1 Introduction

1.1 : Introduction to Soft Computing

Q.1 Define soft computing. ☞[RGPV : June-16, Marks 2]

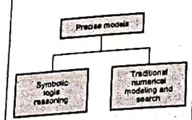
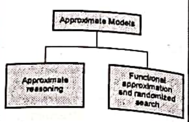
Ans. : • Definition : "Soft computing is an emerging approach to computing which parallel the remarkable ability of human mind to reason and learn in a environment of uncertainty and imprecision".

- The idea behind soft computing is to model cognitive behavior of human mind. Soft computing is foundation of conceptual intelligence in machines. Unlike hard computing, soft computing is tolerant of imprecision, uncertainty, partial truth, and approximation.
- Soft computing is a collection of methodologies that aim to exploit the tolerance for imprecision and uncertainty to achieve tractability, robustness, and low solution cost.

Q.2 Differentiate between soft computing and hard computing. ☞[RGPV : Dec.-16, Marks 2]

OR Compare soft computing and hard computing. ☞[RGPV : June-14, Marks 7]

Ans. :

Sr. No.	Hard computing	Soft computing
1.	It uses binary logic.	It uses fuzzy logic.
2.	It is based on numerical analysis.	It is based on genetic algorithms.
3.	Crisp system is used in had computing.	Neurocomputing is used in soft computing.
4.	It takes help of differential equations.	It takes help of probabilistic reasoning.
5.	Approximation theory.	Evidential reasoning.
6.	Functional analysis.	Machine learning.
7.		

Q.3 List and characterize the constituents of soft computing.

Ans. : 1. One of the characteristics of soft computing methods is that they are typically used in problems where mathematical models are not available or are intractable or too cumbersome to be viable.

2. Another characteristic is that uncertainty inherent in many situations under study is taken into account rather than ignored.
3. Soft computing often provides a good solution as opposed to an optimal solution.
4. Soft computing uses human concept like if-then rules, cases and conventional knowledge representations.
5. Biologically inspired computing models (NN).
6. New optimization techniques are applied by soft computing.
7. Model free learning : Models are constructed based on the target system only.
8. New application domains : Mostly computation intensive like adaptive signal processing, adaptive control, nonlinear system identification etc.
9. Fault tolerance : Deletion of a neuron or a rule does not destroy the system. The system performs with lesser quality.
10. Goal driven characteristics : Only the goal is important and not the path.

Q.4 List strength and weakness of soft computing.

Ans. : Strength :

1. Low cost and less time consuming solutions are provided by soft computing.
2. Soft computing techniques can tolerate imprecision, uncertainty and partial truth to produce High Machine Intelligent Quotient.
3. It provides good solutions.
4. It uses human concept.

Weakness :

1. It requires extensive computation.
2. Soft computing does not perform much symbolic manipulation.
3. Soft computing is not precisely defined.

1.2 : Application Areas of Soft Computing

Q.5 What type of problem is solved by using soft computing ?

Ans. : • Soft computing techniques have become one of promising tools that can provide practice and reasonable solution

1. Feature Selection : • In machine learning and statistics, feature selection, also known as variable selection, attribute selection or variable subset selection, is the process of selecting a subset of relevant features for use in model construction.

- Feature selection techniques are a subset of the more general field of feature extraction. Feature extraction creates new features from functions of the original features, whereas feature selection returns a subset of the features.

2. Image Processing : • In imaging science, image processing is any form of signal processing for which the input is an image, such as a photograph or video frame; the output of image processing may be either an image or a set of characteristics or parameters related to the image.

- Most image-processing techniques involve treating the image as a two-dimensional signal and applying standard signal processing techniques to it.

3. Medical Diagnosis : • Medical diagnosis refers both to the process of attempting to determine or identify a possible disease and to the opinion reached by this process. From the point of view of statistics the diagnostic procedure involves classification tests.

4. Pattern Recognition : • Pattern recognition generally aim to provide a reasonable answer for all possible inputs and to perform "most likely" matching of the inputs, taking into account their statistical variation.

- Pattern recognition is studied in many fields, including psychology, psychiatry, and ethology, cognitive science, and traffic flow and computer science.

5. Smart Instrumentation : • As intelligent devices become ubiquitous the challenge is to connect sensors and actuators through smart systems. A major issue is the growing number of communication protocols, with no single standard.

- The challenge is to intelligently connect smart instrumentation so that devices can communicate across multiple protocols. At the same time, increases in the volume and importance of data means that privacy, security and robustness of systems is paramount.

Q.6 Explain shortest path algorithm.

Ans. : • In many applications such as transportation, routing, communications, economical, and so on, graphs emerge naturally as a mathematical model of the observed real world system. Indeed, many problems can be reformulated as a quest for a path between two nodes in a graph which is optimal in the sense of a number of preset criteria.

- The object is to find a path of minimum cost or length from a specified source node s to another specified sink node t, assuming that each arc (i, j) ∈ A has an associated cost (or length) c_{ij}.
- Directed graph G = (V, A) where V is a set of nodes, A is a set of links.
- c_{ij} is a cost associated with each arc(i, j) and

  Source node : node 1
- Destination node : node n
- Indicator variable :

$$x_{ij} = \begin{cases} 1, & \text{if link (i, j) is included in the path} \\ 0, & \text{otherwise} \end{cases}$$

Scanned with CamScanner

- *Fuzzy logic and fuzzy graph theory provides proper tools to use in these cases. The basic idea behind path finding is searching a graph, starting at one point, and exploring adjacent nodes from there until the destination node is reached.*
- *A path finding algorithm for transit network is proposed to handle the special characteristics of transit networks, such as city emergency handling and drive guiding system, in where the optimal paths have to be found.*
- *In Dijkstra's algorithm the input of the algorithm consists of a weighted directed graph G and a source vertexs in Graph. Let's denote the set of all vertices in the graph G as V. Each edge of the graph is an ordered pair of vertices (u, v) representing a connection from vertex u to vertex v.*
- *The set of all edges is denoted E. Weights of edges are given by a weight function w : E → [0, ∞]; therefore w (u, v) is the non-negative cost of moving from vertex u to vertex v. The cost of an edge can be thought of as the distance between those two vertices.*
- *The cost of a path between two vertices is the sum of costs of the edges in that path. For a given pair of vertices s and t in V, the algorithm finds the path from s to t with lowest cost (i.e. the shortest path). It can also be used for finding costs of shortest paths from a single vertex s to all other vertices in the graph.*
- *In a fuzzy graph G, let λ be the set of all paths from vertex a to vertex b and the fuzzy length of a path be l_q.*

$$l_q = \text{Length } (Q)$$

$$= w_1 + w_2 + \ldots + w_1 = \sum_{e_k \in q} w_k,\ Q \in \lambda$$

- *In the above equation, e_k are edges of graph G. A fuzzy set S on λ with memberships σ_s is the fuzzy set of shortest paths :*

$$\sigma_s\ (Q) = \min\ \{\mu_{l_Q \le l_P}\},\quad \text{where } Q, P \in \lambda$$

1.3 : Classification of Soft Computing Techniques

Q.7 List various types of soft computing techniques. [RGPV : Dec.-16, Marks 2]

Ans. : Types of soft computing techniques are fuzzy logic, neural network, support vector machine, evolutionary computation, probabilistic reasoning and machine learning etc

Q.8 List and explain in brief constituents of soft computing.

Ans. : • The principal constituents, i.e., tools, techniques, of Soft Computing (SC) are Fuzzy Logic (FL), Neural Networks (NN),

Support Vector Machines (SVM), Evolutionary Computation (EC), and Machine Learning (ML) and Probabilistic Reasoning (PR).

- Fuzzy theory plays a leading role in soft computing and this stems from the fact that human reasoning is not crisp and admits degrees.

Genetic algorithms

- Genetic algorithms are inspired by Darwin's theory of natural evolution. In the natural world, organisms that are poorly suited for an environment die off, while those well-suited, prosper.
- Genetic algorithms search the space of individuals for good candidates. The chance of an individual's being selected is proportional to the amount by which its fitness is greater or less than its competitors' fitness.

Neural Networks

- Neural networks consists of many number of simple elements (neurons) connected between them in system. Whole system is able to solve of complex tasks and to learn for it like a natural brain.
- For user NN is black box with Input vector (source data) and Output vector (result).

Probabilistic reasoning

- Uncertainty is described by probabilities. Probability may be use for simulation of fuzziness.
- Relations between events are described as conditional probabilities (Bayesian nets) or probabilities of transition probabilities (Markovian process).

Q.9 Describe evolutionary computing approach of soft computing.

Ans. : The domain of evolutionary computation involves the study of the foundations and the applications of computational techniques based on the principles of natural evolution.

- Generally speaking, evolutionary techniques can be viewed either as search methods, or as optimization techniques.
- An abstract task can be solved in such an optimized way that the best solution can be found as a result of search through a space of potential solutions.
- An abstraction from the theory of an abstraction from the theory of biological evolution that is used to biological evolution that is used to create optimization procedures or create optimization procedures or methodologies, usually implemented on methodologies, usually implemented on computers, that are used to solve computers, that are used to solve problems.
- Components of Evolutionary Computing are as follows :

1. Genetic Algorithms : invented by John Holland in the 1960's
2. Evolution Strategies : invented by Ingo Rechenberg in the 1960's

- EA exhibit an adaptive behavior that allows them to handle non-linear, high dimensional problems without requiring differentiability or explicit knowledge of the problem structure.
- EA are very robust to time-varying behavior, even though they may exhibit low speed of convergence.
- Evolutionary algorithms are ubiquitous nowadays, having been success-fully applied to numerous problems from different domains, including optimization, automatic programming, machine learning, operations research, bioinformatics, and social systems. Fig. Q.9.1 shows flow chart of an evolutionary algorithm.
- A population of candidate solutions is initialized. New solutions are created by applying reproduction operators. The fitness of the resulting solutions are evaluated and suitable selection strategy is then applied to determine which solutions will be maintained into the next generation.

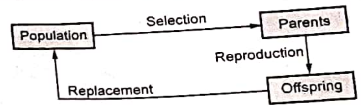


Fig. Q.9.1 Flow chart of an evolutionary algorithm

- The basic evolutionary programming method involves the following steps :

1. Choose an initial population. The number of solutions in a population is highly relevant to the speed of optimization, but no definite answers are available as to how many solutions are appropriate (other than>1) and how many solutions are just wasteful.
2. New offspring's are created by mutation. Each offspring solution is assessed by computing its fitness.

1.4 : Concept of Learning/Training

Q.10 Explain supervised learning.

Ans. : Supervised learning : • Supervised learning is the machine learning task of inferring a function from supervised training data. The training data consist of a set of training examples. The task of the supervised learner is to predict the output behavior of a system for any set of input values, after an initial training phase.

- **Supervised learning** in which the network is trained by providing it with input and matching output patterns. These input-output pairs are usually provided by an external teacher.
- Human learning is based on the past experiences. A computer does not have experiences.
- A computer system learns from data, which represent some "past experiences" of an application domain.
- To learn a target function that can be used to predict the values of a discrete class attribute, e.g., approve or not-approved, and high-risk or low risk. The task is commonly called : Supervised learning, Classification or inductive learning.

Scanned with CamScanner

- Training data includes both the input and the desired results. For some examples the correct results (targets) are known and are given in input to the model during the learning process. The construction of a proper training, validation and test set is crucial. These methods are usually fast and accurate.
- Have to be able to generalize : give the correct results when new data are given in input without knowing a priori the target.
- Supervised learning is the machine learning task of inferring a function from supervised training data. The training data consist of a set of training examples. In supervised learning, each example is a pair consisting of an input object and a desired output value.
- A supervised learning algorithm analyzes the training data and produces an inferred function, which is called a classifier or a regression function. Fig. Q.10.1 shows supervised learning process.
- The learned model helps the system to perform task better as compared to no learning.
- Each input vector requires a corresponding target vector.

Training Pair = (Input Vector, Target Vector)

- Fig. Q.10.2 shows input vector.
- Supervised learning denotes a method in which some input vectors are collected and presented to the network. The output computed by the net-work is observed and the deviation from the expected answer is measured. The weights are corrected according to the magnitude of the error in the way defined by the learning algorithm.
- Supervised learning is further divided into methods which use reinforcement or error correction. The perceptron learning algorithm is an example of supervised learning with reinforcement.
- In order to solve a given problem of supervised learning, following steps are performed :

1. Find out the type of training examples.
2. Collect a training set.
3. Determine the input feature representation of the learned function.
4. Determine the structure of the learned function and corresponding learning algorithm.
5. Complete the design and then run the learning algorithm on the collected training set.

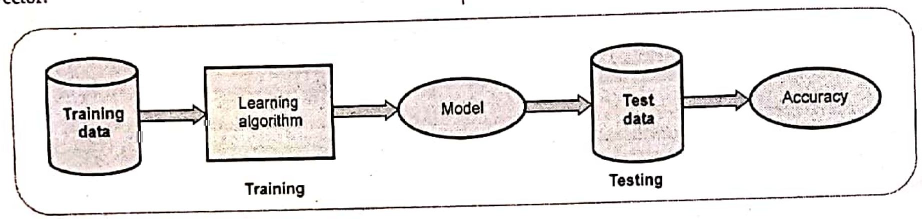


Fig. Q.10.1 Supervised learning process

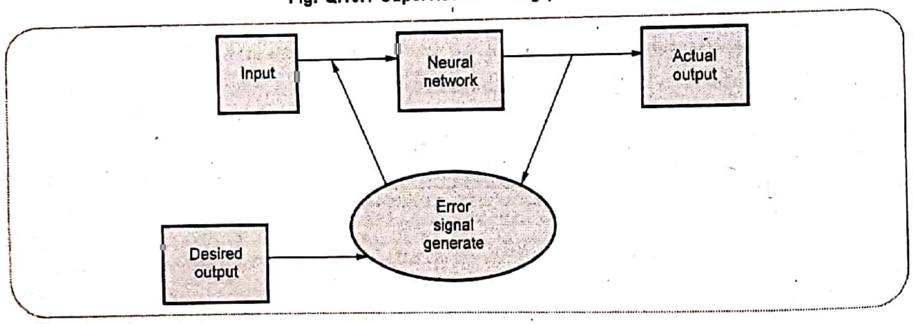


Fig. Q.10.2 Input vector

6. Evaluate the accuracy of the learned function. After parameter adjustment and learning, the performance of the resulting function should be measured on a test set that is separate from the training set.

Q.11 Explain unsupervised learning.

Ans. : Unsupervised learning : • The model is not provided with the correct results during the training. It can be used to cluster the input data in classes on the basis of their statistical properties only. Cluster significance and labeling.

- The labeling can be carried out even if the labels are only available for a small number of objects representative of the desired classes. All similar inputs patterns are grouped together as clusters.
- If matching pattern is not found, a new cluster is formed. There is no error feedback.
- External teacher is not used and is based upon only local information. It is also referred to as **self-organization.**
- They are called unsupervised because they do not need a teacher or super-visor to label a set of training examples. Only the original data is required to start the analysis.
- In contrast to supervised learning, unsupervised or self-organized learning does not require an external teacher. During the training session, the neural network receives a number of different input patterns, discovers significant features in these patterns and learns how to classify input data into appropriate categories.
- Unsupervised learning algorithms aim to learn rapidly and can be used in real-time. Unsupervised learning is frequently employed for data clustering, feature extraction etc.
- Another mode of learning called recording learning by Zurada is typically employed for associative memory networks. An associative memory networks is designed by recording several idea patterns into the networks stable states.

Q.12 What is semi-supervised learning ?

Ans. : Semi-supervised Learning : • Semi-supervised learning uses both labeled and unlabeled data to improve supervised learning. The goal is to learn a predictor that predicts future test data better than the predictor learned from the labeled training data alone.

- Semi-supervised learning is motivated by its practical value in learning faster, better, and cheaper.
- In many real world applications, it is relatively easy to acquire a large amount of unlabeled data x.
- For example, documents can be crawled from the Web, images can be obtained from surveillance cameras, and speech can be collected from broadcast. However, their corresponding labels y for the prediction task, such as sentiment orientation, intrusion detection, and phonetic transcript, often requires slow human annotation and expensive laboratory experiments.
- In many practical learning domains, there is a large supply of unlabeled data but limited labeled data, which can be expensive to generate. For example : text processing, video-indexing, bioinformatics etc.
- Semi-supervised Learning makes use of both labeled and unlabeled data for training, typically a small amount of labeled data with a large amount of unlabeled data. When unlabeled data is used in conjunction with a small amount of labeled data, it can produce considerable improvement in learning accuracy.
- Semi-supervised learning sometimes enables predictive model testing at reduced cost.
- **Semi-supervised classification :** Training on labeled data exploits additional unlabeled data, frequently resulting in a more accurate classifier.
- **Semi-supervised clustering :** Uses small amount of labeled data to aid and bias the clustering of unlabeled data.

Q.13 Compare and contrast Supervised, Unsupervised and Reinforcement learning.

☞[RGPV : May-18, Marks 7]

Ans. : • In supervised learning, the class labels in the dataset, which is used to build the classification model, are known. For example, a dataset for spam filtering would contain spam messages as well as "ham" (=not-spam) messages. In a supervised learning problem, we would know which message in the training set is spam or ham, and we'd use this information to train our model in order to classify new unseen messages.

Scanned with CamScanner

- In contrast, unsupervised learning task deal with unlabeled instances, and the classes have to be inferred from the unstructured dataset. Typically, unsupervised learning employs a clustering technique in order to group the unlabeled samples based on certain similarity (or distance) measures.
- In reinforcement learning, the model is learned from a series of actions by maximizing a "reward function". The reward function can either be maximized by penalizing "bad actions" and/or rewarding "good actions". A popular example of reinforcement learning would be the training of self-driving car using feedback from the environment.

Q.14 Distinguish between supervised learning and unsupervised learning. ☞[RGPV : Dec.-17, Marks 7]

Ans. :

Sr. No.	Supervised Learning	Unsupervised Learning
1.	Desired output is given.	Desired output is not given.
2.	It is not possible to learn larger and more complex models than with supervised learning	It is possible to learn larger and more complex models with unsupervised learning
3.	Use training data to infer model	No training data is used
4.	Every input pattern that is used to train the network is associated with an output pattern.	The target output is not presented to the network.
5.	Trying to predict a function from labeled data	Try to detect interesting relations in data.
6.	Supervised learning requires that the target variable is well defined and that a sufficient number of its values are given.	For unsupervised learning typically either the target variable is unknown or has only been recorded for too small a number of cases.
7.	Example: Optical character recognition	Example: Find a face in an image.
8.	We can test our model	We can not test our model
9.	Supervised learning is also called classification	Unsupervised learning is also called clustering.

Q.15 Explain training dataset and testing dataset.

Ans. : • Machine learning is about learning some properties of a data set and applying them to new data. This is why a common practice in machine learning to evaluate an algorithm is to split the data at hand in two sets, one that we call a training set on which we learn data properties and one that we call a testing set, on which we test these properties.

- In training data, data are assign the labels. In test data, data labels are unknown but not given. The training data consist of a set of training examples.
- The real aim of supervised learning is to do well on test data that is not known during learning. Choosing the values for the parameters that minimize the loss function on the training data is not necessarily the best policy.
- The training error is the mean error over the training sample. The test error is the expected prediction error over an independent test sample.
- Problem is that training error is not a good estimator for test error. Training error can be reduced by making the hypothesis more sensitive to training data, but this may lead to over fitting and poor generalization.
- **Training set :** A set of examples used for learning, where the target value is known.
- **Test set :** It is used only to assess the performances of a classifier. It is never used during the training process so that the error on the test set provides an unbiased estimate of the generalization error.
- Training data is the knowledge about the data source which we use to construct the classifier.

1.5 : Model of an Artificial Neuron

Q.16 Define the following :
a. Cell b. genotype c. phenotype

Ans. :

a. Cell : Animal or human cell is a complex of many small factories that work together. The center of all this is the cell nucleus. The genetic information is contained in the cell nucleus.

b. For a particular individual, the entire combination of genes is called genotype.

c. The phenotype describes the physical aspect of decoding a genotype to produce the phenotype

Q.17 Define structure and function of single neuron. ☞[RGPV : Dec.-16, Marks 2]

Ans. : • Artificial neural systems are inspired by biological neural systems. The elementary building block of biological neural systems is the neuron.

