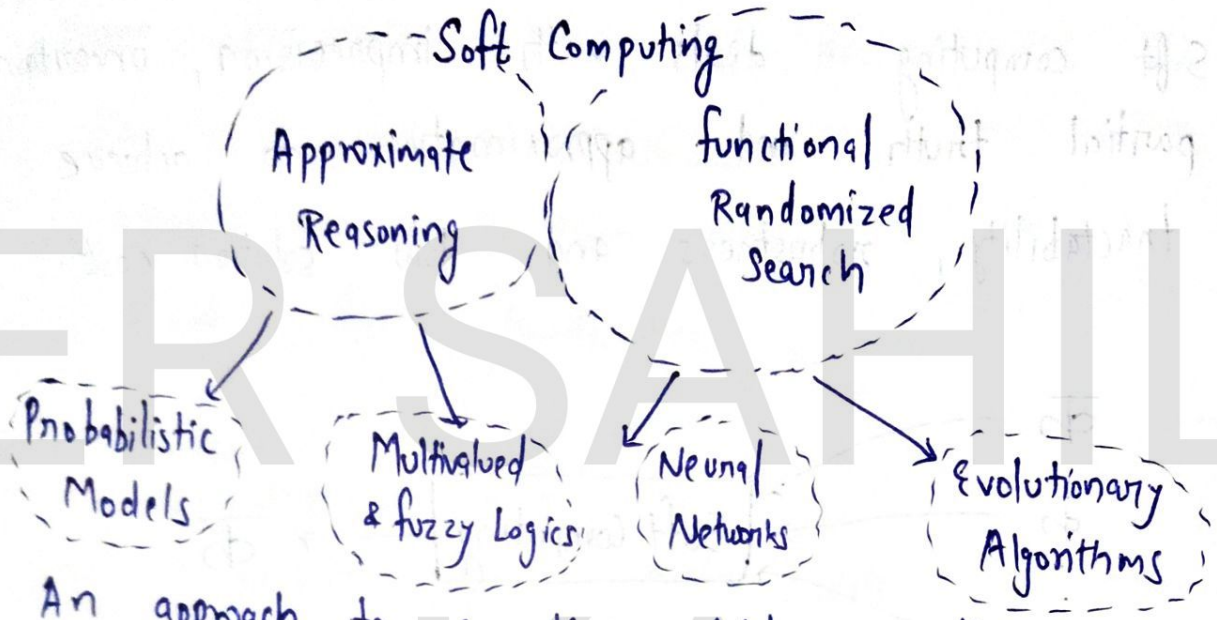
Soft Computing

Soft computing is a collection of artificial intelligence based computational techniques.



→ An approach to computing which parallels the remarkable ability of human mind to reason and learn in an environment of uncertainty & imprecision.

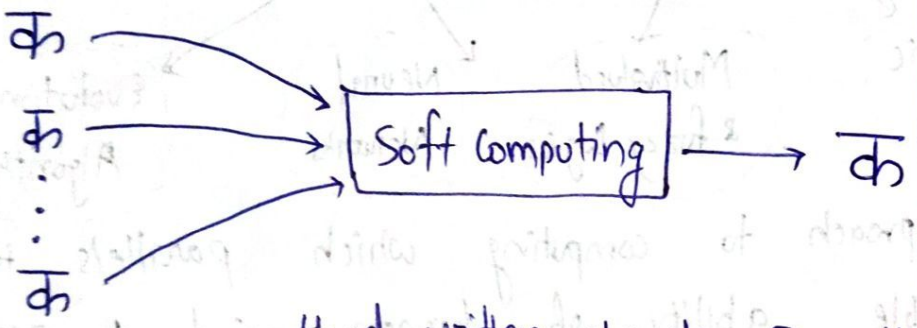
SC = Fuzzy Logic + Neural Network + Genetic Algorithm

- Fuzzy Logic → For knowledge representation
- Neural Network → For learning & Adaptation
- Genetic Algorithm → For evaluation of computation evolutionary

Aims of Soft Computing :-

- It provides cost-effective solutions to the complex real-life problems for which hard computing solution does not exist.
- A. Zadeh coined the term of soft computing in 1992. The objective of soft computing is to provide precise approximation & quick solutions for complex real-life problems.

- More complex systems arising in biology, medicine and management systems remain intractable to conventional mathematical and analytical methods.
- Soft computing deals with imprecision, uncertainty, partial truth and approximation to achieve tractability, robustness and low solution cost.



Hand written character Recognition
(Artifical Neural Network)

Characteristics :-

- It provides an approximate but precise solution
- Algorithms are adaptive.
- The concept of soft computing is based on "learning from experimental data".
- It is based on fuzzy logic, genetic algorithms, machine learning, ANN and expert systems.

Example - s1 = "xyz" , s2 = "xyw"

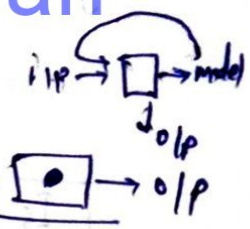
Q.1 Are s1 & s2 same? 0.67 ✓

Q.2 How much s1 & s2 are same?

Applications of soft computing:-

- Used in gaming products like Poker & Checker.
- In kitchen appliances - Microwave & Rice cooker.
- Home appliances - WM, Heater, AC---
- Robotics work
- Image processing & Data compression
- Used for Handwritten recognition

Soft Computing vs Hard Computing:-



Soft Computing

imprecise Models

AR FRS

→ Takes less computation time
→ It depends on approximation & dispositional
→ Parallel computation
→ Approximate Result
→ Neural Networks such as Madaine, Adaine, Ant Networks

Hard Computing

precise Models

Symbolic logic Traditional Numerical Modelling & search

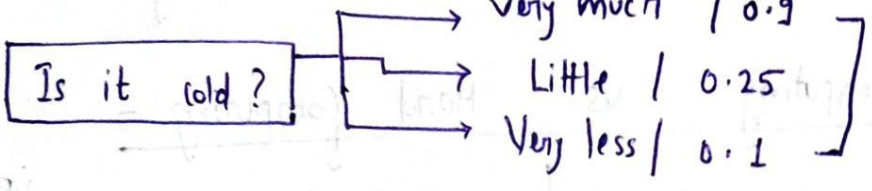
→ Takes more
→ Mainly based on binary logic & numerical systems.
→ Sequential Compution
→ Exact and precise Result
→ Any numerical problem on traditional methods of solving using personal computers.

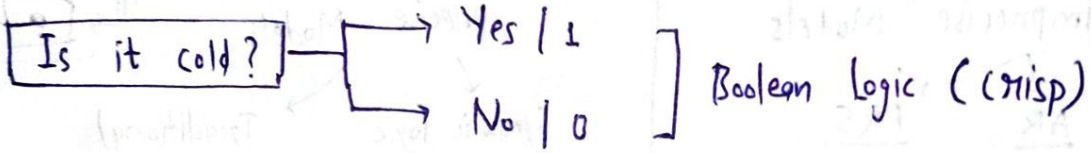
Fuzzy Logic :- Fuzzy logic theory proposed in 1965 by A. Zadeh is a generalization of classical set theory.

In classical set theory, an element either belong to or does not belong to the set & hence, such set are termed as crisp set. [0 & 1] Yes or No

But in fuzzy set, many degrees of membership (blw 0 & 1) are allowed.

Eg-





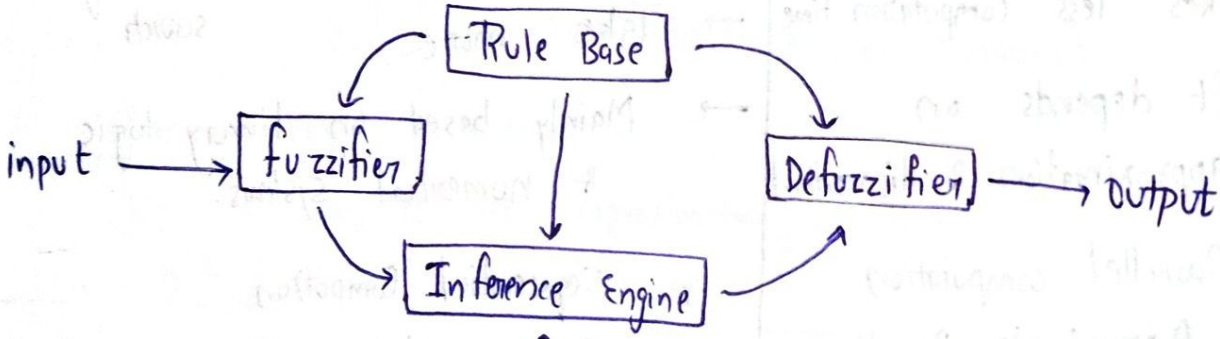
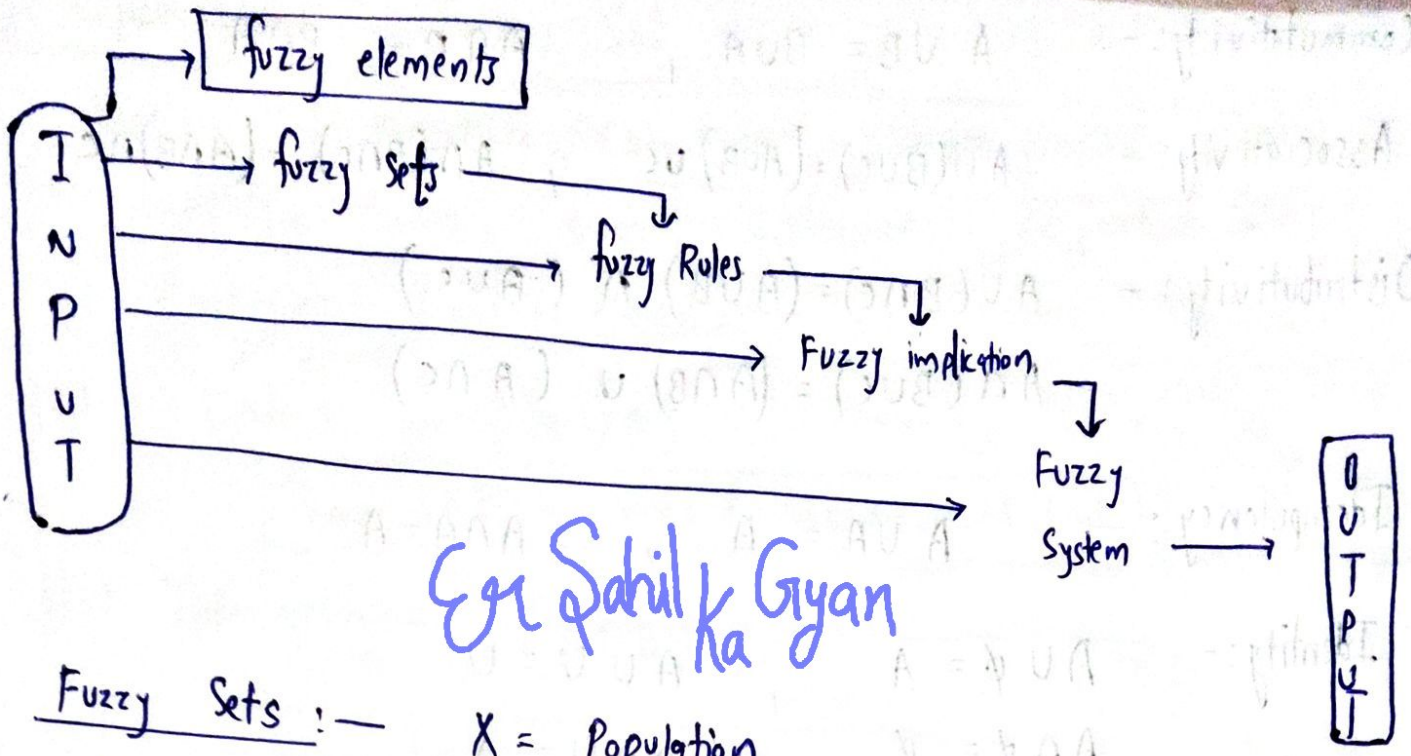


fig:- Fuzzy Logic Architecture

- The word "fuzzy" means vagueness. Fuzziness occurs when the boundary of a piece of information is not clear-out.
- Fuzzy sets theory described with the aid of a membership function valued in the real unit interval [0, 1].

