PART-A

Q.1 *What is soft computing?*

Ans. Soft Computing: Soft computing is a collection of artificial intelligence-based computational techniques. An approach to computing which parallels the remarkable ability of the human mind to reason and learn in an environment of uncertainty and imprecision. It is characterized by the use of inexact solutions to computationally hard tasks such as the solution of nonparametric complex problems for which an exact solution can't be derived in polynomial of time.

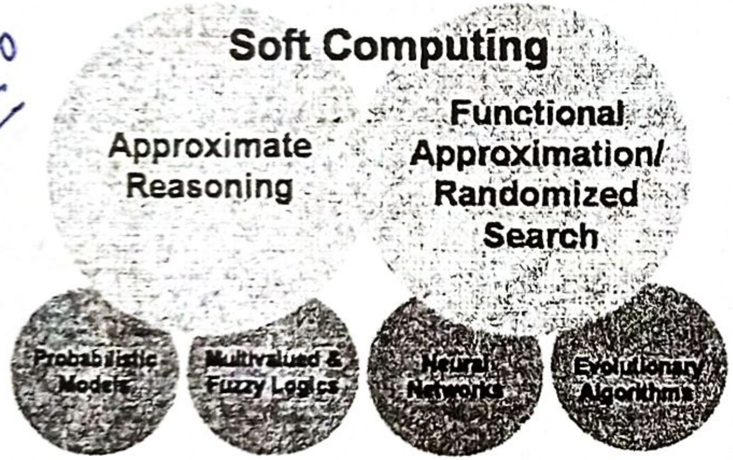


Fig.

Q.2 *Why soft computing approach?*

Ans. Mathematical model and analysis can be done for relatively simple systems. 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. It extends its application to various disciplines of engineering and science. Typically human can:

1. Take decisions
2. Inference from previous situations experienced
3. Expertise in an area
4. Adapt to changing environment
5. Learn to do better
6. Social behaviour of collective intelligence.

Q.3 *What do you mean by non-interactive fuzzy sets?*

Ans. The independent events in probability theory are analogous to noninteractive fuzzy sets in fuzzy theory. A noninteractive fuzzy set is defined as follows. We are defining fuzzy set $\underset{\sim}{A}$ on the cartesian space $X = X_1 \times X_2$. Set $\underset{\sim}{A}$ is separable into two noninteractive fuzzy sets called orthogonal projections, if and only if

$$\underset{\sim}{A} = OPr_x(\underset{\sim}{A}) \times OPr_x(\underset{\sim}{A})$$

$$\text{where } \mu_{OPr_{x_1}(\underset{\sim}{A})}(x_1) = \max_{x_2 \in X_2} \mu_{\underset{\sim}{A}}(x_1, x_2) \qquad \forall x_1 \in X_1$$

$$\mu_{OPr_{x_2}(\underset{\sim}{A})}(x_2) = \max_{x_1 \in X_1} \mu_{\underset{\sim}{A}}(x_1, x_2) \qquad \forall x_2 \in X_2$$

The equations represent membership functions for the orthographic projection of $\underset{\sim}{A}$ on universes X_1 and X_2, respectively.

Q.4 *What is fuzzy control?*

Ans. Fuzzy Control:

1. It is a technique to embody human-like thinking into a control system.
2. It may not be designed to give accurate reasoning but it is designed to give acceptable reasoning.
3. It can emulate human deductive thinking, that is, the process people use to infer conclusions from what they know.
4. Any uncertainties can be easily dealt with the help of fuzzy logic.

Q.5 *Write the characteristics of neuro - fuzzy and soft computing.*

Ans. Characteristics of Neuro - Fuzzy and Soft Computing:

1. Human Expertise
2. Biologically inspired computing models
3. New Optimization Techniques
4. Numerical Computation
5. New Application domains
6. Model-free learning
7. Intensive computation
8. Fault tolerance
9. Goal driven characteristics
10. Real world applications Intelligent Control Strategies (Components of Soft Computing): The popular soft computing components in designing intelligent control theory are:
    (i) Fuzzy Logic
    (ii) Neural Networks
    (iii) Evolutionary Algorithms

PART-B

Q.6 *What do you mean by fuzzy logic?*

Ans. Fuzzy Logic : The term **fuzzy** refers to things whic are not clear or are vague. In the real world many time we encounter a situation when we can't determine whethe the state is true or false, their fuzzy logic provides a ver valuable flexibility for reasoning. In this way, we ca consider the inaccuracies and uncertainties of any situatio

In boolean system truth value, 1.0 represent absolute truth value and 0.0 represents absolute false valu But in the fuzzy system, there is no logic for absolute trut and absolute false value. But in fuzzy logic, there i intermediate value too present which is partially true an partially false.

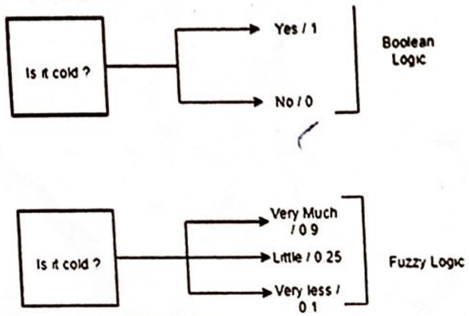


Fig. 1

Architecture : Its Architecture contains four parts :

1. **Rule Base:** It contains the set of rules and the IF THEN conditions provided by the experts to gover the decision making system, on the basis of linguisti information. Recent developments in fuzzy theor offer several effective methods for the design an tuning of fuzzy controllers. Most of thes developments reduce the number of fuzzy rules.
2. **Fuzzification:** It is used to convert inputs i.e. cris numbers into fuzzy sets. Crisp inputs are basicall the exact inputs measured by sensors and passe into the control system for processing, such a temperature, pressure, rpm's, etc.
3. **Inference Engine:** It determines the matchin degree of the current fuzzy input with respect t each rule and decides which rules are to be fire according to the input field. Next, the fired rule are combined to form the control actions.
4. **Defuzzification:** It is used to convert the fuzzy set obtained by inference engine into a crisp value. There are several defuzzification methods available and

the best suited one is used with a specific expert system to reduce the error.

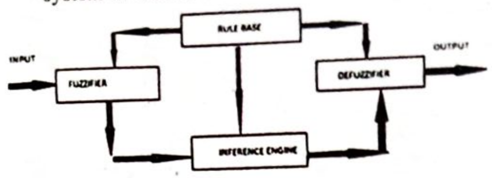


Fig. 2 : Fuzzy Logic Architecture

Membership function : A graph that defines how each point in the input space is mapped to membership value between 0 and 1. Input space is often referred as the universe of discourse or universal set (u), which contain all the possible elements of concern in each particular application.

There are largely three types of fuzzifiers:

1. Singleton fuzzifier
2. Gaussian fuzzifier
3. Trapezoidal or triangular fuzzifier

Advantages of Fuzzy Logic System

(i) This system can work with any type of inputs whether it is imprecise, distorted or noisy input information.

(ii) The construction of Fuzzy Logic Systems is easy and understandable.

(iii) Fuzzy logic comes with mathematical concepts of set theory and the reasoning of that is quite simple.

(iv) It provides a very efficient solution to complex problems in all fields of life as it resembles human reasoning and decision making.

(v) The algorithms can be described with little data, so little memory is required.

Disadvantages of Fuzzy Logic Systems

(i) Many researchers proposed different ways to solve a given problem through fuzzy logic which lead to ambiguity. There is no systematic approach to solve a given problem through fuzzy logic.

(ii) Proof of its characteristics is difficult or impossible in most cases because every time we do not get mathematical description of our approach.

(iii) As fuzzy logic works on precise as well as imprecise data so most of the time accuracy is compromised.

Application

(i) It is used in the aerospace field for altitude co of spacecraft and satellite.

(ii) It has used in the automotive system for s control, traffic control.

(iii) It is used for decision making support system personal evaluation in the large company bus

(iv) It has application in chemical industry for contr the pH, drying, chemical distillation process.

(v) Fuzzy logic are used in Natural language proce and various intensive applications in Arti Intelligence.

(vi) Fuzzy logic are extensively used in modern co systems such as expert systems.

(vii) Fuzzy Logic is used with Neural Networks mimics how a person would make decisions, much faster. It is done by Aggregation of data changing into more meaningful data by for partial truths as fuzzy sets.

Q.7 ***Design a computer software to perform i processing to locate objects within a scene. Th fuzzy sets representing a plane and a train are***

$$Plane = \left\{ \frac{0.2}{train} + \frac{0.5}{bike} + \frac{0.3}{boat} + \frac{0.8}{plane} + \frac{0.1}{hous} \right.$$

$$Train = \left\{ \frac{1}{train} + \frac{0.2}{bike} + \frac{0.4}{boat} + \frac{0.5}{plane} + \frac{0.2}{hous} \right.$$

Find the following:

(a) $Plane \cup Train$

(b) $Plane \cap Train$

(c) $\overline{Plane}$

(d) $\overline{Train}$

(e) $Plane \mid Train$

(f) $\overline{Plane \cup Train}$;

(g) $\overline{Plane \cap Train}$;

(h) $Plane \cup \overline{Plane}$;

(i) $Plane \cap \overline{Plane}$;

(j) $Train \cup \overline{Train}$;

(k) $Train \cap \overline{Train}$;

PART-C

Q.9 ***What are fuzzy sets? Describe in detail about fuzzy set operations.***

Ans. Fuzzy Sets : Fuzzy sets may be viewed as an extension and generalization of the basic concepts of crisp sets. An important property of fuzzy set is that it allows partial membership. A fuzzy set is a set having degrees of membership between 1 and 0. The membership in a fuzzy set need not be complete, i.e., member of one fuzzy set can also be member of other fuzzy sets in the same universe. Fuzzy sets can be analogous to the thinking of intelligent people. If a person has to be classified as friend or enemy, intelligent people will not resort to absolute classification as friend or enemy. Rather, they will classify the person somewhere between two extremes of friendship and enmity. Similarly, vagueness is introduced in fuzzy set by eliminating the sharp boundaries that divide members from nonmembers in the group. There is a gradual transition between full membership and nonmembership, not abrupt transition.

A fuzzy set $\underset{\sim}{A}$ in the universe of discourse U can be defined as a set of ordered pairs and it is given by

$$\underset{\sim}{A} = \{(x, \mu_{\underset{\sim}{A}}(x)) \mid x \in U\}$$

where $\mu_{\underset{\sim}{A}}(x)$ is the degree of membership of x in $\underset{\sim}{A}$ and it indicates the degree that x belongs to $\underset{\sim}{A}$. The degree of membership $\mu_{\underset{\sim}{A}}(x)$ assumes values in the range from 0 to 1, i.e., the membership is set to unit interval [0, 1] or $\mu_{\underset{\sim}{A}}(x) \in [0, 1]$.

There are other ways of representation of fuzzy sets; all representations allow partial membership to be expressed. When the universe of discourse U is discrete and finite, fuzzy set is given as follows:

$$\underset{\sim}{A} = \left\{\frac{\mu_{\underset{\sim}{A}}(x_1)}{x_1} + \frac{\mu_{\underset{\sim}{A}}(x_2)}{x_2} + \frac{\mu_{\underset{\sim}{A}}(x_3)}{x_3} + \ldots\right\}$$

$$= \left\{\sum_{i=1}^{n} \frac{\mu_{\underset{\sim}{A}}(x_i)}{x_i}\right\}$$

where "n" is a finite value. When the universe of discourse U is continuous and infinite, fuzzy set $\underset{\sim}{A}$ is given by

$$\underset{\sim}{A} = \left\{\int \frac{\mu_{\underset{\sim}{A}}(x)}{x}\right\}$$

In the above two representations of fuzzy sets for discrete and continuous universe, the horizontal bar is not a quotient but a delimiter. The numerator in each representation is the membership value in set that is associated with the element of the universe present in the denominator. For discrete and finite universe of discourse U, the summation symbol in the representation of fuzzy set does not denote algebraic summation but indicates the collection of each element. Thus the summation sign ("+") used is not the algebraic "add" but rather it is a discrete function-theoretic union. Also, for continuous and infinite universe of discourse U, the integral sign in the representation of fuzzy set $\underset{\sim}{A}$ is not an algebraic integral but is a continuous function-theoretic union for continuous variables.

A fuzzy set is universal fuzzy set if and only if the value of the membership function is 1 for all the members under consideration. Any fuzzy set $\underset{\sim}{A}$ defined on a universe U is a subset of that universe. Two fuzzy sets $\underset{\sim}{A}$ and $\underset{\sim}{B}$ are said to be equal fuzzy sets if $\mu_{\underset{\sim}{A}}(x) = \mu_{\underset{\sim}{B}}(x)$ for all $x \in U$. A fuzzy set $\underset{\sim}{A}$ is said to be empty fuzzy set if and only if the value of the membership function is 0 for all possible members considered. The universal fuzzy set can also be called whole fuzzy set.