- Fig. Q.17.1 shows biological neural systems.
- The single cell neuron consists of the cell body or soma, the dendrites and the axon. The dendrites receive signals from the axons of other neurons.
- The small space between the axon of one neuron and the dendrite of another is the synapse. The afferent dendrites conduct impulses toward the soma. The efferent axon conducts impulses away from the soma.
- Basic Components of Biological Neurons

1. The majority of neurons encode their activations or outputs as a series of brief electrical pulses.
2. The neuron's cell body (soma) processes the incoming activations and converts them into output activations.
3. The neuron's nucleus contains the genetic material in the form of DNA. This exists in most types of cells, not just neurons.
4. Dendrites are fibres which emanate from the cell body and provide the receptive zones that receive activation from other neurons.
5. Axons are fibres acting as transmission lines that send activation to other neurons.
6. The junctions that allow signal transmission between the axons and dendrites are called synapses. The process of transmission is by diffusion of chemicals called neurotransmitters across the synaptic cleft

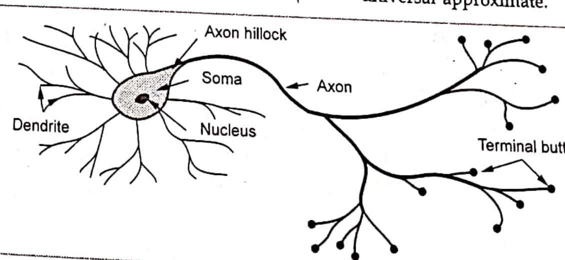


Fig. 1.17.1 Schematic of biological neuron

Q.18 What are the characteristics and application of ANN ? ☞[RGPV : Dec.-16, Marks 2]

Ans. : Characteristics of Artificial Neural Networks

1. Large number of very simple processing neuron-like processing elements.
2. Large number of weighted connections between the elements.
3. Distributed representation of knowledge over the connections.
4. Knowledge is acquired by network through a learning process.

Application of ANN :

1. Controlling the movements of a robot based on self-perception and other information;
2. Deciding the category of potential food items in an artificial world;
3. Recognizing a visual object;
4. Predicting where a moving object goes, when a robot wants to catch it.

Q.19 List out the strength and weakness of artificial neural network.

Ans. : Strength :

1. The greatest power of Neural Networks is that it is endowed with a finite number of hidden units, can yet approximate any continuous function to any desired degree of accuracy. This has been commonly referred to as the property of universal approximate.

Scanned with CamScanner

2. No prior knowledge of the data generating process is needed for implementing NN.
3. Problem of model misspecification does not occur.
4. In case of NN since no specifications are used as the network merely learns the hidden relationship in the data.
5. Adaptive learning : An ability to learn how to do tasks based on the data given for training or initial experience.
6. Self-Organisation : An ANN can create its own organisation or representation of the information it receives during learning time.

Weakness :

1. The addition of too many hidden units incites the problem of over fitting the data.
2. The construction of the NN model can be a time consuming process.

Q.20 Difference between digital computer and neural network.

Ans. :

Sr. No.	Digital computer	Neural network
1.	Deductive reasoning : We apply known rules to input data to produce output.	Inductive reasoning : Given input and output data (training examples), we construct the rules.
2.	Computation is centralized, synchronous and serial.	Computation is collective, asynchronous and parallel.
3.	Memory is packetted, literally stored and location addressable.	Memory is distributed, internalized and content addressable.
4.	Not fault tolerant. One transistor goes and it no longer works.	Fault tolerant, redundancy and sharing of responsibilities.
5.	Fast. Measured in millionths of a second.	Slow. Measured in thousandths of a second.
6.	Exact.	Inexact.
7.	Static connectivity.	Dynamic connectivity.
8.	Applicable if well defined rules with precise input data.	Applicable if rules are unknown or complicated or if data is noisy or partial.

Q.21 What is ANN ? Explain characteristics and applications of ANN. [RGPV : June-16, Marks 3]

Ans. : • Artificial Neural Network (ANN) is a computational system inspired by the structure, processing method, learning ability of a biological brain. An artificial neural network is composed of many artificial neurons that are linked together according to specific network architecture.

- The objective of the neural network is to transform the inputs into meaningful outputs.
- ANNs do not execute programmed instructions; they respond in parallel to the pattern of inputs presented to it. There are also no separate memory addresses for storing data.
- Instead, information is contained in the overall activation 'state' of the network. 'Knowledge' is thus represented by the network itself, which is quite literally more than the sum of its individual components.
- Fig. Q.21.1 shows artificial neural network.

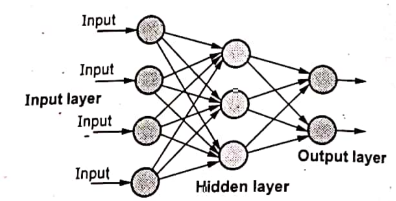


Fig. Q.21.1 Artificial neural network

- Elements of ANN are processing units, topology and learning algorithm.
- Also refer Q.18.

Q.22 Explain the neural network architectures. [RGPV : June-14, Marks 7]

Ans. : • The architecture of the neural network refers to the arrangement of the connection between neurons, processing element, number of layers, and the flow of signal in the neural network.

- There are mainly two category of neural network architecture :

a. Feed-forward
b. Feedback (recurrent) neural networks.

1. Architecture and Learning Rule

- In late 1950s, Frank Rosenblatt introduced a network composed of the units that were enhanced version of McCulloch-Pitts Threshold Logic Unit (TLU) model.
- Rosenblatt's model of neuron, a perceptron, was the result of merger between two concepts from the 1940s, McCulloch-Pitts model of an artificial neuron and Hebbian learning rule of adjusting weights.
- In addition to the variable weight values, the perceptron model added an extra input that represents bias. Thus, the modified equation is now as follows :

$$\text{Sum} = \sum_{i=1}^{N} I_i W_i + b,$$

where **b** represents the bias value.

- Fig. Q.22.1 shows a typical perception setup for pattern recognition applications, in which visual patterns are represented as matrices of elements between 0 and 1.

1. First layer act as a set of feature detectors that are hardwired to the input signals to detect specific features.
2. Second layer i.e. output layer takes the outputs of the feature detectors in the first layer and classifies the given input pattern.

- Learning is initiated by making adjustments to the relevant connection strengths and a threshold value θ.
- Here we consider only two class problem. Here output layer usually has only a single node. For an n-class problem (n > 3), the output layer usually has n-nodes, each corresponding to a class and the output node with the largest value indicates which class the input vector belongs to.
- In the first stage, the linear combination of inputs is calculated. Each value of input array is associated with its weight value, which is normally between 0 and 1. Also, the summation function often takes an extra input value Theta with weight value of 1 to represent threshold or bias of a neuron.

a. The term x_i is referred to as **active or excitatory** if its value is 1.
b. If the value is 0 then it is **inactive.**
c. If the value is –1 then it is **inhibitory.**

- The output unit is a linear threshold element with a threshold value θ :

$$0 = f\left(\sum_{i=1}^{n} w_i\, x_i - \theta\right)$$

$$= f\left(\sum_{i=1}^{n} w_i\, x_i + w_0\right), w_0 \equiv -\theta$$

$$= f\left(\sum_{i=1}^{n} w_i\, x_i\right), x_0 \equiv 1$$

where w_i is a modifiable weight associated with an incoming signal x_i.

- Fig. Q.22.2 shows the bias term w_0.
- The function y = f(x) describes relationship, an input-output mapping from x to y.

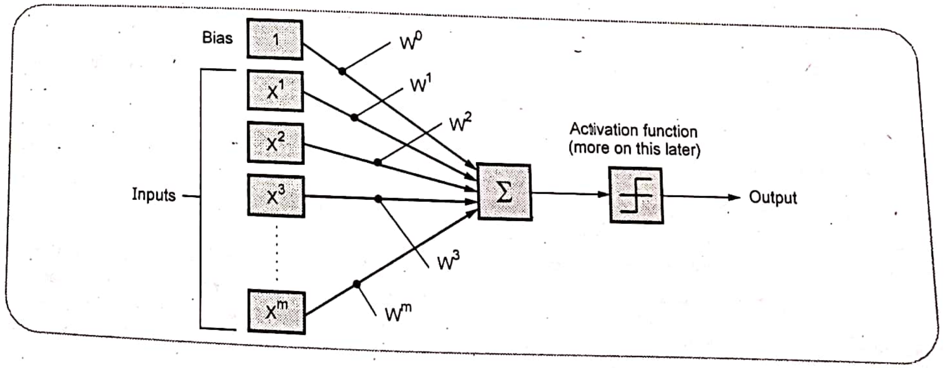


Fig. Q.22.1

Scanned with CamScanner

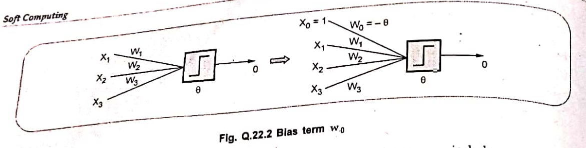


Fig. Q.22.2 Bias term w_0

- The equation (1), the f(.) is the **activation function** of the perceptron and it is typically either a **signum function** sgn(x) or **step function** step(x) :

$$\text{sgn}(x) = \begin{cases} 1 & \text{if } x > 0, \\ -1 & \text{otherwise} \end{cases}$$

$$\text{step}(x) = \begin{cases} 1 & \text{if } x > 0, \\ 0 & \text{otherwise} \end{cases} \qquad \text{... (1)}$$

- The sum-of-product value is then passed into the second stage to perform the activation function which generates the output from the neuron. The activation function "squashes" the amplitude of the output in the range of [0, 1] or [–1, 1] alternately. The behavior of the activation function will describe the characteristics of an artificial neuron model.
- The basic learning algorithm for a single layer perceptron repeats the following steps until the weights converge :

1. Select an input vector x from the training data set.
2. If the perceptron gives an incorrect response, modify all connection weights w_i according to

$$\Delta w_i = \eta t_i X_i$$

Where t_i is a target output and η is a learning state.

Q.23 Explain application of neural network.

Ans. : Neural network applications can be grouped in following categories :

1. **Clustering :** A clustering algorithm explores the similarity between patterns and places similar patterns in a cluster. Best known applications include data compression and data mining.
2. **Classification/Pattern recognition :** The task of pattern recognition is to assign an input pattern (like handwritten symbol) to one of many classes. This category includes algorithmic implementations such as associative memory.
3. **Function approximation :** The tasks of function approximation is to find an estimate of the unknown function f() subject to noise. Various engineering and scientific disciplines require function approximation.
4. **Prediction/Dynamical systems :** The task is to forecast some future values of a time-sequenced data. Prediciton has a significant impact on decision support systems. Prediction differs from function approximation by considering time factor.

1.6 : Activation Functions

Q.24 Define activation function.

Ans. : Activation function decides, whether a neuron should be activated or not by calculating weighted sum and further adding bias with it. The purpose of the activation function is to introduce non-linearity into the output of a neuron

Q.25 What is necessity of activation functions ?

Ans. : In a neural network, each neuron has an activation function which specifies the output of a neuron to a given input. Neurons are switches that output 1 when they are sufficiently activated and a 0 when not.

Q.26 Define activation function. Explain the purpose of activation function in multilayer neural networks. Give any two activation functions.

☞ [RGPV : May-18, Marks 7]

Ans. : • An activation function f performs a mathematical operation on the signal output. The activation functions are chosen depending upon the type of problem to be solved by the network. There are a number of common activation functions in use with neural networks.

- Fig. Q.26.1 shows position of activation function. Unit step function is one of the activation function.

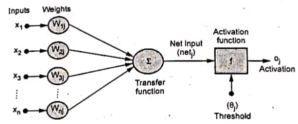


Fig. Q.26.1 Position of activation function

- The cell body itself is considered to have two functions. The first function is integration of all weighted stimuli symbolized by the summation sign. The second function is the activation which transforms the sum of weighted stimuli to an output value which is sent out through connection y.
- Typically the same activation function is used for all neurons in any particular layer. In a multi-layer network if the neurons have linear activation functions the capabilities are no better than a single layer network with a linear activation function. Hence in most cases nonlinear activation functions are used.

1. **Linear activation function :** The linear activation function will only produce positive numbers over the entire real number range. The linear activation function value is 0 if the argument is less than a lower boundary, increasing linearly from 0 to +1 for arguments equal or larger than the lower boundary and less than an upper boundary, and +1 for all arguments equal or greater than a given upper boundary.
2. **Sigmoid activation function :** The sigmoid function will only produce positive numbers between 0 and 1. The sigmoid activation function is most used for training data that is also between 0 and 1. It is one of the most used activation functions. A sigmoid function produces a curve with an "S" shape. Logistic and hyperbolic tangent functions are commonly used sigmoid functions. The sigmoid functions are extensively used in back propagation neural networks because it reduces the burden of complication involved during training phase.

$$\text{sig}(t) = \frac{1}{1+e^{-t}}$$

3. **Binary sigmoid :** The logistic function, which is a sigmoid function between 0 and 1 are used in neural network as activation function where the output values are either binary or varies from 0 to 1. It is also called as binary sigmoid or logistic sigmoid.

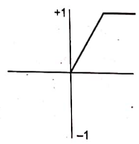


Fig. Q.26.2 (a) Linear

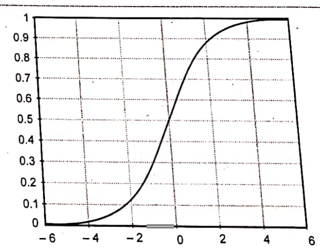


Fig. Q.26.2 (b) Sigmoid functions

Scanned with CamScanner

$$f(x) = \frac{1}{1+e^{-x}}$$

4. **Bipolar sigmoid :** A logistic sigmoid function can be scaled to have any range of values which may be appropriate for a problem. The most common range is from –1 to 1. This id called bipolar sigmoid.

$$f(x) = -1 + \frac{2}{1+e^{-x}}$$

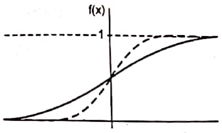


Fig. Q.26.2 (c) Binary sigmoid function

The bipolar sigmoid is also closely related to the hyperbolic tangent function.

$$\tan h(x) = \frac{\exp(x) - \exp(-x)}{\exp(x) + \exp(-x)}$$

$$= \frac{1 - \exp(-2x)}{1 + \exp(-2x)}$$

1.7 : Perceptron

Q.27 What is perceptron ?

Ans. : An arrangement of one input layer of McCulloch-Pitts neurons feeding forward to one output layer of McCulloch-Pitts neurons is known as a Perceptron.

- The perceptron is a feed - forward network with one output neuron that learns a separating hyper - plane in a pattern space.
- The "n" linear Fx neurons feed forward to one threshold output Fy neuron. The perceptron separates linearly set of patterns.

Q.28 Define the architecture of a perceptron. What do you mean by linear separability ?

[RGPV : May-18, Marks 7]

Ans. : Architecture of a perceptron :

- The perceptron is a feed-forward network with one output neuron that learns a separating hyper-plane in a pattern space. The "n" linear Fx neurons feed forward to one threshold output Fy neuron. The perceptron separates linearly set of patterns.
- SLP is the simplest type of artificial neural networks and can only classify linearly separable cases with a binary target (1 , 0).
- We can connect any number of McCulloch-Pitts neurons together in any way we like. An arrangement of one input layer of McCulloch-Pitts neurons feeding forward to one output layer of McCulloch-Pitts neurons is known as a **Perceptron.**
- A single layer feed-forward network consists of one or more output neurons, each of which is connected with a weighting factor W_{ij} to all of the inputs X_i.
- The Perceptron is a kind of a single-layer artificial network with only one neuron. The Percepton is a network in which the neuron unit calculates the linear combination of its real-valued or boolean inputs and passes it through a threshold activation function. Fig. Q.28.1 shows Perceptron.
- The Perceptron is sometimes referred to a Threshold Logic Unit (TLU) since it discriminates the data depending on whether the sum is greater than the threshold value.
- In the simplest case the network has only two inputs and a single output. The output of the neuron is :

$$y = f\left(\sum_{i=1}^{2} W_i X_i + b\right)$$

- Suppose that the activation function is a threshold then

$$f = \begin{cases} 1 & \text{if } s > 0 \\ -1 & \text{if } s \le 0 \end{cases}$$

- The Perceptron can represent most of the primitive boolean functions: AND, OR, NAND and NOR but can not represent XOR.
- In single layer perceptron, initial weight values are assigned randomly because it does not have previous knowledge. It sum all the weighted inputs. If the sum is greater than the threshold value then it is activated i.e. output = 1.

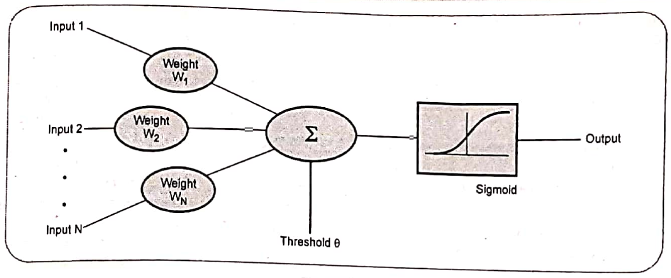


Fig. Q.28.1

Output

$$W_1X_1 + W_2X_2 + \ldots + W_nX_n > \theta \Rightarrow 1$$

$$W_1X_1 + W_2X_2 + \ldots + W_nX_n \le \theta \Rightarrow 0$$

- The input values are presented to the perceptron, and if the predicted output is the same as the desired output, then the performance is considered satisfactory and no changes to the weights are made.
- If the output does not match the desired output, then the weights need to be changed to reduce the error.
- The weight adjustment is done as follows :

$$\Delta W = \eta \times d \times x$$

where

X = Input data

d = Predicted output and desired output.

η = Learning rate

- If the output of the perceptron is correct then we do not take any action. If the output is incorrect then the weight vector is $W \rightarrow W + \Delta W$.
- The process of weight adaptation is called **learning.**
- Perceptron Learning Algorithm :

1. Select random sample from training set as input.
2. If classification is correct, do nothing.
3. If classification is incorrect, modify the weight vector W using

$$W_i = W_i + \eta d(n) X_i(n)$$

Repeat this procedure until the entire training set is classified correctly.

linear separability :

- Consider two - input pattern (X_1, X_2) being classified into two classes as shown in Fig. Q.28.2 Each point with either symbol of x or 0 represents a pattern with a set of values (X_1, X_2).

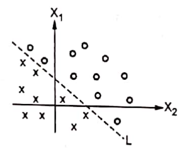


Fig. Q.28.2

- Each pattern is classified into one of two classes. Notice that these classes can be separated with a single line L. They are known as linearly separable patterns.
- Linear separability refers to the fact that classes of patterns with η - dimensional vector $x = (x_1, x_2, \ldots, x_n)$ can be separated with a single decision surface. In the case above, the line L represents the decision surface.

Scanned with CamScanner

- If two classes of patterns can be separated by a decision boundary, represented by the linear equation then they are said to be linearly separable. The simple network can correctly classify any patterns.
- Decision boundary (i.e. W, b or q) of linearly separable classes can be determined either by some learning procedures or by solving linear equation systems based on representative patterns of each classes.
- In such a decision boundary does not exist, then the two classes are said to be linearly inseparable.
- Linear inseparable problems cannot be solved by te simple network, more sophisticated architecture is needed.

Q.29 Discuss perceptron training algorithm. **[RGPV : Dec.-16, Marks 3]**

Ans. : Refer Q.28

Q.30 What is multilayer perceptron ? Explain different types of activation functions. **[RGPV : Dec.-16, Marks 7]**

Ans. : • A multilayer perceptron (MLP) has the same structure of a single layer perceptron with one or more hidden layers. An MLP is a network of simple neurons called perceptrons.

- A typical multilayer perceptron network consists of a set of source nodes forming the input layer, one or more hidden layers of computation nodes, and an output layer of nodes.
- It is not possible to find weights which enable single layer perceptrons to deal with non - linearly separable problems like XOR :

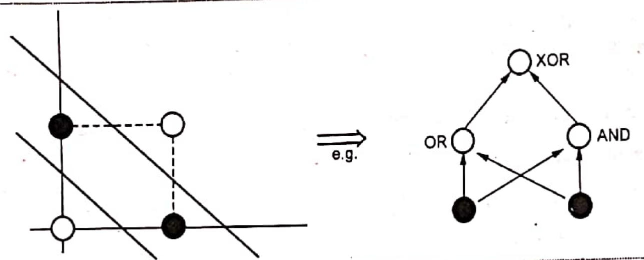


Fig. Q.30.1

- Multi - layer perceptrons are able to cope with non - linearly separable problems.
- Each neuron in one layer has direct connections to all the neurons of the subsequent layer. MLP can be implement non linear discriminants (for classification) and non linear regression functions (for regression).
- Historically, the problem was that there were no known learning algorithms for training MLP's. Fortunately; it is now known to be quite straightforward. The procedure for finding a gradient vector in the network structure is generally referred to as **backpropagation**. Because the gradient vector is calculated in the direction opposite to the flow of the output of each node.