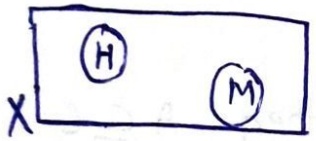


Er Sahil Ka Gyan

Fuzzy Sets :— X = Population

Eg- H = All male = { $h_1, h_2 \dots h_n$ }

M = All female = { $m_1, m_2 \dots m_n$ }



Here all are the sets of finite numbers of individuals such a set is called Crisp set.

Eg- X = all students in College

S = All good students

$S = \{ s, g(s) \mid s \in X \}$ $g(s)$ is measurement of goodness of students.

Eg- $S = \{ (R, 0.8), (K, 0.7) \}$, etc.

So $F = (s, \mu(s) \mid s \in X)$ & $\mu(s)$ is degree of s.

→ Collection of ordered pairs.

Properties of Fuzzy Sets :—

$A \cup \bar{A} \neq U$

$A \cap \bar{A} \neq \emptyset$

(i) Commutativity:- $A \cup B = B \cup A$, $A \cap B = B \cap A$

(ii) Associativity:- $A \cup (B \cup C) = (A \cup B) \cup C$, $A \cap (B \cap C) = (A \cap B) \cap C$

(iii) Distributivity:- $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$

$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$

(iv) Idempotency:- $A \cup A = A$, $A \cap A = A$

(v) Identity:- $A \cup \phi = A$, $A \cup U = U$

$A \cap \phi = \phi$, $A \cap U = A$

(vi) Involution:- $\overline{\overline{A}} = A$

(vii) Transitivity:- if $A \subseteq B \subseteq C$ then $A \subseteq C$

(viii) De Morgan's Law:-

$$\overline{A \cup B} = \overline{A} \cap \overline{B}$$

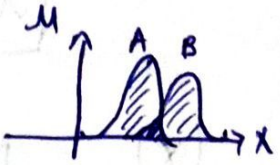
$$\overline{A \cap B} = \overline{A} \cup \overline{B}$$

★ Operations on Fuzzy Set :-

Union, Intersection, Complement, Vector product, Scalar product, Equality, Power, sum, difference, disjuctive Sum, Cartesian Product ($\times$ & +)

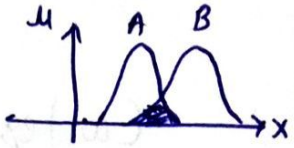
(i) Union :- $(A \cup B)$.

$$\mu_{(A \cup B)}(x) = \max\left(\mu_A(x), \mu_B(x)\right)$$



(ii) Intersection :- $(A \cap B)$

$$\mu_{(A \cap B)}(x) = \min\left(\mu_A(x), \mu_B(x)\right)$$



Eg- $A = \{ (x_1, 0.9) \quad (x_2, 0.6) \}$

$B = \{ (x_1, 0.2) \quad (x_2, 0.7) \}$

$(A \cup B) = \{ (x_1, 0.9) \quad (x_2, 0.7) \}$ ✓ max

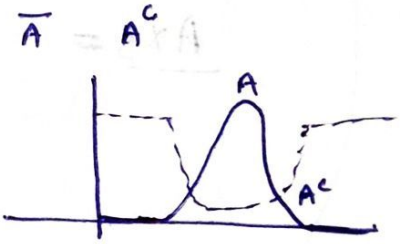
$(A \cap B) = \{ (x_1, 0.2) \quad (x_2, 0.6) \}$ ✓ min

$\mu_{AB}(x_1) = 0.9 \times 0.2 = 0.18$

$\mu_{AB}(x_2) = 0.6 \times 0.7 = 0.42$

(iii) Complement (A^c) :-

$$\mu_{(A^c)}(x) = 1 - \mu_A(x)$$



Eg- $A = \{ (x_1, 0.6) \quad (x_2, 0.7) \}$

$A^c = \{ (x_1, 0.4) \quad (x_2, 0.3) \}$

(iv) Vector Product $(A \cdot B)$:-

$$\mu_{AB}(x) = \mu_A(x) \cdot \mu_B(x)$$

(vi) Equality $(A = B)$

$$\mu_A(x) = \mu_B(x)$$

(v) Scalar Product $(\alpha \times A)$:-

$$\mu_{\alpha A}(x) = \alpha \cdot \mu_A(x)$$

$\alpha = 2$

$\mu_{\alpha A}(x_1) = 2 \times 0.6 = 1.2$

(vii) Power (A^α)

$$\mu_{A^\alpha}(x) = (\mu_A(x))^\alpha$$

$(0.6)^2 = 0.36$ ✓

(viii) Sum (A+B) :-

$$\mu_{A+B}(x) = \mu_A(x) + \mu_B(x) - \mu_A(x) \cdot \mu_B(x)$$

(ix) Difference (A - B) :-

$$\mu_{A-B}(x) = \mu_{A \cap B^c}(x) = (\min[\mu_A(x), \mu_{B^c}(x)])$$

$\mu_{B^c}(x) \downarrow$

$1 - \mu_B(x)$

(x) Disjunctive Sum ($A \oplus B$) :-

$$\mu_{(A \oplus B)}(x) = (A^c \cap B) \cup (A \cap B^c)$$

Q. $A = \{ (x_1, 0.4), (x_2, 0.3) \}$

$B = \{ (x_1, 0.2), (x_2, 0.6) \}$ $\quad 1 - \mu_A(x)$

Find- $\boxed{(A \cap B)^c = A^c \cup B^c}$ $\quad 1 - \mu_B(x)$

Ans - LHS $(A \cap B) = \{ (x_1, 0.2) \; (x_2, 0.3) \}$

$(A \cap B)^c = \{ (x_1, 0.8) \; (x_2, 0.7) \}$ ——— (1)

RHS $A^c = \{ (x_1, 0.6), (x_2, 0.7) \}$

$B^c = \{ (x_1, 0.8), (x_2, 0.4) \}$ $\quad \max$

$A^c \cup B^c = \{ (x_1, 0.8), (x_2, 0.7) \}$ ——— (2)

Cartesian Product ($A \times B$):-

$$\mu_{A \times B}(x, y) = \min\left(\mu_A(x), \mu_B(y)\right)$$

Eg- $A(x) = \{(x_1, 0.2), (x_2, 0.3), (x_3, 0.5)\}$

$B(y) = \{(y_1, 0.8), (y_2, 0.6), (y_3, 0.3)\}$

$A \times B =$

	y_1	y_2	y_3
x_1	0.2	0.2	0.2
x_2	0.3	0.3	0.3
x_3	0.5	0.5	0.5

Ek Sahil Ka Gyan

Fuzzy Relation:- fuzzy relations relate elements of one universe (x) to those of another universe (y) through Cartesian product.

- **Operations on fuzzy Relations:-**

(i) **Union:-** $\mu_{R \cup S}(x, y) = \max\left(\mu_R(x, y), \mu_S(x, y)\right)$

(ii) **Intersection:-** $\mu_{R \cap S}(x, y) = \min\left(\mu_R(x, y), \mu_S(x, y)\right)$

(iii) Complement :-

$$\mu_{\bar{R}}(x,y) = 1 - \mu_R(x,y)$$

(iv) Containment :-

$$R \subset S \Rightarrow \mu_R(x,y) \leq \mu_S(x,y)$$

- Fuzzy Composition :-

$$A \times B = R \quad , \quad R \subset A \times B$$

$$\mu_R(x,y) = \mu_{A\times B}(x,y) = \min(\mu_A(x), \mu_B(y))$$

2 types of fuzzy composition techniques ⇒

(i) Fuzzy max-min composition :- Er Sahil Ka Gyan

$R = X \times Y$, $S = Y \times Z$

max-min composition of R & S denoted by T(x, z) as

$$\mu_{R\circ S}(x,z) = \max(\min(\mu_R(x,y), \mu_S(y,z)))$$ ✓

(ii) Fuzzy max-product composition :-

$$\mu_T(x,z) = \mu_{R\cdot S}(x,z)$$

$$= \max(\mu_R(x,y) \cdot \mu_S(y,z))$$

Q** Let R, S be defined on sets $\{1,3,5\} \times \{1,3,5\}$ and

$$R : \{(x,y) \mid y = x+2\}$$

$$S : \{(x,y) \mid x < y\}$$

Ans - $X \times X = \{(1,1)\ (1,3)\ (1,5)\ (3,1)\ (3,3)\ (3,5)\ (5,1)\ (5,3)\ (5,5)\}$

$R = \{(1,3), (3,5)\}$

$S = \{(1,3)\ (1,5)\ (3,5)\}$

The relation matrices are –

$$R = \begin{array}{c} \\ x_1\ 1 \\ x_2\ 3 \\ x_3\ 5 \end{array} \begin{array}{c} \begin{array}{ccc} y_1 & y_2 & y_3 \\ 1 & 3 & 5 \end{array} \\ \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \end{array}, \quad S = \begin{array}{c} \\ 1 \\ 3 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccc} 1 & 3 & 5 \end{array} \\ \begin{bmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \end{array}$$

$RUS(x,y) = \max(R(x,y), S(x,y))$ $\quad RUS(1,1) = \max(0,0) = 0$

$RUS(x_1 y_1) = 0$, $RUS(x_1 y_2) = 1$, $RUS(x_1 y_3) = 1$

$RUS(x_2 y_1) = 0$, $RUS(x_2 y_2) = 0$, $RUS(x_2 y_3) = 1$

$RUS(x_3 y_1) = 0$ $\quad RUS(x_3 y_2) = 0$, $RUS(x_3 y_3) = 0$

$$RUS(x,y) = \begin{array}{c} \\ 1 \\ 3 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccc} 1 & 3 & 5 \end{array} \\ \begin{bmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \end{array} \checkmark$$

Ek Sahil Ka Gyan

$\min RMS(x,y) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$ (min)

$$\overline{R} = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

$$\begin{array}{c} \\ 1 \\ 3 \\ 5 \end{array} R = \begin{array}{c} \begin{array}{ccc} 1 & 3 & 5 \\ y_1 & y_2 & y_3 \end{array} \\ \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \end{array} \qquad S = \begin{array}{c} \begin{array}{ccc} 1 & 3 & 5 \\ z_1 & z_2 & z_3 \end{array} \\ \begin{bmatrix} 0 & 1 & 1 \\ 0 & 0 & -1 \\ 0 & 0 & 0 \end{bmatrix} \end{array}$$