The collection of all fuzzy sets and fuzzy subsets on universe U is called fuzzy power set $\underset{\sim}{P}(U)$. Since all the fuzzy sets can overlap, the cardinality of the fuzzy power set, $n_{\underset{\sim}{P}(U)}$ is infinite, i.e., $n_{\underset{\sim}{P}(U)} = \infty$

On the basis of the above discussion we have

$$\underset{\sim}{A} \subseteq U \Rightarrow \mu_{\underset{\sim}{A}}(x) \le \mu_U(u)$$

Also, for all $x \in U$

$$\mu_{\phi}(x) = 0; \mu_U(x) = 1$$

1. Fuzzy Set Operations : The generalization of operations on classical sets to operations on fuzzy sets is not unique. The fuzzy set operations being discussed in this section are termed standard fuzzy set operations. These are the operations widely used in engineering applications. Let A and B be fuzzy sets in the universe of discourse U. For a given element x on the universe, the following function theoretic Operations of union, intersection and complement are defined for fuzzy sets $\underset{\sim}{A}$ and $\underset{\sim}{B}$ on U.

(i) Union : The union of fuzzy sets $\underset{\sim}{A}$ and $\underset{\sim}{B}$, denoted by $\underset{\sim}{A} \cup \underset{\sim}{B}$, is defined as

$$\mu_{\underset{\sim}{A} \cup \underset{\sim}{B}}(x) = \max[\mu_{\underset{\sim}{A}}(x), \mu_{\underset{\sim}{B}}(x)]$$

$$= \mu_{\underset{\sim}{A}}(x) \vee \mu_{\underset{\sim}{B}}(x) \qquad \text{for all } x \in U$$

where $\vee$ indicates max operation. The Venn diagram for union operation of fuzzy sets $\underset{\sim}{A}$ and $\underset{\sim}{B}$ is shown in Figure 1.

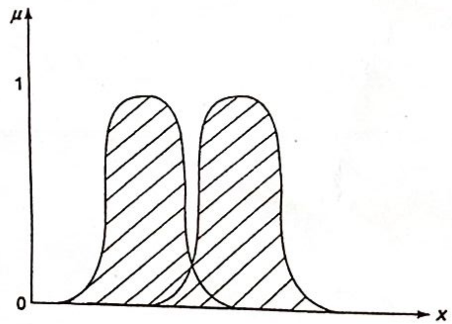


Fig. 1 : Union of fuzzy sets $\underset{\sim}{A}$ and $\underset{\sim}{B}$

(ii) Intersection : The intersection of fuzzy sets $\underset{\sim}{A}$ and $\underset{\sim}{B}$, denoted by $\underset{\sim}{A} \cap \underset{\sim}{B}$, is defined by

$$\mu_{\underset{\sim}{A} \cap \underset{\sim}{B}}(x) = \min[\mu_{\underset{\sim}{A}}(x), \mu_{\underset{\sim}{B}}(x)]$$

$$= \mu_{\underset{\sim}{A}}(x) \wedge \mu_{\underset{\sim}{B}}(x) \qquad \text{for all } x \in U$$

where $\wedge$ indicates min operator. The Venn diagram for intersection operation of fuzzy sets $\underset{\sim}{A}$ and $\underset{\sim}{B}$ is shown in Figure 2.

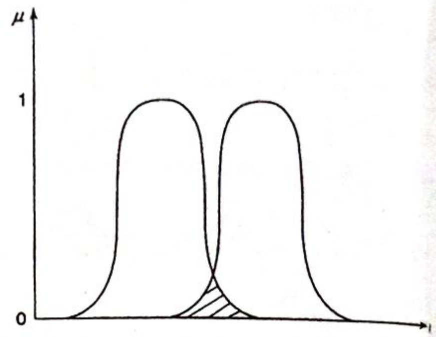


Fig. 2 : Intersection of fuzzy sets $\underset{\sim}{A}$ and $\underset{\sim}{B}$.

(iii) Complement : When $\mu_{\underset{\sim}{A}}(x) \in [0, 1]$, the complement of $\underset{\sim}{A}$, denoted as $\overline{\underset{\sim}{A}}$ is defined by

$$\mu_{\overline{\underset{\sim}{A}}}(x) = 1 - \mu_{\underset{\sim}{A}}(x) \qquad \text{for all}$$

The Venn diagram for complement operation of set $\underset{\sim}{A}$ is shown in Figure 3.

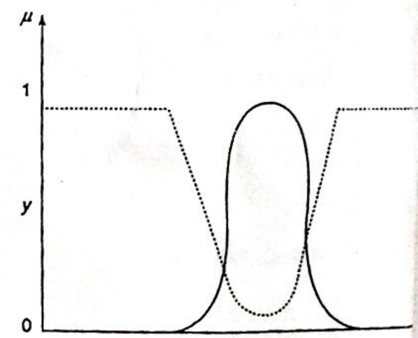


Fig. 3 : Complement of fuzzy set $\underset{\sim}{A}$

(iv) More Operations on Fuzzy Sets :

(a) Algebraic sum: The algebraic sum ($\underset{\sim}{A} + \underset{\sim}{B}$) sets, fuzzy sets $\underset{\sim}{A}$ and $\underset{\sim}{B}$ is defined as

$$\mu_{\underset{\sim}{A}+\underset{\sim}{B}}(x) = \mu_{\underset{\sim}{A}}(x) + \mu_{\underset{\sim}{B}}(x) - \mu_{\underset{\sim}{A}}(x)\mu_{\underset{\sim}{B}}(x)$$

(b) Algebraic prod
two fuzzy sets

$\mu_{\underset{\sim}{A}.\underset{\sim}{B}}(x) =$

(c) Bounded sum: Th
sets $\underset{\sim}{A}$ and $\underset{\sim}{B}$ is

$\mu_{\underset{\sim}{A} \oplus \underset{\sim}{B}}(x) =$

(d) Bounded differenc
of two fuzzy sets

$\mu_{\underset{\sim}{A} \ominus \underset{\sim}{B}}(x) =$

2. Properties of Fuzzy S
properties as crisp sets exc
and law of contradiction T

$\underset{\sim}{A} \cup \overline{\underset{\sim}{A}} \ne U;$

Frequently used prop
follows:

1. Commutativity

$\underset{\sim}{A} \cup \underset{\sim}{B} = \underset{\sim}{B} \cup$

2. Associativity

$\underset{\sim}{A} \cup (\underset{\sim}{B} \cup \underset{\sim}{C}) =$

$\underset{\sim}{A} \cap (\underset{\sim}{B} \cap \underset{\sim}{C}) =$

3. Distributivity

$\underset{\sim}{A} \cup (\underset{\sim}{B} \cap \underset{\sim}{C}) =$

$\underset{\sim}{A} \cap (\underset{\sim}{B} \cup \underset{\sim}{C}) =$

4. Idempotency

$\underset{\sim}{A} \cup \underset{\sim}{A} = \underset{\sim}{A};$

5. Identity

$\underset{\sim}{A} \cup \phi = \underset{\sim}{A}$ and

$\underset{\sim}{A} \cap \phi = \phi$ and

6. Involution (double negation

$\overline{\overline{\underset{\sim}{A}}} = \underset{\sim}{A}$

(b) Algebraic product: The algebraic product ($\underset{\sim}{A}.\underset{\sim}{B}$) of two fuzzy sets $\underset{\sim}{A}$ and $\underset{\sim}{B}$ is defined as

$$\mu_{\underset{\sim}{A}.\underset{\sim}{B}}(x) = \mu_{\underset{\sim}{A}}(x)\mu_{\underset{\sim}{B}}(x)$$

(c) Bounded sum: The bounded sum ($\underset{\sim}{A} \oplus \underset{\sim}{B}$) of two fuzzy sets $\underset{\sim}{A}$ and $\underset{\sim}{B}$ is defined as

$$\mu_{\underset{\sim}{A}\oplus\underset{\sim}{B}}(x) = \min\{1, \mu_{\underset{\sim}{A}}(x) + \mu_{\underset{\sim}{B}}(x)\}$$

(d) Bounded difference: The bounded difference ($\underset{\sim}{A} \ominus \underset{\sim}{B}$) of two fuzzy sets $\underset{\sim}{A}$ and $\underset{\sim}{B}$ is define

$$\mu_{\underset{\sim}{A}\ominus\underset{\sim}{B}}(x) = \max\{0, \mu_{\underset{\sim}{A}}(x) - \mu_{\underset{\sim}{B}}(x)\}$$

2. Properties of Fuzzy Sets : Fuzzy sets follow the same properties as crisp sets except for the law of excluded middle and law of contradiction That is, for fuzzy set $\underset{\sim}{A}$

$$\underset{\sim}{A} \cup \bar{\underset{\sim}{A}} \neq U;\ \underset{\sim}{A} \cap \bar{\underset{\sim}{A}} \neq \phi$$

Frequently used properties of fuzzy sets are given as follows:

1. Commutativity

$$\underset{\sim}{A} \cup \underset{\sim}{B} = \underset{\sim}{B} \cup \underset{\sim}{A};\quad \underset{\sim}{A} \cap \underset{\sim}{B} = \underset{\sim}{B} \cap \underset{\sim}{A}$$

2. Associativity

$$\underset{\sim}{A} \cup (\underset{\sim}{B} \cup \underset{\sim}{C}) = (\underset{\sim}{A} \cup \underset{\sim}{B}) \cup \underset{\sim}{C}$$

$$\underset{\sim}{A} \cap (\underset{\sim}{B} \cap \underset{\sim}{C}) = (\underset{\sim}{A} \cap \underset{\sim}{B}) \cap \underset{\sim}{C}$$

3. Distributivity

$$\underset{\sim}{A} \cup (\underset{\sim}{B} \cap \underset{\sim}{C}) = (\underset{\sim}{A} \cup \underset{\sim}{B}) \cap (\underset{\sim}{A} \cup \underset{\sim}{C})$$

$$\underset{\sim}{A} \cap (\underset{\sim}{B} \cup \underset{\sim}{C}) = (\underset{\sim}{A} \cap \underset{\sim}{B}) \cup (\underset{\sim}{A} \cap \underset{\sim}{C})$$

4. Idempotency

$$\underset{\sim}{A} \cup \underset{\sim}{A} = \underset{\sim}{A};\quad \underset{\sim}{A} \cap \underset{\sim}{A} = \underset{\sim}{A}$$

5. Identity

$$\underset{\sim}{A} \cup \phi = \underset{\sim}{A} \text{ and } \underset{\sim}{A} \cup U = U \text{ (universal set)}$$

$$\underset{\sim}{A} \cap \phi = \phi \text{ and } \underset{\sim}{A} \cap U = \underset{\sim}{A}$$

6. Involution (double negation)

$$\bar{\bar{\underset{\sim}{A}}} = \underset{\sim}{A}$$

7. Transitivity

If $\underset{\sim}{A} \subseteq \underset{\sim}{B} \subseteq \underset{\sim}{C}$ then $\underset{\sim}{A} \subseteq \underset{\sim}{C}$

8. De Morgan's law

$$\overline{\underset{\sim}{A} \cup \underset{\sim}{B}} = \bar{\underset{\sim}{A}} \cap \bar{\underset{\sim}{B}},\ \overline{\underset{\sim}{A} \cap \underset{\sim}{B}} = \bar{\underset{\sim}{A}} \cup \bar{\underset{\sim}{B}}$$

Q.10 *Describe in detail about fuzzy relation.*

Ans. Fuzzy Relation : Fuzzy relations relate elements of one universe (say X) to those of another universe (say Y) through the Cartesian product of the two universes. These can also be referred to as fuzzy sets defined on universal sets, which are Cartesian products. A fuzzy relation is based on the concept that everything is related to some extent or unrelated.

A fuzzy relation is a fuzzy set defined on the Cartesian product of classical sets $\{X_1, X_2, ..., X_n\}$ where tuples $(x_1, x_2, ..., x_n)$ may have varying degrees of membership $\mu_R(x_1, x_2, ..., x_n)$ within the relation. That is,

$$R(X_1, X_2, ..., X_n) = \int_{X_1 \times X_2 \times ... \times X_n} \mu_R(x_1, x_2 ... x_n) | (x_1, x_2 ... x_n),\quad x_i \in X_i$$

A fuzzy relation between two sets X and Y is called binary fuzzy relation and is denoted by R(X, Y). A binary relation R(X, Y) is referred to as bipartite graph when $X \neq Y$. The binary relation on a single set X is called directed graph or digraph. This relation occurs when X = Y and is denoted as R(X, X) or $R(X^2)$.

Let

$$\underset{\sim}{X} = \{x_1, x_2 ... x_n\} \text{ and } \underset{\sim}{Y} = \{y_1, y_2 ... y_n\}$$

Fuzzy relation $\underset{\sim}{R} = (\underset{\sim}{X}, \underset{\sim}{Y})$ can be expressed by an n × m matrix as follows:

$$R(X, Y) = \begin{bmatrix} \mu_R(x_1, y_1) & \mu_R(x_1, y_2) & \mu_R(x_1, y_m) \\ \mu_R(x_2, y_1) & \mu_R(x_2, y_2) & \mu_R(x_2, y_m) \\ \vdots & \vdots & \vdots \\ \mu_R(x_n, y_1) & \mu_R(x_n, y_2) & \mu_R(x_n, y_m) \end{bmatrix}$$

The matrix representing a fuzzy relation is called fuzzy matrix. A fuzzy relation $\underset{\sim}{R}$ is a mapping from Cartesian space X × Y to the interval [0, 1] where the mapping strength is expressed by the membership function of the relation for ordered pairs from the two universes [$\mu_R(x, y)$].