Types of activation functions : Refer Q.26

Q.31 What is linear separability ? Explain why single layer neural network is unable to solve the problem of linear separability. Also state how it is solved ? **[RGPV : June-16, Marks 7]**

Ans. : Refer Q.28

Important Points to Remember

- Soft computing is an emerging approach to computing which parallel the remarkable ability of human mind to reason and learn in a environment of uncertainty and imprecision.
- Soft computing are of two types: soft computing and hard computing
- One of the characteristics of soft computing methods is that they are typically used in problems where mathematical models are not available or are intractable or too cumbersome to be viable.

- The principal constituents, i.e., tools, techniques, of Soft Computing (SC) are Fuzzy Logic (FL), Neural Networks (NN), Support Vector Machines (SVM), Evolutionary Computation (EC), and Machine Learning (ML) and Probabilistic Reasoning (PR).
- Evolutionary techniques can be viewed either as search methods, or as optimization techniques
- Uncertainty is described by probabilities. Probability may be use for simulation of fuzziness
- Neural networks consists of many number of simple elements (neurons) connected between them in system.
- Hard computing uses binary logic and soft computing uses fuzzy logic.

END...

Scanned with CamScanner

2 Multilayer Neural Networks

2.1 : Multilayer Neural Networks : Feed Forward Network

Q.1 Explain in brief architecture of multilayer feed-forward neural network.

Ans. : • A multilayer feed-forward neural network is a network consisting of multiple layers of units, all of which are adaptive.

- The network is not allowed to have cycles from later layers back to earlier layers, hence the name "feed-forward".
- Let us consider a network with a single complete hidden layer. i.e., the network consists of some input nodes, some output nodes, and a set of hidden nodes.
- Every hidden node takes inputs from each of the input nodes, and feeds into each of the output nodes.
- Fig. Q.1.1 shows multilayer feed forward neural network.

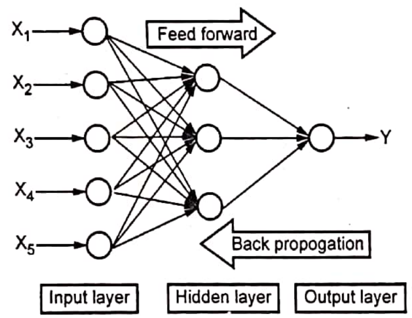


Fig. Q.1.1 Multilayer feed forward neural network

- This structure is called multilayer because it has a layer of processing units (i.e., the hidden units) in addition to the output units.
- These networks are called feedforward because the output from one layer of neurons feeds forward into the next layer of neurons. There are never any backward connections, and connections never skip a layer.
- Each connection between nodes has a weight associated with it. In addition, there is a special weight (called w_0) that feeds into every node at the hidden layer and a special weight (called z_0) that feeds into every node at the output layer.
- These weights are called the bias, and set the thresholding values for the nodes. Initially, all of the weights are set to some small random values near zero.
- Every node in the hidden layer and in the output layer processes its weighted input to produce an output. This can be done slightly differently at the hidden layer, compared to the output layer.
- Input units : The input data you provide your network comes through the input units. No processing takes place in an input unit, it simply feeds data into the system.
- Hidden units : The connections coming out of an input unit have weights associated with them. A weight going to hidden unit z_h from input unit x_j would be labeled w_{hj} . The bias input node (x_0) is connected to all the hidden units, with weights w_{h0}.
- Each hidden node calculates the weighted sum of its inputs and applies a thresholding function to determine the output of the hidden node. The weighted sum of the inputs for hidden node z_h is calculated as :

$$\sum_{j=0}^{d} w_{hj} x_j$$

- The thresholding function applied at the hidden node is typically either a step function or a sigmoid function. The sigmoid function is sometimes called the "squashing" function, because it squashes its input (i.e., a) to a value between 0 and 1.

Scanned with CamScanner

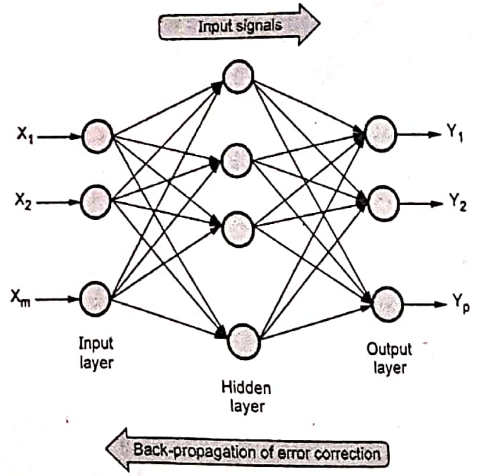


Fig. Q.4.2 Two layer backpropagation MLP

- The above backpropagation MLP will refer to as a 3-4-3 network, corresponding to the number of nodes in each layer.
- The backward error propagation also known as the backpropagation (BP) or the Generalized Delta Rule (GDR). A squared error measure for the p^{th} input-output pair is defined as

$$E_p = \sum_k (d_k - x_k)^2$$

Where d_k is the desired output for node k and x_k is the actual output for node k when the input part of the p^{th} data pair is presented.

- To find the gradient vector, an error term $\bar{\epsilon}_i$ for node i is defined as :

$$\bar{\epsilon}_i = \frac{\partial + E_p}{\partial \overline{X}_i}$$

- The partial derivative can be rewritten as product of two terms using chain rule for partial differentiation :

$$\frac{\partial \bar{E}(t)}{\partial w_{ij}(t)} = \frac{\partial E(t)}{\partial a_i(t)} \bullet \frac{\partial a_i(t)}{\partial w_{ij}(t)}$$

Q.5 Write the characteristics and applications of error back propagation algorithm.

☞[RGPV : May-18, Marks 7]

OR Describe error back propagation algorithm with its characteristics. ☞[RGPV : Dec-16, Marks 7]

Ans. : Characteristics :

- It is an algorithm for supervised learning of artificial neural networks using gradient descent.
- Learns weights for a multilayer network, given a fixed set of units and interconnections.
- In multilayer networks the error surface can have multiple minima, but in practice backpropagation has produced excellent results in many real-world applications
- The algorithm is for two layers of sigmoid units and does stochastic gradient descent
- It uses gradient descent to minimize the squashed error between the network outputs and the target values for these outputs.

Applications :

- The fast development of artificial satellite technology has increased the importance of three dimensional (3D) positioning and therefore, satellite geodesy.
- Particularly, the Global Positioning System (GPS) provides more practical, rapid, precise and continuous positioning results anywhere on the Earth in geodetic applications when compared to the traditional terrestrial positioning methods.
- Due to the increasing use of GPS positioning techniques, a great attention has been paid to the precise determination of local/regional geoid aiming at replacing the geometric leveling with GPS measurements.
- Therefore, BPANN method is easily programmable with decreased and increased number of reference points when generating a local GPS, it performs flexible modelling.
- Also, BPANN method is open to updating which could be accepted as an important advantage. Thus it is believed that BPANN method is more convenient for generating local GPS when compared to other methods.

Q.6 List out merits and demerits of EBP.

Ans. : Merits/Strength :

1. Computing time is reduced if weight chosen are small at the beginning.
2. It minimize the error

Scanned with CamScanner

- Locally, the negated gradient is the steepest descent direction, i.e., the direction that x would need to move in order to decrease "f" the fastest. The algorithm typically converges to a local minimum, but may rarely reach a saddle point, or not move at all if x_1 lies at a local maximum.
- The gradient will give the slope of the curve at that x and its direction will point to an increase in the function. So we change x in the opposite direction to lower the function value :

$$X_{k+1} = x_k - \lambda \nabla f(x_k)$$

The $\lambda > 0$ is a small number that forces the algorithm to make small jumps

Limitations of Gradient Descent :

- Gradient descent is relatively slow close to the minimum: technically, its asymptotic rate of convergence is inferior to many other methods.
- For poorly conditioned convex problems, gradient descent increasingly 'zigzags' as the gradients point nearly orthogonally to the shortest direction to a minimum point.

Steepest Descent :

- Steepest descent is also known as gradient method.
- This method is based on first order Taylor series approximation of objective function. This method is also called saddle point method. Fig. Q.8.1 shows steepest descent method.
- The Steepest Descent is the simplest of the gradient methods. The choice of direction is where f decreases most quickly, which is in the direction opposite to $\nabla f(x_i)$. The search starts at an arbitrary point x_0 and then go down the gradient, until reach close to the solution.

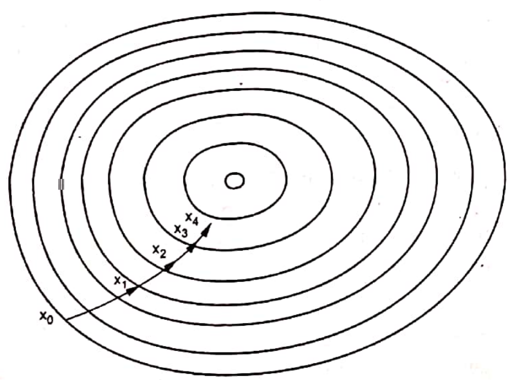


Fig. Q.8.1 Steepest descent method

- The method of steepest descent is the discrete analogue of gradient descent, but the best move is computed using a local minimization rather than computing a gradient. It is typically able to converge in few steps but it is unable to escape local minima or plateaus in the objective function.
- The gradient is everywhere perpendicular to the contour lines. After each line minimization the new gradient is always orthogonal to the previous step direction. Consequently, the iterates tend to zig-zag down the valley in a very inefficient manner.
- The method of Steepest Descent is simple, easy to apply, and each iteration is fast. It also very stable; if the minimum points exist, the method is guaranteed to locate them after at least an infinite number of iterations.

2.3 : Feedback Network - Hopfield Nets

Q.9 What is discrete hopfield network ?

Ans. : A Hopfield network which operates in a discrete line fashion or in other words, it can be said the input and output patterns are discrete vector, which can be either binary (0,1) or bipolar (+1, – 1) in nature. The network has symmetrical weights with no self-connections i.e., $w_{ij} = w_{ji}$ and $w_{ii} = 0$.

Q.10 What is continuous hopfield network ?

Ans. : In comparison with discrete hopfield network continuous network has time as a continuous variable. It is also used in auto association and optimization problems such as travelling salesman problem.

Q.11 Explain with neat diagram the architecture of hopfield neural network.

Ans. : • The Hopfield model is a single-layered recurrent network. Like the associative memory, it is usually initialized with appropriate weights instead of being trained.

$$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}}\int_0^x \exp(-s^2)\,ds$$

Therefore, given N and p we can find out the probability P_{error} of error for a single neuron of a stored pattern.

Q.13 Give the limitation and application of Hopfield network. [RGPV : June-13, Marks 7]

Ans. : Limitations of Hopfield networks

- Severely limited in the number of patterns p that can be stored reliably in N nodes.
- Generally, the maximum number of patterns must be below 0.15N for reliable performance;
- Avalanche in error occurs above 0.138N.
- Too many patterns will result in spurious outputs; i.e., outputs not corresponding to any stored pattern.
- Storing similar patterns can cause errors in output.

Application

- Common applications are those where pattern recognition is useful, and Hopfield networks have been used for image detection and recognition, enhancement of X-Ray images, medical image restoration, etc.
- The Hopfield network is commonly used for auto-association and optimization tasks.

Q.14 Discuss energy function of Hopfield network.

Ans. : • An energy function is defined as a function that is bonded and non-increasing function of the state of the system.

- Energy function also called Lyapunov function determines the stability of discrete Hopfield network.
- There is a relationship between dynamics of Hopfield network and an energy function.
- Consider the function :

$$H = -\frac{1}{2}\sum_{i,j=1}^{N} w_{ij} S_i S_j$$

- Consider that our weights are picked such that the stored patterns represent local minimums of the function H.
- Matching an input to a stored pattern is like minimization of H; "fall into the closest minimum". The stability condition is akin to being stuck in the local minimum.

Q.15 Explain training and testing algorithm

Ans. : Training algorithm :

- There exist several rules to train a Hopfield network all of which are based on the same process ; the modification of the synaptic weights.
- Given an input to learn, the connectivity weight between nodes with the same state is increased whereas the weight between nodes with opposite states is decreased.
- A learning rule can have two useful properties : being local and being incremental.
- It is local if for each update it uses only the information from both nodes whose connectivity weight is updated.
- It is incremental if it does not require any information about the other patterns previously learnt by the network.
- Another major property of a learning rule is the capacity. It corresponds to the number of patterns per neuron that can be recalled by the network.
- During training of discrete Hopfield network weights will be updated. As we know that we can have the binary input vectors as well as bipolar input vectors. Hence, in both the cases, weight updates can be done with the following relation

Case 1 : Binary input patterns

- For a set of binary patterns **s(p), p = 1 to P. Here** $s(p) = s_1(p), s_2(p), \ldots, s_i(p), \ldots, s_n(p)$.
- Weight matrix is given by

$$w_{ij} = \sum_{p=1}^{P}[2s_i(p)-1][2s_j(p)-1] \quad \text{for } i \neq j$$

Case 2 : Bipolar input patterns

- For a set of binary patterns s(p), p = 1 to P
- Here, $s(p) = s_1(p), s_2(p), \ldots, s_i(p), \ldots, s_n(p)$
- Weight matrix is given by

$$w_{ij} = \sum_{p=1}^{P}[s_i(p)][s_j(p)] \quad \text{for } i \neq j$$

Q.16 State and explain the Hebb's learning rule for neural network. [RGPV : May-18, Marks 7]

Ans. : • In 1949, Donald Hebb proposed one of the key ideas in biological learning, commonly known as **Hebb's Law.** Hebb's Law states that if neuron i is near enough to excite neuron j and repeatedly participates in its activation, the synaptic connection between these two neurons is strengthened and neuron j becomes more sensitive to stimuli from neuron i.

- Hebb's Law can be represented in the form of two rules :
  1. If two neurons on either side of a connection are activated synchronously, then the weight of that connection is increased.
  2. If two neurons on either side of a connection are activated asynchronously, then the weight of that connection is decreased.
- Hebb's Law provides the basis for learning without a teacher. Learning here is a **local phenomenon** occurring without feedback from the environment.
- Using Hebb's Law we can express the adjustment applied to the weight w_{ij} at iteration p in the following form :

$$\Delta w_{ij}(p) = F[y_i(p), x_i(p)]$$

- As a special case, we can represent Hebb's Law as follows :

$$\Delta w_{ij}(p) = \alpha\, y_i(p)\, x_i(p)$$

where α is the learning rate parameter. This equation is referred to as the **activity product rule.**

- Hebbian learning implies that weights can only increase. To resolve this problem, we might impose a limit on the growth of synaptic weights. It can be done by introducing a non-linear **forgetting factor** into Hebb's Law :

$$\Delta w_{ij}(p) = \alpha\, y_i(p)\, x_i(p) - \varphi\, y_i(p)\, w_{ij}(p)$$

where φ is the forgetting factor.

- Forgetting factor usually falls in the interval between 0 and 1, typically between 0.01 and 0.1, to allow only a little "forgetting" while limiting the weight growth.

Hebbian learning algorithm

Step 1 : Initialisation

Set initial synaptic weights and thresholds to small random values, say in an interval [0, 1].

Step 2 : Activation

Compute the neuron output at iteration p

$$y_j(p) = \sum_{i=1}^{n} x_i(p)\, w_{ij}(p) - \theta_j$$

where n is the number of neuron inputs, and θ_j is the threshold value of neuron j.

Step 3 : Learning

Update the weights in the network :

$$w_{ij}(p+1) = w_{ij}(p) + \Delta w_{ij}(p)$$

where $w_{ij}(p)$ is the weight correction at iteration p. The weight correction is determined by the generalised activity product rule :

$$w_{ij}(p) = \varphi\, y_j(p)\,[\lambda x_i(p) - w_{ij}(p)]$$

Step 4 : Iteration

Increase iteration p by one, go back to Step 2.

Hebbian learning example

- To illustrate Hebbian learning, consider a fully connected feed forward network with a single layer of five computation neurons. Each neuron is represented by a McCulloch and Pitts model with the sign activation function. The network is trained on the following set of input vectors :

$$x_1 = \begin{bmatrix}0\\0\\0\\0\\0\end{bmatrix} \quad x_2 = \begin{bmatrix}0\\1\\0\\0\\1\end{bmatrix}$$

$$x_3 = \begin{bmatrix}0\\0\\0\\1\\0\end{bmatrix} \quad x_4 = \begin{bmatrix}0\\0\\1\\0\\0\end{bmatrix} \quad x_5 = \begin{bmatrix}0\\1\\0\\0\\1\end{bmatrix}$$

$$\begin{array}{c|ccccc} & ① & ② & ③ & ④ & ⑤ \\ \hline 1 & 1 & 0 & 0 & 0 & 0 \\ 2 & 0 & 1 & 0 & 0 & 0 \\ 3 & 0 & 0 & 1 & 0 & 0 \\ 4 & 0 & 0 & 0 & 1 & 0 \\ 5 & 0 & 0 & 0 & 0 & 1 \end{array}$$

(Rows: Input layer; columns: Output layer)

$$\begin{array}{c|ccccc} & ① & ② & ③ & ④ & ⑤ \\ \hline 1 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 2.0204 & 0 & 0 & 2.0204 \\ 3 & 0 & 0 & 1.0200 & 0 & 0 \\ 4 & 0 & 0 & 0 & 0.9996 & 0 \\ 5 & 0 & 2.0204 & 0 & 0 & 2.0204 \end{array}$$

(Rows: Input layer; columns: Output layer)

Initial and final weight matrices

- A test input vector, or probe, is defined as

$$X = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

- When this probe is presented to the network, we obtain :

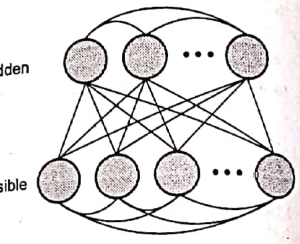
$$Y = \left\{ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 2.0204 & 0 & 0 & 2.0204 \\ 0 & 0 & 1.0200 & 0 & 0 \\ 0 & 0 & 0 & 0.9996 & 0 \\ 0 & 2.0204 & 0 & 0 & 2.0204 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} - \begin{bmatrix} 0.4940 \\ 0.2661 \\ 0.0907 \\ 0.9478 \\ 0.0737 \end{bmatrix} \right.$$

$$= \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

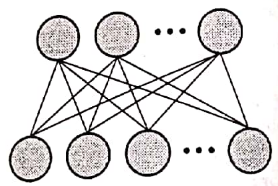
Q.17 What is Boltzmann machine ? With neat sketch explain its architecture.

Ans. : • The Boltzmann machine is among the first multilayer learning machine inspired by statistical mechanics. The primary goal of Boltzmann learning is to produce a neural network that correctly models input patterns according to a Boltzmann distribution.

- A Boltzmann machine is a network of symmetrically coupled stochastic binary units.
- The Boltzmann machine consists of stochastic neurons. A stochastic neuron resides in one of two possible states (± 1) in a probabilistic manner.
- The use of symmetric synaptic connections between neurons. The stochastic neurons partition into two functional groups: visible and hidden.



(a) Boltzmann machines



(b) Restricted Boltzmann machines

Fig. Q.17.1

- During the training phase of the network, the visible neurons are all clamped onto specific states determined by the environment.
- The hidden neurons always operate freely; they are used to explain underlying constraints contained in the environmental input vectors.
- This is accomplished by capturing higher-order statistical correlations in the clamping vectors. The network can perform pattern completion provided that it has learned the training distribution properly.
- Boltzmann machine is characterized by an energy function :

$$E = -\frac{1}{2}\sum_{j}\sum_{k \neq j} w_{kj}\, x_k\, x_j$$

- Free neurons change state with probability :

$$P(+x_k \to -x_k) = \frac{1}{1+\exp(-\Delta E_k / T)}$$

- The learning rule is given by :

$$\Delta w_{kj} = \eta\,(p_{kj}^{+} - p_{kj}^{-}) \quad j \neq k$$

- Property of the Boltzmann Machines (BM)
  1. If we run BM by repeatedly choosing a neuron and finding its next state with the above mentioned formula, the probability of the state will become stationary, and will follow the Boltzmann distribution.
  2. To guarantee the convergence of the state, we should set a relatively high temperature, and reduce it gradually. This process is called simulated annealing.

