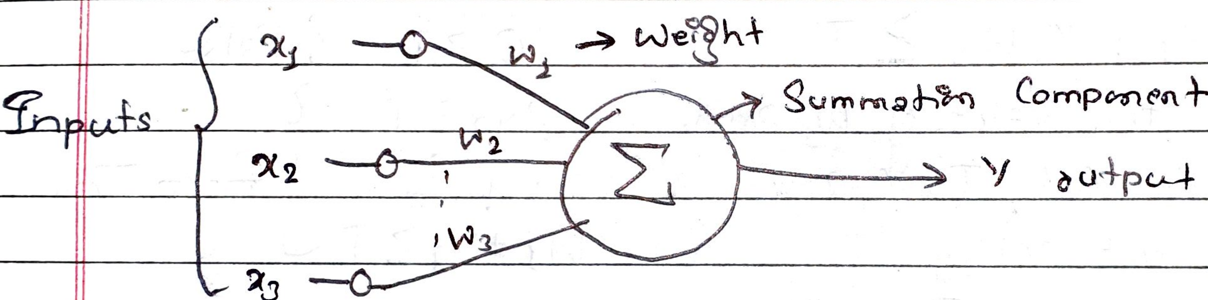
Chapter 7 : Application of AI.

(1.) Neural Network :



Biological Neural network.



Artificial Neural Network.

Parameter	BNN	ANN
Component	Dendrites	i/p
	Axon	o/p
	Soma	node
	Synapses	weight
	Nucleus	Summation Component
Processing Speed	Slow	~~Rla~~ Fast

1943 McCulloch & Pitt Neural Network

$\sum x_i w_i \geq T$, fires the neural network
else doesn't fire the N.N.

Realise OR gate

X_1	X_2	Y
0	0	0
0	1	1
1	0	1
1	1	1

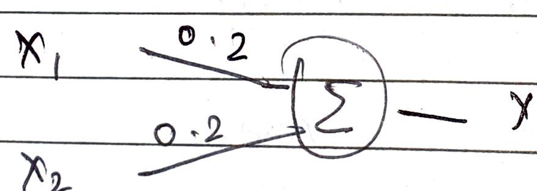
$X_1W_1 + X_2W_2 \geq T$

$0 \geq T$ } False

$W_2 \geq T$
$W_1 \geq T$
$W_1 + W_2 \geq T$ } ~~False~~ True

Choosing the value of W_1 and W_2

- $W_1 = W_2 = 0.1$, $T = 0.2$
  Doesn't satisfy
- $W_1 = W_2 = 0.2$, $T = 0.1$



AND Gate:

X_1	X_2	Y
0	0	0
0	1	0
1	0	0
1	1	1

$0 \geq T$
$W_2 \geq T$
$W_1 \geq T$ } False

$W_1 + W_2 \geq T$ } True

Choosing the value of W_1 and W_2

$W_1 = W_2 = 0.2$ $T = 0.23$

(1954) Hebb's learning

Step 1: Initialize all weight & bias as 0.

Step 2: Update the weight & bias.

$$W_{new} = W_{old} + t * X_i$$

$$b_{new} = b_{old} + t$$

Step 3: Repeat the process unless it satisfies all training input. Check the condition using

$$\sum X_i W_i + b \geq 0 \text{ to trigger.}$$

Eg:

X_1	X_2	Y
1	1	1
-1	1	-1
1	-1	-1
-1	-1	-1

$w_1 = w_2 = b = 0$

For training set (1, 1) 1

$W_1 new = 0 + 1 \times 1 = 1$

$W_2 new = 0 + 1 \times 1 = 1$

$b new = 0 + 1 = 1$

For 2nd training set (-1, 1) -1

$W_1 new = 1 + (-1)(-1) = 2$

$W_2 new = 1 + (1)(-1) = 0$

$b new = 1 + (-1) = 0$

For 3rd training set (1, -1) → t = 1

$W_1 new = 2 + (1)(-1) = 1$

$W_2 new = 0 + (-1)(-1) = 1$

$b new = 0 + (-1) = -1$

DATE :

Fourth training: $(-1,-1)$, $t=-1$

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

$W_{2new} = 1 + (-1)(-1) = 2$

$b_{new} = -1 - 1 = 2$

1 epoch

Test: $\Sigma x_i w_i + b \geq 0$ to trigger.