$T(x,z)$

Composition :–

$$ROS\begin{pmatrix} x,z \\ 1,1 \end{pmatrix} = \max(\min(R(x,y), S(y,z)))$$

$ROS(1,1) = \max(\min(0,0), \min(1,0), \min(0,0))$

$= \max(0,0,0) = 0$

$ROS(1,3) = \max(\min(0,1), \min(1,0), \min(0,0))$

$= 0$

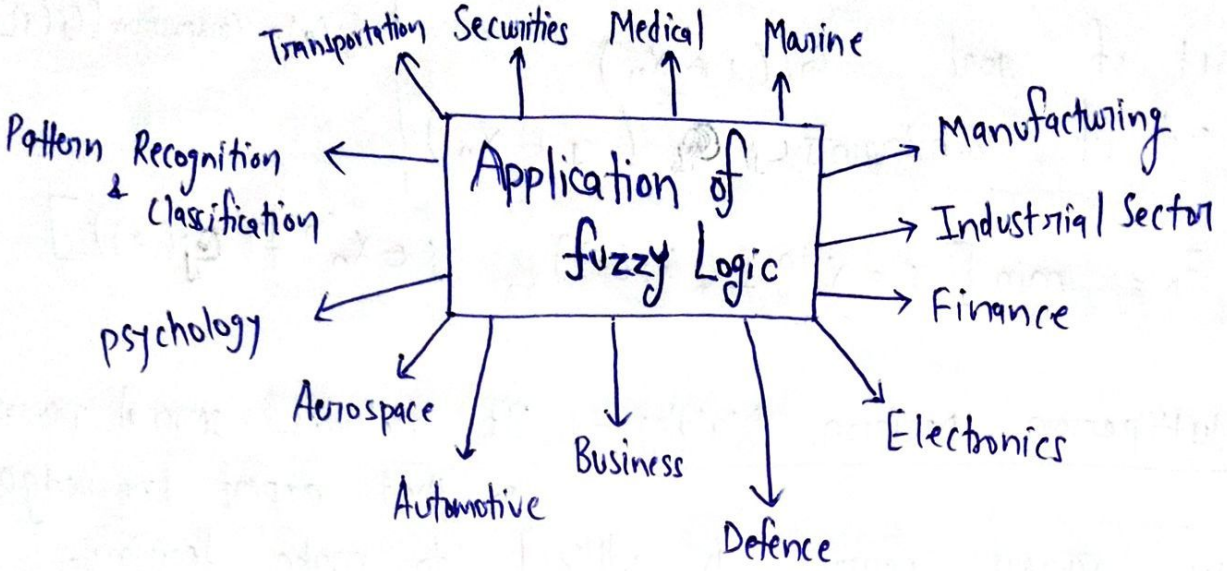
$ROS(1,5) = \max(0,1,0) = 1$

$ROS(3,1) = \max(0,0,0) = 0$

$ROS(3,3) = \max(0,0,0) = 0$, $ROS(3,5) = 0$

$ROS(5,1) = 0$, $ROS(5,3) = 0$, $ROS(5,5) = 0$

$$ROS = \begin{array}{c} \\ 1 \\ 3 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccc} 1 & 3 & 5 \end{array} \\ \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \end{array}$$



Fuzzy Modelling ⇒

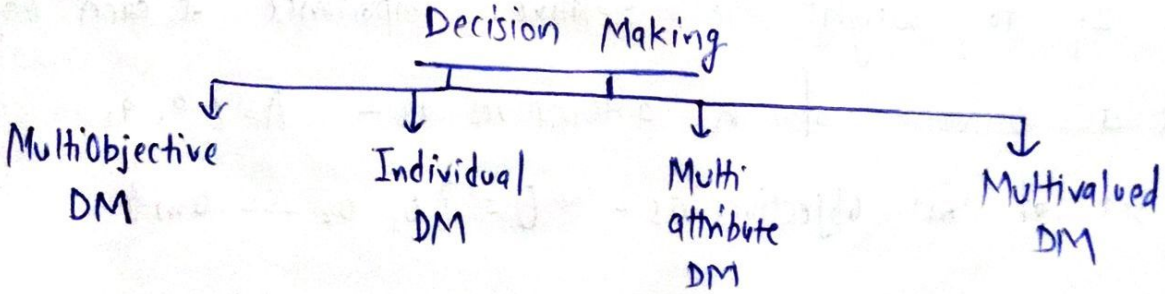
Fuzzy Logic Decision Making:- Decision Making activities are the steps taken to choose a suitable alternative from those that are needed for realizing a certain goal.

Steps for DM :-

(i) Determine the set of alternatives

(ii) Evaluating alternatives

(iii) Comparison blw alternatives



(i) Individual Decision Making:- Only a single person is responsible for taking decisions.

The decision making model in this kind can be characterized as -

- set of possible actions
- set of goal $G_i (i \in X_n)$
- set of constraints C_j $(j \in X_n)$

$G_i(a) = \text{Composition}[G_i(a)]$
$C_j(a) = \text{Composition}[G_j(a)]$

$$F_D = \min [i \in X_n^{in} f(G_i(a)), \; j \in X_n^{in} f(G_j(a))]$$

(ii) Multiperson Decision Making:- It includes several persons so that expert knowledge from various persons is utilized to make decision.

$SC: X \times X \rightarrow [0, 1]$

No of persons $= N(x_i, x_j)$

Total no. of decision maker $= n$

Then $SC(x_i, x_j) = \dfrac{N(x_i, x_j)}{n}$

(iii) Multi Objective Decision Making:-

When there are several objectives to be realized, then it is called multiobjective DM.

There is 2 issues:-

1. to acquire proper information related to the satisfaction of the objectives by various alternatives.
2. to weight the relative importance of each objective

- Let a universe of n alternatives as - $A = \{a_1, a_2 \dots a_n\}$ & set of 'm' objectives as - $O = \{o_1, o_2 \dots o_m\}$

Decision function = $D_1 \cap D_2 \cap \dots D_i \cap \dots D_m$

$$m_{DF}(a^*) = \max_{a \in A} [\mu_{DF}(a)]$$

(iv) Multi-Attribute DM:- It takes place when the evaluation of alternatives can be carried out based on several attributes of the object. The attributes can be numerical data, linguistic data & qualitative data.

$$Y = A_1X_1 + A_2X_2 + \dots + A_nX_n$$

Fuzzy Logic Control System :-

A control system is an arrangement of physical components designed to alter another physical system so that this system exhibits certain desired characteristics.

Following are some reasons of using fuzzy logic in control systems:

(i) While applying traditional control, one needs to know about the model and the objective function formulated in precise terms. This makes it very difficult to apply in many cases.

(ii) By applying fuzzy Logic for control we can utilize the human expertise and experience for designing a controller.

(iii) The fuzzy control rules, basically the IF-THEN rules, can be best utilized in designing a controller.

Architecture of fuzzy Logic Control:-

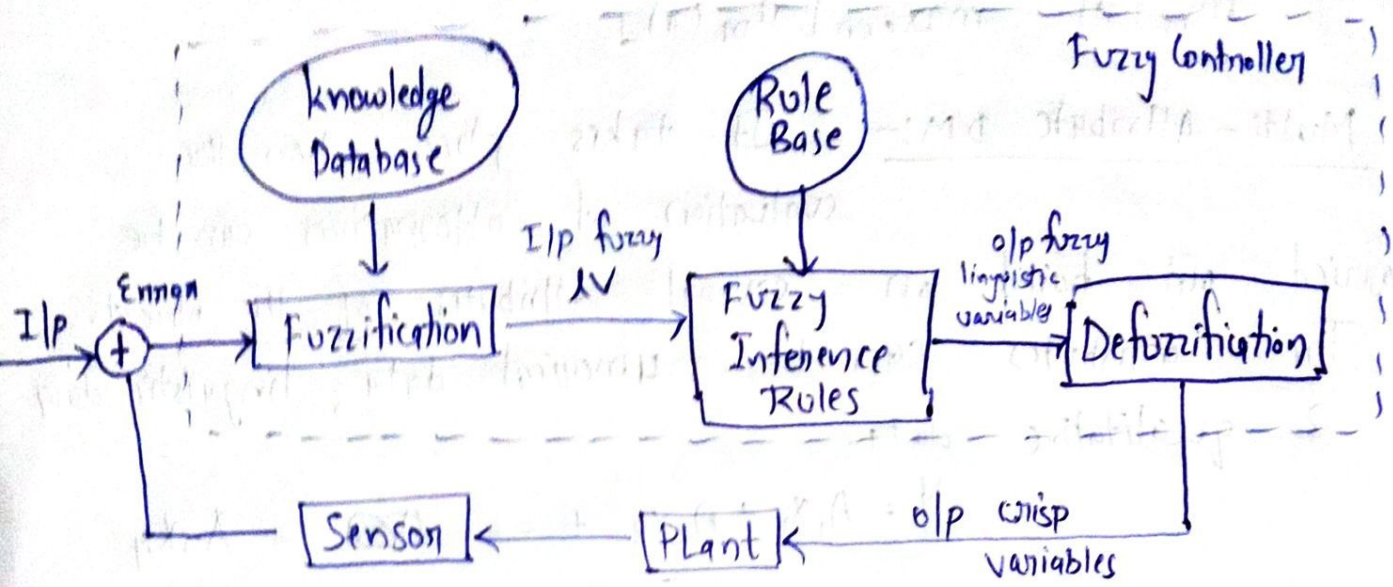


fig:- Architecture of FLC systems

Components of Fuzzy logic control:-

Fuzzifier:- It is to convert the crisp i/p values into fuzzy values.

Fuzzy knowledge Base:- Normalization of parameters involved, partitioning of i/p & o/p spaces, selection of membership functions of a primary fuzzy set.

Fuzzy Rule Base:- selection of i/o variables; types of fuzzy control rules; completeness of fuzzy control rules;

Inference Engine:- It acts as kernal of any FLC. Basically it simulates human decisions by performing approximate reasoning.

Defuzzifier:- The role of defuzzifier is to convert the fuzzy values into crisp values getting from fuzzy inference engine.

Advantages of FLC:-

- Cheaper
- Robust
- Customizable
- Reliability
- Emulate human deductive thinking
- Efficiency

Disadvantages:-

- Requires lots of data
- Useful in case of moderate historical data
- Need high human expertise
- Needs regular updating of rules
- Some fuzzy time logic is confused with probability theory & the terms.

Fuzzy Bayesian Decision Making:-

In classical Bayesian decision making, the future states of the nature are characterised as probability events. Conventionally, the probabilities sum to unity.