2.4 : Self-organizing Maps

Q.18 What is self-organizing feature maps ? List and explain its components.

Ans. : • In competitive learning, neurons compete among themselves to be activated. While in Hebbian learning, several output neurons can be activated simultaneously, in competitive learning, only a single output neuron is active at any time. The output neuron that wins the "competition" is called the winner-takes-all neuron.

- Such competition can be implemented by having lateral inhibition connections between the neurons. The result is that the neurons are forced to organise themselves. For obvious reasons, such a network is called a Self Organizing Map (SOM).
- The Self-Organizing Map is based on unsupervised learning, which means that no human intervention is needed during the learning and that little needs to be known about the characteristics of the input data.
- It provides a topology preserving mapping from the high dimensional space to map units. Map units or neurons, usually form a two-dimensional lattice and thus the mapping is a mapping from high dimensional space onto a plane.
- Brain is a self-organizing system that can learn by itself by changing (adding, removing, strengthening) the interconnections between neurons. Formation of feature maps in the brain that have a linear or planar topology.
- Use the SOM for clustering data without knowing the class memberships of the input data. The SOM can be used to detect features inherent to the problem and thus has also been called the **Self-Organizing Feature Map.**
- The property of topology preserving means that the mapping preserves the relative distance between the points. Points that are near each other in the input space are mapped to nearby map units in the SOM. The SOM can thus serve as a cluster analyzing tool of high-dimensional data. Also, the SOM has the capability to generalize.
- Generalization capability means that the network can recognize or characterize inputs it has never encountered before. A new input is assimilated with the map unit it is mapped to. Fig. Q.18.1 shows organization of the mapping.

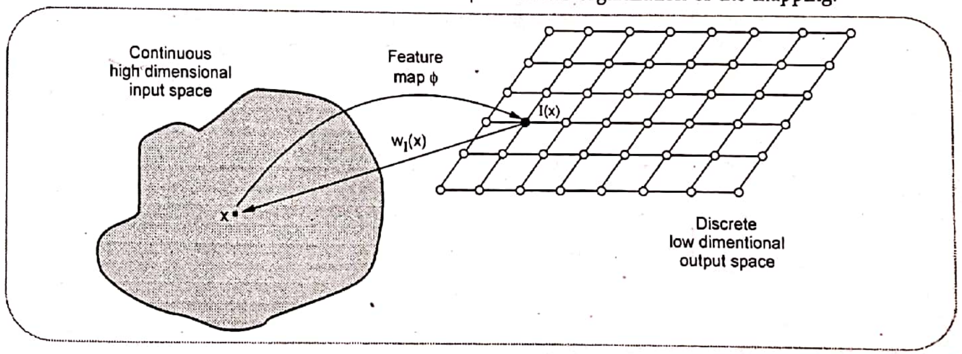


Fig. Q.18.1 Organization of the mapping

Scanned with CamScanner

- Here points x in the input space mapping to points I(x) in the output space. Each point "I" in the output space will map to a corresponding point w(I) in the input space.

Components of Self Organization

- The self-organization process involves four major components :

1. Initialization : All the connection weights are initialized with small random values.
2. Competition : For each input pattern, the neurons compute their respective values of a discriminant function which provides the basis for competition. The particular neuron with the smallest value of the discriminant function is declared the winner.
3. Cooperation : The winning neuron determines the spatial location of a topological neighbourhood of excited neurons, thereby providing the basis for cooperation among neighbouring neurons.
4. Adaptation : The excited neurons decrease their individual values of the Discriminant function in relation to the input pattern through suitable adjustment of the associated connection weights, such that the response of the winning neuron to the subsequent application of a similar input pattern is enhanced.

Q.19 Write short note on Kohonen self organizing.

Ans. : • Kohonen self organizing networks are also called **Kohonen features maps** or **topology preserving maps** are used to solve competition based network paradigm for data clustering.

- The Kohonen model provides a topological mapping. It places a fixed number of input patterns from the input layer into a higher-dimensional output or Kohonen layer.
- Training in the Kohonen network begins with the winner's neighbourhood of a fairly large size. Then, as training proceeds, the neighbourhood size gradually decreases.
- Fig. Q.19.1 shows a simple Kohonen self organizing network with 2 inputs and 49 outputs. The learning feature map is similar to that of competitive learning networks.
- A similarity measure is selected and the winning unit is considered to be the one with the largest activation. For this **Kohonen features** maps all the weights in a neighborhood around the winning units are also updated. The neighborhood's size generally decreases slowly with each iteration.
- Step for how to train a Kohonen self organizing network is as follows :

For n-dimensional input space and m output neurons :

1. Choose random weight vector w_i for neuron i, $i = 1, \ldots, m$
2. Choose random input x
3. Determine winner neuron k : $||w_k - x|| = \min_i ||w_i - x||$ (Euclidean distance)

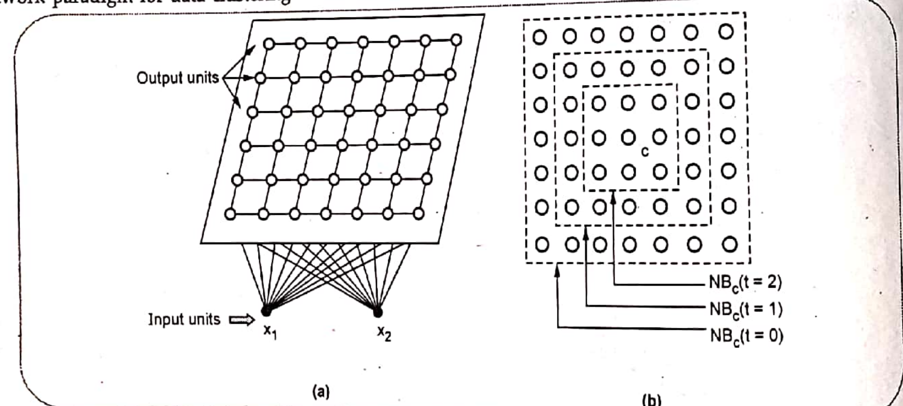


Fig. Q.9.1 Simple Kohonen self organizing network

4. Update all weight vectors of all neurons i in the neighborhood of neuron k : $w_i := w_i + \eta \cdot \varphi(i, k) \cdot (x - w_i)$ (w_i is shifted towards x)
5. If convergence criterion met, STOP. Otherwise, narrow neighborhood function and learning parameter η and go to (2).

Competitive learning in the Kohonen network

- To illustrate competitive learning, consider the Kohonen network with 100 neurons arranged in the form of a two-dimensional lattice with 10 rows and 10 columns. The network is required to classify two-dimensional input vectors - each neuron in the network should respond only to the input vectors occurring in its region.
- The network is trained with 1000 two-dimensional input vectors generated randomly in a square region in the interval between –1 and +1. The learning rate parameter a is equal to 0.1.

2.5 : Introduction to CNN and RNN Network

Q.20 What is Convolutional Neural Networks (CNN) ?

Ans. : • Convolutional Neural Network is a class of deep neural network that is used for computer vision or analyzing visual imagery.

- The Convolutional Layer makes use of a set of learnable filters. A filter is used to detect the presence of specific features or patterns present in the original image (input). It is usually expressed as a matrix (MxMx3), with a smaller dimension but the same depth as the input file.
- This filter is convolved across the width and height of the input file, and a dot product is computed to give an activation map.
- Different filters which detect different features are convolved on the input file and a set of activation maps is outputted which is passed to the next layer in the CNN.
- Convolutional networks perceive images as volumes; i.e. three-dimensional objects, rather than flat canvases to be measured only by width and height.
- That's because digital color images have a red-blue-green (RGB) encoding, mixing those three colors to produce the color spectrum humans perceive.
- A convolutional network ingests such images as three separate strata of color stacked one on top of the other.
- So a convolutional network receives a normal color image as a rectangular box whose width and height are measured by the number of pixels along those dimensions, and whose depth is three layers deep, one for each letter in RGB. Those depth layers are referred to as channels.
- Now, for each pixel of an image, the intensity of R, G and B will be expressed by a number, and that number will be an element in one of the three, stacked two-dimensional matrices, which together form the image volume.
- Those numbers are the initial, raw, sensory features being fed into the convolutional network, and the ConvNets purpose is to find which of those numbers are significant signals that actually help it classify images more accurately.
- Rather than focus on one pixel at a time, a convolutional net takes in square patches of pixels and passes them through a filter. That filter is also a square matrix smaller than the image itself, and equal in size to the patch. It is also called a kernel, which will ring a bell for those familiar with support-vector machines, and the job of the filter is to find patterns in the pixels.

Q.21 Why use Convolutional Neural Networks (CNN) ?

Ans. :

1. Ruggedness to shifts and distortion in the image : Detection using CNN is rugged to distortions such as change in shape due to camera lens, different lighting conditions, different poses, presence of partial occlusions, horizontal and vertical shifts, etc.
2. Fewer memory requirements
3. Easier and better training : In a CNN, since the number of parameters is drastically reduced, training time is proportionately reduced.

Scanned with CamScanner

Q.22 What is RNN ?

Ans. : • A recurrent neural network (RNN) is a type of artificial neural network commonly used in speech recognition and Natural Language Processing (NLP).

- RNNs are designed to recognize a data's sequential characteristics and use patterns to predict the next likely scenario. RNNs are used in deep learning and in the development of models that simulate the activity of neurons in the human brain.
- The simplest form of fully recurrent neural network is an MLP with the previous set of hidden unit activations feeding back into the network along with the inputs :

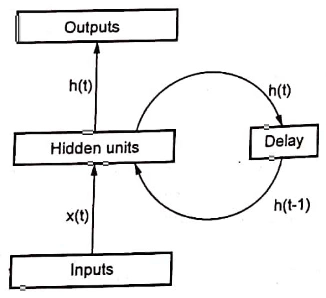


Fig. Q.22.1

- Note that the time t has to be discretized, with the activations updated at each time step.
- The time scale might correspond to the operation of real neurons, or for artificial systems any time step size appropriate for the given problem can be used. A delay unit needs to be introduced to hold activations until they are processed at the next time step.
- If the neural network inputs and outputs are the vectors x(t) and y(t), the three connection weight matrices are W_{IH}, W_{HH} and W_{HO}, and the hidden and output unit activation functions are f_H and f_O, the behaviour of the recurrent network can be described as a dynamical system by the pair of non-linear matrix equations :

$$h(t) = f_H(W_{IH}x(t) + W_{HH}h(t-1))$$

$$y(t) = f_O(W_{HO}h(t))$$

- In general, the state of a dynamical system is a set of values that summarizes all the information about the past behaviour of the system that is necessary to provide a unique description of its future behaviour, apart from the effect of any external factors.
- In this case the state is defined by the set of hidden unit activations h(t).
- Thus, in addition to the input and output spaces, there is also a state space. The order of the dynamical system is the dimensionality of the state space, the number of hidden units.

Important Points to Remember

- A multilayer feed-forward neural network is a network consisting of multiple layers of units, all of which are adaptive.
- The back-propagation neural network is a solution to the problem of training multi-layer perceptrons.
- Error Back propagation algorithm is for supervised learning of artificial neural networks using gradient descent.
- Hopfield network which operates in a discrete line fashion
- An energy function is defined as a function that is bonded and non-increasing function of the state of the system
- Convolutional Neural Network is a class of deep neural network that is used for Computer Vision or analyzing visual imagery
- A recurrent neural network (RNN) is a type of artificial neural network commonly used in speech recognition and natural language processing (NLP).
- RNNs are designed to recognize a data's sequential characteristics and use patterns to predict the next likely scenario

END...

3 Fuzzy Systems

3.1 : Fuzzy Set Theory

Q.1 What is fuzzy set ?

Ans. : • A fuzzy set is a collection of objects with graded membership. Two examples of "Sets" :

1. All employees of XYZ who are over 1.8 meter in height.
2. All employees of XYZ who are tall.

- The first example is a classical set, we have a universe (all XYZ employees) and a membership rule that divides the universe into members (those over 1.8 m) and nonmembers.
- The second example is a fuzzy set, some employees are definitely in the set and some are definitely not in the set, but some are "borderline". This distinction between the "ins", the "outs", and the "borderlines" is made more exact by the membership function, $\mu_A(x)$.

Q.2 Differentiate between crisp relations and fuzzy relations. [RGPV : Dec-16, Marks 2]

Ans. : • Crisp Relation R from a set A to a set B assigns to each ordered pair exactly one of the following statements:)"a is related to b" or "a is not related to b".

- Fuzzy relations map elements of one universe (U) to those of another universe (V) through the Cartesian product of the two universes. However, the "strength" of the relation between ordered pairs of the two universes is measured with a membership function expressing various "degrees" of strength of the relation on the unit interval [0,1].

Q.3 How to represent a fuzzy set in a computer ?

Ans. : The membership function must be determined first. A number of methods learned from knowledge acquisition can be applied here. For example, one of the most practical approaches for forming fuzzy sets relies on the knowledge of a single expert.

Q.4 Give the properties of fuzzy sets and also explain the operations involved in it. [RGPV : June - 17, Marks 7]

Ans. : Properties of fuzzy sets are as follows : Let A, B and C be fuzzy sets in a universe of discourse U, then we have,

Idempotent laws	$A \cup A = A$, $A \cap A$
Commutative laws	$A \cup B = B \cup A$, $A \cap B = B \cap A$
Associative laws	$(A \cup B) \cup C = A \cup (B \cup C)$, $(A \cap B) \cap C = A \cap (B \cap C)$
Absorption laws	$A \cup (A \cap B) = A$, $A \cap (A \cup B) = A$
Distributive laws	$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$, $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$
Involution law	$\bar{\bar{A}} = A$
De Morgan's laws	$\overline{A \cup B} = \bar{A} \cap \bar{B}$, $\overline{A \cap B} = \bar{A} \cup \bar{B}$
Complement laws	$A \cup \bar{A} \neq U$, $A \cap \bar{A} \neq \emptyset$

Operations on Fuzzy Sets

- Union, intersection, and complement are the most basic operations on classical sets. On the basis of these three operations, a number of identities can be established. Corresponding to the ordinary set operations of union, intersection, and complement, fuzzy sets have similar operations.

Scanned with CamScanner

1. **Containment or subset :** Fuzzy set A is contained in fuzzy set B (or, equivalently, A is a subset of B, or A is smaller than or equal to B, $A \subseteq B$) if and only if $\mu_A(x) \le \mu_B(x)$ for all x.

$$A \subseteq B \Leftrightarrow \mu_A \le \mu_B$$

A is contained in B

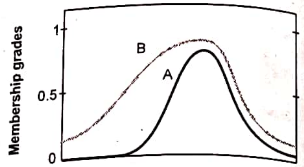


Fig. Q.4.1 Concept of A ⊆ B

2. **Union (disjunction) :** The **union** of two fuzzy sets A and B is a fuzzy set C, written as $C = A \cup B$ or $C = A$ OR B, whose MF is related to those of A and B by $\mu_C(x) = \max(\mu_A(x), \mu_B(x))$.

$$C = A \cup B \Leftrightarrow \mu_c(x) = \max(\mu_A(x), \mu_B(x)) = \mu_A(x) \vee \mu_B(x)$$

3. **Intersection (conjunction) :** The **intersection** of two fuzzy sets A and B is a fuzzy set C, written as $C = A \cap B$ or $C = A$ AND B, whose MF is related to those of A and B by $\mu_C(x) = \min(\mu_A(x), \mu_B(x))$.

$$C = A \cap B \Leftrightarrow \mu_C(x) = \min(\mu_A(x), \mu_B(x)) = \mu_A(x) \wedge \mu_B(x)$$

4. **Complement (negation) :** The complement of fuzzy set A, denoted by $\overline{A}$ or NOT A, is defined as $\mu_A(x) = 1 - \mu_A(x)$.

5. **Cartesian product and co-product :** Let A and B be the fuzzy sets in X and Y respectively. The cartesion product of A and B denoted by $A \times B$, is a fuzzy set in the product space $X \times Y$ with the membership function

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

Cartesian co-product A + B is a fuzzy set with the membership function

$$\mu_{A+B}(x, y) = \max(\mu_A(x), \mu_B(x)).$$

Fig. Q.4.2 shows the basic three operation on two fuzzy set A and B.

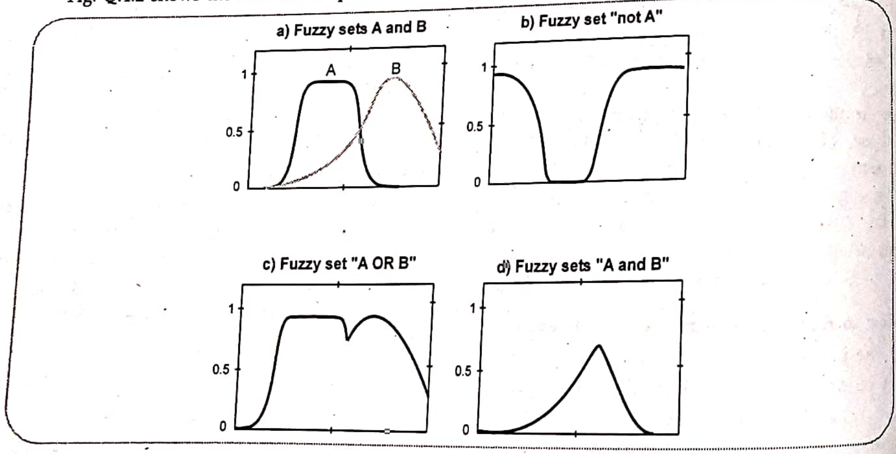


Fig. Q.4.2 Operation on fuzzy sets

Q.5 Explain fuzzy vs CRISP sets ? [RGPV : June - 14, Marks 7]

Ans. : • Fuzzy set and crisp set are the part of the distinct set theories, where the fuzzy set implements infinite-valued logic while crisp set employs bi-valued logic.

• In a crisp set, an element is either a member of the set or not. For example, a jelly bean belongs in the class of food known as candy. Mashed potatoes do not.

• Fuzzy sets, on the other hand, allow elements to be partially in a set. Each element is given a degree of membership in a set. This membership value can range from 0 (not an element of the set) to 1 (a member of the set). It is clear that if one only allowed the extreme membership values of 0 and 1, that this would actually be equivalent to crisp sets.

Parameters	Crisp Set	Fuzzy Set
Basic	Defined by precise and certain characteristics	Prescribed by vague or ambiguous properties
Property	Element is either the member of a set or not	Elements are allowed to be partially included in the set
Application	Digital design	Used in fuzzy controllers
Logic	Bi-valued	Infinite-valued

Q.6 Define crisp sets with its fundamental concepts. [RGPV : June-13, Marks 10]

Ans. : • A **classical set** is either an element belongs to the set or it does not. For example, for the set of integers, either an integer is even or it is not (it is odd). However, either you are in the Indian or you are not.

• Another example is for black and white photographs, one cannot say either a pixel is white or it is black. However, when you digitize a black/white figure, you turn all the black/white and gray scales into 256 discrete tones.

• Classical sets are also called **crisp** sets.

Lists : A = {apples, oranges, cherries, mangoes, bananas}

A = {a1, a2, a3, a4}

A = {2, 4, 6, 8, 10,12,14 ...}

Formulas : A = {x | x is an even natural number}

A = {x | x = 2n, n is a natural number}

• *Fuzzy set is any set that allows its members to have different degree of membership, called membership function in the interval [0, 1].*

• Fuzzy sets are characterized by **Membership Functions.** A fuzzy set has a graphical description that expresses how the transition from one to another takes place. This graphical description is called a **membership function.**

• The **membership function** assigns to each element x in a fuzzy set a number, A(x), in the closed unit interval [0, 1]. The number, A(x), represents the degree of membership of x in A. Hence, membership functions are functions of the form :

$$A : X \to [0, 1]$$

• In the case of crisp sets, the members of a set are either out of the set, with membership degree of zero, or in the set, with the value one being the degree of membership. Therefore, Crisp Sets ⊆ Fuzzy Sets or in other words, **crisp sets are special cases of fuzzy sets.**

• An element can be in the set with a degree of membership and out of the set with a degree of membership. Hence, crisp sets are a special case, or a subset, of fuzzy sets, where elements are allowed a membership degree of 100 % or 0 % but nothing in between.