OR Gate:

x_1	x_2	y
1	1	1
-1	1	1
1	-1	1
-1	-1	-1

Initially, $w_1 = w_2 = b = 0$

For training set, $(1,1)$ $t=1$

$W_{1new} = 0 + 1 \times 1 = 1$

$W_{2new} = 0 + 1 \times 1 = 1$

$b_{new} = 0 + 1 = 1$

For 2nd training set, $(-1,1)$ 1

$W_{1new} = 0$

$W_{2new} = 2$

$b_{new} = 2$

3rd $(1,-1)$ 1

$W_{1new} = 1$

$W_{2new} = 1$

$b_{new} = 3$

4th $(-1,-1)$ -1

$W_{1new} = 2$

$W_{2new} = 2$

$b_{new} = \cancel{3}\ 2$

PAGE NO. :
DATE :

Perceptron

(1958) Rosenblat

Step 1: Initialize the weight & bias randomly

Step 2: Update the weight & bias using

$$W_{new} = W_{old} + \alpha * x_i * t$$

$$b_{new} = b_{old} + \alpha * t$$

α is normally taken as 0.1

Step 3: Repeat step (2) until it satisfies all the conditions of input and output data to find the network using $\sum x_i w_i + b \geq 0$.

For AND Gate:

X_1	X_2	Y
1	1	1
-1	1	-1
1	-1	-1
-1	-1	-1

Initially, $w_1 = w_2 = b = 0$.

For training set, (1,1), 1

$W_1new = 0 + 0.1 \times 1 \times 1 = 0.1$

$W_2new = 0 + 0.1 \times 1 \times 1 = 0.1$

$b_{new} = 0 + 0.1 \times 1 = 0.1$

$(-1, 1) \rightarrow (-1)$

$W_1new = 0.1 + 0.1 \times (-1)(-1)$

$= 0.2$

$W_2new = 0.1 + 0.1 \times (1) \times (-1)$

$= 0$

$b_{new} = 0.1 + 0.1 \times (-1) = 0$

3rd $(1, -1), -1$

$W_1new = 0.2 + 0.1 \times (1) \times (-1) = 0.1$

$W_2new = 0 + 0.1 \times (-1) \times (-1) = 0.1$

$b_{new} = 0 + 0.1 \times (-1) = -0.1$

4th $(-1, -1), (-1)$

$W_1new = \cancel{0.1 +} \; 0.2$

$W_2new = 0.2$

$b_{new} = -0.2$

Test:

$\sum x_i w_i + b \geq 0$

$x_1 w_1 + x_2 w_2 + b \geq 0$

$(1, 1), 1$ $\quad 1 \times 0.2 + 1 \times 0.2 - 0.2 \geq 0$

$0.2 \geq 0$ [True]

$(-1, 1)$ $\quad -1 \times 0.2 + 1 \times 0.2 - 0.2 \geq 0$

$-0.2 \geq 0$ [False]

$(1, -1)$ $\quad 1 \times 0.2 + (-1) \times 0.2 - 0.2 \geq 0$

$-0.2 \geq 0$ [False]

$(-1, -1)$ $\quad -1 \times 0.2 - 1 \times 0.2 - 0.2 \geq 0$

$-0.6 \geq 0$ [False]

Adaline Neural Network:

1. Assign all weight & bias randomly.
2. Calculate neural i/p as

$$Y_{in} = \sum x_i w_i + b$$

3. Update weight & bias using delta rule,

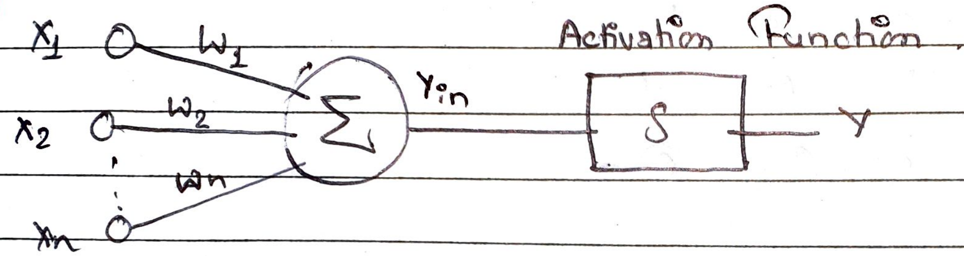
$$W_{new} = W_{old} + \alpha (t - Y_{in}) x_i$$

$$b_{new} = b_{old} + \alpha (t - Y_{in})$$

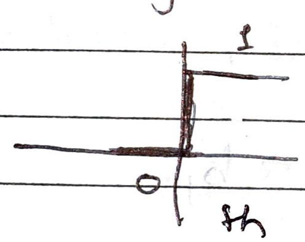
DATE :

(4.) Continue step (2) and (3) until all the condition or training sets are satisfied. Check to ~~fix~~ the neural network using condition $\sum x_i w_i + b \geq 0$.