The problem with fuzzy Bayesian DM is that events are ambiguous.

$$P = \{ P(s_1), P(s_2) \cdots P(s_n) \} \quad \text{where} \sum_{i=1}^{n} P(s_i) = 1$$

$$A = \{ a_1 \; a_2 \cdots a_m \}$$

Expected utility $EX(U_j) = \sum_{i=1}^{n} U_j \, P(s_i)$ (cost of each alternative)

$$EX(u^*) = \max EX(U_i)$$

$$P(s_i \mid x_n) = \frac{P(s_i \mid x_n)}{P(x_n)} P(s_i) \quad [\text{Conditional probabilities}]$$

$$\boxed{P(x_n) = \sum_{i=1}^{n} P(x_n \mid s_i) \, P(s_i)}$$

$$EX(u_j|x_n) = \sum_{i=1}^{m} u_{ji} P(s_i|x_n), \quad EX(u^*|x_n) = \max EX(u_i|x_n)$$

$$E_x(u^*_n) = \sum_{i=1}^{m} EX(u^*|x_n) P(x_n)$$

value of perfect information $V(x) = EX(u^*_{xp}) - EX(u^*)$

and

$$P(s_i|E) = \frac{\sum_{r=1}^{n} P(x_n|s_i) \mu E(x_n) P(s_i)}{P(E)} = \frac{P(E|s_i) P(s_i)}{P(E)}$$

$$\phi = \{E_1, E_2 \dots E_m\}$$

The value of fuzzy information

$$\boxed{V(\phi) = E(u^*_\phi) - E(u^*)}$$

Fuzzy Information Processing - Fuzzy Robotics :—

- Robotics is the branch of engineering that deals with the design, construction, operation & application of robots.
- These technologies deal with automated machines that can take place of humans in dangerous environments or manufacturing processes.
- On the basis of situational reactive behaviours, behaviour based control has been proposed for robot navigation.

Eg- Robot motion to reach a target

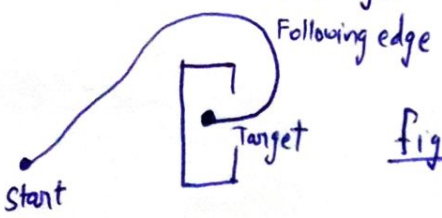


fig: Robot motion inside a U-shape object

Robot must efficiently weight multiple reactive behaviors, such as avoiding obstacle, following edge and moving to target

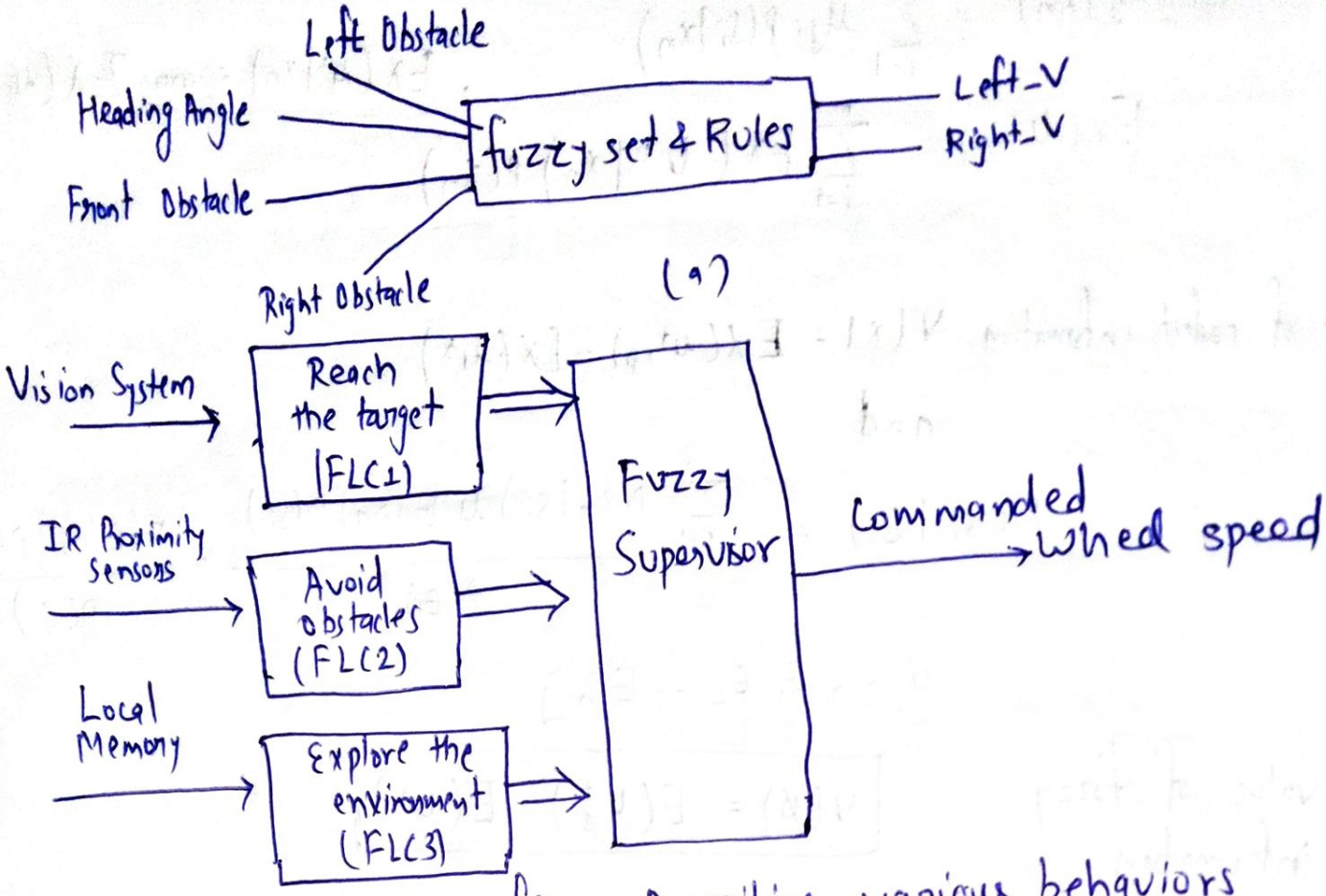


fig:- Describing various behaviors for Robot.

Advantages :-

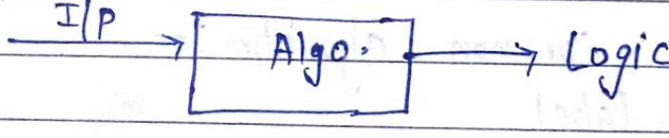
- Robots never get sick or rest.
- Work in dangerous Region
- Can work repetitive & unrewarding

Disadvantages :-

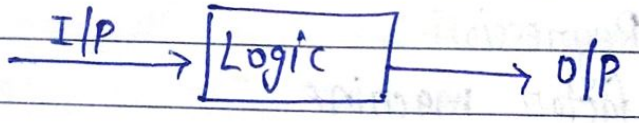
- Not creative or innovative
- Do not think independently
- Don't make complicated decisions.
- Don't adapt quickly to changes in their surrounding.

Machine learning is a subject of AI. The goal of ML generally is to understand the structure of data & fit that data into models that can be understood & utilized by people.
In traditional computing, algorithms are that of explicitly programmed instructions so used by computers to calculate or problem solve. ML algorithms instead allows for computer to trained on data inputs and used statistical analysis in order to output values that fall within specific
↳ range because of this ML facilitates computers in building model from sample data in order to automate decision making process based on data input.

Training phase:-



Testing phase:-



Type of ML Algorithm :⇒ There are variations of how to define ML algorithm but commonly they can be divided into categories according to there purpose & main category are following:-

1. Supervised
2. Unsupervised
3. Semi-supervised
4. Reinforcement

0.3439 0.1183 -0.3295

(1.) Supervised Learning :- Supervised is most common sub-branch of ML. Supervised ML algorithms are designed to learn by
Eg - The name supervised learning originate from idea that training : This type of algo like having a teacher supervised the whole process.

keypoint :-
(i) Predictive Model
(ii) Labeled data

The main type of SL problem is to regression and classification problem.

Supervised ML

Regression
[No labels defined]

Classification
[Labels defined]

- List of common algorithm :-

(i) Nearest label
(ii) Decision Tree
(iii) Linear Regression
(iv) Support factor machine
(v) Neural Network
(vi) Naive Bayes

(2) UnSupervised Learning :-

(A) The computer prints with unlabeled data.
(B) There is no teacher at all, actually computer might be able to teach you new things after it learns from itself. These algos. are particularly used in cases

where human experts don't know what to look for data.

(c) These algo. are the family of ML algo. which are used in pattern detection and disrespecting modelling. However there are no o/p categories or labels that are based on which algo. can → try to model relationships (d) These algos. try to use techniques on data to mine the rules, detects patterns and the data points which helps in deriving inside & describe the data better to the user.

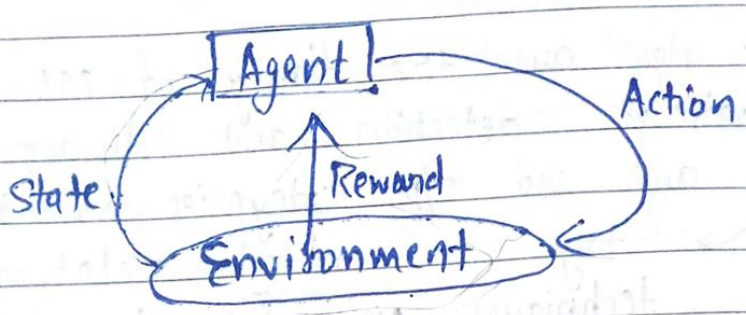
keypoint:- (i) Descriptive Model
(ii) Main type of Unsupervised are:-
(A) Association Rule Mining
(B) Clustering Algorithm

List of common algo.:-
(i) K-mean clustering
(ii) Association Rule Mining

(3) Reinforcement Learning:- This method aims to using observation gathered from the interaction with environment to take actions that could maximize the reward or minimise the risk. RL algo. (called Agent) continuously learns from env. in a interaction session in process the agent learnt from its experience of env. until it explores the full range of possible state.

→ RL is a type of ML & therefore also branch of AI.
→ It allows machine & slw agents to automatically determine the behavior with specific context in order to max. performance. Simple reward feedback is required for agent to learn its behavior

This is known as Reinforcement Agent.



In order to produce intelligent programs (Agents) RL goes to following steps:—

(i) Input state is observed by agent.
(ii) Decision making function is used to make agent perform a function.
(iii) After action is performed, agent receives the reward or reinforcement from env.
(iv) The state action pair information about the reward is stored.

List of common algorithm:—

(A) Q Learning
(B) Temporal Difference
(C) Deep Adversarial Network

Use cases: Some applications of RL algo are⇒

→ Computer played board game
→ Robotics Hands
→ Self Driving cars

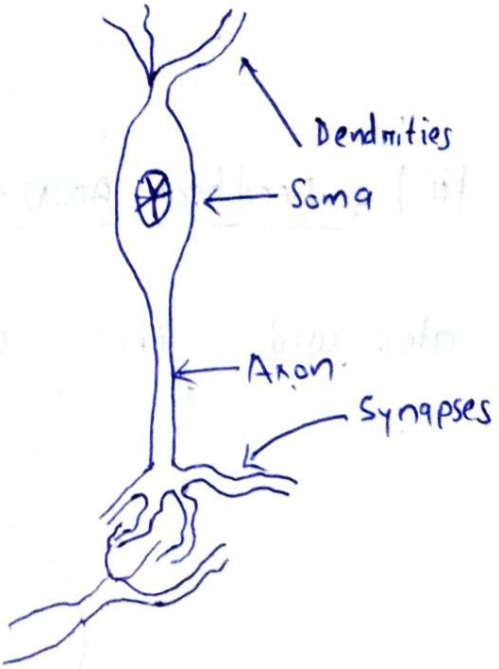
Artificial Neural Networks [ANN]:— ANNs has developed from a biological model of the brain.