• Fig. Q.6.1 shows the use of fuzzy sets to represent the notion of a tall person. It also shows how we can differentiate between the notions of tall and very tall, resulting in a more accurate model than the classical set theory.

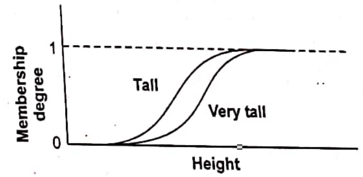


Fig. Q.6.1 Fuzzy set representation

Scanned with CamScanner

- The above figure is graphical representation of fuzzy logic. There are four ways of representing fuzzy membership functions, namely,
  a. Graphical representation, b. Tabular and list representation,
  c. Geometric representation, d. Analytic representation
- Tabular and list representations are used for finite sets.
- The third method of representation is the geometric representation and is also used for representing finite sets.
- Analytical representation is another alternative to graphical representation in representing infinite sets, e.g. a set of real numbers.

1. **α - cut (alpha cut) :**
   - α-cuts reduce a fuzzy set into an extracted crisp set. The value α represents a membership degree, i.e., $\alpha \in [0, 1]$. The α-cut of a fuzzy set A is the crisp set $(A - \alpha)$, i.e., the set of all elements whose membership degrees in A are $\geq \alpha$.
2. **Support, crossover point, and fuzzy singleton :** The support of a fuzzy set F is the crisp of all points in the universe of Discourse $\cup$ such that the membership function of F is non-zero. In particular, the element u in U at which F(u) = 0.5, is called the crossover point and a fuzzy set whose support is a single point in U with F(u) = 1.0 is referred to as fuzzy singleton (s).
3. **Normalized fuzzy set :** A fuzzy set is called normalized when at least one of its elements attains the maximum possible membership grade. If membership grades range in the closed interval between 0 and 1, for instance, then at least one element must have a membership grade of 1 for the fuzzy set to be considered normalized.
4. **Fuzzy set with a discrete nonordered universe.**
   - Let X = **{Delhi, Mumbai, Bangalore, Ahmedabad}** be the set of cities one may choose to live in.
   - Then the fuzzy set C = **"desirable city to live in"** may be described as follows :

     C = { (Delhi, 0.7) , (Mumbai, 0.8) , (Bangalore, 0.4), (Ahmedabad, 0.9) }
   - The universe of discourse X is discrete and it contains nonordered objects.
5. **Fuzzy set with a discrete ordered universe.**
   - Let X = { 0, 1, 2, 3, 4, 5, 6} be the set of numbers of children a family may choose to have.
   - Then the fuzzy set A = "sensible number of children in a family" may be described as follows :

     A = { (0, 0.1) , (1, 0.3), (2, 0.6), (3, 1) , (4, 0.7), (5, 0.2), (6, 0.1)}

Here we have discrete ordered universe X ; the MF for the fuzzy set A is shown in Fig. Q.6.2.

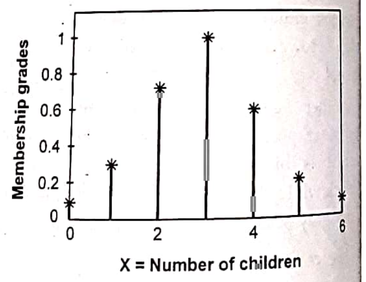


Fig. Q.6.2 Sensible number of children in a family

6. **Fuzzy sets with a continuous universe**
   - Let $X = R^+$ be the set of possible ages for human beings.
   - Then the fuzzy set C = "about 50 years old" may be expressed as

$$C = \{(x, \mu_C(x)) \mid x \in X\},$$

where $$\mu_C(x) = \frac{1}{1+\left(\frac{x-50}{10}\right)^4}$$

- Fig. Q.6.3 shows MF on a continuous universe.
- The specification of membership functions is subjective, which means that the membership functions specified for the same concept by different persons may vary considerably.
- This subjectivity comes from individual differences in perceiving or expressing abstract concepts and has little to do with randomness.
- The **subjectivity** and **nonrandomness** of fuzzy sets is the primary difference between the study of fuzzy sets and probability theory, which deals with objective treatment of random phenomena.

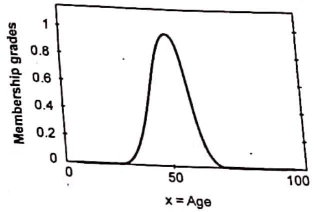


Fig. Q.6.3 MFON a continuous universe

7. **Core :** The core of a fuzzy set A is the set of all points x in X such that $\mu_A(X) = 1$:

   core (A) = $\{ x \mid \mu_A(X) = 1\}$
8. **Normality :** A fuzzy set A is normal if its core is nonempty. In other words, we can always find a point.

   $x \in X$ such that $\mu_A(X) = 1$.
9. **Fuzzy numbers :** A fuzzy number A is a fuzzy set in the real line (R) that satisfies the conditions for normality and convexity.

Q.7 Explain essential characteristics of fuzzy logic.

Ans. : • Some of the essential characteristics of fuzzy logic :

1. In fuzzy logic, exact reasoning is viewed as a limiting case of approximate reasoning.
2. In fuzzy logic, everything is a matter of degree.
3. In fuzzy logic, knowledge is interpreted a collection of elastic or, equivalently, fuzzy constraint on a collection of variables.
4. Inference is viewed as a process of elastic constraints.
5. Any logical system can be fuzzified.

Q.8 Consider two fuzzy sets A and B

$$A = \left\{\frac{1}{2}+\frac{0.5}{3}+\frac{0.3}{4}+\frac{0.2}{5}\right\} \quad B = \left\{\frac{0.5}{2}+\frac{0.7}{3}+\frac{0.2}{4}+\frac{0.4}{5}\right\}$$

Perform the following operating on fuzzy sets

i) $A \cup B$ **ii) $A \cap B$**

iii) Compement of fuzzy set A **iv) Difference $\left(\frac{A}{B}\right)$**

v) Algebraic sum of given fuzzy sets.

vi) Bounded sum of the given fuzzy set.

vii) Algebraic product of the given fizzy sets.

viii) $\overline{A \cup B}$ ix) $A \cup \overline{B}$

Scanned with CamScanner

3.2 : Membership Functions

Q.9 What are the features of fuzzy membership function ? ☞[RGPV : June-16, Dec.-16, Marks 2]

Ans. : Features of fuzzy membership function are defined by three properties : Core, support and boundary.

Q.10 What is degree of membership ?

Ans. : For any element x of universe X, membership function $\mu_A(x)$ equals the degree to which x is an element of set A. This degree, a value between 0 and 1, represents the degree of membership, also called membership value, of element x in set A.

Q.11 Define membership function with its features. Also state its importance in Fuzzy logic. ☞[RGPV : May-18, Marks 7]

Ans. : • Membership Function (MF) is a curve that defines how each point in the input space is mapped to a membership value (or degree of membership) between 0 and 1. The input space is sometimes referred to as the universe of discourse.

- Feature of membership function is defined by three properties :

1. Core 2. Support 3. Boundary

1. **Core :** If the region of universe is characterized by full membership in the set A then this gives the core of the membership function of fuzzy at A. The elements which have the membership function as 1 are the elements of the core. Here

$$\mu_A(x) = 1$$

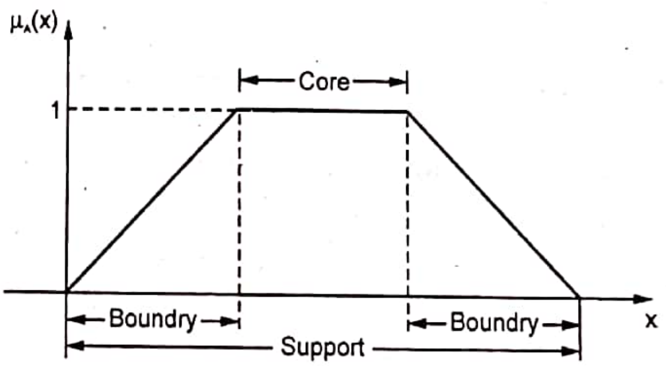


Fig. Q.11.1 Membership function feature

2. **Support :** if the region of universe is characterized by nonzero membership in the set A. This defines the support of a membership function for fuzzy set A. The support has the elements whose membership is greater than 0.

$$\mu_A(x) > 1$$

3. **Boundary :** If the region of universe has a nonzero membership but not full membership, this defines the boundary of a membership; this define the boundary of a membership function for fuzzy set A. The boundary has the elements whose membership is between 0 and 1,

$$0 < \mu_A(x) < 1.$$

- **Normal fuzzy set :** A fuzzy set A is normal if its maximal degree of membership is unity, i.e., there must exist at least one x for which $\mu A(x) = 1$. On the other hand, non-normal fuzzy sets have maximum degree of membership less than one. Fig. Q.11.2 shows normal fuzzy set.

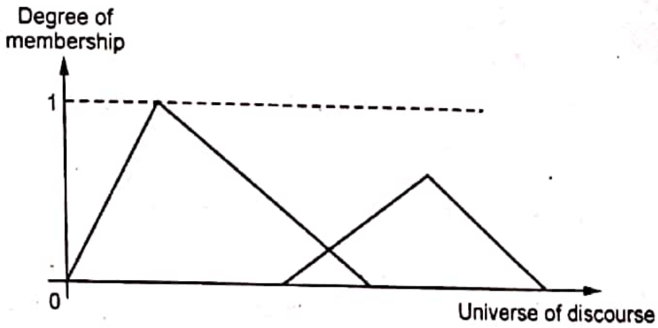


Fig. Q.11.2 Normal fuzzy set

- The height of a fuzzy set is the largest membership degree among all elements of the uni-verse. Fuzzy sets whose height equals one for at least one element X in the domain X are called **normal fuzzy sets**. The height of sub-normal fuzzy sets is thus smaller than one for all elements in the domain.

Various types of membership functions

1. S-shaped function
2. Trapezoidal membership function
3. Gaussian membership function
4. Exponential - like function
5. Triangular membership function

- Fig. Q.11.3 shows various shapes of membership function.

Scanned with CamScanner

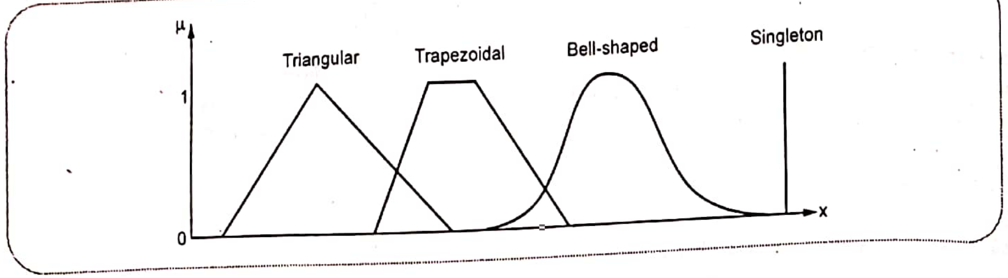


Fig. Q.11.3 Various shapes of membership function

- Following **criteria are valid for all membership functions :**

1. The membership function must be a real valued function whose values are between 0 and 1.
2. The membership values should be 1 at the center of the set, i.e., for those members that definitely belong to the set.
3. The membership function should fall off in an appropriate way from the center through the boundary.
4. The points with membership value 0.5 (crossover point) should be at the boundary of the crisp set, i.e., if we would apply a crisp classification, the class boundary should be represented by the crossover points.

Q.12 Explain different types of membership function used in fuzzification process ? ☞[RGPV : June-17, Marks 7]

Ans. : • Fuzzification is the process of making a crisp quantity fuzzy. In the real world, hardware such as a digital voltmeter generates crisp data, but these data are subject to experimental error.

- For each input and output variable selected, we define two or more Membership Functions (MF), normally three but can be more. **Fuzzification** is the process of changing a real scalar value into a fuzzy value. This is achieved with the different types of **fuzzifiers (membership functions).**
- The purpose of fuzzification is to partition the feature space into fuzzy subspaces and generate rules for each fuzzy subspace. Note that all fuzzy subspaces normally overlap some degree with each other.
- To carry out the process of fuzzification, one first has to define membership functions in order to calculate the membership grade for the input pixels.

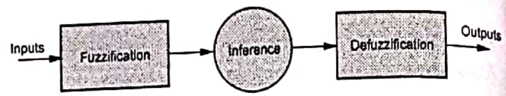


Fig. Q.12.1 Fuzzification

- A problem which converts exact logic problem into a fuzzy logic problem. Such problem is sometimes called fuzzification. A problem which deals with originally a fuzzy logic nature of problem and converts it into a fuzzy logic problem.
- The fuzzification process consists of two basic steps :

1. During the first step the interval of each concept is analyzed into trapezoidal membership functions.
2. Building a fuzzy knowledge base.

Q.13 Explain features of membership functions. ☞[RGPV : June-13, Marks 10]

Ans. : Refer Q.9.

Q.14 List and explain methods employed for membership value assignment.

Ans: : Membership Value assignments Methods :

1. Intuition
2. Inference
3. Rank ordering
4. Neural networks

- Intuition : Intuition involves contextual and semantic knowledge about an issue; it can also involve linguistic truth values about this knowledge.

Example : R = "x is considerably larger than y"

R (X, Y) = Relation between sets X and Y

R (x, y) = Membership function for the relation R (X, Y)

$$R(X, Y) = \{R(x, y) / (x, y) \setminus (x, y) \in (X \times Y)\}$$

$$R(x, y) = \begin{cases} 0 & \text{for } x \le y \\ (x-y)/(10-y) & \text{for } y < x \le 11y \\ 1 & \text{for } x > 11y \end{cases}$$

- Let X = Y = R+ and R = "y is much greater than x". The member function of the fuzzy relation R can be subjectively defined as

$$\mu_R(x, y) = \begin{cases} \dfrac{y-x}{x+y+2} & \text{if } y . x \\ 0 & \text{if } y \le x \end{cases}$$

Q.16 What are fuzzy relations ? Explain following operation on fuzzy relations.
i) Intersection ii) Containment

Ans. : • Union, intersection, and complement are the most basic operations on classical sets. On the basis of these three operations, a number of identities can be established.

- Corresponding to the ordinary set operations of union, intersection, and complement, fuzzy sets have similar operations.

i) Intersection (conjunction) : The intersection of two fuzzy sets A and B is a fuzzy set C, written as $C = A \cap B$ or C = A And B, whose MF is related to those of A and B by $\mu_C(x) = \min(\mu_A(x), \mu_B(x))$

$$C = A \cap B \Leftrightarrow \mu_C(x) = \min(\mu_A(x), \mu_B(x)) = \mu_A(x) \wedge \mu_B(x)$$

ii) Containment : Fuzzy set A is contained in fuzzy set B (equivalently, A is a subset of B, or A is smaller than or equal to B, $A \subseteq B$) if and only if $\mu_A(x) \le \mu_B(x)$ for all x.

$$A \subseteq B \Leftrightarrow \mu_A \le \mu_B$$

A is contained in B

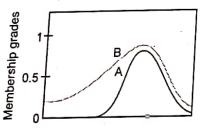


Fig. Q.16.1 Concept of A ⊆ B

Q.17 Explain fuzzy logic operations.

Ans. : Refer Q.4

Q.18 Consider two fuzzy sets X and Y, find Complement, Union, Intersection, Difference :

$$X = \left\{\frac{0.5}{2}, \frac{0.4}{3}, \frac{0.1}{4}, \frac{0.1}{5}, \frac{0.3}{6}, \frac{0.7}{7}, \frac{0.8}{8}\right\}$$

$$Y = \left\{\frac{0.3}{2}, \frac{0.8}{3}, \frac{0.6}{4}, \frac{0.8}{5}, \frac{0.2}{6}, \frac{0.2}{7}, \frac{0.3}{8}\right\}$$

Ans. : 1) Complement :

$$\text{Set } X = \left\{\frac{0.5}{2}, \frac{0.6}{3}, \frac{0.9}{4}, \frac{0.9}{5}, \frac{0.7}{7}, \frac{0.3}{7}, \frac{0.2}{8}\right\}$$

$$\text{Set } Y = \left\{\frac{0.7}{2}, \frac{0.2}{3}, \frac{0.4}{4}, \frac{0.2}{5}, \frac{0.8}{6}, \frac{0.8}{7}, \frac{0.7}{8}\right\}$$

2) Untion :

$$X \cup Y = \left\{\frac{0.5}{2}, \frac{0.8}{3}, \frac{0.6}{4}, \frac{0.8}{5}, \frac{0.3}{6}, \frac{0.7}{7}, \frac{0.8}{8}\right\}$$

3) Intersection :

$$X \cap Y = \left\{\frac{0.3}{2}, \frac{0.4}{3}, \frac{0.1}{4}, \frac{0.1}{5}, \frac{0.2}{6}, \frac{0.2}{7}, \frac{0.3}{8}\right\}$$

4) Difference

$$(X/Y) = X \cap \overline{Y} = X - (X \cap Y)$$

$$= \left\{\frac{0.2}{2}, \frac{0}{3}, \frac{0}{4}, \frac{0}{5}, \frac{0.1}{6}, \frac{0.5}{7}, \frac{0.5}{8}\right\}$$

3.4 : Concept of Fuzzy Measures

Q.19 What is T-norm, T-conorm ? Explain order of T-norms.

Ans. : 1. T-norm

- A t-norm (triangular norm) is a kind of binar operation used in the framework of probabilist metric spaces and in multi-valued logic, specifical in fuzzy logic. A t-norm generalizes intersection i a lattice and conjunction in logic.
- Fuzzy intersection of two fuzzy sets can b specified in a more general way by a binar operation on the unit interval, i.e., a function of th form :

$$T : [0, 1] \times [0, 1] \rightarrow [0, 1]$$

- In order for a functionTto qualify as a fuzzy intersection, it must have appropriate properties. Functions known ast-norms (triangular norms) posses the properties re-quired for the intersection. Similarly, functions calledt-conorms can be used for the fuzzy union.
- A t-norm T is a binary operation on the unit interval that satisfies at least the following axioms for all a; b; c ∈ [0, 1].
- Basic t-norms have following order:

$$t_{min}(a, b) \le \max(0, a+b-1) \le ab \le \min(a, b)$$

(T1) T (x, y) = T (y, x),

i.e., the t-norm is commutative,

(T2) T (T (x, y), z) = T (x, T (y, z)),

i.e., the t-norm is associative

(T3) $x \le y \Rightarrow T(x, z) \le T(y, z)$,

i.e., the t-norm is monotone,

(T4) T (x, 1) = x,

i.e., a neutral element exists, which is 1.

- **Some frequently used t-norms are :**

1. standard (Zadeh) intersection :
   T(a; b) = min (a; b)
2. algebraic product (probabilistic intersection) :
   T(a; b) = ab
3. Likewise (bold) intersection :
   T(a; b) = max (0; a + b – 1)

- Although continuous t-norms are most commonly used in fuzzy logics and have an elegant representation

2. T-conorm

- A t-conorm S is a binary operation on the unit interval that satisfies at least the following axioms for all a; b; c ∈ [0; 1]
- Some frequently used t-conorms are :

1. Standard (Zadeh) union: **S(a; b) = max(a; b);**
2. Algebraic sum (probabilistic union) :
   S(a; b) = a + b – ab;
3. Like wise (bold) union: **S(a; b) = min(1; a + b)**

The minimum is the biggest t-norm and maximum is the smallest t-conorm.

3.5 : Fuzzy Logic : Fuzzy Rules, Inferencing

Q.20 State fuzzy inference.

Ans. : Fuzzy inference can be defined as a process of mapping from a given input to an output, using the theory of fuzzy sets.

Q.21 What is a fuzzy rule ?

Ans. : A fuzzy rule can be defined as a conditional statement in the form :

IF x is A

THEN y is B

where x and y are linguistic variables; and A and B are linguistic values determined by fuzzy sets on the universe of discourses X and Y, respectively.

Q.22 What is fuzzy logic ? Explain its importance.

[RGPV : Dec.-16, Marks 7]

OR What is fuzzy logic ? Explain in brief any one application of fuzzy logic.