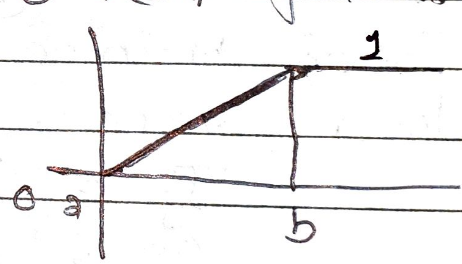
X_1	X_2	Y
1	1	1
-1	1	1
1	-1	1
-1	-1	-1



Type (1) Step function.



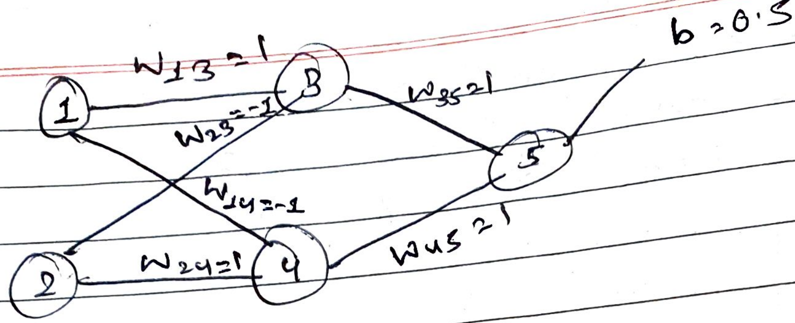
Type (2) Ramp function



Type (3) Sigmoid function.

$$\frac{1}{1+e^{-x}}$$

DATE :



When $\sum x_i w_i \leq 0 \rightarrow 0$

$\sum x_i w_i > 0 \rightarrow 1$

x_1	x_2	3	4	5
0	0	0	0	0
0	1	0	1	1
1	0	1	0	1
1	1	0	0	0

V V Imp **Back Propagation**

1. Initialise weight & bias randomly.
2. Taking training set $I_i = O_i$, calculate the hidden layer values,

$$I_j = \sum W_{ij} * O_i + b_j$$

$$O_j = \frac{1}{1+e^{-I_j}}$$

3. For output layer O_K,

$$I_K = \sum (W_{jk} * O_j) + b_K$$

$$O_K = \frac{1}{1+e^{-I_K}}$$

4. Calculate error, $E_K = O_j(1-O_j)(T_j - O_j)$ &

$$E_j = O_j(1-O_j)\sum_K E_K * W_{jk}.$$

PAGE NO. :

DATE :

(5) Weight updation using following formula.

$\Delta W_{ij} = \alpha \times E_j * O_i$

$W_{ij}new = W_{ij}old + \Delta W_{ij}$

~~Δb_j~~ = $\Delta b_j = \alpha E_j$

$b_j new = b_j old + \Delta b_j$

$\Delta W_{jk} = \alpha E_k * O_j$

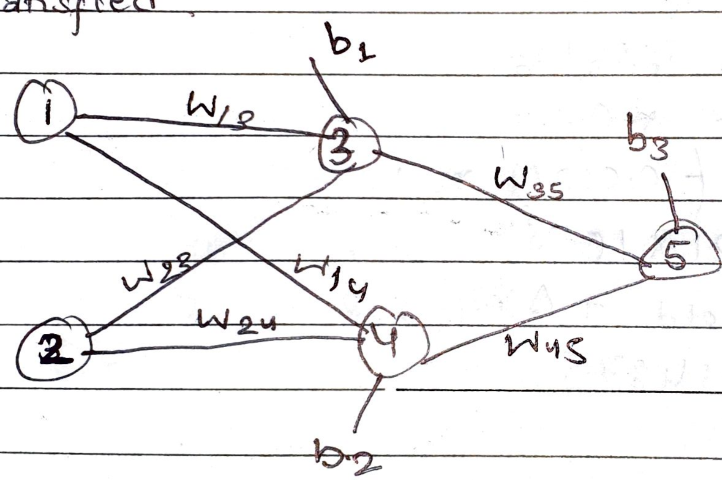
$W_{jk} = W_{jk} + \Delta W_{jk}$

$\Delta b_k = \alpha E_k$

$b_k = b_k + \Delta b_j$

(6) Continue step 2 to 5 until all the training sets are satisfied.