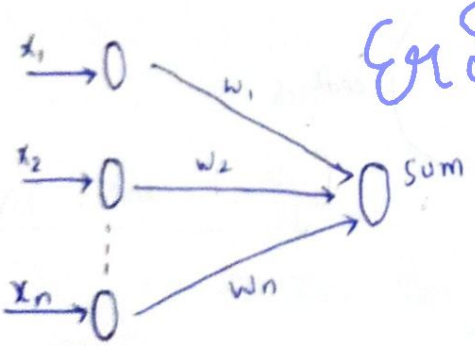
A neural net consists of a set of connected cells as the neurons. The neurons receive impulses from either cells or output cells.

- ANNs are programs designed to solve any problem by trying to mimic the structure & the function of our nervous system.
- A neural network acquires knowledge through learning.

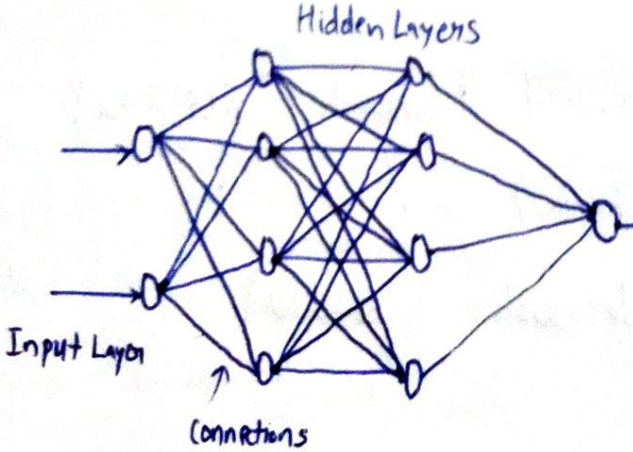
Biological Neuron model:—

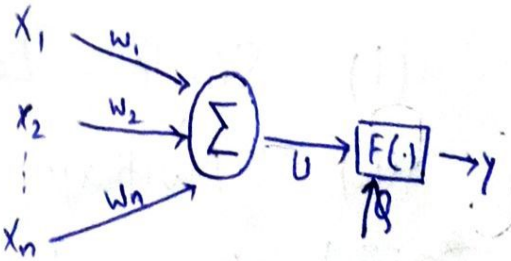
- Dendrities :— Accepts the inputs
- Soma :— process the inputs
- AXON :— Turns the processed i/p into o/ps.
- Synapses :— The electrochemical contact b/w neurons





$$\text{sum} = w_1x_1 + w_2x_2 + \cdots + w_nx_n$$





$$U = \sum_{i=1}^{n} w_j x_i$$

Q = desired o/p

$$y = F(U - Q)$$

Types of ANN :-

(i) **Feed Forward ANN** ⇒ In this network, flow of information is undirectional. A unit used to send information to another unit that does n't receive any information. Also, no feedback loops are present in this.

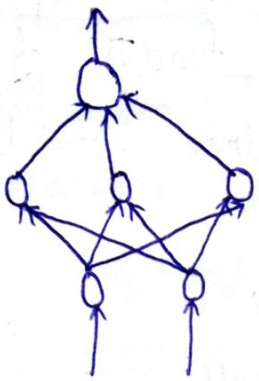


fig :- Feedforward ANN

(ii) **Feedback ANN** ⇒ In this particular artificial neural network, it allows feedback loops, also used in content addressable memories.

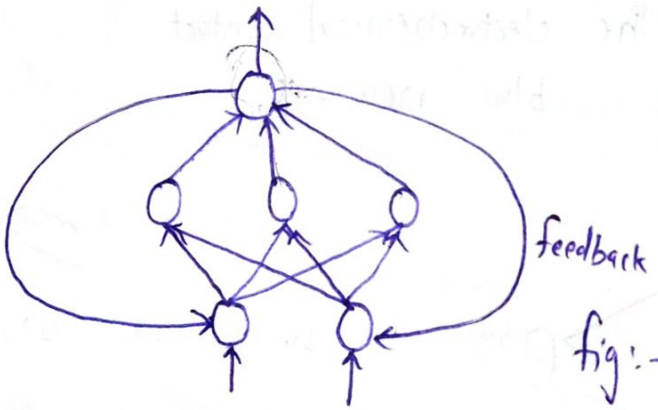


fig :- Feedback ANN

Classification of ANN :-

(i) Single Layer Feedforward Network (SLFFN)

(ii) Multilayer FN (MLFFN)

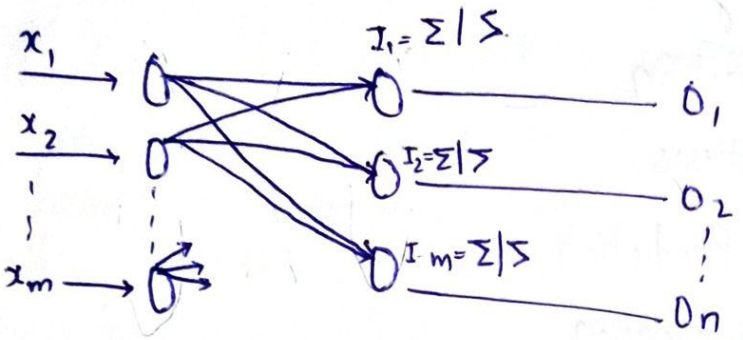
(iii) Recurrent Neural Network (RNN)

(i) **SLFFN:-** There is only one computational layer (op Layer) so it is single Layer FFN.

→ The input $x_1, x_2 \dots x_m$ are connected to layers of neurons through weight matrix

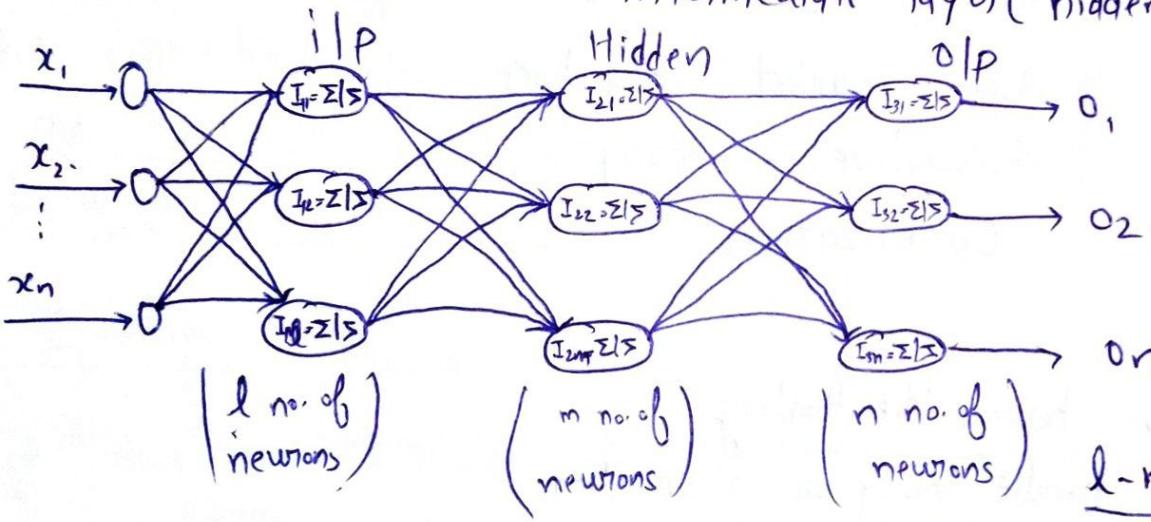
$$W = \begin{bmatrix} w_{11} & w_{12} & w_{13} & \dots & w_{1n} \\ \vdots & & & & \\ w_{m1} & w_{m2} & w_{m3} & \dots & w_{mn} \end{bmatrix}_{m \times n}$$

o/p of k^{th} neuron $O_k = f_k\left(\sum_{i=1}^{n} (w_{ik} \cdot x_i) + \theta_k \right)$ → Threshold value



Gyan Sahil Ka Gyan

(ii) **MLFFN:-** There is an input layer, and one or more intermediate layers (hidden layer).

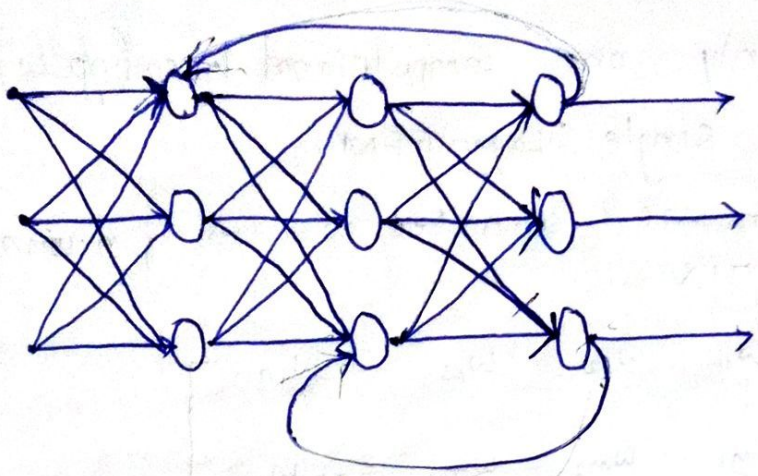


l-m-n FFN

$$O_i^l = f_i^l \left(\sum x_i w^l + \theta_i^l \right)$$

(iii) **RNN:-** The network differ from feedback network architecture in the sense that there is at least one "feedback Loop".

So there could exist one layer with feedback connection.



Application of ANNs :-

(i) Signal Processing
(ii) Pattern Recognition
(iii) Speech Synthesis
(iv) Forecasting & Prediction
(v) Control & Automation
(vi) Radar Interpretation
(vii) Interpreting Brain Scans
(viii) Stock market Prediction
(ix) Associative Memory
(x) Optimization

Er Sahil Ka Gyan

Adv :-

- Involves human like thinking
- They handle noisy or missing data
- They can work with large no. of variables
- They provide general solutions with accuracy
- They deal with the non-linearity in the world in which we live.

Radial Basis function Network (RBFN):-

→ • It is used for approximate function & recognition pattern (classification).

→ • It uses gaussian potential function.