[RGPV : June -16, Marks 7]

Ans. : • Fuzzy set theory was developed by Lotfi A. Zadeh, professor for computer science at the University of California in Berkeley, to provide a mathematical tool for dealing with the concepts used in natural language.

- Fuzzy logic is not logic that is fuzzy, but logic that is used to describe fuzziness. Fuzzy logic is the theory of fuzzy sets, sets that calibrate vagueness.
- Fuzzy Logic is basically a multi-valued logic that allows intermediate values to be defined between conventional evaluations. Fuzzy logic is a form of many-valued logic in which the truth values of variables may be any real number between 0 and 1.
- Fuzzy logic is an extension of Boolean logic used to describe environments where there is no absolute truth and there is uncertainty.

Scanned with CamScanner

Scanned with CamScanner

- Fuzzy logic was specifically designed to mathematically represent uncertainty and vagueness and to provide formalized tools for dealing with the imprecision intrinsic to many problems.
- Fuzzy logic provides an inference morphology that enables approximate human reasoning capabilities to be applied to knowledge-based systems. The theory of fuzzy logic provides a mathematical strength to capture the uncertainties associated with human cognitive processes, such as thinking and reasoning.
- Fuzzy systems are **knowledge-based or rule-based systems.** The heart of a fuzzy system is a knowledge base consisting of the so-called fuzzy IF-THEN rules.
- FL is a problem-solving control system methodology that lends itself to implementation in systems ranging from simple, small, embedded micro-controllers to large, networked, multi-channel PC or workstation-based data acquisition and control systems.
- The conventional approaches to knowledge representation lack the means for representation the meaning of fuzzy concepts. As a consequence, the approaches based on first order logic and classical probability theory do not provide an appropriate conceptual framework for dealing with the representation of commonsense knowledge, since such knowledge is by its nature both lexically imprecise and non-categorical.

Essential characteristics of fuzzy logic :

1. In fuzzy logic, exact reasoning is viewed as a limiting case of approximate reasoning.
2. Everything is a matter of degree in fuzzy logic.
3. Knowledge is interpreted a collection of elastic or, equivalently, fuzzy constraint on a collection of variables.
4. Inference is viewed as a process of propagation of elastic constraints.
5. Any logical system can be fuzzified.

- Fuzzy systems are suitable for uncertain or approximate reasoning, especially for the system with a mathematical model that is difficult to derive. Fuzzy logic allows decision making with estimated values under incomplete or uncertain information.
- Fuzzy logic is based on the idea that all things admit of degrees. For example, temperature, height, speed, distance and beauty all come on a sliding scale. The motor is running *really* hot. Ram is a *very* tall guy.

What is fuzziness ?

- For example, if pressure takes values between 0 and 50, one might label the range 20 to 30 as medium pressure.
- Medium is known as a linguistic variable. Therefore, with Boolean logic 15.0 (or even 19.99) is not a member of the medium pressure range. As soon as the pressure equals 20, then it becomes a member.
- Fuzziness is not vague and Multi-valued logic.
- Fuzzy logic may be considered as an extension of multi-valued logic but they are somewhat different. Multi-valued logic is still based on exact reasoning whereas fuzzy logic is approximate reasoning.
- Fuzzy logic has rapidly become one of the most successful of today's technologies for developing sophisticated control systems. The reason for which is very simple.
- Fuzzy logic addresses such applications perfectly as it resembles human decision making with an ability to generate precise solutions from certain or approximate information.
- It fills an important gap in engineering design methods left vacant by purely mathematical approaches (e.g. linear control design), and purely logic-based approaches (e.g. expert systems) in system design.
- While other approaches require accurate equations to model real-world behaviors, fuzzy design can accommodate the ambiguities of real-world human language and logic.
- It provides both an intuitive method for describing systems in human terms and automates the conversion of those system specifications into effective models.

- Fuzzy system applications can be found in many fields of human activity, especially in control tasks of nonlinear and complex systems, where expert knowledge in the form of fuzzy rules is known.
- They can be classified according to the tasks to be performed and the domain of application.
- Also known as TSK fuzzy model. Takagi, Sugeno and Kang was proposed generation of fuzzy rules from a given input-output data set.
- A TSK fuzzy rule is of the form : **"If x is A and y is B then z = f(x, y)"**

where A and B are fuzzy sets in the antecedent, while $z = f(x, y)$ is a crisp function in the consequent.

- f(.,.) is very often a polynomial function w.r.t. x and y.
- If f(.,.) is a first order polynomial, then the resulting fuzzy inference is called a first order Sugeno fuzzy model.
- If f(.,.) is a constant then it is a zero-order Sugeno fuzzy model (special case of Mamdani model).
- Case of two rules with a first-order Sugeno fuzzy model.
  1. Each rule has a crisp output.
  2. Overall output is obtained via weighted average.
  3. No defuzzyfication required.

Example 1 : Single output-input Sugeno fuzzy model with three rules.

If X is small then $Y = 0.1X + 6.4$

If X is medium then $Y = -0.5X + 4$

If X is large then $Y = X - 2$

If "small", "medium" and "large" are nonfuzzy sets then the overall input-output curve is a piece wise linear. Following Fig. Q.22.1 shows comparison between fuzzy and nonfuzzy rules of eample 1.

- However, if we have smooth membership functions (fuzzy rules) the overall input-output curve becomes a smoother one.

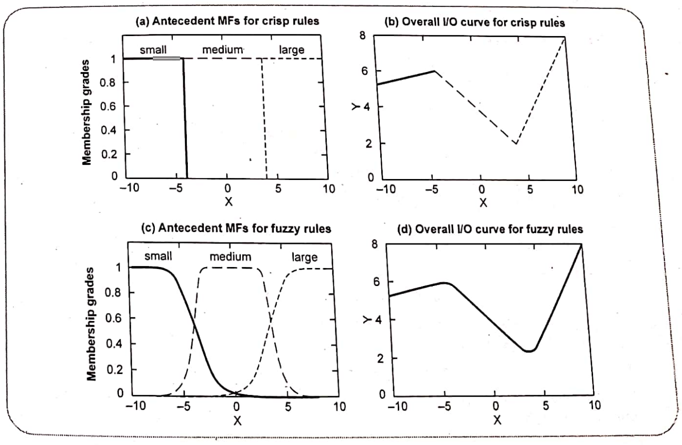


Fig. Q.22.1

Example 2 : Two-input single output fuzzy model with 4 rules

R1 : If X is small and Y is small then $z = -x + y + 1$

R2 : If X is small and Y is large then $z = -y + 3$

R3 : If X is large and Y is small then $z = -x + 3$

R4 : If X is large and Y is large then $z = x + y + 2$

$$R1 \rightarrow (x \wedge s) \text{ and } (y \wedge s) \rightarrow w1$$

$$R2 \rightarrow (x \wedge s) \text{ and } (y \wedge l) \rightarrow w2$$

$$R3 \rightarrow (x \wedge l) \text{ and } (y \wedge s) \rightarrow w3$$

$$R4 \rightarrow (x \wedge l) \text{ and } (y \wedge l) \rightarrow w4$$

$$\text{Aggregated consequent} \rightarrow F[(w1, z1); (w2, z2); (w3, z3); (w4, z4)]$$

$$= \text{Weighted average}$$

Q.23 With neat diagram, explain the working of Fuzzy Inference System (FIS). ☞ **[RGPV : May-18, Marks 7]**

Ans. : • A Fuzzy Inference System (FIS) is a system that uses fuzzy set theory to map inputs to outputs. FIS uses a collection of fuzzy membership functions and rules, instead of Boolean logic, to reason about data.

- For instance : *"If the service is good, even if the food is not excellent, the tip will be generous"*.
- FIS are used to solve decision problems, i.e. to make a decision and act accordingly.

Structure of a fuzzy inference system

- In general, a fuzzy inference system consists of four modules :

1. Fuzzification module
2. Knowledge base
3. Inference engine
4. Defuzzification module

- Fig. Q.23.1 shows block diagram of FIS.

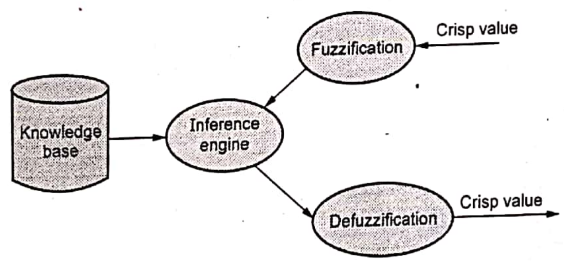


Fig. Q.23.1 Block diagram of FIS

1. **Fuzzification module :** Transforms the system inputs, which are crisp numbers, into fuzzy sets. This is done by applying a fuzzification function.
2. **Knowledge base :** Stores IF-THEN rules provided by experts.
3. **Inference engine :** Simulates the human reasoning process by making fuzzy inference on the inputs and IF-THEN rules.

Scanned with CamScanner

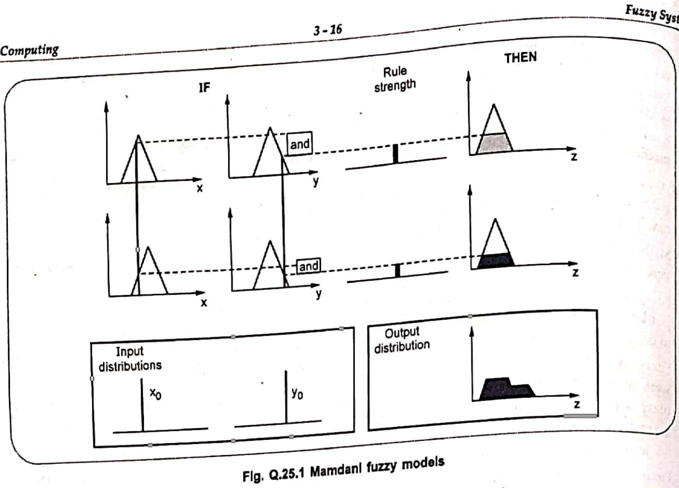


Fig. Q.25.1 Mamdani fuzzy models

Q.26 Consider two fuzzy sets A and B

$$A = \left\{\frac{1}{2}+\frac{0.5}{3}+\frac{0.3}{4}+\frac{0.2}{5}\right\} \quad B = \left\{\frac{0.5}{2}+\frac{0.7}{3}+\frac{0.2}{4}+\frac{0.4}{5}\right\}$$

Perform the following operating on fuzzy sets

i) $A \cup B$ **ii) $A \cap B$**

iii) Compement of fuzzy set A **iv) Difference $\left(\frac{A}{B}\right)$**

v) Algebraic sum of given fuzzy sets. **vi) Bounded sum of the given fuzzy set.**

vii) Algebraic product of the given fuzzy sets.

viii) $\overline{A \cup B}$ ix) $A \cup \overline{B}$

Ans. : i) $A \cup B$

$$A \cup B = \frac{1}{2}+\frac{0.7}{3}+\frac{0.3}{4}+\frac{0.4}{5}$$

ii) $$A \cap B = \frac{0.5}{2}+\frac{0.5}{3}+\frac{0.2}{4}+\frac{0.2}{5}$$

iii) Complement of fuzzy set A : $1-\mu_A(x)$

$$= \left\{\frac{0}{2}+\frac{0.5}{3}+\frac{0.7}{4}+\frac{0.8}{5}\right\}$$

iv) Difference $\left(\frac{A}{B}\right) = A \cap \overline{B} = A - (A \cap B)$

$$= \left\{\frac{0.5}{2}+\frac{0}{3}+\frac{0.1}{4}+\frac{0}{5}\right\}$$

v) Algebraic sum $= \left\{\frac{1.5}{2}+\frac{1.2}{3}+\frac{0.5}{4}+\frac{0.6}{5}\right\}-\left\{\frac{0.5}{2}+\frac{0.35}{3}+\frac{0.06}{4}+\frac{0.08}{5}\right\}$

$$= \left\{\frac{1}{2}+\frac{0.85}{3}+\frac{0.44}{4}+\frac{0.52}{5}\right\}$$

vi) Bounded sum $= \min\left\{1,\left\{\frac{1.5}{2}+\frac{1.2}{3}+\frac{0.5}{4}+\frac{0.6}{5}\right\}\right\}$

$$= \left\{\frac{1.5}{2}+\frac{1.2}{3}+\frac{0.5}{4}+\frac{0.6}{5}\right\}$$

vii) Algebraic product $= \left\{\frac{0.5}{2}+\frac{0.35}{3}+\frac{0.06}{4}+\frac{0.08}{5}\right\}$

viii) $\overline{A \cup B} = 1-\max\{\mu_A(x),\mu_B(x)\}$

$$= 1-\left\{\frac{1}{2}+\frac{0.7}{3}+\frac{0.3}{4}+\frac{0.4}{5}\right\}$$

$$= \left\{\frac{0}{2}+\frac{0.3}{3}+\frac{0.7}{4}+\frac{0.6}{5}\right\}$$

ix) $A \cup \overline{B} = \max[\mu_A(x),\mu_B(x)]$

Calculate, $\overline{B} = 1-\mu_B(x)$

$$= 1-\left\{\frac{0.5}{2}+\frac{0.7}{3}+\frac{0.2}{4}+\frac{0.4}{5}\right\}$$

$$= \left\{\frac{0.5}{2}+\frac{0.3}{3}+\frac{0.8}{4}+\frac{0.6}{5}\right\}$$

$$A \cup \overline{B} = \left\{\frac{1}{2}+\frac{0.5}{3}+\frac{0.8}{4}+\frac{0.6}{5}\right\}$$

Q.27 Write short note on Sugeno fuzzy model.

Ans. : Refer Q.22

3.6 : Fuzzy Control System

Q.28 What do you mean defuzzification ?

Ans. : The last step in the fuzzy inference process is defuzzification. Fuzziness helps us to evaluate the rules, but the final output of a fuzzy system has to be a crisp number. The input for the defuzzification process is the aggregate output fuzzy set and the output is a single number.

Q.29 What is called the principle of incompatibility ?

Ans. : Conventional techniques for system analysis are intrinsically unsuited for dealing with humanistic systems, whose behavior is strongly influenced by human judgment, perception and emotion. This is the manifestation of what might be called the principle of incompatibility.

Q.30 Explain architecture of fuzzy control system. What is rule base strategy ? Explain steps in fuzzy control system.

Ans. : • Fig. Q.30.1 shows architecture of a fuzzy control system.

• A typical fuzzy system can be split up into four main parts : Fuzzifier, a knowledge base, an inference engine and a defuzzifier.

• The fuzzy controller has four main components :

1. **Rule base :** It holds the knowledge with the set of rules. It helps for controlling the system.
2. **Inference mechanism :** Evaluates which control rules are relevant at the current time and then decides which type of input given to the plant.
3. **The fuzzification interface :** It simply modifies the inputs so that they can be interpreted and compared to the rules in the rule-base system.
4. **Defuzzification interface** converts the conclusions reached by the inference mechanism into the inputs to the plant.

Rule base strategy :

• Rule base strategy has to do with the fuzzy control rules themselves. The property of completeness is incorporated into fuzzy control rules through design experience and engineering knowledge.

• An additional rule is added whenever a fuzzy condition is not included in the rule base or whenever the degree of partial match between some inputs and predefined fuzzy conditions is lower than some level.

• Fig. Q.30.2 shows pure fuzzy system.

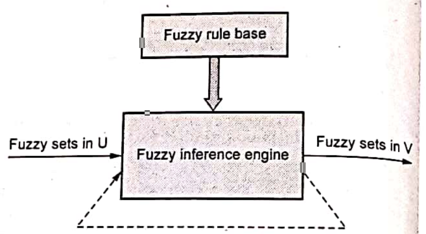


Fig. Q.30.2 Basic configuration of pure fuzzy systems

Steps in fuzzy control system :

• A FC operates by repeating the cycle :

1. Measurements are taken of all relevant variables
2. Fuzzification : Conversion of the measurements into fuzzy sets to express their uncertainties
3. Fuzzified measurements are used by the inference engine to evaluate the control rules. The result of this evaluation is one or more fuzzy sets defined on the universe of possible actions

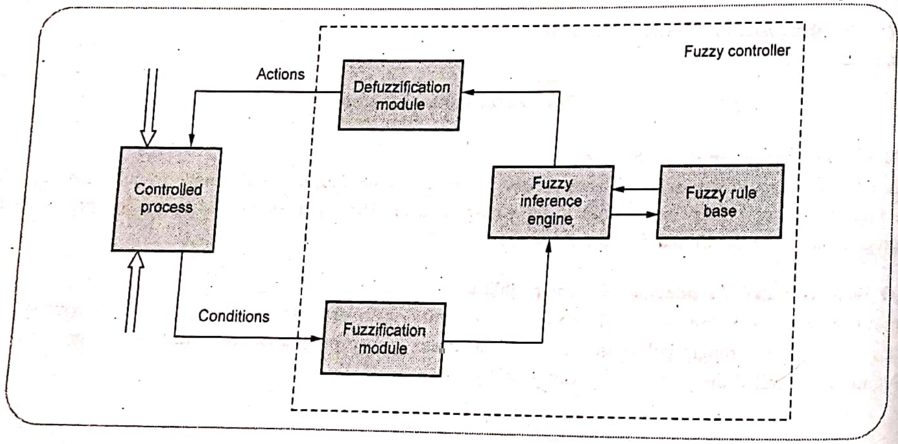


Fig. Q.30.1 Architecture of a fuzzy control system

4. Defuzzification : The outcome is converted into a single crisp value.

• In order to process the input to get the output reasoning there are six steps involved in the creation of a rule based fuzzy system :

1. Identify the inputs and their ranges and name them.
2. Identify the outputs and their ranges and name them.
3. Create the degree of fuzzy membership function for each input and output.
4. Construct the rule base that the system will operate under
5. Decide how the action will be executed by assigning strengths to the rules
6. Combine the rules and defuzzify the output.

• To design the fuzzy controller, the control engineer must gather information on how the artificial decision maker should act in the closed-loop system.

Q.31 Define fuzzy rule base system. ☞[RGPV : Dec.-16, Marks 3]

OR What is fuzzy rule base system ? Explain its working. ☞[RGPV : June-16, Marks 7]

Ans. : • A rule-based fuzzy logic system is comprised of four elements: rules, fuzzifier, inference engine and output processor.

• Fuzzy rules are linguistic IF-THEN - constructions that have the general form "IF A THEN B" where A and B are propositions containing linguistic variables.

• A is called the premise and B is the consequence of the rule. In effect, the use of linguistic variables and fuzzy IF-THEN- rules exploits the tolerance for imprecision and uncertainty. In this respect, fuzzy logic mimics the crucial ability of the human mind to summarize data and focus on decision-relevant information.

• In a more explicit form, if there are I rules each with K premises in a system, the i^{th} rule has the following form.

If a_1 is $A_{i,1}\,\Theta\,a_2$ is $A_{i,2}\,\Theta \ldots \Theta\, a_k$ is $A_{i,k}$ then B_i.

• In the above equation a represents the crisp inputs to the rule and A and B are linguistic variables. The operator 1 can be AND or OR or XOR.

Q.32 Explain open loop and closed loop control system.

Ans. : • There are two kinds of control systems : **Open-loop control systems and closed-loop control systems.**

1. Open loop control system :

• Fig. Q.32.1 shows open loop control.

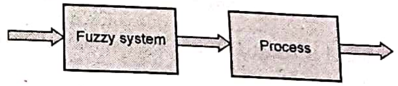


Fig. Q.32.1 Open loop control

• Systems in which the output has no effect on the control action are called open-loop control systems.

• In an open-loop control system, there is no feedback path but only forward path between the input and the output.

• When used as an open-loop controller, the fuzzy system usually sets up some control parameters and then the system operates according to these control parameters.

• Open-loop system is easier to build and lower in cost.

• The open-loop control system can only be used when the requirement for accuracy is low and the disturbance can be neglected, such as the control of washing machines, stepper motors.

• Many applications of fuzzy systems in consumer electronics belong to this category.

2. Closed loop control system :

• The control system in which the output has an effect on the control action is called as **closed-loop control system.** When used as a closed-loop controller, the fuzzy system measures the outputs of the process and takes control actions on the process continuously. Applications of fuzzy systems in industrial processes belong to this category.

• Fig. Q.32.2 shows close loop control system.