Eg.



$W_{35} = 0.35$	$b_1 = 0.1$	$W_{14} = 0.25$
$W_{45} = 0.4$	$W_{13} = 0.15$	$W_{24} = 0.3$
$b_3 = 0.45$	$W_{23} = 0.2$	$b_2 = 0.35$

$I_1 = 1,\ I_2 = 1,\ T_5 = 0$

$I_3 = 0.15 \times 1 + 0.2 \times 1 + 0.1$
$= 0.45$

$O_3 = \dfrac{1}{1+e^{-0.45}} = 0.61$

$I_4 = 0.25 \times 1 + 0.3 \times 1 + 0.35 = 0.9$

$O_4 = 0.71$

$I_5 = 0.71 \times 0.4 + 0.61 \times 0.35 + 0.45 = 0.9475$

$O_5 = \dfrac{1}{1+e^{-0.9475}} = 0.72$

(4) $E_5 = O_5(1-O_5)(0-O_5)$

$= 0.72(1-0.72)(0-0.72)$

$= -0.1451$

$E_3 = 0.61 \times (1-0.61) * (-0.1451 * 0.35)$

$= -0.0121$

$E_4 = 0.71 \times (1-0.71) *$

$= -0.0119$

Step 5: Weight updation:

$\Delta W_{13} = \alpha \times E_j * O_i$

$= 0.1 \times (-0.0121) * 1$

$= -1.21 \times 10^{-3}$

$W_{13new} = W_{13old} + \Delta W_{13}$

$= 0.14879$

$\Delta b_j = \alpha E_j$

$= 0.1 \times -0.0121$

$= -0.00121$

$b_{1new} = 0.1 - 0.00121 = 0.09879$

$\Delta W_{23} = \alpha \times E_j \times O_i$

$= 0.1 \times (-0.0121) \times 1$

$= -0.00121$

$W_{23new} = W_{13old} + \Delta W_{23}$

$= 0.2 - 0.00121$

$= 0.19879$

DATE :