A fuzzy graph is a graphical representation of a binary fuzzy relation. Each element in $\underset{\sim}{X}$ and $\underset{\sim}{Y}$ corresponds to a node in the fuzzy graph. The connection links are established between the nodes by the elements of $\underset{\sim}{X} \times \underset{\sim}{Y}$ with non-zero membership grades in $\underset{\sim}{R}(\underset{\sim}{X}, \underset{\sim}{Y})$. The links may also be present in the form of arcs. These links are labeled with the membership values as $\mu_R(x_i, y_j)$ When $X \neq Y$, the link connecting the two nodes is an undirected binary graph called bipartite graph. Here, each of the sets X and Y can be represented by a set of nodes such that the nodes corresponding to one set are clearly differentiated from the nodes representing the other set. When X = Y, a node is connected to itself and directed links are used; in such a case, the fuzzy graph is called directed graph. Here, only one set of nodes corresponding to set X is used.

The domain of a binary fuzzy relation R(X, Y) is the fuzzy set, dom R(X, Y), having the membership function as

$$\mu_{\text{domainR}}(x) = \max_{y \in Y} \mu_R(x, y)\quad \forall x \in X$$

The range of a binary fuzzy relation R(X, Y) is the fuzzy set, ran R(X, Y), having the membership function as

$$\mu_{\text{RangeR}}(y) = \max_{x \in X} \mu_R(x, y)\quad \forall y \in Y$$

Consider a universe $X = \{x_1, x_2, x_3, x_4\}$ and the binary fuzzy relation on X as

	x_1	x_2	x_3	x_4
x_1	0.2	0	0.5	0
x_2	0	0.3	0.7	0.8
x_3	0.1	0	0.4	0
x_4	0	0.6	0	1

(R(X, X) =)

The bipartite graph and simple fuzzy graph of $\underset{\sim}{R}$ (X, X) is shown in Figure (1(A)) and (1(B)), respectively.

Let $\underset{\sim}{X} = (x_1, x_2, x_3, x_4)$ and $\underset{\sim}{Y} = (y_1, y_2, y_3, y_4)$

Let $\underset{\sim}{R}$ be a relation from $\underset{\sim}{X}$ to $\underset{\sim}{Y}$ given by

$$\underset{\sim}{R} = \frac{0.2}{(x_1, y_1)} + \frac{0.4}{(x_1, y_2)} + \frac{0.1}{(x_2, y_2)} + \frac{0.6}{(x_2, y_3)} + \frac{1.0}{(x_3, y_3)} + \frac{0.5}{(x_3, y_1)}$$

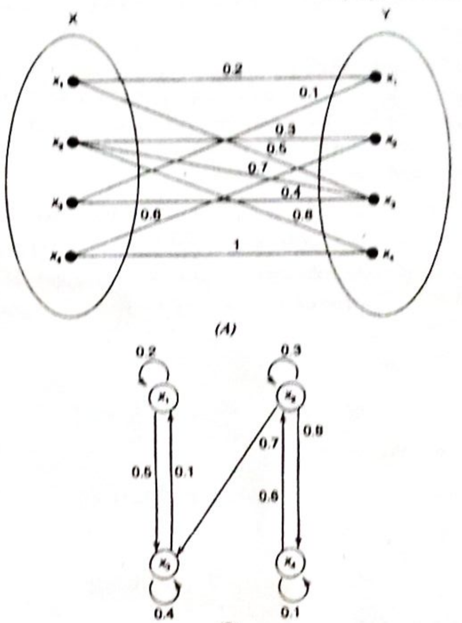


Fig. 1 : Graphical representation of fuzzy relations: (A) Bipartite graph; (B) simple fuzzy graph.

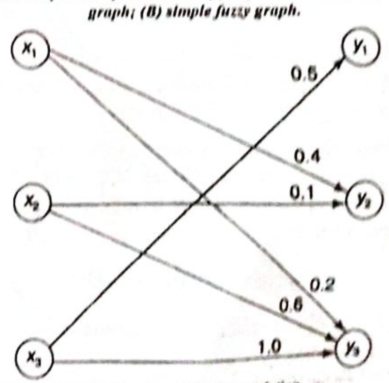


Fig. 2 : Graph of fuzzy relation

The corresponding Fuzzy matrix for relation $\underset{\sim}{R}$ is

$$\underset{\sim}{R} = \begin{array}{c} \\ x_1 \\ x_2 \\ x_3 \end{array}\begin{array}{c} \begin{array}{ccc} y_1 & y_2 & y_3 \end{array} \\ \begin{bmatrix} 0 & 0.4 & 0.2 \\ 0 & 0.1 & 0.6 \\ 0.5 & 0 & 1.0 \end{bmatrix} \end{array}$$

The graph of the above relation $\underset{\sim}{R}(\underset{\sim}{X} \times \underset{\sim}{Y})$ if is shown in Figure 2.

1. Cardinality of Fuzzy Relations : The cardinality of fuzzy sets on any universe is infinity; hence the cardinality of a fuzzy relation between two or more universes is also infinity. This is mainly a result of the occurrence of partial membership in fuzzy sets and fuzzy relations.

2. Operations on Fuzzy Relations : The basic operations on fuzzy sets also apply on fuzzy relations. Let $\underset{\sim}{R}$ and $\underset{\sim}{S}$ be fuzzy relations on the Cartesian space X × Y. The operations that can be performed on these fuzzy relations are described below:

(i) Union

$$\mu_{\underset{\sim}{R} \cup \underset{\sim}{S}}(x,y) = \max[\mu_{\underset{\sim}{R}}(x,y), \mu_{\underset{\sim}{S}}(x,y)]$$

(ii) Intersection

$$\mu_{\underset{\sim}{R} \cap \underset{\sim}{S}}(x,y) = \min[\mu_{\underset{\sim}{R}}(x,y), \mu_{\underset{\sim}{S}}(x,y)]$$

(iii) Complement

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

(iv) Containment

$$\underset{\sim}{R} \subset \underset{\sim}{S} \Rightarrow \mu_{\underset{\sim}{R}}(x,y) \le \mu_{\underset{\sim}{S}}(x,y)$$

(v) **Inverse:** The inverse of a fuzzy relation R on X × Y is denoted by R^{-1}. It is a relation on Y × X defined by $R^{-1}(y, x) = R(x, y)$ for all pairs $(y, x) \in Y \times X$.

(vi) **Projection:** For a fuzzy relation R(X, Y), let [R ↓ Y] denote the projection of R onto Y. Then [R ↓ Y] is a **fuzzy relation** in Y whose **membership function** is **defined by**

$$\mu_{[R \downarrow Y]}(x,y) = \max \mu_{\underset{\sim}{R}}(x,y)$$

The projection concept can be extended to an n–ary relation $R(x_1, x_2, \ldots, x_n)$

3. Properties of Fuzzy Relations : Like classical relations, the properties of commutativity, associativity, distributivity, idempotency and identity also hold good for fuzzy relations. De Morgan's laws hold good for fuzzy relations as they do for classical relations. The null relation ϕ_R and complete relation E_R are analogous to the null set ϕ and the whole set $\underset{\sim}{E}$, respectively, in set theoretic form. The excluded middle laws are not satisfied in fuzzy relations as for fuzzy sets. This is because a fuzzy relation $\underset{\sim}{R}$ is also a fuzzy set, and there exists an overlap between a relation and its complement. Hence

$$\underset{\sim}{R} \cup \bar{\underset{\sim}{R}} \ne \underset{\sim}{E} \text{ (whole set)}$$

$$\underset{\sim}{R} \cap \bar{\underset{\sim}{R}} \ne \phi \text{ (null set)}$$

4. Fuzzy Composition : Before understanding the fuzzy composition techniques, let us learn about the fuzzy Cartesian product rule. Let $\underset{\sim}{A}$ be a fuzzy set on universe X and $\underset{\sim}{B}$ be a fuzzy set on universe Y. The Cartesian product over $\underset{\sim}{A}$ and $\underset{\sim}{B}$ results in fuzzy relation $\underset{\sim}{R}$, which is contained within the entire (complete) Cartesian space, i.e.,

$$\underset{\sim}{A} \times \underset{\sim}{B} = \underset{\sim}{R}$$

where $\underset{\sim}{R} \subset X \times Y$

The membership function of fuzzy relation is given by

$$\mu_{\underset{\sim}{R}}(x,y) = \mu_{\underset{\sim}{A} \times \underset{\sim}{B}}(x,y) = \min[\mu_{\underset{\sim}{A}}(x)\mu_{\underset{\sim}{B}}(y)]$$

The Cartesian product is not an operation similar to arithmetic product. Cartesian product $\underset{\sim}{R} = \underset{\sim}{A} \times \underset{\sim}{B}$ is obtained in the same way as the cross-product of two vectors. For example, for a fuzzy set $\underset{\sim}{A}$ that has three elements (hence column vector of size 3 × 1) and a fuzzy set $\underset{\sim}{B}$ that has four elements (hence row vector of size 1 × 4), the resulting fuzzy relation $\underset{\sim}{R}$ will be represented by a matrix of size 3 × 4, i.e., $\underset{\sim}{R}$ will have three rows and four columns.

Now let's discuss the composition of fuzzy relations. There are two types of fuzzy composition techniques:

1. Fuzzy max-min composition
2. Fuzzy max-product composition

There also exists fuzzy min-max composition method, but the most commonly used technique is fuzzy max-min composition. Let $\underset{\sim}{R}$ be fuzzy relation on Cartesian space X × Y, and $\underset{\sim}{S}$ be fuzzy relation on Cartesian space Y × Z.

The max-min composition of R(X, Y) and S(Y, Z), denoted by R(X, Y) ∘ S(Y, Z) is defined by T(X, Z) as

$$\mu_{\underset{\sim}{R} \circ \underset{\sim}{S}}(x,z) = \max_{y \in Y}\{\min[\mu_{\underset{\sim}{R}}(x,y), \mu_{\underset{\sim}{S}}(x,y)]\}$$

$$= \bigvee_{y \in Y}[\mu_{\underset{\sim}{R}}(x,y) \wedge \mu_{\underset{\sim}{S}}(x,y)]$$

The min-max composition of R(X, Y) and S(Y, Z), denoted as R(X, Y) ∘ S(Y, Z), is defined by T(X, Z) as

$$\mu_T(x,z) = \mu_{\underset{\sim}{R} \circ \underset{\sim}{S}}(x,z)$$

$$= \min_{y \in Y}\{\max[\mu_{\underset{\sim}{R}}(x,y), \mu_{\underset{\sim}{S}}(y,z)]\}$$

$$= \bigwedge_{y \in Y}[\mu_{\underset{\sim}{R}}(x,y) \vee \mu_{\underset{\sim}{S}}(y,z)] \quad \forall x \in X, z \in Z$$

From the above definitions it can be noted that

$$\overline{R(X,Y)} \cdot \overline{S(Y,Z)} = \overline{R(X,Y) \circ S(Y,Z)}$$

The max-min composition is most widely used, hence the problems discussed in this chapter are limited to max-min composition. The max-product composition of R(X, Y) and S(Y, D,) denoted as R(X, Y) . S(Y, Z), is defined by T(X, Z) as

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

$$= \max_{y \in Y}[\mu_{\underset{\sim}{R}}(x,y) \circ \mu_{\underset{\sim}{S}}(y,z)]$$

$$= \bigvee_{y \in Y}[\mu_{\underset{\sim}{R}}(x,y) . \mu_{\underset{\sim}{S}}(y,z)]$$

The properties of fuzzy composition can be given as follows:

$$R \circ S \ne S \circ R$$

$$(\underset{\sim}{R} \circ \underset{\sim}{S})^{-1} = \underset{\sim}{S}^{-1} \circ \underset{\sim}{R}^{-1}$$

$$(\underset{\sim}{R} \circ \underset{\sim}{S}) \circ \underset{\sim}{M} = \underset{\sim}{R} \circ (\underset{\sim}{S} \circ \underset{\sim}{M})$$

Q.4 *Write a short not on individual decision making.*

Ans. Individual Decision Making : A decision-making model is characterized by the following:

1. set of possible actions;
2. set of goals $G_i(i \in X_n)$;
3. set of constraints $C_j(j \in X_m)$.

The goals and constraints are expressed in terms of fuzzy sets. These fuzzy sets in individual decision making are not defined directly on the set of actions, but by means of other sets that characterize relevant states of nature. Consider a set A. Then the goal and constraint for this set are given by

$$G_i(a) = \text{Composition}[G_i(a)] = G_i^1(G_i(a)) \text{ with } G_i^1$$

$$C_j(a) = \text{Composition}[C_j(a)] = C_j^1(C_j(a)) \text{ with } C_j^1$$

for $a \in A$. The fuzzy decision in this case is given by

$$F_D = \min[\inf_{i \in X_n} G_i(a), \inf_{j \in X_m} C_j(a)]$$

Q.5 *Why is use fuzzy logic in control systems?*

Ans. Use of Fuzzy Logic in Control Systems : 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 :

1. 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.
2. By applying fuzzy logic for control we can utilize the human expertise and experience for designing a controller.
3. The fuzzy control rules, basically the IF-THEN rules, can be be st uti'ized in designing a controller.

PART-B

Q.6 *Explain the fuzzy decision making.*

Ans. Fuzzy Decision Making : Decision making is a very important social, economical and scientific endeavor. Decision-making activities are the steps taken to choose a suitable alternative from those that are needed for realizing a certain goal. The decision-making process involves three steps:

1. Determining the set of alternatives;
2. Evaluating alternatives;
3. Comparison between alternatives.

In any decision process, the information about the outcome is considered and a suitable path has to be chosen from two or more alternatives for subsequent action; when good decisions are made, good output is expected. If a decision is made under certainty, then the outcome for each process can be determined precisely; one should note that whenever decision is made, it is under risk condition. The prime domain for fuzzy decision making is the existing uncertainty. There are several situations under the decision-making process. There may be situations when even though decisions made are good, the output may be adverse or vice-versa. When good decisions are made continuously for a longer period, advantageous situations may prevail.