Scanned with CamScanner

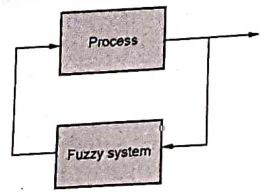


Fig. Q.32.2 Close loop control system

- A system that compares the output and the reference input and uses the difference as a means of control is called a closed-loop control system or feedback control system.
- By the use of feedback, the closed-loop control system is relatively insensitive to external disturbances and internal variations in system parameters.
- The closed-loop control system is generally high in cost and complexity.

Q.33 Explain fuzzy control system design problem.

Ans. : Control System Design Problem

- Fuzzy control system design essentially amounts to

1. Choosing the fuzzy controller inputs and outputs,
2. Choosing the preprocessing that is needed for the controller inputs and possibly post processing that is needed for the outputs,
3. Designing each of the four components of the fuzzy controller.

- Since fuzzy control is a relatively new technology, it is often quite important to determine what value it has relative to conventional methods.
- Fig. Q.33.1 shows basic control system. Process is the object to be controlled. Its inputs are u(t), outputs are y(t) and reference input is r(t).

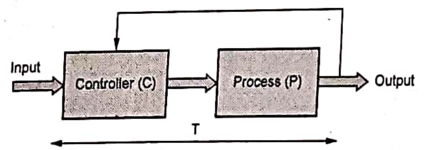


Fig. Q.33.1 Control system

- Consider the example of cruise control problem :

a. u(t) is the throttle input
b. y(t) is the speed of the vehicle
c. r(t) is the desired speed that is specified by the driver.

- The plant is the vehicle itself. The controller is the computer in the vehicle that actuates the throttle based on the speed of the vehicle and the desired speed that was specified.
- Fuzzy control design methodology can be used to construct fuzzy controllers for challenging real world application. Feedback control system design problem is defined by using nonlinear vector valued function h( ). The h( ) is defined with time reference.

$$u(t) = h[t, x(t), r(t)]$$

where $u(t)$ = Process control input

$r(t)$ = System reference input

$x(t)$ = System state vector

- In the process of designing a linear or nonlinear control system, it is usually necessary to obtain a number of parameters of the controller in order to define a good design that meets a number of performance requirements under certain practical constraints.
- In control system design, the structure of a controller is usually determined by the control scheme or control law that the design engineer opts to apply.
- The feedback control law (h) is supposed to stabilize the feedback control system and give good performance.
- Full state feedback or output feedback is expressed as :

$$u(t) = h\,[x(t)]$$

$$u(t) = h\left[y(t), \dot{y}, \int y\,dt\right]$$

- Here y( ) is the system output or response function.
- Function "h" is also applied to single input with single output system. It is expressed as follows :

$$u(t) = e(t)\,K_p$$

- Fuzzy controller design consists of turning intuitions, and any other information about how to control a system, into set of rules. These rules can then be applied to the system. If the rules adequately control the system, the design work is done. If the rules are inadequate, the way they fail provides information to change the rules.

Q.34 What is defuzzyfication ? Explain with example. [RGPV : June-15, Marks 7]

OR What is defuzzyfication and explain widely used methods COM, COS and MOM. [RGPV : June-14, Marks 7]

Ans. : • Defuzzification means the fuzzy-to-crisp conversions. The fuzzy results generated cannot be used to the applications. It is necessary to convert the fuzzy quantities into crisp quantities for further processing.

- Defuzzification has the capability to reduce a fuzzy set to a crisp single-valued quantity or as a set.
- Defuzzification is the process by which a fuzzy consequent is reduced to a singleton or crisp scalar value in order to provide an interface to a typically scalar 'action'. Actions are typically associated with a scalar defining a specific measurable property.
- The task of defuzzification is to find one single crisp value that summarizes the fuzzy set that enters it from the inference block. Examples might include speed of movement, volume of voice, (angular) direction, or a decrease/increase in health/energy.
- Defuzzification operates on the implied fuzzy sets produced by the inference mechanism and combines their effects to provide the "most certain" controller output.
- Defuzzification methods are as follows :

1. Maxima method
2. Centroid method
3. Weighted average method
4. Middle-of-maxima method
5. First-of-maxima or last-of-maxima

Max Membership Principle

- For peaked output function, max membership is used. It is also called as height method. Fig. Q.34.1. shows maximum membership principle.

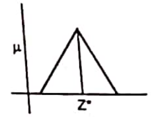


Fig. Q.34.1

- The method is expressed as :

$$\mu_c(Z^*) \geq \mu_c(z)$$

Where Z* is the defuzzified value.

Centroid Method

- In the centroid method, the crisp value of the output variable is computed by finding the variable value of the centre of gravity of the membership function for the fuzzy value. Fig. Q.34.2 shows centroid method.

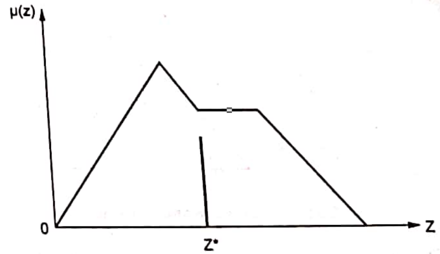


Fig. Q.34.2 Centroid method

- This most familiar defuzzification scheme. The composite output fuzzy set is built by taking the union of all clipped or scaled output fuzzy sets.
- The crisp output value is then obtained by deriving the centroid or centre of mass of the shape represented by the composite output fuzzy set.
- The x co-ordinate of the centre of mass is the required crisp output value. Because of the variety of shapes that are encountered, the method is computationally intensive.

$$z^* = \frac{\int \mu_c(z) \cdot z\,dz}{\int \mu_c(z)\,dz}$$

Where $\int$ denotes an algebraic integration.

Weighted Average Method

- Weighted average defuzzification instead of using a composite output membership function, uses the individual clipped or scaled output fuzzy sets.
- The method takes the peak value of each clipped/scaled output fuzzy set and builds the weighted (with respect to the peak heights) sum of these peak values. The output is given by :

$$z^* = \frac{\sum \mu_{\underline{C}}(\bar{z})\,\bar{z}}{\sum \mu_{\underline{C}}(\bar{z})}$$

$$z^* = \frac{a(0.8)+b(0.6)}{0.8+0.6}$$

$$z^* = \frac{0.8a+0.6a}{1.4}$$

- The weighted average method is formed by weighting each membership function in the output by its respective maximum membership value.

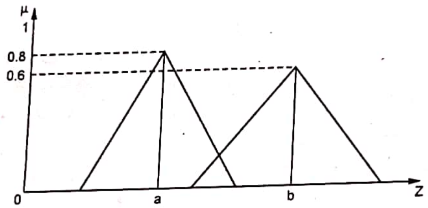


Fig. Q.34.3 Weighted average method

- This method can be used only for symmetrical output membership functions.

Other Methods

a. Centre of sums

- As with the centroid method, the clipped or scaled output fuzzy sets are combined to form a composite output fuzzy set. The contribution of each clipped or scaled fuzzy set, however, is considered individually.
- Unlike the Centroid method which builds the output composite set using the union operation the centre of sums method takes the SUM of the clipped/scaled output Fuzzy sets and then computes the centre of mass of the resulting shape.

- Thus overlapping areas, if they exist are reflected more than once by this technique.

b. Singleton method

- The singleton method is quite a simple method that is widely used in hardware implementations. The technique uses output membership functions which are just vertical lines of height 1 in output space, or singletons.
- Each singleton has a single crisp output value associated with it. The final crisp output is a weighted average of all qualified output sets.

Q.35 What is the need of fuzzyfication ? Explain the methods used for defuzzyfication.

[RGPV : June-16, Marks 3]

Ans. : Refer Q.12 and Q.34

Q.36 Explain with example linguistic variable and linguistic values.

Ans. : • Linguistic variable is "a variable whose values are words or sentences in a natural or artificial language".

- Each linguistic variable may be assigned one or more linguistic values, which are in turn connected to a numeric value through the mechanism of membership functions.
- Conventional techniques for system analysis are intrinsically unsuited for dealing with systems based on human judgment, perception and emotion.
- A linguistic variable is characterized by a quintuple (x, T(x), X, G, M) in which x is the name of the variable, T(x) is the term set of x, X is the universe of discourse; G is syntactic rule which generates the terms in T(x) ; and M is a semantic rule which associates with each linguistic value A its meaning M(A) , where M(A) denotes a fuzzy set in X.
- A fuzzy rule can be defined as a conditional statement in the form :

If x is A THEN y is B

where x and y are linguistic variables; and A and B are linguistic values determined by fuzzy sets on the universe of discourses X and Y, respectively.

- Example : **If temperature is cold and oil is cheap then heating is high**
- If age is interpreted as a linguistic variable, then its term set T(age) could be

T(age) = { young, not young, very young, not very young, ...
middle aged, not middle aged,..........,
old, not old, very old, more or less old, not very old, ...,
not very young and not very old, ...},

where each term in T(age) is characterized by a fuzzy set of a universe of discourse X = [0, 100] . It is shown in Fig. Q.36.1.

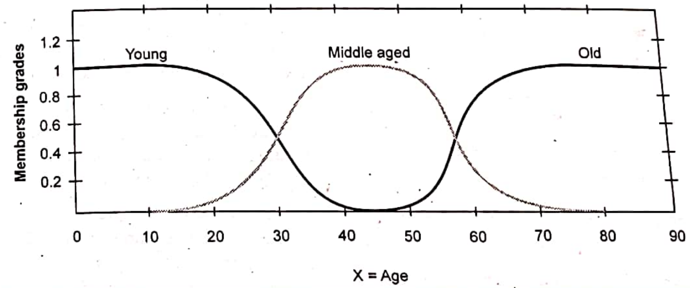


Fig. Q.36.1 Membership function of the term set T(age)

- Here "age is young" is used to denote the assignment of the linguistic value "young" to the linguistic age.
- Let A be a linguistic value characterized by a fuzzy set with membership function $\mu_A(.)$. Then A^k is interpreted as a modified version of the original linguistic value expressed as

$$A^k = \int_x [\mu_A(x)]^k / x$$

- The operation of **concentration** is defined as

$CON(A) = A^2$

While that of dilation is expressed by

$DIL(A) = A^{0.5}$

Scanned with CamScanner

Q.37 Explain fuzzy If-Then rules.

Ans. : • Fuzzy sets and fuzzy sets operations are the subjects and verbs of fuzzy logic. If-Then rule statements are used to formulate the conditional statements that comprise fuzzy logic.

- A fuzzy if-then rule is also called fuzzy rule, fuzzy implication or fuzzy conditional statement. It assume the form

If x is A then y is B

Where A and B are linguistic values defined by fuzzy sets on universes of discourse X and Y respectively.

- Often "x is A" is called the **antecedent or premise** while "y is B" is called the **consequence or conclusion.**
- The antecedent describes to what degree the rule applies, while the conclusion assigns a fuzzy function to each of one or more output variables.

Examples :

- If pressure is high, then volume is small.
- If the road is slippery, then driving is dangerous.
- If a tomato is red, then it is ripe.
- If the speed is high, then apply the brake a little.

Example :

Speed and pressure of a steam engine can be expressed with the following linguistic conditional statement

If **Speed** is **Slow** Then **Pressure** should be **High.**

Graphically, this statement looks like as shown in Fig. Q.37.1.

- The fuzzy rule "If x is A then y is B" may be abbreviated as $A \to B$ and is interpreted as $A \times B$. A fuzzy if then rule may be defined (Mamdani) as a binary fuzzy relation R on the product space $X \times Y$.

$$R = A \to B = A \times B$$

$$= \int \mu_A(x) * \mu_A(y) | (x, y)$$

3.7 : Applications of Fuzzy System

Q.38 Explain one application of fuzzy system.

Ans. :

- Aircraft landing is the process during which the aircraft descends from certain altitude and touches the ground. The aircraft landing system is supposed to control the vertical speed during the aircraft landing process, so that the aircraft touches the ground "smoothly" but descends rapidly.
- If the aircraft touches the ground with high negative vertical speed, that is considered a crash landing and should be avoided by the controller. Fig. Q.38.1 shows landing problem of aircraft.
- Vertical velocity changes only if control force is applied. Negative control force directs the aircraft to the ground while positive control force directs the aircraft from the ground.

1. **Antecedent :** Height "h" above ground and vertical velocity of the aircraft, "v"
2. **Consequent :** Control forces that, when applied to the aircraft, will alter its height, h and velocity, v.

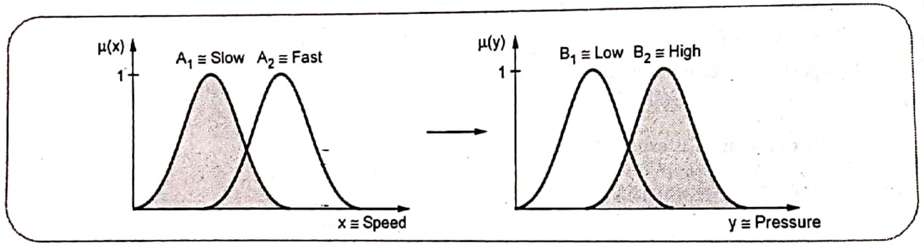


Fig. Q.37.1 If-then rule

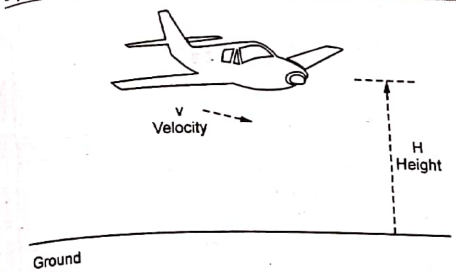


Fig. Q.38.1 Landing problem of aircraft

3. **Membership functions :**
   a. **Height :** L (large), M (medium), S (small), NZ (near zero)
   b. **Velocity :** UL(up large), US (up small), Z (zero), DS(down small), DL (down large)
   c. **Force :** UL (up large), US (up small), Z (zero), DS (down small), DL (down large)

- If current vertical velocity is negative, than aircraft will descend, if it is positive it will increase altitude and if it is zero, then aircraft will remain at the same altitude.
- All three different types of controllers have two inputs : Height in feet and vertical velocity in feet per second.
- The output is also the same variable for all three types of controllers : Control force in pound-force. Control force can change current vertical velocity, and change of vertical velocity, changes rate of descent or climb, thus affects current height.
- Fig. Q.38.2 shows fuzzy aircraft landing control system.

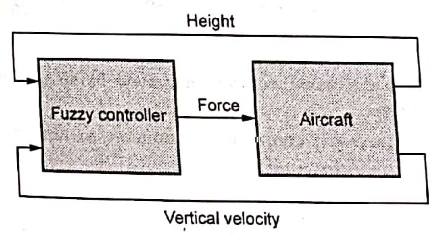


Fig. Q.38.2 Fuzzy aircraft landing control system

- Fuzzy rules are as follows :

		Velocity				
		DL	DS	ZERO	US	UL
Height	L	Z	DS	DL	DL	DL
	M	US	Z	DS	DL	DL
	S	UL	US	Z	DS	DL
	NZ	UL	UL	Z	DS	DS

- Control surface graphically represents all possible inputs and outputs.

Important Points to Remember

- A fuzzy set is a collection of objects with graded membership.
- In a crisp set, an element is either a member of the set or not.
- Fuzzy relations map elements of one universe (U) to those of another universe (V) through the cartesian product of the two universes.
- Features of fuzzy membership function are defined by three properties : core, support and boundary.
- Fuzzy inference can be defined as a process of mapping from a given input to an output, using the theory of fuzzy sets.
- A rule-based fuzzy logic system is comprised of four elements: rules, fuzzifier, inference engine and output processor
- In defuzzification, the fuzzy output set is converted to a crisp number

END...

Scanned with CamScanner

4 Genetic Algorithm

4.1 : Concept of Genetic Algorithm

Q.1 What is Genetic algorithm ?

Ans. : A genetic algorithm is a search technique used in computing to find true or approximate solutions to optimization and search problems.

Q.2 What do you mean genetic programming ?

Ans. : Genetic programming is genetic algorithm wherein the population contains programs rather than bit strings.

Q.3 Explain the "Darwinian theory of survival" ☞[RGPV : June-16, Marks 2]

Ans. : More individuals are produced each generation that can survive. Phenotypic variation exists among individuals and the variation is heritable. Those individuals with heritable traits better suited to the environment will survive.

Q.4 State the importance of genetic algorithm.

Ans. : Genetic algorithm is that the problem solving strategy involves using "the strings' fitness to direct the search; therefore they do not require any problem-specific knowledge of the search space, and they can operate well on search spaces that have gaps, jumps, or noise". As each individual string within a population directs the search, the genetic algorithm searches, in parallel, numerous points on the problem state space with numerous search directions.

Q.5 Compare and contract genetic algorithm with traditional algorithm. ☞[RGPV : Dec.-16, Marks 2]

Ans. :

Genetic algorithm	Traditional algorithm
GA generates a population of points at each iteration. The best point in the population approaches an optimal solution.	It generates a single point at each iteration. The sequence of points approaches an optimal solution.
Selects the next population by computation which uses random number generators	Selects the next point in the sequence by a deterministic computation.
Convergence in each iteration in problem independent	Improvement in each iteration is problem specific
Rules are probabilistic.	Rules are fully deterministic.

Q.6 What two requirements should a problem satisfy it by a genetic algorithm ? ☞[RGPV : Dec.-16, Marks 2]

Ans. : GA can only be applied to problems that satisfy the following requirements :

i) The tness function can be well(dened.

ii) Solutions should be decomposable into steps (building blocks) which could be then encoded as chromosomes

Q.7 What is use of crossover operator ?

Ans. : A crossover operator is used to recombine two strings to get a better string. In crossover operation, recombination process creates different individuals in the successive generations by combining material from two individuals of the previous generation.

Q.8 Explain two point crossover.

Ans. : Two crossover points are selected, binary string from the beginning of the chromosome to the first crossover point is copied from the first parent, the part from the first to the second crossover point is copied from the other parent and the rest is copied from the first parent again.

Q.9 List the basic components used in all genetic algorithms.

Ans. : The basic components common to almost all genetic algorithms are :

1. Fitness function for optimization
2. a population of chromosomes
3. selection of which chromosomes will reproduce

4. crossover to produce next generation of chromosomes
5. random mutation of chromosomes in new generation

Q.10 Explain the steps of genetic algorithm (GA). ☞[RGPV : May-18, Marks 7]

OR Describe working principle of genetic algorithm. ☞[RGPV : June-13, Marks 10]

Ans. : 1. **[Start]** Generate random population of n chromosomes (suitable solutions for the problem).

2. **[Fitness]** Evaluate the fitness f(x) of each chromosome x in the population.

3. **[New population]** Create a new population by repeating following steps until the new population is complete.

a. **[Selection]** Select two parent chromosomes from a population according to their fitness (the better fitness, the bigger chance to be selected).

b. **[Crossover]** with a crossover probability cross over the parents to form a new offspring (children). If no crossover was performed, offspring is an exact copy of parents.

c. **[Mutation]** with a mutation probability mutate new offspring at each locus (position in chromosome).

d. **[Accepting]** Place new offspring in a new population.

4. **[Replace]** Use new generated population for a further run of algorithm.

5. **[Test]** If the end condition is satisfied, stop and return the best solution in current population.

6. **[Loop]** Go to step 2.