$\Delta W_{24} = \alpha \times E_j \times O_i$

$= 0.1 \times (-0.0119) \times 1$

$= -1.19 \times 10^{-3}$

$W_{24new} = W_{24} + \Delta W_{24}$

$= 0.3 - 1.19 \times 10^{-3}$

$= 0.29881$

$\Delta W_{14} = \alpha \times E_4 \times O_1$

$= 0.1 \times (-0.0119) \times 1$

$= -1.19 \times 10^{-3}$

$W_{14new} = W_{14} + \Delta W_{14}$

$= 0.25 - 1.19 \times 10^{-3}$

$= 0.24881$

$\Delta W_{35} = \alpha \times E_5 \times O_3$

$= 0.1 \times (-0.1451) \times 0.61$

$= -8.85 \times 10^{-3}$

$W_{35new} = W_{35} + \Delta W_{35}$

$= 0.35 - 8.85 \times 10^{-3}$

$= 0.3411$

$\Delta W_{45} = \alpha \times E_5 \times O_4$

$= 0.1 \times -0.1451 \times 0.71$

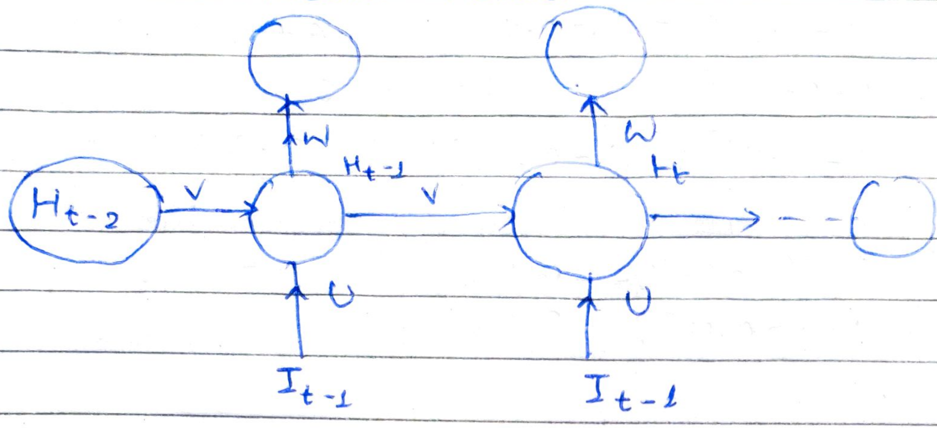
$= -0.01035$

$W_{45new} = W_{45} + \Delta W_{45}$

$= 0.4 - 0.01035$

$= 0.3869.$

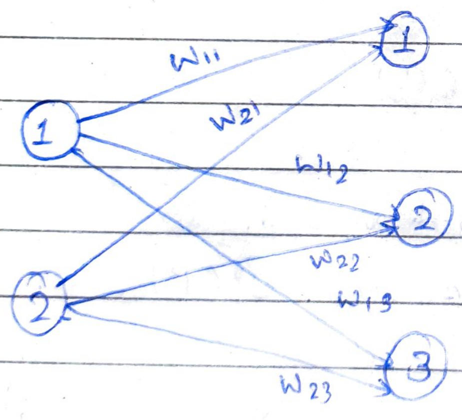
Recurrent Neural Network.



$O_i = H_{t-1} * V + I_{t-1} * U$

$O = \text{Activation}(O_i)$

Kohonen Neural Network



$W_{11} = 0.2$

$W_{21} = 0.25$

$W_{12} = 0.3$

$W_{22} = 0.35$

$W_{13} = 0.4$

$W_{23} = 0.45$

Calculate d using target vector & weight using formula

$$d = \sqrt{(X_1 - W_{ij})^2 + (X_2 - W_{ij})^2}$$

Then update the n/w with the least value as

$$W_{ij}new = W_{ij}old + \Delta W_{ij}$$

where $\Delta W_{ij} = \alpha(X_1 - W_{ij})$

Here

$d_1 = \sqrt{(1-0.8)^2 + (1-0.35)^2} = \cancel{0.96}\ 1.096$

$d_2 = \sqrt{(1-0.8)^2 + (1-0.35)^2} = 0.955$

$d_3 = \sqrt{(1-0.4)^2 + (1-0.45)^2} = 0.814$

$\Delta W_{13} = 0.1(1-0.4) = 0.06$

$\Delta W_{23} = 0.1(1-0.45) = 0.055$

$W_{13_{new}} = 0.4 + 0.06 = 0.46$

$W_{23_{new}} = 0.45 + 0.055 = 0.505$

Hopfield N.N

For string a set of bipolar i/p patterns, weights are stored in matrix as,

$$W_{ij} = \sum_{p=1}^{P} S_i(P)\ S_j(P)$$

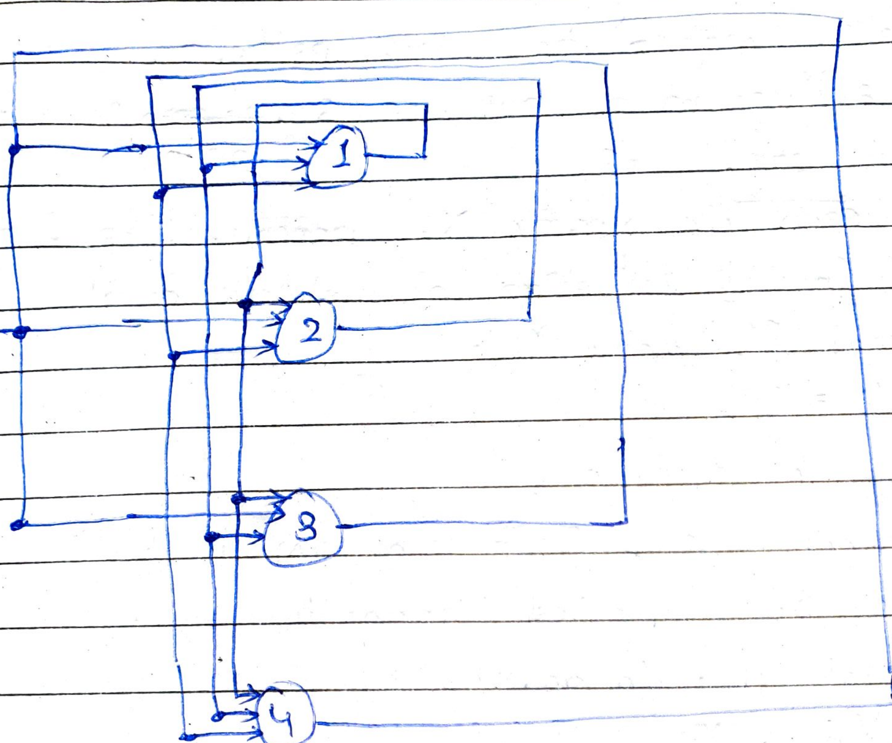
where, $S_j(P)$ is given input vector &
$S_i(P)$ is transverse of $S_j(P)$.