When there are several objectives to be realized in making a decision, the decision making is called multiobjective decision making. The knowledge of experts becomes very essential when decision making is very tedious. The information may be available for the following: the possible outcomes, change in conditions with respect to time about value of new information, when the priority for each action is typically ambiguous, vague and otherwise fuzzy. Obtaining an evaluation structure for selecting alternatives and establishing selection standards are very important stages. The evaluation of alternatives may be carried out based on several attributes of the object; such a decision making is called multiattribute decision making.

Q.7 *What is multiperson decision making?*

Ans. Multiperson Decision Making : Decision making in this case includes several persons. The expert knowledge from various persons is utilized to make decisions. The difference between the individual decision making and multiperson decision making is: The goals of individual decision makers differ, i.e., each places a different ordering arrangement. On the other hand, in multiperson decision making, the decision makers have access to different information upon which to base their decision.

Here, each member of a group of "n" individual decision makers has a preference ordering PO_k, $k \in x_n$, which totally or partially orders a set X. A social choice (sc) function has to be found, given the individual preference ordering. The fuzzy relation for a social choice preference function is given by

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

which has a membership of SC (X_i, X_j), which indicates the preference of alternative X_i over X_j. Let

Number of persons preferring X_i to $X_j = N(X_i, X_j)$

Total number of decision makers = n

Then,

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

The mulriperson decision making is also given by

$$SC(x_i, x_j) = \begin{cases} 1 & \text{if } x_i >_k x_j \text{ for some k} \\ 0 & \text{otherwise} \end{cases}$$

Q.8 ***Write the applications of fuzzy logic.***

Ans. Applications of Fuzzy Logic : Fuzzy logic is used in following areas :

1. **Aerospace :** In aerospace, fuzzy logic is used in the following areas :
   - (i) Altitude control of spacecraft
   - (ii) Satellite altitude control
   - (iii) Flow and mixture regulation in aircraft de-icing vehicles
2. **Automotive :** In automotive, fuzzy logic is used in the following areas :
   - (i) Trainable fuzzy systems for idle speed control
   - (ii) Shift scheduling method for automatic transmission
   - (iii) Intelligent highway systems
   - (iv) Traffic control
   - (v) Improving efficiency of automatic transmissions
3. **Business :** In business, fuzzy logic is used in the following areas :
   - (i) Decision-making support systems
   - (ii) Personnel evaluation in a large company

4. **Defence :** In defence, fuzzy logic is used in the following areas :
   - (i) Underwater target recognition
   - (ii) Automatic target recognition of thermal infrared images
   - (iii) Naval decision support aids
   - (iv) Control of a hypervelocity interceptor
   - (v) Fuzzy set modelling of NATO decision making
5. **Electronics :** In electronics, fuzzy logic is used in the following areas :
   - (i) Control of automatic exposure in video cameras
   - (ii) Humidity in a clean room
   - (iii) Air conditioning systems
   - (iv) Washing machine timing
   - (v) Microwave ovens
   - (vi) Vacuum cleaners
6. **Finance :** In the finance field, fuzzy logic is used in the following areas :
   - (i) Banknote transfer control
   - (ii) Fund management
   - (iii) Stock market predictions
7. **Industrial Sector :** In industrial sector, fuzzy logic is used in following areas :
   - (i) Cement kiln controls heat exchanger control
   - (ii) Activated sludge wastewater treatment process control
   - (iii) Water purification plant control
   - (iv) Quantitative pattern analysis for industrial quality assurance
   - (v) Control of constraint satisfaction problems in structural design
   - (vi) Control of water purification plants
8. **Manufacturing :** In the manufacturing industry, fuzzy logic is used in following areas :
   - (i) Optimization of cheese production
   - (ii) Optimization of milk production

9. **Marine :** In the marine field, fuzzy logic is used in the following areas :
   - (i) Autopilot for ships
   - (ii) Optimal route selection
   - (iii) Control of autonomous underwater vehicles
   - (iv) Ship steering
10. **Medical :** In the medical field, fuzzy logic is used in the following areas :
    - (i) Medical diagnostic support system
    - (ii) Control of arterial pressure during anaesthesia
    - (iii) Multivariable control of anaesthesia
    - (iv) Modelling of neuropath logical findings in Alzheimer's patients
    - (v) Radiology diagnoses
    - (vi) Fuzzy inference diagnosis of diabetes and prostate cancer
11. **Securities :** In securities, fuzzy logic is used in following areas :
    - (i) Decision systems for securities trading
    - (ii) Various security appliances
12. **Transportation :** In transportation, fuzzy logic is used in the following areas :
    - (i) Automatic underground train operation
    - (ii) Train schedule control
    - (iii) Railway acceleration
    - (iv) Braking and stopping
13. **Pattern Recognition and Classification :** In Pattern Recognition and Classification, fuzzy logic is used in the following areas :
    - (i) Fuzzy logic based speech recognition
    - (ii) Fuzzy logic based
    - (iii) Handwriting recognition
    - (iv) Fuzzy logic based facial characteristic analysis
    - (v) Command analysis
    - (vi) Fuzzy image search
14. **Psychology :** In Psychology, fuzzy logic is used in following areas :
    - (i) Fuzzy logic based analysis of human behaviour
    - (ii) Criminal investigation and prevention based on fuzzy logic reasoning

Q.9 ***Explain the architecture and operation of fuzzy logic control system in detail.***

Ans. The basic architecture of a fuzzy logic controller is shown in figure. The principal components of an FLC system are: a fuzzifier, a fuzzy rule base, a fuzzy knowledge base, an inference engine and a defuzzifier. It also includes parameters for normalization. When the output from the defuzzifier is not a control action for a plant, then the system is a fuzzy logic decision system. The fuzzifier present converts the crisp quantities into fuzzy quantities. The fuzzy rule base stores the knowledge about the operation of the process of domain expertise. The fuzzy knowledge base stores the knowledge about all the input-output fuzzy relationships. It includes the membership functions defining the input variables to the fuzzy rule base and the output ariables to the plant under control. The inference engine is the kernel of an FLC system, and it possess the capability to simulate human decisions by performing approximate reasoning to achieve a desired control strategy. The defuzzifier converts the fuzzy quantities into crisp quantities from an inferred fuzzy control action by the inference engine.

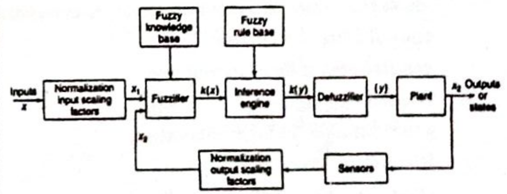


Fig. : Basic architecture of an FLC system.

The various steps involved in designing a fuzzy logic controller are as follows:

Step 1: Locate the input, output and state variables of the plant under consideration.

Step 2: Split the complete universe of discourse spanned by each variable into a number of fuzzy subsets, assigning each with a linguistic label. The subsets include all the elements in the universe.

Step 3: Obtain the membership function for each fuzzy subset.

Step 4: Assign the fuzzy relationships between the inputs or states of fuzzy subsets on one side and the outputs of fuzzy subsets on other side, thereby forming the rule base.

Step 5: Choose appropriate scaling factors for the input and output variables for normalizing the variables between [0, 1] and [– 1, 1] interval.

Step 6: Carry out the fuzzification process.

Step 7: Identify the output contributed from each rule using fuzzy approximate reasoning.

Step 8: Combine the fuzzy outputs obtained from each rule.

Step 9: Finally, apply defuzzification to form a crisp output.

The above steps are performed and executed for a simple FLC system. The following design elements are adopted form designing a general FLC system:

1. Fuzzification strategies and the interpretation of a fuzzifier.
2. Fuzzy knowledge base:
   normalization of the parameters involved;
   partitioning of input and output spaces;
   selection of membership functions of a primary fuzzy set.
3. Fuzzy rule base:
   selection of input and output variables;
   source from which fuzzy control rules are to be derived;
   types of fuzzy control rules;
   completeness of fuzzy control rules.
4. Decision-making logic:
   proper definition of fuzzy implication;
   interpretation of connective "and";
   interpretation of connective "or";
   inference engine.
5. Defuzzification strategies and the interpretation of a defuzzifier.

   When all the above five design parameters are fixed, the FLC system is simple. Based on all this, the features of a simple FLC system are as follows:

- fixed and uniform input and output scaling factors for normalization;
- fixed and noninteractive rules;
- fixed membership functions;
- only limited number of rules, which increases exponentially with the number of input variables;
- fixed expertise knowledge;
- no hierarchical rule structure and low-level control.

constructed. The look-up table comprises the information about the control surface which can be downloaded into a read-only memory chip This chip would constitute a fixed controller for the plant.

PART-C

Q.12 *Explain in detail about multiobjective decision making.*

Ans. Multiobjective Decision Making : In making a decision when there are several objectives to be realized, then the decision making is called multiobjective decision making. Many decision processes may be based on single objectives such as cost minimization, time consumption, profit maximization and so on. However, if all the above-mentioned objectives are to be considered for decision-making process, then it becomes multiobjective decision making. The main issues in multiobjective decision making are:

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

Multiobjective decision making involves selection of one alternative a_i, from universe of alternatives A given a collection of objectives {0} that are important for a decision maker. It is necessary to evaluate how best each alternative satisfies each objective. The main aim here is to combine the weighted objectives into an overall decision function in some way. The decision function represents a mapping of alternatives in A to an ordinal set of ranks. In order to make suitable decisions, the process needs to weigh the relative importance of each objective.

Let us define a universe of n alternatives as

$$A = \{a_1, a_2, ... a_i, ... a_n\}$$

and a set of "m" objectives as

$$O = \{o_1, o_2, ... o_i, ... o_m\}$$

where o_i indicates the i^{th} objective. The degree of membership of alternative a in o_i, denoted $\mu o_i(a)$, is the degree to which alternative a satisfies the criteria mentioned for this objective. A decision function is formed, which simultaneously satisfies all the decision objectives. As a result, the decision function, DF is given by the intersection of all the set of objectives, i.e.,

$$DF = o_1 \cap o_2 \cap ... \cap o_i \cap ... \cap o_m$$

The grade of membership that DF has for each alternative a is defined by

$$m_{DF}(a) = \min[\mu o_1(a), \mu o_2(a) ... \mu o_i(a) ... \mu o_m(a)]$$

The optimal decision, a*, will then be the alternative that satisfies the equation

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

Let {P} be the set of preferences – linear and ordinal. The element of the preference set will possess linguistic values or will have values in the interval [0, 1], or in the intervals [– 1, 1}, [1, 101, etc. The preferences are attached to each of the objectives in order to notify the decision maker about the influence that each objective should possess on the chosen alternative. The preference set P contains the parameters b_i, i = 1 to m, i.e.,

$$\{P\} = \{b_1, b_2, ... b_i ... b_n\}$$

Thus for each object, we have a measure as to say how important it is to the decision maker for a given decision. The decision function is then defined by decision measure (DM), which involves objectives and preferences. The intersection of m-tuples of DM gives the decision function:

$$DM = DM(o_i, b_i) \rightarrow DM\ (\text{objective, preference})$$

$$DF = DM(o_1, b_1) \wedge DM(o_2, b_2) \wedge ... \wedge DM(o_i, b_i) \wedge ... \wedge DM(o_m, b_m)$$

The DM for a particular alternative, a_i, is given by

$$DM(o_i(a), b_i) = b_i \rightarrow o_i(a) = \bar{b}_i \vee o_i(a)$$

where $\bar{b}_i = 1 - b_i$ and $b_i = o_i$ indicates a distinct relationship between a preference and its corresponding objective. Neverthless, several objectives can have the same preferences weighting in a cardinal sense; however, they will be distinct in an ordinal sense, even though the equality situation $b_i = b_j$ for $i \neq j$ can exist for certain objectives. A joint intersection of "m" decision measures will give an appropriate decision model:

$$DF = \prod_{i=1}^{m} (\bar{b}_i \cup o_i)$$

The optimal solution, a*, is the alternative that maximizes the decision function. When we define

$$C_i = \bar{b}_i \cup o_i$$

$$\mu_{C_i}(a) = \max[\mu_{\bar{b}_i}(a), \mu_{o_i}(a)]$$

the optimal solution in membership form is given by

$$\mu_{DF}(a^*) = \max_{a \in A}\{\min[\mu_{C_1}(a), \mu_{C_2}(a), ... \mu_{C_i}(a), ..., \mu_{C_m}(a)]\}$$

When i^{th} objective becomes very important in the final decision, b_i increases, so $\bar{b}_i$ tends to decrease. As a result $C_i(a)$ decreases, thereby increasing the likelihood that $C_i(a) - o_i(a)$, where $o_i(a)$ at present will be the value of the decision function, is DF, denoting alternative a. When this process is repeated for several alternatives a, the largest value $o_i(a)$ for other alternatives will automatically result in the choice of the optimum solution, a*. The multiobjective decision making process works in this manner.