The radial base function or multi quadratic function

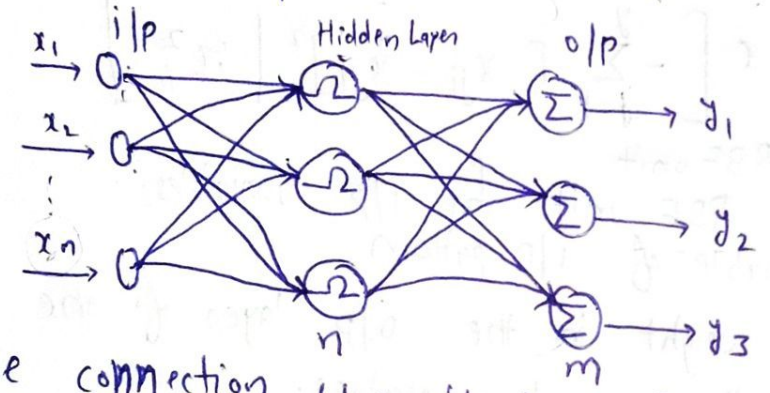
$$\phi(r) = (r^2 + c)$$

$c \rightarrow$ true cost $r \in R$

$$\phi(r) = r$$

$$\phi(r) = r^2, \quad \phi(r) = r^3, \quad \phi(r) = \exp(-r^2)$$

→ It is an multilayer feedforward network. It consists of n no. input neuron & m no. of o/p neuron.



m > n

The connection blw i/p layer & hidden layer for no. n the hyperbolial connection & hidden layer & o/p layer from m the weighted connection.

Tanning Algorithm:-

It uses the gaussian activation function the response of such function is non-negative for all values of x.

$$f(x) = \exp(-x^2)$$

S_1:- Initialize weight and basis (random values)

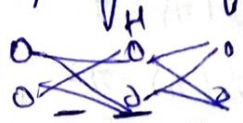
S_2:- While Stopping criteria is false do step-3 to 10.

S_3:- for each input do step 4 to 9.

S_4 :- For each input unit x_i ($i = 1$ to n) receives input signal to all units in layer above.

S_5 :- Calculate the radial basis function.

S_6 :- Select the centre of RSF the centre of selected from the set of input vectors. The no. of selections have to be selected in order to insure adequate sampling of the input vector space.



S_7 :- The output of In unit $V_i(xe)$ in hidden layer.

$$V_i(x_i) = e\left[-\sum_{j=1}^{y}\left[x_{ji} - \hat{x}_{ji}\right]^2 \Big| r_i^2\right]$$

r_i = bit of RBF unit

where x_{ji} = Center of RBF unit for i/p variables.

x_{ji} = jth variable of i/p pattern

S_8 :- Initialize the weight in the o/p layer of the m to small random values.

S_9 :- Calculate the o/p of the neural network.

$$y_{net} = \sum_{i=1}^{H} W_{im}(V_i(xp) + W_o$$

where

H = no. of hidden layer holds

y_{net} = o/p value of nth node in o/p layer

W_{im} = weight b/w ith RBF unit and o/p node

W_o = bias at nth o/p node.

S_{10} :- Calculate error & test stop condition, the stop condition may be weight change no. of epochs etc.

Adaptive Resonance Theory :- (ART)

It is a type of neural network technique developed by Stephen & Grail in 1987. The basic ART uses unsupervised learning technique.

The term "adaptive" & "~~resource~~ "resonance" used in this suggests that they are open to new learning without discarding the previous or old information.

- ART networks are known to solve the stability (nature of memorizing learning) and plasticity (flexible to gain new information) dilemma.
- ART networks implement a clustering algorithm.

ART phase ⇒

Recognition Phase ⟶ Comparison Phase ⟶ Search Phase

ART1 :- It is a type of ART which is designed to cluster binary vectors.

Architecture of ART :- It consists of 2 unit.

(i) Computational unit :-

(a) Input unit ⇒

⇒ F_1 a layer input portion :- There would be no processing in this portion rather than having the input vectors only. It is connected to F_2 b.

⇒ F_1 b layer Interfaceportion:- The portion combines signal from i/p portion with that of F_2 Layer.

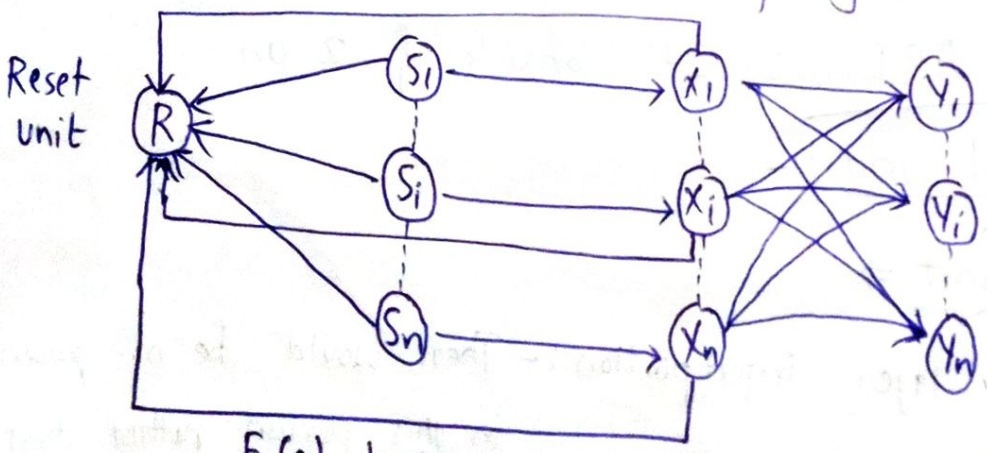
(b) Cluster unit (F_2 Layer):- This is a competitive layer. The unit having the largest net input is selected to learn the i/p pattern. The activation of all other cluster unit are set to 0.

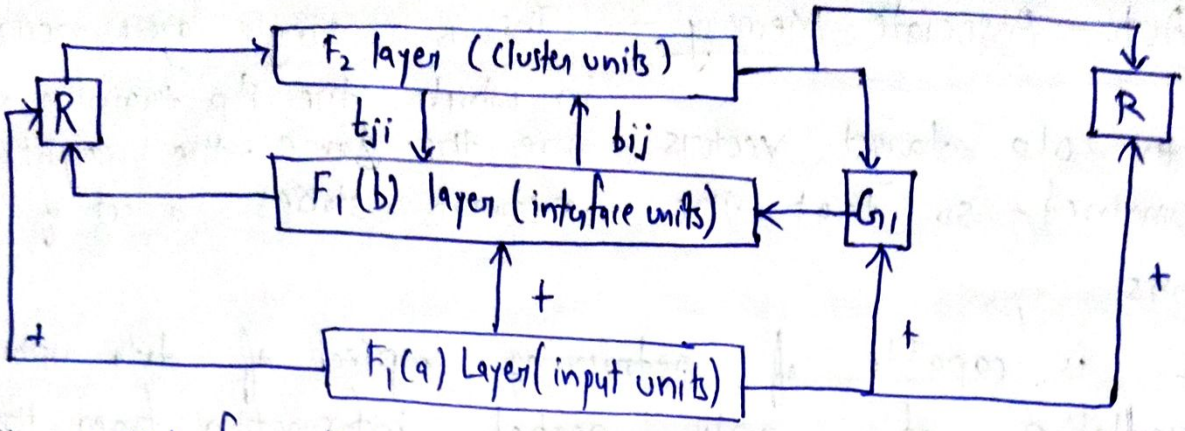
(c) Reset Mechanism:- It is based upon ~~simplicity~~ similarity b/w top-down weight & the i/p vector how if degree of this ~~simplicity~~ similarity is less than vigilance, then the cluster is not allowed to learn the pattern & a reset would happen.

(2) Supplement Unit ⇒ due to issue with reset mechanisme G_1 & G_2 (sumplemental unit) are added along with reset unit R. They are called gain control units.

't' indicate on excitatory signal.

'-' indicate on inhibitory signal.





bij = weight from F_1 b to F_2 layer i.e. bottom-up weights

tji = Top-down weights

Applications of ART:-

(i) Target Recognition
(ii) Mobile Control Robot
(iii) Signature verification
(iv) Medical Diagnosis

Adv.

- Coordinated with technologies
- Ensure stability
- Recognitions, robotics
- stability-plasticity Dilemma

Associative Neural Memory:-

It is obtained by its content, adjacent to an explicit in the traditional compute memory system the memory enables the recollection of information based on incomplete knowledge of its content..

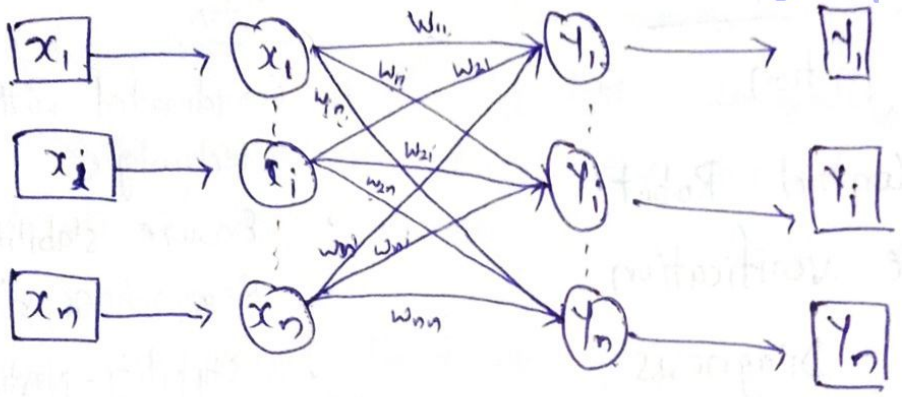
→ Associate Memory are divided into 2 parts:—

- Auto- Associate Memory
- Hetro- Associate Memory

(i) **Auto-Associate Memory:-** This is a single layer neural n/w in which the i/p training vector and the o/p target vectors are the same. the weights are determined so that the network stores a set of patterns.

→ It is capable of retrieving a piece of data upon presentation of only partial information from that place of data.

→ Neural n/w using autoassociative memory are called associative network.



weight matrix Calculation = $W_{new} = W_{old} + x_i y_j$

$W_{new} = [s]^T [\epsilon]$

Training Algorithm:-

S_1:- Initialize all weights to zero as $w_{ij} = 0$, $i = 1$ to n, $j = 1$ to n

S_2:- Perform step-3&4 for each i/p vector

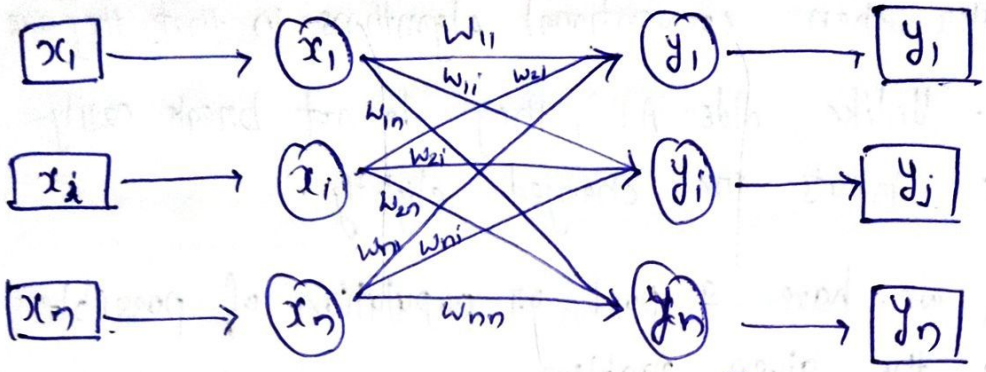
S_3:- Activate each i/p unit as follows $x_i = s_i$ $(i = 1 \text{ to } n)$

S_4:- Activate each o/p unit as follows $y_j = s_j$ $(j = 1 \text{ to } n)$

S_5:- Adjust the weights:

$$w_{ij}(new) = w_{ij}(old) + x_i y_j$$

(ii) Hetro Associative Memory:- In this network, the i/p training vector & o/p target vectors are not same. The weights are determined so that network stores a set of pattern. It is static in nature, hence there would be no non-linear & delay operations.