For no self connection we make diagonal element 0.

This is proposed by John Hopfield.

→ can be used to store the corresponding weight of each link.

→ Each neuron is connected with every other neuron.



$$W_{ij} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ -1 \end{bmatrix}_{4\times 1} * \begin{bmatrix} 1 & 1 & 1 & -1 \end{bmatrix}_{1\times 4} \quad S_j(P) = \begin{bmatrix} 1 & 1 & 1 & -1 \end{bmatrix}$$

$$W_{ij} = \begin{bmatrix} 1 & 1 & 1 & -1 \\ 1 & 1 & 1 & -1 \\ 1 & 1 & 1 & -1 \\ -1 & -1 & -1 & 1 \end{bmatrix}_{4\times 4}$$

Hopfield N.N

Step 0 : Initialize weight to store pattern.

Step 1 : If activation of network are ~~a~~ not converged perform step 2-8.

Step 2 : For each i/p vector X_i perform step 3-7.

Step 3 : Make initial activation of network equal to external i/p vector X_i.

$$y_i = X_i$$

where $i = 1$ to n.

Step 4 : Perform step 5-7 for each unit Y_i units are updated randomly.

Step 5 : Calculate net i/p of the n/w

$$y_{in} = X_i + \sum y_j W_i$$

Step 6 : Apply activation over net i/p to calculate o/p as,

$$y_i = \begin{cases} 1 & \text{if } y_{in} > \theta_i \\ y_i & \text{if } y_{in} = \theta_i \\ 0 & \text{if } y_{in} < \theta_i \end{cases}$$

θ_i is taken 0 basically.

Step 7 : Now feedback or transmit the obtained o/p Y_i to all other units. Thus activation vectors are updated.

Step 8 : Finally test the n/w for convergence.

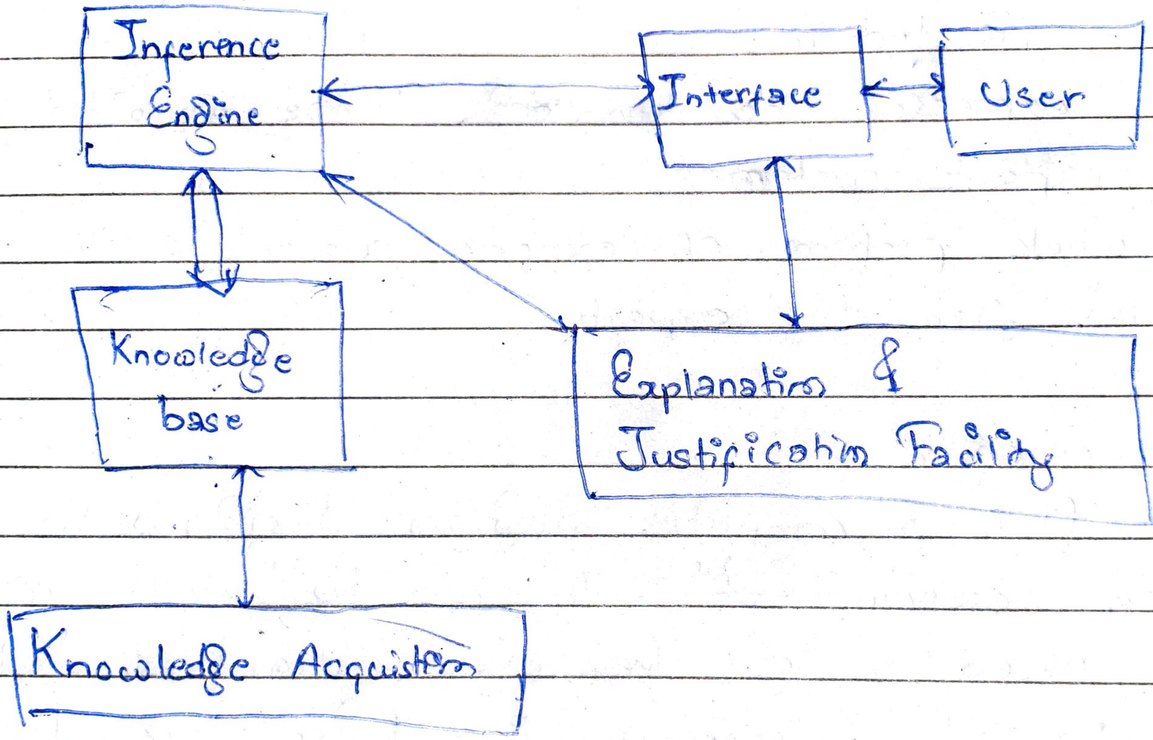
~~Input~~ Target : [1 1 1 -1] , i/ps [-1 -1 1 -1]