Q.13 *Explain the robotic control with fuzzy logic in detail.*

Ans. Robotic Control with Fuzzy Logic: The ultrasonic sensors provide distance information between the robot and obstacles for behaviour control of the mobile robot, while the vision system identifies some sub goals for determining a good motion direction to avoid robot trap in local region. If a mobile robot moves in unknown environments to reach a specified target without collisions with obstacles, sensors must be used to acquire information about the real world. Using such information, it is very difficult to build a precise world model in real-time for preplanning a collision-free path. On the basis of situational reactive behaviours, behaviour based control has been proposed for robot navigation. Since this method does not need building an entire world model and complex reasoning process, it is suitable for robot control in dynamic environments. A key issue in behaviour based control is how to coordinate conflicts and competitions among multiple reactive behaviours efficiently. The example in Fig.1 shows that the robot must efficiently weight multiple reactive behaviours, such as avoiding obstacle, following edge, and moving to target and so on, according to range information, when it reaches a target inside a U-shaped object. The usual approach for implementing behaviour control is artificial potential fields.

A drawback to this approach is that during pre programming much effort must be made to test and to adjust some thresholds regarding potential fields for avoiding obstacle, wandering, and moving to target and so on. In particular, these thresholds frequently depend on environments. Unlike behaviour control based on artificial potential fields, this method is to compute weights of multiple reactive behaviours in dynamic environments by a fuzzy logic algorithm rather than simply to inhibit some reactive behaviour with lower levels. The ultrasonic sensors provide distance information between the robot and obstacles for robot navigation by reactive behaviours, such as avoiding obstacles and following edges, while the vision system identifies some sub goals for determining a good motion direction to avoid robot trap in local region. This method differs from the fuzzy control approaches for obstacle avoidance. Since perception and decision units in this method are integrated in one module by the use of the idea of reactive behaviors and are directly oriented to a dynamic environment, this strategy has the better real-time response and reliability. To demonstrate the effectiveness and the robustness of the proposed strategy, we report a lot of simulation results on robot navigation in uncertain environments, such as avoiding obstacle in real-time, decelerating at curved and narrow roads, escaping from a U shaped object and moving to target and so on.

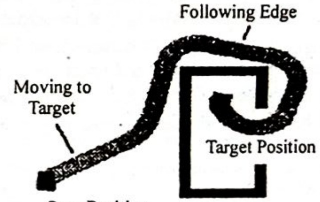


Fig. 1 : Robot motion to reach a target

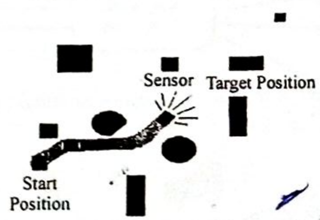


Fig. 2 : Robot motion inside a U-shape object

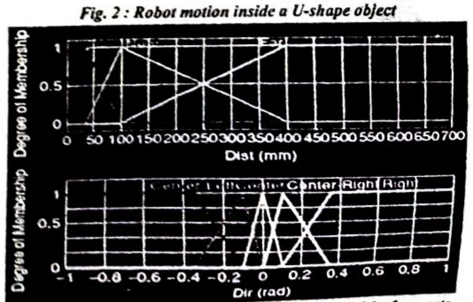


Fig. 3 : This graph shows the membership grades of the fuzzy sets distance and direction

In order to acquire information about dynamic environments, 15 ultrasonic sensors are mounted on the THMR-11 mobile robot. The sonar reflection from a sensor i represents the distance di, measured by the sensor i, between the robot and obstacles in the real world.

These ultrasonic sensors are divided into three groups to detect obstacles to the right (sensor i = 1,....., 6), front (sensor i = 7, ..., 9), and left locations (sensor i = 10,..., 15). Using such information, obviously, it is difficult to build a precise and entire world model in real-time for preplanning a collision-free path. Here, we use the sonar data di (i = 1, ..., 15) to build a simple model for representation of the distances between the robot and obstacles in the real world as follows:

$$\text{Right- obs} = \text{Min}\ \{di\}\ i = 1, ..., 6 \quad ...(1)$$

$$\text{Front- obs} = \text{Min}\ \{di\}\ i = 7, ..., 9 \quad ...(2)$$

$$\text{Left-obs} = \text{Min}\ \{di\}\ i = 10,...., 15 \quad ...(3)$$

Where the minimum values, right-obs, front-obs, and left-obs, derived from the sensor data di (i = 1, ...,15), express the distances between the robot and obstacles to the right, front, and left locations, respectively. The mobile robot is equipped with two wheel encode units to determine its current coordinates.

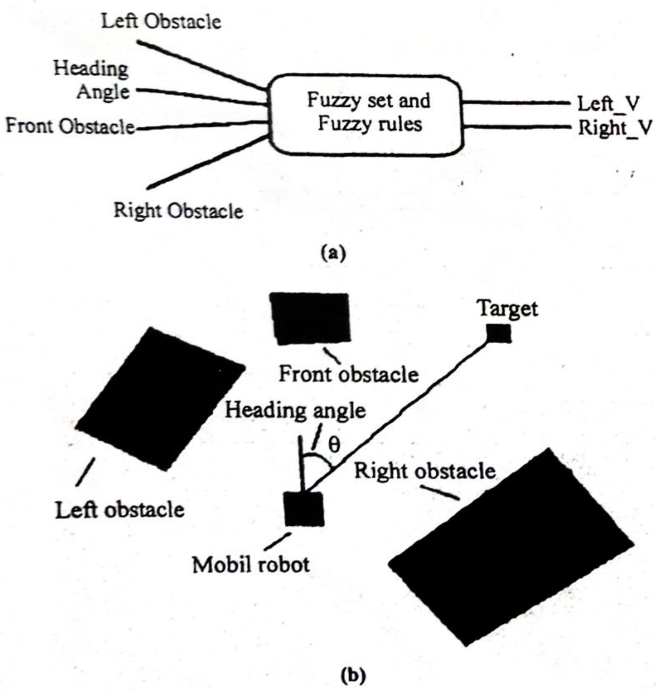


Fig. 4 : Fuzzy logic scheme for perception-action behavior control

At a start position, a counter is reset to zero. When the robot moves, its current coordinates can be roughly computed by counting the numbers of pulses from the wheel encodes that are attached on driving motors.

The THMR-I1 mobile robot with 1.0 m length and 0.8 m width is equipped with two driving wheels and one driven wheel. The velocities of the driving wheels are controlled by a motor drive unit.

The input signals to fuzzy logic scheme are the distances between the robot and obstacles to the left, front, and right locations as well as the heading angle between the robot and a specified target, denoted by left-obs, front-obs, right-obs and head-ang, respectively, as shown in figure. When the target is located to the left side of the mobile robot, a heading angle head-ang is defined as negative; while the target is located to the right side of the mobile robot, a heading angle head-ang is defined as positive, as shown in figure. According to acquired range information, reactive behaviors are weighted by the fuzzy logic algorithm to control the velocities of the two driving wheels of the robot, denoted by left-v and right-v, respectively. The linguistic variables far, med (medium) and near are chosen to fuzzify left-obs, front-obs and right-obs. The linguistic variables P (positive), 2 (zero) and N (negative) are used to fuzzify head-ang; the linguistic variables fast, med, and slow are used to fuzzy the velocities of the driving wheels left-v and right-v. In analogy to artificial potential fields, the distances between the robot and obstacles serve as a repulsive force for avoiding obstacle, while the heading angle serves as an attractive force for moving to target.

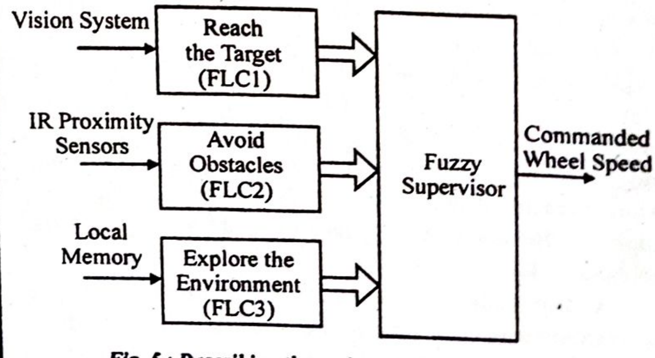


Fig. 5 : Describing the various behaviors for robot

(A) Reactive Behaviors using Fuzzy Logic

In order to reach a specified target in a complex environment, the mobile robot at least needs the following reactive behaviors:

ata point in the

rmation universe
n approach for
quivalents for the
d utility and the
zy event E_t,

n be determined as

□□□

3 ARTIFICIAL NEURAL NETWORKS

IMPORTANT QUESTIONS

PART-A

Q.1 ***What are the different types of activation functions?*** *[R.T.U. 2016]*

Ans. Types of Activation Functions : Artificial Neural Networks functions are of three types :

(i) Linear

(ii) Threshold

(iii) Sigmoid

(i) Linear Activation Function : In this function output activity is proportional to the total weighted output.

(ii) Threshold Units : In this the output is set at one of two levels but depends upon the total input is greater than or less than threshold value.

(iii) Sigmoid Unit : The output varies continuously but not linearly as the input change this unit hear a greater resemblance to real neurons then do linear or threshold untis.

Q.2 ***What are artificial neural networks?***

Ans. Artificial Neural Networks (ANNs) : The terminology of ANNs has developed from a biological model of the brain. A neural net (NN) consists of a set of connected cells as the neurons. The neurons receive impluses from either cells or other neurons or the output cells. The NNs are built from layer of neurons connected so that one layer receive inpu from the preceding layer of neurons and passes the output o the subsquent layer.

Q.3 ***Write the applications of artificial neura networks(ANNs).***

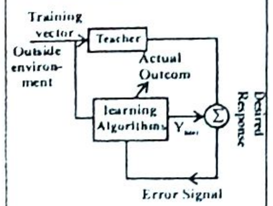
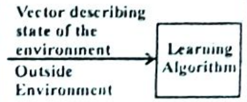
Ans. Applications of ANNs :

1. Signal processing, e.g. adaptive echo cancellation
2. Pattern recognition, e.g. character recognition
3. Speech synthesis (e.g. text-to-speech) and recognitior
4. Forecasting and prediction
5. Control and automation (neuro-controllers) e.g. broom balancing
6. Radar interpretation
7. Interpreting brain scans
8. Stock market prediction
9. Associative memory
10. Optimization

PART-B

Q.4 ***What is difference between supervised an unsupervised learning?*** *[R.T.U. 2016]*

Ans. Difference between Supervised and Unsupervised Learning :

S.N.	Supervised learning	Unsupervised learning
1.	Feedback paths are used	Feedback paths are not used
2.	Presence of teacher, input and outputs are defined	No teacher present. Only inputs are defined.
3.	Use pattern class information of each training pattern.	Adaptively clusters pattern to generate decision class without any prior pattern class information.
4.	Global error signal govern learning.	Local information is used for learning.
5.	Learning is usually off line.	Locality allows synapses to learn in real time.
6.		

Q.5 ***What is activation functions and discuss the importance of activation function in learning of neural network?*** *[R.T.U 2012]*

OR

Explain different types of activation functions of artificial neural networks. *[R.T.U. 2013]*

Ans. Activation Function or Squashing Function : An activation function is used for limiting the amplitude of the output of a neuron.

Another important element of this model is **bias**, b_k. The bias, b_k has the effect of increasing or lowering the net input of the activation function. Depending on whether it is positive or negative, respectively. The outcome of neuron k in mathematical terms is :

$$u_k = \sum_{j=1}^{m} w_{k_j} x_j \quad \ldots (1)$$

and,

$$y_k = \phi(u_k + b_k) \quad \ldots (2)$$

where,

$x_1, x_2, \ldots, x_m$ are input signal.
$w_{k1}, w_{k2}, \ldots w_{km}$ are synaptic weights;
u_k is linear combiner output;
b_k is bias;

Whereas these are probably historically the earliest systematic principles, they do not all apply to today's state-of-the-art of ANN design.

The Hebbian Learning Law (Hebbian Rule) due to Donald Hebb (1949) is also a widely applied principle. The Habbian Learning Law states that :

"When an axon of cell A is near-enough to excite cell B and when it repeatedly and persistently takes part in firing it, then some growth process of metabolic change takes place in one or both these cells such that the efficiency of cell A [Hebb, 1949] is increased" (*i.e.* – the weight of the contribution of the output of cell A to the above firing of coil B is increased).

The Hebbian rule can be explained in terms of the following example : Suppose that cell S caused salivation and is excited by cell F which, in turn, is excited by the sight of food. Also, suppose that cell L, which is excited by hearing a bell ring, connects to cell S but cannot alone causes S to fire.

Now, after repeated firing of S by cell F while also cell L is firing, then L will eventually be able to cause S to fire without having cell F fire. This will be due to the eventually increase in the weight of the input from cell L into cell S. Here cells L and S play the role of cells A, B respectively, as in the formulation of the Hebbian rule above.

Also the Hebbian rule need not be employed in all ANN designs. Still, it is implicitly used in designs.

However, the employment of weights at the input to any neuron of an ANN and the variation of these weights according to some procedure is common to all ANNs.

It takes place in all biological neurons. In the latter, weights variation takes place through complex biochemical processes at the dendrite side of the neutral cell, at the synaptic junction and in the biochemical structures of the chemical messengers that pass through that junction. It is also influenced by other biochemical changes outside the cell's membrane in close proximity to the membrane.