Training algorithm:-

S_1:- Initialize all the weights to zero as

$$w_{ij} = 0 \quad , i = 1 \text{ to } n, \quad j = 1 \text{ to } n$$

S_2:- Perform step 3-4 for each i/p vector

S_3:- Activate each i/p unit as

$$x_i = s_i \quad (i = 1 \text{ to } n)$$

S_4:- Activate each o/p units as

$$y_j = s_i \quad (j = 1 \text{ to } m)$$

S_5:- Adjust the weight as follows -

$$w_{ij}(\text{new}) = w_{ij}(\text{old}) + x_i y_j$$

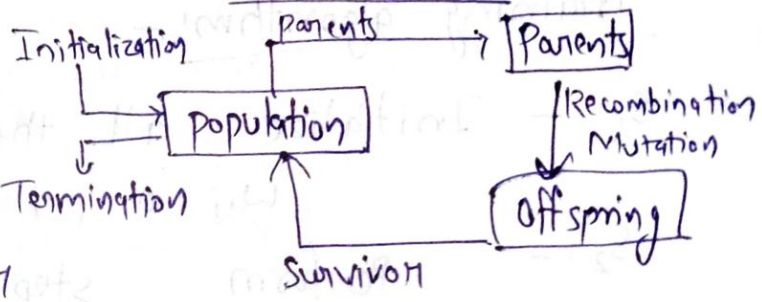
Genetic Algorithm:-

→ GAs are search based algorithm (adaptive heuristic) based on the evolutionary ideas of natural selection & genetics.

→ They are better than conventional algorithms in that they are more robust. Unlike older AI, they do not break easily even if the inputs are changed slightly.

→ In GAs, we have a pool on population of possible solution to the given problem.

Cycle of GA



→ Advantages :-

(i) Parallelism
(ii) Liability
(iii) Solution space is wider
(iv) Easy to discover global optimum.
(v) The problem has multiobjective function.

Limitations:-

(i) The problem of identifying fitness function
(ii) Definition of representation for the problem
(iii) Premature convergence occurs.
(iv) The problem of choosing various parameter such as the size of population, mutation rate, crossover rate, the selection method & its strength.

→ GAs have the ability to deliver a "good-enough" solution "fast-enough". This makes GA to solve optimization problems.

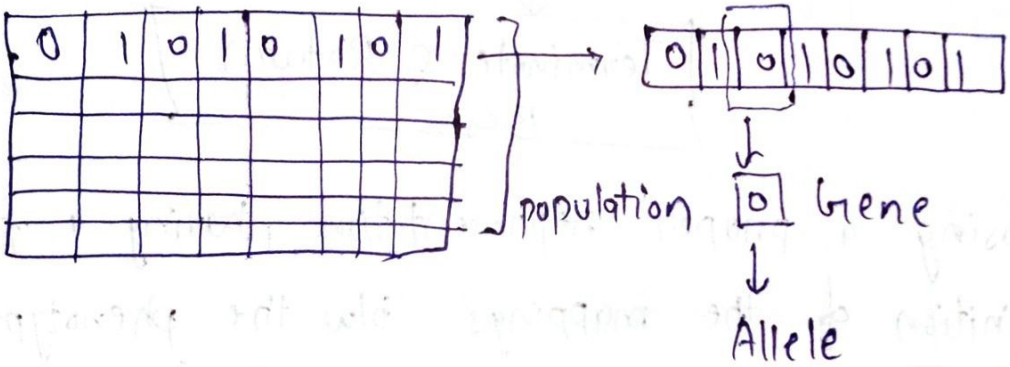
The reasons why GA are needed are as follows –

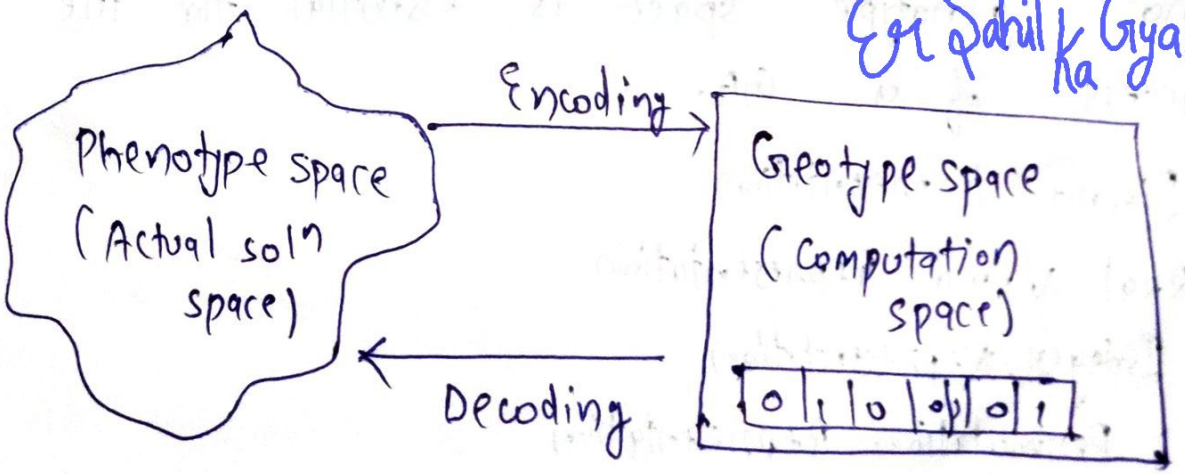
(i) Solving difficult Problems

(ii) Failure of Gradient Based Method

(iii) Getting a Good solution fast

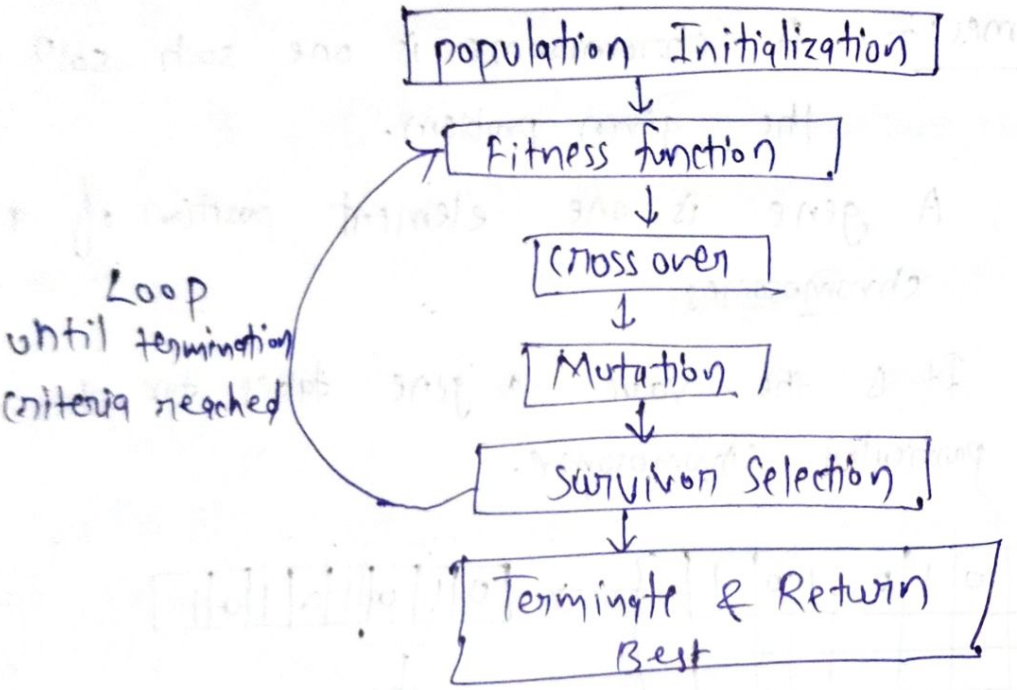
- **Population** :- It is subset of all possible solutions to the given problem.
- **Chromosomes** :- A chromosome is one such soln for the given problem.
- **Gene** :- A gene is one element position of a chromosomes.
- **Allele** :- It is the value a gene takes for a particular chromosome.





Structure of GA :- We start with an initial population, select parents from this population from mating.

- Apply crossover & mutation operators on the parents to generate new off-springs. And finally these off springs replace the existing individual in the population. and the process repeats.



→ Choosing a proper representation, having a proper definition of the mappings blw the phenotype and genotype space is essential for the success of a GA.

(i) Binary Representation
(ii) Real valued Representation
(iii) Integer Representation
(iv). Permutation Representation

→ fitness function takes solution as i/p & produces suitability of soln as o/p.

→ Genetic function alters genetic composition of offspring. These includes crossover, mutation, selection etc.

Parameter / Operators of Genetic Algorithm:- ⁂

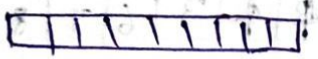
Genetic operator is used to guide the algorithm towards a solution to a given problem.

(i) Mutation
(ii) Crossover
(iii) selection

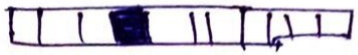
Er Sahil Ka Gyan

(i) Mutation :- Mutation is the part of GA which is related to the "exploration" of the search space. It has been observed that mutation is essential to the convergence of GA while crossover is not.

BitFlip



Before Mutation

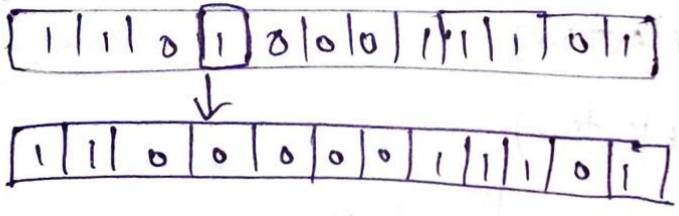


After Mutation

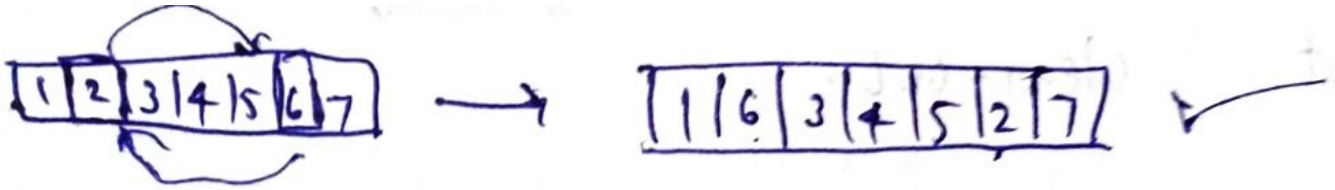
* Binary Mutation ⇒ Mutation operator changes 1 to 0 or vice-versa with a mutation probability

- The mutation probability is generally kept low for steady convergence.