Step D:

$$\begin{bmatrix} 1 & 1 & 1 & -1 \\ 1 & 1 & 1 & -1 \\ 1 & 1 & 1 & -1 \\ 1 & 1 & 1 & -1 \end{bmatrix}$$

Ques

Expert System :



Architecture of ES

An expert system is a computer system that emulates the decision making ability of human expert. They are designed to solve complex problem by reasoning about knowledge like an expert. An expert system is mainly divided into two part. One is fixed i.e. inference engine and the other is variable i.e. knowledge based.

(I) Knowledge base:

It is expressed with if ---- then ---- rules in natural language

Example: If temp is above 100°F then the patient is having fever.

If the fever is occuring at evening then it is typhoid.

Rules expressed the knowledge to be exploited by the expert system.

The whole problem of expert system is to collect knowledge from the expert.

(II) Inference engine

It is a computer program designed to ~~program~~ produce reasoning on rules ~~facts~~. In order to produce reasoning it is based on logic. There are several types of logic like propositional logic, predicate logic, fuzzy logic, etc. With logic, the engine is able to generate new information from the knowledge contained in the rule base.

Date
Page

(III) User Interface:

The user interacts with a system through a User Interface which may use Natural language or GUI. They may even ask cross-question and ask for validity.

(IV) Explanation & Justification Facility:

This system allows the program to explain the reasoning to the user.

7.3 NLP

Natural Language Processing — NLU (Understanding) / NLG (Generation)

NLP refers to AI method of communicating with an intelligent system using a natural language such as English. NLP is the area of computer science & AI concerned with interaction between computer & human language. In particular how program computer to process & analyze large amount of natural language data. The i/p & o/p of NLP can be speech or written text. It has mainly two components NLU & NLG.

NLU refers to speech/text to ~~mean~~ understanding meaning while NLG refers to conversion from meaning to text/speech. NLU involves the following task.

i. Mapping the given i/p in natural language into useful representation

ii. Analyzing different aspects of the language.

Problems in NLU:

1. Lexical Ambiguity

It is at very primitive level known as word level. A word having different meaning can cause ambiguity for the machine. Example Treating the word like board as noun or verb.

PAGE NO. :
DATE :

(ii) Syntax level Ambiguity:
In ~~this~~ ~~level~~ syntax level ambiguity, it involves the analysis of words in the sentence for grammar & arranging words in a manner that shows the relationship among the words.
Eg: The man saw a person with telescope.

(iii) Referential ambiguity
The ambiguity related with pronouns are classified in referrential ambiguity.
E.g: Ram went to Hari & he said "I want pen".

NLG:
It is the process of producing meaningful phrases & sentences in the form of natural language from some internal representation. It involves

i. Text planning:
It includes retriving the relevant content from the knowledge base

ii. Sentence planning
It includes choosing required words, forming many full phrases, setting tone of the sentence.

iii. Text Realization
It is mapping sentence plan into sentence structure.

NLU is harder than NLG

PAGE NO. :
DATE :

Lexical or Morphological Analysis

↓

Context free Grammar — Syntatic Analysis — Top down Parser

↓

Semantic Analysis

↓

Disclosure Integration

↓

Pragmatic Analysis

① Lexical Analysis:

It involves identifying & analyzing the structure of words. Lexicon of a language means the collection of words & phrases in a language. lexical analysis is dividing the whole chunk of text into paragraphs, sentences & words.

① Syntatic Analysis: [Write Short notes]

It involves analysis of words in the sentences for grammar & arranging words in a manner that shows the relationship among the words. The sentence such as " The school goes to boy " is rejected by English syntatic analyzer.

PAGE NO. :
DATE :

(iii) Semantic Analysis:

It does the exact meaning or dictionary meaning from the text. The text is checked for meaningfullness. It is done by mapping syntatic structure in the task domains.