PART-C

Q.7 **(a)** ***What are different types of learning schemes used in training of artificial neural networks? Explain each of them clearly.***

(b) ***Explain the following terms with respect to neural networks:***

(i) Stability
(ii) Plasticity
(iii) Learning
(iv) Architecture

[R.T.U. 2015]

Ans.(a) Types of Learning Schemes : A neural network has to be configured in such a way that the application of a set of inputs produces the desired set of outputs. There are various methods to set the strengths of the connections.

- One way is to set the weights explicitly, using a priori knowledge.
- Another way is to "train the neural network" by feeding it teaching patterns and letting it change its weights according to some learning rule. Various categories of learning are as follows:

1. Supervised Learning or Associative Learning : In this type of learning, every input pattern that is used to train the network is associated with an output pattern, which is the target or desired pattern. A teacher is assumed to be present during the learning process, when a comparison is made between the network's computed output and the correct expected to determine the error. The error can be used to change the network parameters (weights), to improve performance.

An important issue concerning supervised learning is the problem of error convergence, i.e. the minimization of error between the desired and computed unit values. The aim is to determine a set of weights which minimizes the error. One well-known method, which is common to many learning paradigms, is the least mean square (LMS) convergence.

Paradigms of supervised learning include error-correction learning, reinforcement learning and stochastic learning.

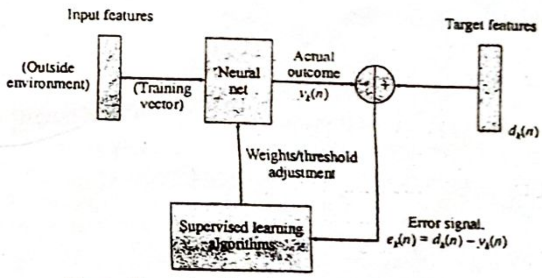


Fig. 1 : Supervised Learing or Associative Learning

2. Unsupervised Learning or Self-Organisation : In this learning method, the target output is not presented to the network. An output unit is trained to respond to clusters of pattern within the input. It is as if there is no teacher to present the desired patterns and hence, the system learns of its own by discovering and adapting to structural features in the input patterns. Paradigms of unsupervised learning are Hebbian learning and competitive learning.

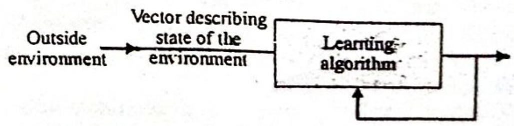


Fig. 2 : Unsupervised Learning or Self-organisation

Q.8 (a) *Explain the basic principle and some rece trends in Artificial Neural Networks.*

(b) *Explain briefly the terms cell body, axo synapse, dendride and neuron with referen to a biological neural network and explain ho it can be used for Artificial Neural Network*

[R.T.U. Dec. 20

Ans.(a) Artificial Neural Networks (ANNs) : *Refer Q.2.*

The general artificial neuron has a set of n inputs {X_1, X_2 .... X_n} representing the synaptic contact between its ce body and the axons of the neurons it is connected with and single output y representing the neurons axon. Each input is multiplied by an associated weight w_i to indicate the streng of the synapse before it is applied to the summation blo labeled by Q.

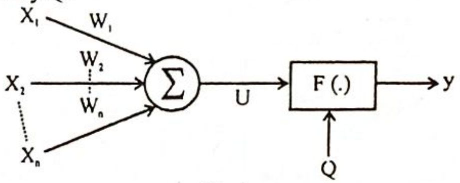


Fig. 1

In addition, it has a threshold valve e that has to be reach or exceeded for the neuron to produce a signal. The summati block, adds all the weighted inputs algebraically. Producing overall unit activation U. This activation is the processed by activation function F to produce the neuron's output. The functi F is usually nonlinear to simulate the property of biological neuro to require a minimum activation above threshold before firi and to reach a certain level of saturation with increasi activation.

So now, $U = \sum_{i=1}^{n} w_j x_i$ activation

$y = F(U - \theta)$ output

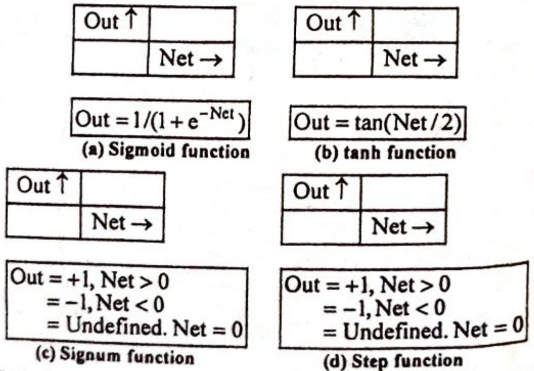


Fig.2 : Common non-linear activation function used for synaptic inhibiti

Ans.(b) Artificial neural networks emerged after the introduction of simplified neurons by Mcculloh and Pitts in 1943. These neurons were presented as models of biological neurons and as conceptual components for circuit that could perform computational task. The basic model of the neuron is founded upon the functionality of a biological neuron. Neuron's are the basic signaling units of the nervous system and each neuron is a discrete cell whose several processes arises from its cell body.

The neuron has four main regions to its structure. The cell body, or some, has two offshoots from it, the Dendrites and the axon, which end in pre-synaptic terminals. The cell body is the heart of the cell, containing the nucleus and maintaining protein synthesis. A neuron may have dendrites, which branch out in a tree like structure and receive signal from other neurons.

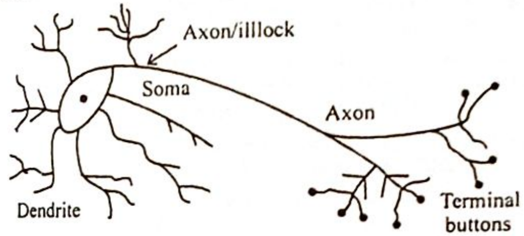


Fig. 1 : Schematic of biological neuron

A neuron usually only has one axon which grows out from a part of the cell body called the axon hillock. The axon conducts electric signals generated at the axon hillock down its length. These electric signals are called action potentials. The other end of the axon may split into several branches, which end in a pre-synaptic terminals.

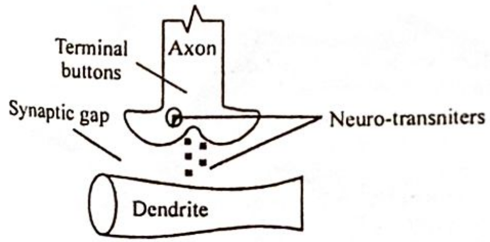


Fig. 2 : Model of synapse

The signal traveling down is regenerated in some places of axon to make it fast and remains constant. the synapse is the area of contact between two neuron. The neurons do not actually physically touch. They are separated by the synaptic deft and electric signals are sent through electrochemical interaction. The neuron sending the signal is called the pre-synaptic cell and the receiving to signal is called the post-synaptic cell.

Most synaptic contact are two types.

(1) Excitatory synapses.

(2) Inhibitory synapses.

Excitatory synapse is asymmetrical in nature. Membrances is thicker on the post-synaptic side than pre-synaptic side.

Inhibitory synapse is symmetrical in nature. Inhibitory or excitatory signals from other neurons are transmitted to neuron its dendrites synapses.

Q.9 *Write short note on use of neural nets.*

[Raj. Univ. 2008, 2007]

Ans. Use of Neural Nets: Neural networks or connectionist models can be divided into the following categories based on the complexity of the problem and the network's behavior :

- Pattern recognizers and associative memories
- Pattern transformers
- Dynamic inferencers

(a) Connectionist Speech :

Speech recognition is a difficult perceptual task. Connectionist networks have been applied to a number of problems in speech recognition. Figure shows how a three-layer backpropagation network can be trained to discriminate between different vowel sounds. The network is trained to output one of ten vowels, given a pair of frequencies taken from the speech waveform. Note the nonlinear decision surfaces created by backpropagation learning.

Speech Production : The problem of translating text into speech rather than vice versa has also been attacked with neural networks. Speech production is easier than speech recognition and high performance programs are available. Net talk, a network that learns to pronounce English text, was one of the first systems to demonstrate that connectionist methods could be applied to real-world tasks.

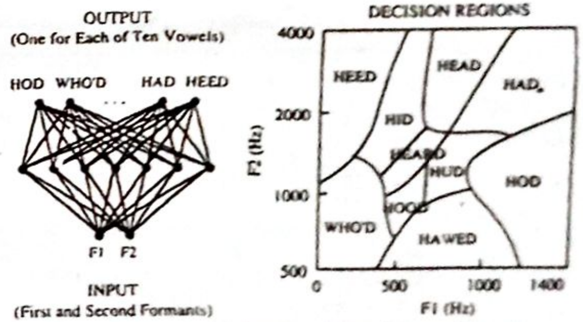


Fig. : A network that learns to distinguish vowel sounds

The rules governing the translation of text into speech units called phonemes. For example, the letter "x" is usually pronounced with a "ks" sound, as in "box" and "axe". A traditional approach to the problem would be to write all these

4 GENETIC ALGORITHMS

IMPORTANT QUESTIONS

PART-A

Q.1 *Write the advantages of particle swarm optimization(PSO).*

Ans. Advantages of Particle Swarm Optimization(PSO) :

1. PSO is based on the intelligence. It can be applied into both scientific research and engineering use.
2. PSO have no overlapping and mutation calculation.
3. The search can be carried out by the speed of the particle. During the development of several generations, only the most optimist particle can transmit information onto the other particles, and the speed of the researching is very fast.
4. The calculation in PSO is very simple. Compared with the other developing calculations, it occupies the bigger optimization ability and it can be completed easily.
5. PSO adopts the real number code, and it is decided directly by the solution. The number of the dimension is equal to the constant of the solution.

Q.2 *Write the disadvantages of particle swarm optimization (PSO).*

Ans. Disadvantages of Particle Swarm Optimization (PSO) :

1. PSO easily suffers from the partial optimism, which causes the less exact at the regulation of its speed and the direction.
2. PSO can not work out the problems of scattering.
3. PSO can not work out the problems of non-coordinate system, such as the solution to the energy field and the moving rules of the particles in the energy field.

Q.3 *What are genetic algorithms (GAs)?*

Ans. Genetic Algorithms : GAs are adaptive heuristic search algorithms based on the evolutionary ideas of natural selection and genetics. As such they represent an intelligent exploitation of a random search used to solve optimization problems. Although randomized, GAs are by no means random; instead they exploit historical information to direct the search into the region of better performance within the search space. The basic techniques of the GAs are designed to simulate processes in natural systems necessary for evolution, especially those that follow the principles first laid down by Charles Darwin, "survival of the fittest' because in nature, competition among individuals for scanty resources results in the fittest individuals dominating over the weaker one.

Q.4 *Why genetic algorithms?*

Ans. They are better than conventional algorithms in that they are more robust. Unlike older artificial intelligence systems, they do not break easily even if the inputs are changed slightly or in the presence of reasonable noise. Also, in searching a large state-space, multimodal state-space or n dimensional surface, GA offers significant benefits over more typical optimization techniques (linear programming, heuristic depth-first, breath-first and praxis.)

Q.5 *What are the advantages and limitations of genetic algorithms?*

Ans. Advantages of Genetic Algorithms : The advantages of genetic algorithm are as follows:

1. Parallelism.
2. Liability
3. Solution space is wider.
4. The fitness landscape is complex.
5. Easy to discover global optimum.
6. The problem has multiobjective function.

Limitations of Genetic Algorithms : The limitations of genetic algorithms are as follows:

1. The problem of identifying fitness function.
2. Definition of representation for the problem.
3. Premature convergence occurs.
4. The problem of choosing various parameters such as the size of the population, mutation rate, crossover rate, the selection method and its strength.

PART-B

Q.6 *Explain the following terms with example in reference to Genetic Algorithm*
(i) Population
(ii) Crossover
(iii) Reproduction
(iv) Mutation **[R.T.U. 2016]**

Ans.(i) Population : It is a subset of solutions in the current generation. It can also be defined as a set chromosomes. There are two models of population :

1. Steady state model (Incremental GA)
2. Generational model

There are two methods to initialize a population in genetic algorithm :

1. Random initialization
2. Heuristic initialization.

The population is generally defined as a two dimensional array of size population, size x, chromosome size.

(ii) Crossover Operator/Recombination : The recombination is based on the principle that by mating two individuals with different but desirable features can produce an offspring that have desirable features of both of its parents. However, the result can be counterproductive also. But when applied many times, there must be some positive outcomes. The recombination should be designed in such a way that it should recombine the information relevant to the problem in hand while recombining genes. In addition, if the mating parents are genetically close then the offspring must also be similar to the parents. This is called similarity requirement. Therefore for different representation schemes, different recombination operators are proposed.

(iii) Reproduction : The next step is to generate a second generation population of solutions from those selected through genetic operators: crossover (also called recombination), and or mutation.

For each new solution to be produced, a pair of "parent" solutions is selected for breeding from the pool selected previously. By producing a "child" solution using the above methods of crossover and mutation, a new solution is created which typically shares many of the characteristics of its "parents". New parents are selected for each new child, and the process continues until a new population of solutions of appropriate size is generated. Although reproduction methods that are based on the use of two parents are more "biology inspired", recent researches suggested more than two "parents" are better to be used to reproduce a good quality chromosome.

These processes ultimately result in the next generation population of chromosomes that is different from the initial generation. Generally the average fitness will have increased by this procedure for the population, since only the best organisms from the first generation are selected for breeding, along with a small proportion of less fit solutions, for reasons already mentioned above.