Bit Flip



(ii) Swap Mutation:-

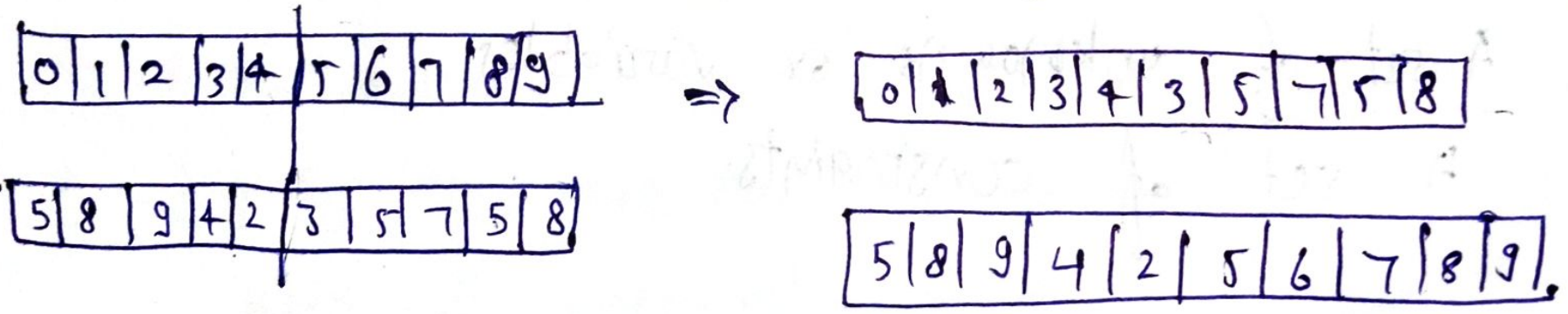


Scramble Mutation

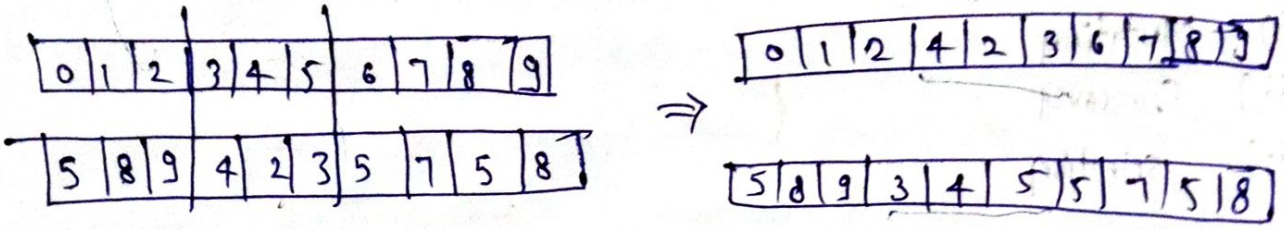
Inversion Mutation

(ii) Crossover Operator :—

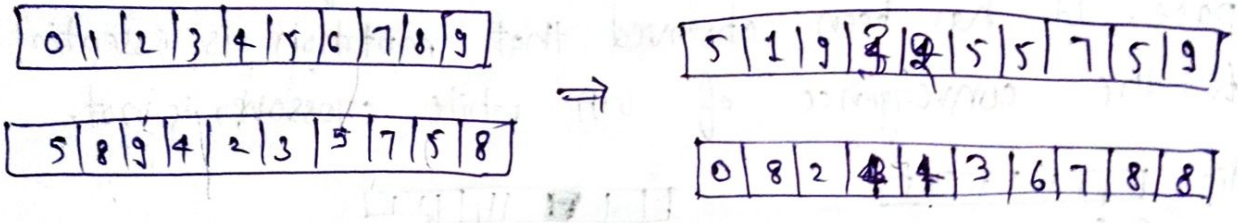
(a) One point Crossover ⇒ A random crossover point is selected & the tails of its two parents are swapped to get new off-springs.



Multi-point crossover :- It is a generalization of the one-point crossover where in alternating segments are swapped to get new off-springs.

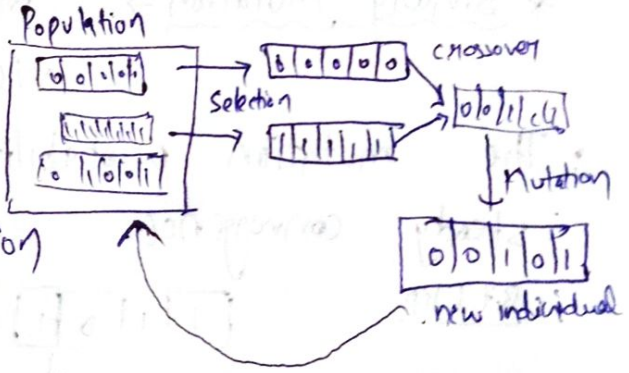


Uniform crossover :-



(iii) Selection :-

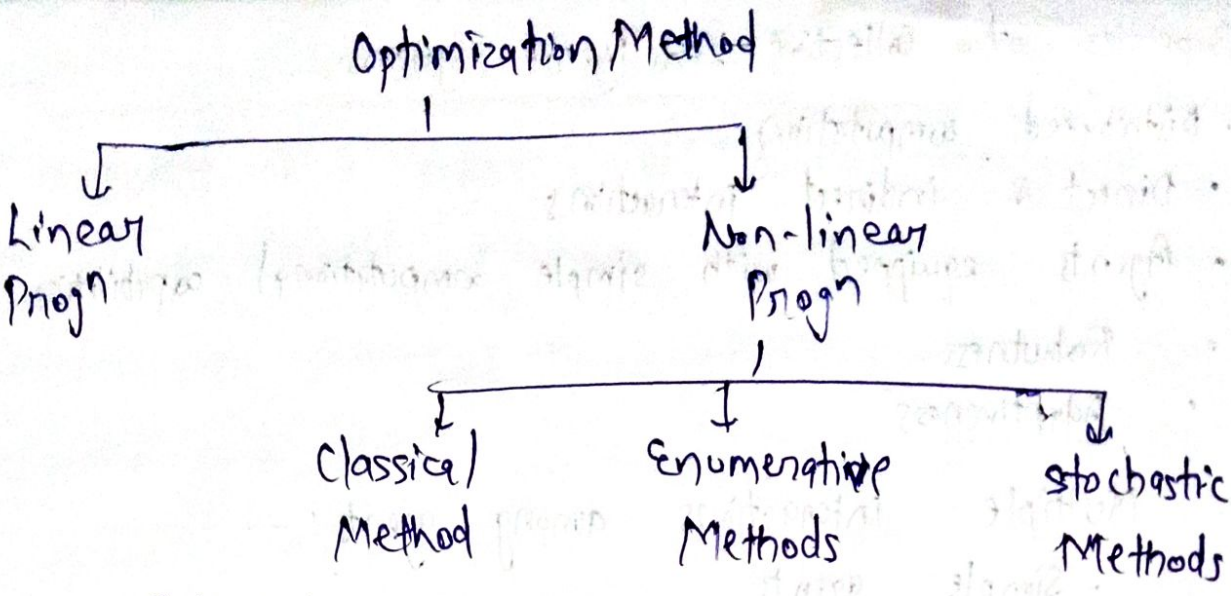
- Roulette-wheel selection
- Elitist selection
- Fitness-proportionate selection
- Scaling selection
- Rank selection



Optimization of GA :-

Optimization is a process that find a best or optimal soln for problem. It is centered around 3 factors.

(i) An objective function
(ii) A set of unknowns or variables
(iii) A set of constraints.



Swarm Optimization:-

- Swarm intelligence is the collective behaviour of decentralized, self organized systems, natural or artificial. SI system consist typically of a population of simple agents or boids interacting locally with one another and with their environment.
- The agents follow very simple rules & although there is no centralized control structure dictating how individual agents should behave, local & to a certain degree random intractions- b/w such agents lead to the emergence of "intelligent" global behaviour, unknown to the individual agents.

2 principles in SO:-

→ Self Organized is based on:-
- Activity amplification by positive feedback.
- Activity balancing by negative feedback.
- Amplification of random fluctuations multiple interactions.

→ Stigmergy:- Stimulation by work → is based on:-
- Work as behavioural response to the environmental state an environment that serves as a work state memory.

→ Properties of Collective Intelligence systems:—

- Distributed computation
- Direct & indirect interactions
- Agents equipped with simple computational capabilities.
- Robustness
- Adaptiveness

→ Multiple interactions among agents:—

- Simple agents
- System composed of many agents.

→ Positive feedback:—

- Amplification of random fluctuations & structure formation.
- Reinforcement of most common behavior patterns.

→ Negative feedback:—

- Saturation
- Competition
- Resource exhaustion

Er Sahil Ka Gyan

Particle Swarm Optimization:—

- PSO can be applied into both scientific research and engineering use.
- PSO have no overlapping & mutation calculation.

→ It is a computational method that optimizes a problem by iteratively trying to improve a candidate solution with regard to a given measure of quality.

→ It uses a number of particles that constitute a swarm moving around in the search spare, looking for the best solution.

Working :-

→ Population initialized by assigning random positions and velocities, potential solutions are then flow n through hyperspace.

→ Each particle keeps track of its "best" (highest fitness) position in hyperspace.

→ This is called "pbest" for an individual particle.

→ It is called "gbest" for the best in population.

→ It is called "lbest" for the best in defined neighborhood.

→ At each time step, each particle stochastically accelerates toward its pbest & gbest or lbest.

PSO Algorithm (General) :-

Searches hyperspace of problem for opimum

- Define problem to search
- How many dimensions?
- Solution Criteria?
- Initialize Population
- Random initial position
- Random initial velocities
- Determine Best Position
- Determine Global best position
- Personal best Position
- Update velocity & position Equations

Step by step Implementation :-

Step-1 Initialize PSO parameters, population size, no. of generations, cognitive constant, social constant, maximum velocity, no. of design variables & respective range for design variables.

Step-2 Generate random population equal to population size, Each population member contains the value of all design variables.

Step-3
- Obtain values of objective function for all population members.
- In first iteration, value of OF indicates pBest.
- Identify particle with best OF value which is gBest.

Step-4
- Update velocity of each particle & check for maximum velocity.
- If velocity obtained exceeds maximum velocity then reduce existing velocity to maximum velocity.

Step-5
- Update the position of particles,
- Check all design ~~particles~~ variables for upper & lower limits.

Step-6
- Obtain value of OF for all particles, new solution replaces the pBest if it has better values.
- Identify the gBest from population.

Step-7 :-
- Best obtained results are saved using elitism.
- All elite members are not modified using crossover & mutation operators.

Step-8 :-
- Repeat the steps (from step 4) until the specified no. of generations or termination criterion is reached.

Adv. & Disdv

PSO in Engineering – Applications of GA:-

GA are useful when

- search space is large, complex or poorly understood.
- Domain knowledge is expert knowledge is difficult to encode to narrow the search space.
- No mathematical analysis is available.
- Traditional Search Methods fail.

GA are applied to many scientific, engineering problem, in business & entertainment including:-

(i) Optimization
(ii) Automatic Programming
(iii) Machine & Robot Learning
(iv) Economic Models
(v) Immune System Model
(vi) Ecological Model
(vii) Population Genetics Models
(viii) Interactions b/w evolution & learning
(ix) Models of social systems