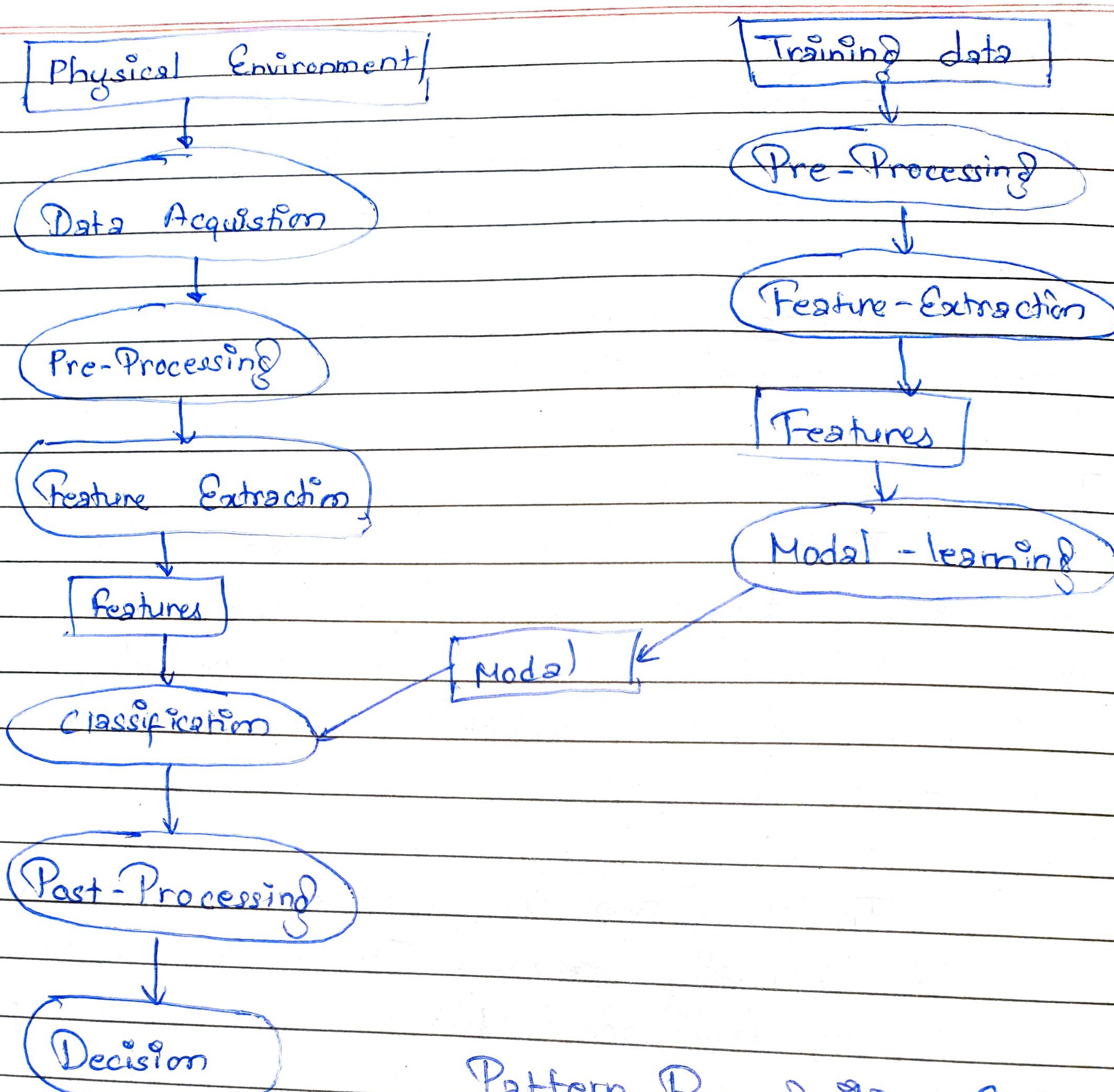
Semantic analyses disregards sentences such as "Hot ice-cream", "Colorless green".

(IV) Disclosure Integration.

The meaning of any sentence depends upon the meaning of the sentence just before. In addition, it also brings about the meaning of immediately succeding sentence. E.g: "Ram went to Hari & he said "Our board is not good".

(V) Pragmatic Analysis:

In this analysis what was said is reinterpated what it actually means. It involves deriving those aspect of language which require real world knowledge.



Pattern Recognition System.