(iv) Mutation : Mutation is used to maintain diversity in the population so that premature convergence may not take place. Moreover to a certain extent, it also helps to explore new areas of the search space. Mutation operator randomly flips some of the bits in the chromosome. For different types of representations different mutation operators have been proposed. All mutation operator generate a new but random search point without any positional bias in the neighbourhood of present search point.

Q.7 *A budget airline company operates 3 plains and employs 5 cabin crews. Only one crew can operate on any plain on a single day, and each crew cannot work for more than two days in a row. The company uses all planes every day. A genetic algorithm is used to work out the best combination of crews on any particular day.*

could represent an ?
alphabet of this ?
for this problem.
n this problem? Is
tic Algorithms for
company operated
d more crews?

[R.T.U. 2015]

mbination of 5 cabin
, we could encode

ing a solution would
ws):

n the chromosome
ding is possible for the
ld encode if a crew is
does not matter which
y off:

ome of 5 genes each
th 0 or 1 representing if

n many different ways.
g used. In the first case,
he alphabet consists of 5
binary representation is

ifferent versions, but it is
nto account the condition
e than 2 days in a row. For
ke into account how many
day off (e.g. 1 or 0). The
sum of these numbers for
, if d is the number of days
f, and m is the number of
s, then the fitness can be

$= \sum_{i=1}^{m} d_i$

(iv) The number of solutions is the number of times 3 crews can be selected out of 5 without replacement and without taking into account their order. The first crew can be selected in 5 different ways, the second in 4 ways and the third in 3 different ways. These numbers multiplied together will give us total number times how 3 crews can be selected randomly out of 5 : 5 × 4 × 3 = 60 times. However, there are 6 possible combinations in which 3 crews can be ordered, and because the order does not matter the answer is 60/6 = 10. Thus, there are 10 possible solutions for this problem.

It is not really necessary to use GA for a problem with such a small population, because solutions can be checked explicitly. However, as the number of crews and airplanes increases, so does the number of solutions, and the use of GA can be the only option. In fact, if n is the number of cabin crews and $k \le n$ is the number of airplanes, then the number of solutions is

$$\frac{n!}{k!(n-k)!}$$

For example, if the company operated 10 airplanes and employed 20 cabin crews, then the number of solutions would be

$$\frac{20!}{10!(20-10)!} = 184,756$$

PART-C

Q.8 ***Write short note on Genetic Algorithm.***

OR

Explain the procedure of Genetic Algorithm with example. *[R.T.U. 2016]*

OR

What are the basic organizations for a genetic algorithm. *[R.T.U. Dec. 2013]*

OR

Explain various properties of genetic algorithm. Discuss their application with examples. *[R.T.U 2012]*

Ans. Genetic Algorithm : A genetic algorithm (GA) is a search technique used in computing to find exact o approximate solutions to optimization and search problems Genetic algorithms are categorized as global search heuristics Genetic algorithms are a particular class of evolutionar algorithms (EA) that use techniques inspired by evolutionar biology such as inheritance, mutation, selection, and crossove

Genetic algorithms are implemented in a computer simulation in which a population of abstract representations (called chromosomes or the genotype of the genome) of candidate solutions (called individuals, creatures, or phenotypes) to an optimization problem evolves toward better solutions. Traditionally, solutions are represented in binary as strings of 0s and 1s, but other encodings are also possible. The evolution usually starts from a population of randomly generated individuals and happens in generations. In each generation, the fitness of every individual in the population is evaluated, multiple individuals are stochastically selected from the current population (based on their fitness), and modified (recombined and possibly randomly mutated) to form a new population. The new population is then used in the next iteration of the algorithm. Commonly, the algorithm terminates when either a maximum number of generations has been produced, or a satisfactory fitness level has been reached for the population. If the algorithm has terminated due to a maximum number of generations, a satisfactory solution may or may not have been reached.

Genetic algorithms find application in bioinformatics, phylogenetics, computational science, engineering, economics, chemistry, manufacturing, mathematics, physics and other fields.

A typical genetic algorithm requires :

1. A genetic representation of the solution domain,
2. A fitness function to evaluate the solution domain.

A standard representation of the solution is as an array of bits. Arrays of other types and structures can be used in essentially the same way. The main property that makes these genetic representations convenient is that their parts are easily aligned due to their fixed size, which facilitates simple crossover operations. Variable length representations may also be used, but crossover implementation is more complex in this case. Tree-like representations are explored in genetic programming and graph-form representations are explored in evolutionary programming.

The fitness function is defined over the genetic representation and measures the quality of the represented solution. The fitness function is always problem dependent. For instance, in the knapsack problem one wants to maximize the total value of objects that can be put in a knapsack of some fixed capacity. A representation of a solution might be an array of bits, where each bit represents a different object, and the value of the bit (0 or 1) represents whether or not the object is in the knapsack. Not every such representation is valid, as the size of objects may exceed the capacity of the knapsack. The fitness of the solution is the sum of values of all objects in the knapsack if the representation is valid, or 0 otherwise. In some problems, it is hard or even impossible to define the fitness expression; in these cases, interactive genetic algorithms are used.

Once we have the genetic representation and the fitness function defined, GA proceeds to initialize a population of solutions randomly, then improve it through repetitive application of mutation, crossover, inversion and selection operators.

Selection : During each successive generation, a proportion of the existing population is selected to breed a new generation. Individual solutions are selected through a fitness-based process, where fitter solutions (as measured by a fitness function) are typically more likely to be selected. Certain selection methods rate the fitness of each solution and preferentially select the best solutions. Other methods rate only a random sample of the population, as this process may be very time-consuming.

Most functions are stochastic and designed so that a small proportion of less fit solutions are selected. This helps keep the diversity of the population large, preventing premature convergence on poor solutions. Popular and well-studied selection methods include roulette wheel selection and tournament selection.

Reproduction : *Refer to Q.6.*

Simple generational genetic algorithm pseudocode

1. Choose the initial population of individuals.
2. Evaluate the fitness of each individual in that population.
3. Repeat on this generation until termination: (time limit, sufficient fitness achieved, etc.).

(a) Select the best-fit individuals for reproduction.

(b) Breed new individuals through crossover and mutation operations to give birth to offspring.

(c) Evaluate the individual fitness of new individuals.

(d) Replace least-fit population with new individuals.

❑❑❑

NEURO-FUZZY TECHNOLOGY AND MATLAB 5

IMPORTANT QUESTIONS

PART-A

Q.1 *What do you mean by neuro fuzzy system?*

Ans. Neuro Fuzzy System : A combination of a neural network and a fuzzy system is called a neuro fuzzy system. In neuro fuzzy control, the parameters of the fuzzy controller are adjusted using a neural network.

Neuro fuzzy systems utilize both the linguistic, human-like reasoning of fuzzy systems and the powerful computing ability of neural networks.

Q.2 *Describe in brief the types of neuro fuzzy system.*

Ans. Types of Neuro-Fuzzy Systems : Types of neuro-fuzzy systems are as follows :

1. Cooperative Neuro-Fuzzy System : Neural networks mechanisms of learning determine some sub-blocks of the fuzzy system. After the fuzzy sub-blocks are calculated the neural network learning methods are taken away.

2. Concurrent Neuro-Fuzzy Model : Neural network and fuzzy system work simultaneously to determine the required parameters.

3. Hybrid Neuro-Fuzzy System : Fuzzy system uses a learning algorithm inspired by the neural networks theory to determine its parameters through the pattern processing.

Q.3 *Why is reinforcement learning needed?*

Ans. Needs of Reinforcement Learning :

1. The parameters of the fuzzy controller could be updated using the back propagation algorithm common in supervised learning in neural networks.
2. Back propagation algorithm cannot be used; instead, a learning algorithm called reinforcement learning is used.
3. In reinforcement learning, the system evaluates whether the previous control action was good or not. If the action had good consequences, the tendency to produce that action is strengthened, that is, reinforced.

Q.4 *Which types of tools are consist in matlab fuzzy toolbox?*

Ans. Matlab Fuzzy Toolbox : Matlab fuzzy toolbox consists of two useful tools :

(i) FIS Editor : This editor in combination with 4 other editors, provides a powerful environment to define and modify Fuzzy Inference System (FIS) variable.

(ii) Fuzzy Controller : This is a block in fuzzy toolbar library in simulink environment. This block admits FIS variable produced by FIS editor and implements the desirable rules.

Q.5 *Write the benefits of fuzzy interference system (FIS).*

Ans. Benefits of Fuzzy Interference System (FIS) :

1. Simply add or remove inputs and/or outputs.
2. Set or modify interference methods.
3. Simply add or remove membership functions and easy management of function's parameters.

4. Easy rule definition and modification.
5. Great visulization area for FIS diagram.
6. 2-D visulization of interference for any input pairs to any output.

PART-B

Q.6 *What is fuzzy controllers? Explain the basic structure of the fuzzy controller with block diagram.* *[R.T.U. 2013]*

OR

What is fuzzy controller? Discuss the basic steps involved in design of fuzzy controller. *[R.T.U. 2016]*

Ans. Fuzzy Controller : A fuzzy controller is a special fuzzy system that can be used as a controller component in a closed-loop system. The integration of a fuzzy system into a closed loop is shown. Special emphasis is put onto the transfer behaviour of fuzzy controllers, which is analysed using different configurations of standard membership functions. An example for the design of a fuzzy controller for a loading crane is given. The module series is closed about the contribution of fuzzy control.

1. Design a fuzzy control system.
2. Know the transfer behaviour of a fuzzy control system.
3. Change the membership functions to influence the transfer behaviour of a fuzzy control system.

Basic Structure of a Fuzzy Controller : A fuzzy controller can be handled as a system that transmits information like a conventional controller with inputs containing information about the plant to be controlled and an output that is the manipulated variable. From outside, there is no vague information visible, both, the input and output values are crisp values. The input values of a fuzzy controller consist of measured values from the plant that are either plant output values or plant states, or control errors derived from the set-point values and the controlled variables.

A control law represented in the form of a fuzzy system is a static control law. This means that the fuzzy rule-based representation of a fuzzy controller does not include any dynamics which makes a fuzzy controller a static transfer element, like the standard state-feedback controller. In addition to this, a fuzzy controller is in general a fixed nonlinear static transfer element, which is due to those computational steps of its computational structure that have nonlinear properties. In what follows the computational structure of a fuzzy controller will be described by presenting the computational steps involved. The computational structure of a fuzzy controller consists of three main steps as illustrated by the three blocks in fig.

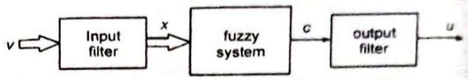


Fig. : Basic structure of a fuzzy controller

1. Signal conditioning and filtering at the input (input filter)
2. Fuzzy system
3. Signal conditioning and filtering at the output (output filter).

The input and output filters are for signal conditioning. The external input signals must be scaled such that they can be fed as signals into the fuzzification part of the fuzzy system. In many cases, the signals in are the control error and its derivative. In this case the input filter contains a differentiating element. Also other dynamical elements can be in the input filter, *e.g.* integrators for the control error. Additionally auxiliary signals from the plant measurements may be used that represent plant states or disturbances acting on the plant. The design of this input filter depends on the application, which will be illustrated later by an example.

The fuzzy system contains the control strategy and consists of those components already discussed in section 16.5. For example, a linguistic formulation of a proportional control strategy would be expressed by the following rules of the fuzzy system :

1. IF (Control error positive) Then (Manipulated variable positive),
2. IF (Control error zero) Then (Manipulated variable zero)
3. IF (Control error negative) Then (Manipulated variable negative)

A proper rule base can be found either by asking experts or by evaluation of measurement data using data mining methods.

The output filter is for the adaptation of the crisp output from the fuzzy system to the manipulated variable of the plant. In principle, there are many dynamical and static operations possible. Often, the output of the fuzzy system describes an increment of the manipulated variable, and thus an integration of this increment must occur.

In COA, the overlapping area is counted once. The center of sum (COS) take into account the overlapping areas of multiple rules more than once. The graphical representation is similar to that of figure but overlapping area is counted more than once.

In discrete case, defuzzification output is definedas :

$$u = \frac{\sum_{i=1}^{N} u_i \sum_{k=1}^{n} \mu_k(u_i)}{\sum_{i=1}^{N} \sum_{k=1}^{n} \mu_k(u_i)}$$

where,

N = number of sample points,

n = number of rules.

This method can be implemented easily and leads to fast inference cycles.

PART-C

Q.10 *What do you mean by membership function with fuzzy set? Also describe various types of membership functions.* *[R.T.U. 2016]*

Ans. Membership Function : Every element in the universe of discourse is a member of the fuzzy set to some grade, may be even zero. The set of elements that have a non-zero membership is called the *support* of the fuzzy set. The function that ties a number to each element of the universe is called the membership function μ(x).

Continuous and Discrete Representations : There are two alternative ways to represent a membership function in a computer: continuous or discrete. In the continuous form the membership function is a mathematical function, possibly a program. A membership function is for example bell-shaped (also called a π-curve), s-shaped (called an *s*-curve), a reverse *s*-curve (called *z*-curve), triangular, or trapezoidal. In the discrete form the membership function and the universe are discrete points in a list (vector). Sometimes it can be more convenient with a sampled (discrete) representation.

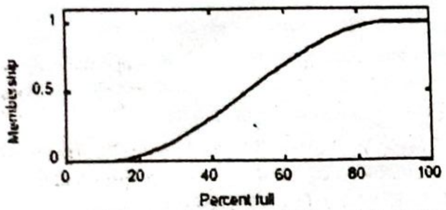


Fig. 1 : Possible definition of the set high levels in the tank

As a very crude rule of thumb, the continuous form is more CPU intensive, but less storage demanding than the discrete form.

Various types of membership function can be used to represent membership functions such as including triangular, trapezoidal, bell shaped and gaussian functions. However, triangular and trapezoidal membership functions are the most popular. The choice of a shape for each particular linguistic variable is both subjective and problem dependent.

Important points of fuzzy membership functions are :

1. Determining fuzzy membership function is the key issue in all fuzzy sets.
2. The membership function fully defines the fuzzy set.
3. A membership function provides a measure of the degree of similarity of an element to a fuzzy set.
4. Membership function can :
   (a) either be chosen by the user arbitrarily, based on the user experience,
   (b) Or, be designed using machine learning methods *i.e.* neural networks, genetic algorithms, etc.

A triangular membership function can be defined by three parameters {a,b,c} as follows:

$$\mu_A(x; a, b, c) = \begin{cases} 0, & \text{if } x \le a \\ \frac{x-a}{b-a}, & \text{if } a \le x \le b \\ \frac{c-x}{c-b}, & \text{if } b \le x \le c \\ 0, & \text{if } x \ge c \end{cases}$$

The pictorial representation of triangular membership function is given fig.2.

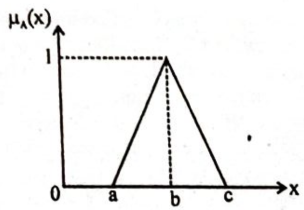


Fig. 2 : Triangular membership function

a,b and c represent the x coordinates of the vertices of $\mu_A(x)$ in a fuzzy set A, where

a = lower boundary

c = upper boundary where membership degree is zero,

b = the centre where membership degree is 1.

A trapezoidal membership function is defined by four parameters {a, b, c, d} as follows :

$$\mu_A(x; a, b, c, d) = \begin{cases} 0, & \text{if } x \le a \\ \frac{x-a}{b-a}, & \text{if } a \le x \le b \\ 1, & \text{if } b \le x \le c \\ \frac{d-x}{d-c}, & \text{if } c \le x \le d \\ 0, & \text{if } d \le x \end{cases}$$

The pictorial representation of trapezoidal membership function is given in fig.3.

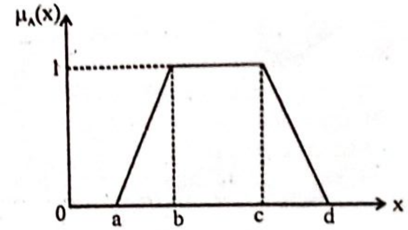


Fig.3 : Trapezoidal membership function

a,b,c and d determines the x coordinates of the corners of a trapezoidal membership function.

The **gaussian membership** function is defined as follows :

$$\mu_A(x, c, s, m) = \exp\left[-\frac{1}{2}\left|\frac{x-c}{s}\right|^m\right]$$

where c is centre, s is width and m is fuzzification factor. The pictorial representation of gaussian membership function is given in fig.4.

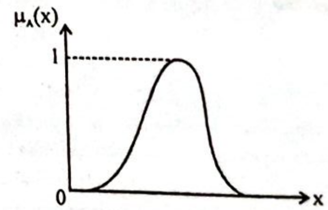


Fig. 4 : Gaussian Membership Function

Q.11 *What is fuzzy relation? Explain the method of defuzzification.* *[R.T.U. 2013]*

Ans. Fuzzy Relation : A standard relation from set *A* to set *B* is a subset of the Cartesian product of *A* and *B*, written as *A*×*B*. The elements of *A*×*B* are ordered pairs (*a*, *b*) where *a*

is an element of ...
ordered pair (Joe ...
of the set of fath...
which includes P...
the Cartesian pr...
case, the ordere...
contains (*a*, *b*, if ...
on the set of mem...
relation is similar ...
sets are fuzzy ...
'much_more_educ...
much_...
(Jeff, Mike), ... }

Matrix Rep...

A fuzzy rela...
underlying sets, c...
let the set of mem...
the same relation...
of the Cartesian...
membership in thi...
mmuch_more_e...
mmuch_more_ed...
membership in th...
Jeff)? It seems re...
whenever the two...
Our relation much...
may look like the f...
0/(Jeff...
much_more_...
Steve) 0.3/(Steve, ...
0.1/(M...

Method of ...
methods of defuzzi...

(i) Maxima ...
an element from t...

Random Ch...

Prob(D(A) = ...

First of max...
Middle of Maxi...

FOM (A) ...
LOM(A) ...
For MOM(A) ...
|core(A)$_{MOM}$ ...
|core(A)$_{MOM}$ ...

nd derivatives :
on into a probability
xpected value. Main

Distributions)

idence in the system.
t Defuzzification)

nce to the system, N is
m_i is the average value

ravity)

hold below which all

uzzy output sets, α_i is the
inference rules and a_i is
with the output fuzzy sets

e de-fuzzification value
ship function in two (more

nimizes the expression

x)

Some other methods are give below :

- AI (adaptive integration)
- CDD (constraint decision defuzzification)
- ECOA (extended center of area)
- EQM (extended quality method)
- FCD (fuzzy clustering defuzzification)
- IV (influence value)
- QM (quality method)
- SLIDE (semi-linear defuzzification)
- WFM (weighted fuzzy mean)

The maxima methods are good candidates for fuzzy reasoning systems. The distribution methods and the area methods exhibit the property of continuity that makes them suitable for fuzzy controllers.

Q.12 ***What is called De-Fuzzification? Mention its types and discuss the different methods of de-fuzzification.***

OR

Write short notes on De-Fuzzification. **[R.T.U 2012]**

Ans. Defuzzification (Convert fuzzy output values to control signal) : The process of moving from a fuzzy set to a point where a control decision can be made is called ' defuzzification'. The process of defuzzification will depend on the fuzzy data being processed and may involve linguistic approximation or an arithmetic method. The most common arithmetic methods involve either the mean of maximum values or a method which calculates the 'centre of gravity' of moments method. These methods will obviously different control results. The choice of method is likely to rely on the type of data being processed. The maximum method will yield relatively few discrete control points whilst the moments method will provide an output with greater continuity.

For example, an advertising company wishing to sell slimming products may wish to select young or heavy people from the global set.

Young_heavy = Young People U Heavy People = { 1/Jim, 0.9/Mike, 0.88/Peter, 0.8/David, 0.7/John, 0.76/Anna, 1/Sally, 0.95/Jane }

Maximum values selected

So this fuzzy set includes all members from either set.

If we wished to target only those people who were Tall and Elderly (old) as part of out marketing campaign we would select.

Tall_old= Tall People Maths symbol Old People = {0.5/Jim, 0.9/Mike, 0/John, 0.8/Mary, 0.7/David, 0/Anna, 0/Peter } = { 0.9/Mike, 0.8/Mary, 0.7/David, 0.5/Jim}

Minimum values selected

When fuzzy logic has identified a solution such as the set of Tall_old people it will probably be necessary to take some action from the result. If action required is to use the resulting set from this example as the source for a mail shot, it would not be helpful to send 0.9 of a letter to Mike, 0.8 of a letter to Mary etc.

In the example shown above, we wish to know, for each of the elements of the resultant fuzzy set:

{ 0.9/Mike, 0.8/Mary, 0.7/David, 0.5/Jim }

Whether to send the advertising mail, or not to send the advertising mail.

Since in this case, the decision is of a Boolean type (send or don't send), then it would be a decision made by the marketing department to select a value of set membership, above which mail would be sent and below which, no mail would be sent.

In a situation where the decision would be applied in an analogue or applied in varying degrees, the defuzzification methods would require application of one of the methods discussed. For instance, we may consider the set of Old and Heavy people who should be advised to eat less. How much less they should eat may be on a sliding scale. The set of Old Heavy people will be:

Old_Heavy = { 0.9/Mike, 0.7/David, 0.5/Jim } NOTE: Minimum values (&)

We may also consult dietary experts and derive a proportion for how much less people should eat, depending on their membership of the set of Old_Heavy people.

Strictly for the sake of continuing this discussion, table and figure represent some fictitious data about proportional reduction of dietary intake for members of the set of Old_Heavy people.

If there was some need to give general advice to members of this set rather than specify dietary information to each person it would be necessary to deduce a specific value representing dietary reduction to all members of the set. For this example, one of the methods of defuzzification would be necessary.

	A	B
1	**Membership**	**%less to eat**
2	1	65
3	0.9	62
4	0.8	56
5	0.7	50
6	0.6	42
7	0.5	30
8	0.4	16
9	0.3	2
10	0.2	0
11	0.1	0

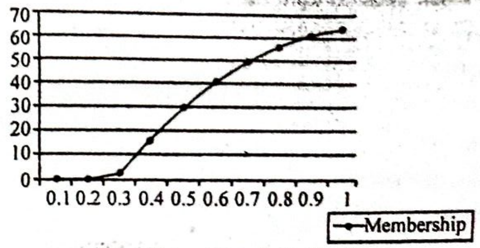


Fig. : Dietary Intake Reduction Graph

If the maximum value were to be used, we may advise all members of the set to eat 65% less food. However, if the centre of gravity is calculated (a simple average for the curve shown) the set of Old_Heavy people would be advised to eat 50% less food. This is because the average value for the three members is 0.7 and from the curve, this equates to a dietary reduction of 50%.

This example is not intended to show precisely how to derive decisions from fuzzy logic. However, it should be clear that in order to take decisions from the results of fuzzy logic, some additional processing is required.

Fuzzy logic is able to represent the shades of particular attributes rather than requiring their classification as either one thing or another. It also allows these shades or fuzzy values to be processed in a similar way to ordinary logic and a result to be derived.

For industrial control problems, where the eventual outcome is to perform specific action, some defuzzification will usually be required.

Types of De-Fuzzification

(a) Triangular (b) Exponential

(c) Gaussian

Methods of De-Fuzification : *Refer to Q.11.*

Q.13 ***To find whether the following relation is equivalence or not using a MATLAB program.***

$$R = \begin{bmatrix} 1 & 0.87 & 0 & 0.13 & 0.35 \\ 0.87 & 1 & 0.46 & 0 & 0.98 \\ 0 & 0.46 & 1 & 0 & 0 \\ 0.13 & 0 & 0 & 1 & 0.54 \\ 0.24 & 0.98 & 0 & 0.54 & 1 \end{bmatrix}$$

Output

enter the matrix [1 0.87 0 0.13 0.35; 0.87 1 0.46 0.98; 0 0.46 1 0 0; 0.13 0 0 1 0.54; 0.24 0.98 0 0.54 1]

p =

1.0000	0.8700	0	0.1300	0.3500
0.8700	1.0000	0.4600	0	0.9800
0	0.4600	1.0000	0	0
0.1300	0	0	1.0000	0.5400
0.2400	0.9800	0	0.5400	1.0000

The given relation is reflexive, not symmetry and not transitivity and hence not an equivalence relation.

Q.14 *Write a MATLAB program for maximizing $f(x) = x^2$ using GA, where x is ranges from 0 to 31. perform 5 iterations only.*

Ans. Steps involved

Step 1: Generate initial four populations of binary string with 5 bits length.

Step 2: Calculate corresponding x and fitness value $f(x) = x^2$.

Step 3: Use the tournament selection method to generate new four populations.

Step 4: Apply crossover operator to the new four populations and generate new populations.

Step 5: Apply mutation operator for each population.

Step 6: Repeat the steps 2-5 for 5 iterations.

Step 7: Finally print the result.

Source Code

```
%proqram for genetic algorithm to maximize the function f(x) = x square
clear all;
    clc;
    %x ranges from 0 to 31 2power5 = 32
    %five bits are enough to represent x in binary representation
    n = input('Enter no. of population in each iteration');
    nit = input('Enter no. of iterations');
    % Generate the initial population
    [oldchrom]=initibp(n, 5)
% The popultion in binary is converted to integer
FieldD = [5 ; 0 ; 31 ; 0 ; 0 ; 1 ; 1]
for i = 1:nit
    phen = bindecod(oldchrom , FieldD , 3); % gives the integer value of the binary population
    % obtain fitness value
    sqx = phen." 2;
    sumsqx = sum(sqx);
    avsqx = sumsqx/n;
    hsqx = max(sqx);
    pselect = sqx./sumsqx;
    sumpselect = sum(pselect);
    avpselecte = sumpselect/n;
    hpselect = max(pselect);
    % apply roulette wheel selection
    FitnV=sqx;
    Nsel = 4;
    newchrix = selrws(FitnV, Nsel);
    newchrom = oldchrom(newchrix, :);
    %Perform Crossover
    crossrate = 1;
    newchromc = recsp(newchrom , crossrate); % population after crossover
    % Perform mutation
    vlub=0:31;
    mutrate = 0.001;
    newchromm = mutrandbin (newchromc,vlub,0.0
    %new population
    after mutation
    disp('For iteration');
i
disp('Population');
oldchrom
disp('x');
phen
disp('f(x)');
sqx
oldchrom = newchromm;
end
```


Output

Enter no. of population in each iteration4

Enter no. of iterations5

At the end of fifth iteration, the output is

For iteration

i=

5

Population

oldchrom =

```
0 0 0 0 1
0 0 0 1 0
0 1 1 0 0
0 1 0 0 0
```

x

phen =

1
2
12
8

f(x)

sqx =

1
4
144
64