GA Steps

Step 1 : Set population size and probability

Step 2 : Define fitness function

Step 3 : Generate initial population

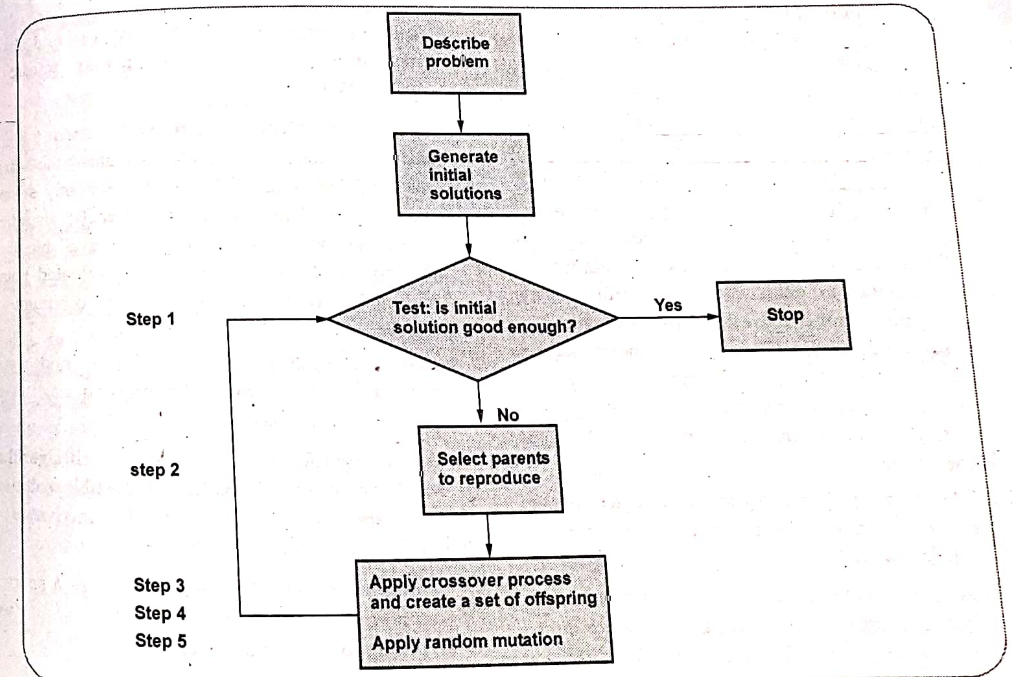


Fig. Q.10.1 Flow diagram of the genetic algorithm process

Scanned with CamScanner

Step 4 : Calculate fitness for each chromosome

Step 5 : Mating of chromosomes

Step 6 : Create offspring-crossover and mutation

Step 7 : Offspring in new population

Step 8 : Repeat Step 5 until new = initial population

Step 9 : Replace initial population with new

Step 10 : To Step 4 and repeat until criteria achieved

Q.11 How does genetic algorithm differ from conventional algorithm? Give the advantages of GA over conventional algorithms.

[RGPV : June-17, Marks 7]

Ans. : 1. Genetic algorithms a coded form of the function values i.e. parameter set, rather than with the actual values themselves. For example, if we want to find the minimum of the function $f(x) = x^3 + x^2 + 5$, the GA would not deal directly with x or y values, but with strings that encode these values. For this case, strings representing the binary x values should be used.

2. Genetic algorithms use a set or population of points to conduct a search, not just a single point on the problem space. This gives GAs the power to search noisy spaces littered with local optimum points. Instead of relying on a single point to search through the space, the GAs looks at many different areas of the problem space at once, and uses all of this information to guide it.

3. Genetic algorithms use only payoff information to guide themselves through the problem space. Many search techniques need a variety of information to guide themselves. Hill climbing methods require derivatives, for example. The only information a GA needs is some measure of fitness about a point in the space. Once the GA knows the current measure of "goodness" about a point, it can use this to continue searching for the optimum.

4. GAs are probabilistic in nature, not deterministic. This is a direct result of the randomization techniques used by GAs.

5. GA is inherently parallel. Here lies one of the most powerful features of genetic algorithms. GA's, by their nature, are very parallel, dealing with a large number of points simultaneously.

Parameters	Genetic Algorithms	Traditional Methods
Work with	Coding of parameter set	Parameters directly
Use information	Payoff i.e. objective function	Payoff plus derivatives etc.
Rules	Probabilistic	Fully deterministic
Search	A population of points	A population of points a single point

- Genetic algorithm is a stochastic algorithm.
- Randomness as an essential role in both selection and reproduction phases.
- Genetic algorithms always consider a population of solutions. A population base algorithm is also very amenable for parallelization.
- There is no particular requirement on the problem before using genetic algorithms, so it can be applied to resolve any problem (optimization).
- GAs are a new field and parts of the theory have still to be properly established. We can find almost as many opinions on GAs as there are researchers in this field.

Q.12 Why use Genetic algorithm ?

Ans. : • Genetic algorithms evaluate the target function to be optimized at some randomly selected points of the definition domain. Genetic algorithms can be used when no information is available about the gradient of the function at the evaluated points. The function itself does not need to be continuous or differentiable.

- Genetic algorithms can still achieve good results even in cases in which the function has several local minima or maxima.
- Genetic algorithms are stochastic search algorithms which act on a population of possible solutions. They are loosely based on the mechanics of population genetics and selection.
- Genetic algorithms allow you to explore a space of parameters to find solutions that score well according to a "fitness function".

Scanned with CamScanner

Ans. : Reproduction selects good strings in a population and forms a mating pool. This is one of the reasons for the reproduction operation to be sometimes known as the selection operator.

Q.20 What are the various methods of selecting chromosomes for parents to crossover ?

Ans. : Various methods are :

1. Roulette-Wheel selection 2. Boltzmann selection

3. Tournament selection 4. Rank selection

5. Steady state selection

Q.21 How is a population with increasing fitness generated ?

Ans. : If two parents have superior fitness, there is a good chance that a combination of their genes will produce an offspring with even higher fitness.

Q.22 Define mutation rate.

Ans. : Mutation rate is the probability of mutation which is used to calculate number of bits to be muted.

Q.23 Explain the role of fitness function in GA and what are the requirements of GA ?

☞[RGPV : June-17, Marks 7]

OR What is fitness function ? Why it is necessary ? ☞[RGPV : June-14, Marks-7]

Ans. : • Fitness is an important concept in genetic algorithms. The fitness of a chromosome determines how likely it is that it will reproduce. Fitness is usually measured in terms of how well the chromosome solves some goal problem. Fitness can also be subjective (aesthetic). E.g., if the genetic algorithm is to be used to sort numbers, then the fitness of a chromosome will be determined by how close to a correct sorting it produces.

- A fitness function quantifies the optimality of a solution (chromosome) so that particular solution may be ranked against all the other solutions. A fitness value is assigned to each solution depending on how close it actually is to solving the problem.
- Ideal fitness function correlates closely to goal plus quickly computable.
- Example : In TSP, f(x) is sum of distances between the cities in solution. The lesser the value, the fitter the solution is.
- The performance of the individual strings is measured by a fitness function. A fitness function is a problem specific user defined heuristic. After each iteration, the members are given a performance measure derived from the fitness function and the "fittest" members of the population will propagate the next iteration.

$$\text{Fitness} = F_i, \ \text{Hit} = \lambda_i, \ \text{Survival} = \phi_i,$$

$$\text{Death} = \delta_i$$

$$F_i = 2\lambda_i - \delta_i + \sum \phi_i$$

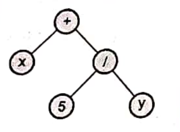
- The fitness function is defined over the genetic representation and measures the ***quality*** of the represented solution. The fitness function is always problem dependent.
- For instance, in the knapsack problem we want to maximize the total value of objects that we can put in a knapsack of some 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.

Q.24 What is encoding ? When to select encoding method ? Explain type of binary encoding methods.

Ans. : • Encoding is the process of representing the solution in the form of a string that conveys the necessary information. It is a process of representing individual genes. The process can be performed using bits, numbers, trees, array or any other objects.

- Just as in a chromosome, each gene controls a particular characteristic of the individual; similarly, each bit in the string represents a characteristic of the solution.

Scanned with CamScanner

Chromosome A	Chromosome B
Chromosome A  (+ x (/ 5 y))	Chromosome B  (do_until step wall)

Q.25 Let a function $f(x) = x - \frac{x^2}{16}$ be defined on the internal [0, 31]. Apply Genetic algorithm for determining the maximum of the given function. (Assume suitable missing data). ☞[RGPV : June-17]

Ans. : • The first step in defining the GA is to code the search space. The encoding used for genetic operations is the binary code for the independent variable x.

• The value of f(x) is the fitness value. Normalized fitness over the whole population determines the probability Pselect of being selected for reproduction.

• Encode the chromosomes and use binary representation for integer. Five bits used to represents integer 31.

• We assume that population size is 4. Initial population is generated randomly. For example : 001101, 11000, 01000, 10011.

• Calculate fitness value. Decode the individual into integer : 001101 → 13, 11000 → 24, 0100 → 8, 10011 → 19.

Then according to function $f(x) = x - x^2/16$

$F(13) = 2.43$, $f(24) = -12$, $f(8) = 4$, $f(19) = -3.56$

Initial Population	Value of X	Fitness function Fi ($f(x) = x - (x^2)/16$)	Pi	Expected count
001101	13	2.43	0.266	1.064
11000	24	– 12	1.31	5.24
01000	8	4	0.438	1.752
10011	19	– 3.56	0.389	1.556
Sum		– 9.13	1	4
Average		– 2.28	0.249	0.998
Max		4	– 0.438	– 1.75

Q.26 Explain different types of mutation function in Genetic algorithm ? ☞[RGPV : June-17, Marks 7]

Ans. : Following are the Types of Mutation function in GA :

1. Insert Mutation- It is used in Permutation encoding.
2. Inversion Mutation- Inversion mutation is used for chromosomes with permutation encoding.
3. Scramble Mutation- Scramble mutation is also used with permutation encoded chromosome.
4. Swap Mutation- It is also used in Permutation encoding
5. Flip Mutation- Based on a generated mutation chromosome, flipping of a bit involves changing 0 to 1 and 1 to 0.
6. Interchanging Mutation - Two random positions of the string are chosen and the bits corresponding to those positions are interchanged.
7. Reversing Mutation - In case of reversing mutation applied for binary encoded chromosome, random position is chosen and the bits next to that position are reversed and child chromosome is produced.
8. Uniform Mutation - The Mutation operator changes the value of chosen gene with uniform random value selected between the user specified upper and lower bound for that gene. It is used in case of real and integer representation.
9. Creep Mutation - In creep mutation, a random gene is selected and its value is changed with a random value between lower and upper bound. It is used in case of real representation

Q.27 Discuss the categorization of bit-wise operator. ☞[RGPV : June-13, Marks 10]

Ans. : 1. One's complement operator : Bitwise one's compliment operator will invert the binary bits. If a bit is 1, it will change it to 0. If the bit is 0, it will change it to 1.

2. Logical bitwise operator : there are three logical bitwise operators. They are Bit-wise AND, Bit-wise Exclusive-OR, Bit-wise OR.

• Bitwise AND (&) operator will take two equal length binary sequence and perform the bitwise AND operation on each pair of bit sequence. AND operator will return 1, if both bits are 1. Otherwise it will return 0.

X	0	0	1	1
Y	0	1	0	1
Result	0	0	0	1

• Bitwise XOR (^) operator will take two equal length binary sequence and perform bitwise XOR operation on each pair of bit sequence. XOR operator will return 1, if both bits are different. If bits are same, it will return 0.

• Bitwise OR (|) operator will take two equal length binary sequence and perform bitwise OR operation on each pair of bit sequence. Bitwise OR will return 1, if any of the bit in the pair is 1. If both bits are 0, then the outcome will be 0.

X	0	0	1	1
Y	0	1	0	1
Result	0	1	1	1

3. Shift Operator : shift operator are of two types, shift left (<<) and shift right (>>).

• Bitwise Left shift (<<) operator is used to shift the binary sequence to the left side by specified position.

• Bitwise Right shift operator (>>) is used to shift the binary sequence to right side by specified position.

Q.28 Write short note on : crossover operator and its types. ☞[RGPV : May-18, Marks 7]

OR What is crossover ? Explain various types of crossover operations used in GA. ☞[RGPV : June-16, Marks 7]

Ans. : • A binary variation operator is called recombination or crossover. A method of mixing good solutions to produce better ones is called crossover.

• Crossover means choosing a random position in the string (say, after 2 digits) and exchanging the segments either to the right or to the left of this point with another string partitioned similarly to produce two new offspring.

• Crossover produces two new offspring from two parent strings by copying selected bits from each parent. Bit at position "i" in each offspring is copied from the bit at position "I" in one of the two parents. The choice which parent contributes bit "I" is determined by an additional string, called cross-over mask.

• In the crossover operator, new strings are created by exchanging information among strings of the mating pool.

• The primary objective of the recombination operator is to emphasize the good solutions and eliminate the bad solutions in a population, while keeping the population size constant.

• "Selects The Best, Discards The Rest". "Recombination" is different from "Reproduction".

• Identify the good solutions in a population. Make multiple copies of the good solutions. Eliminate bad solutions from the population so that multiple copies of good solutions can be placed in the population. It is the process in which two chromosomes (strings) combine their genetic material (bits) to produce a new offspring which possesses both their characteristics. Two strings are picked from the mating pool at random to crossover. The method chosen depends on the encoding method.

• Cross over can be rated complicated and very depends on encoding of the chromosome. Specific crossover made for a specific problem can improve performance of the genetic algorithm.

Steps of crossover :

1. The reproduction operator selects at random a pair of two individual strings for the mating.
2. A cross site is selected at random along the string length.
3. Finally, the position values are swapped between the two strings following the cross site.

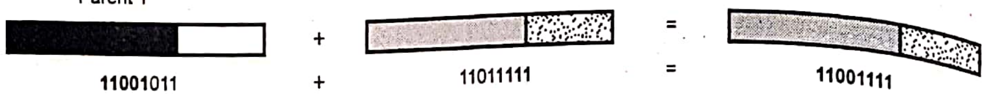
Scanned with CamScanner

Crossover Types

1. Single-point crossover,
2. Two-point crossover,
3. Uniform crossover

One point crossover

- One point crossover is the most basic crossover operator. Crossover point on the genetic code is selected at random and two parent chromosomes are interchanged at this point. Binary string from beginning of chromosome to the crossover point is copied from one parent, the rest is copied from the second parent.



- Example :

Parent 1 :	X X I X X X X X
Parent 2 :	Y Y I Y Y Y Y Y
Offspring 1 :	X X Y Y Y Y Y
Offspring 2 :	Y Y X X X X X

Two Point Crossover

- Two-point crossover is very similar to single-point crossover except that two cut-points are randomly generated instead of one.
- Two point crossover : Two crossover points are selected and the part of the chromosome string between these two points is then swapped to generate two children. Binary string from beginning of chromosome to the first crossover point is copied from one parent, the part from the first to the second crossover point is copied from the second parent and the rest is copied from the first parent.

- Example :

Parent 1 :	X X I X X X I X X
Parent 2 :	Y Y I Y Y Y I Y Y
Offspring 1 :	X X Y Y Y X X
Offspring 2 :	Y Y X X X Y Y

Uniform Crossover

- In uniform crossover, a value of the first parent's gene is assigned to the first offspring and the value of the second parent's gene is to the second offspring with a probability value p_c.

Scanned with CamScanner

Parent 1

1	3	9	8
5	4	7	2
6	12	11	10

Parent 2

4	7	11	9
10	1	5	3
2	12	6	8

Offspring 1

1	3	11	9
5	4	5	3
6	12	6	8

Offspring 2

4	7	9	8
10	1	7	2
2	12	11	10

Fig. Q.28.2 Vertical substring crossover

- Alternatively, the two-dimensional crossover operator can be easily modified to generate four offspring chromosomes from a pair of parents by executing the horizontal and the vertical crossovers at the same time.
- The new offspring chromosomes that result from executing the crossover operation may become infeasible for some application problems.

Q.29 What are different mutation operators ? [RGPV : June-14, Marks 7]

OR Write short note on mutation operator. [RGPV : June-13, Marks 10]

Ans. : • Mutation operators are mutation and mutation rate.

Mutation :

- It is the process by which a string is deliberately changed so as to maintain diversity in the population set. **Mutation probability** determines how often the parts of a chromosome will be mutated.
- Mutation adds new information in a random way to the genetic search process and ultimately helps to avoid getting trapped at local optima.
- It is an operator that introduces diversity in the population whenever the population tends to become homogeneous due to repeated use of reproduction and crossover operators.
- Mutation may cause the chromosomes of individuals to be different from those of their parent individuals.
- If crossover is supposed to exploit the current solution to find better ones, mutation is supposed to help for exploration of the whole search space.
- Mutation in a way is the process of randomly disturbing genetic information. They operate at the bit level; when the bits are being copied from the current string to the new string, there is probability that each bit may become mutated.
- The need for mutation is to create a point in the neighborhood of the current point, thereby achieving a local search around the current solution. The mutation is also used to maintain diversity in the population.
- For example, the following population having four eight bit strings may be considered :

01101011

00111101

00010110

01111100

- It can be noticed that all four strings have a 0 in the left most bit position. If the true optimum solution requires in that position, then neither reproduction nor crossover operator described above will be able to create 1 in that position.

Mutation rate :

- Mutation rate is the probability of mutation which is used to calculate number of bits to be muted.
- Mutation rates are not constant and are not limited to a single type of mutation, therefore there are many different types of mutations.

- Mutation rates are given for specific classes of mutations. Point mutations are a class of mutations which are small or large scale insertions or deletions.
- There are also missense and nonsense mutations, which are variations of point mutations. The rate of these types of substitutions can be further subdivided into a mutation spectrum which describes the influence of the genetic context on the mutation rate

4.3 : Reproduction

Q.30 Under the reproduction phase of genetic algorithm what are the various selection techniques and which is the most effective of the rest. [RGPV : Dec.-16]

OR What is reproduction ? Give various methods of selecting chromosomes for parents to cross over. [RGPV : June-14, Marks 7]

Ans. : • Reproduction is usually the first operator applied on population. Chromosomes are selected from the population to be parents to cross over and produce offspring.

- Reproduction selects good strings in a population and forms a mating pool. This is one of the reasons for the reproduction operation to be sometimes known as the **selection operator.**
- Selection is the process of choosing two parents from the population for crossing. Thus, in reproduction operation the process of natural selection causes those individuals that encode successful structures to produce copies more frequently.
- To sustain the generation of a new population, the reproduction of the individuals in the current population is necessary. For better individuals, these should be from the fittest individuals of the previous population.
- Through **reproduction,** genetic algorithms produce new generations of improved solutions by selecting parents with higher fitness ratings or by giving such parents a greater probability of being contributors and by using random selection.
- If the selection pressure is too low, the convergence rate will be slow and the GA will take unnecessarily longer to find the optimal solution. If the selection pressure is too high, there is an increased change of the GA prematurely converging to an incorrect solution.
- Selection scheme are of two types : **proportionate based selection and ordinal based selection.**

Proportionate Selection

- In proportionate selection, individuals are assigned a probability of being selected based on their fitness :

$$p_i = f_i \sum f_i$$

where p_i is the probability that individual i will be selected,

fi is the fitness of individual i and

$\sum$ fj represents the sum of all the fitnesses of the individuals with the population.

- This type of selection is similar to using a roulette wheel where the fitness of an individual is represented as proportionate slice of wheel. The wheel is then spun and the slice underneath the wheel when it stops determines which individual becomes a parent.
- There are a number of disadvantages associated with using proportionate selection :

1. Cannot be used on minimization problems,
2. Loss of selection pressure (search direction) as population converges,
3. Susceptible to super individuals

Ordinal based selection

In ordinal based selection, individuals are assigned subjective fitness based on the rank within the population :

$$sf_i = (P - r_i)(\max - \min)/(P - 1) + \min$$

where r_i is the rank of indvidual i,

P is the population size,

Max represents the fitness to assign to the best individual,

Min represents the fitness to assign to the worst individual.

Scanned with CamScanner

- $p_i = sf_i / \sum sf_i$ Roulette Wheel Selection can be performed using the subjective fitness.
- One disadvantage associated with linear rank selection is that the population must be sorted on each cycle.
- Selection has to balance with variation from crossover and mutation.

1. Too strong selection means sub optimal highly fit individuals will take over the population, reducing the diversity needed for change and progress.
2. Too weak selection will result in too slow evolution.

- The various methods of selecting chromosomes for parents to crossover are :

1. Roulette-Wheel selection
2. Boltzmann selection
3. Tournament selection
4. Rank selection
5. Steady state selection

Q.31 Write short note on : tournament and rank selection. [RGPV : May-18, Marks 7]

Ans. : Tournament selection :

- Subgroups of individuals are chosen from the larger population and members of each subgroup compete against each other. Only one individual from each subgroup is chosen to reproduce.
- Tournament selection is a method of selecting an individual from a population of individuals in a genetic algorithm.
- Tournament selection involves running several "tournaments" among a few individuals (or "chromosomes") chosen at random from the population. The winner of each tournament is selected for crossover.
- This techniques founds to be computationally more faster than both Roulette-Wheel and Rank-based selection scheme.

Rank selection :

- To overcome the problem with Roulette-Wheel selection, a rank-based selection scheme has been proposed.
- The process of ranking selection consists of two steps.

1. Individuals are arranged in an ascending order of their fitness values. The individual, which has the lowest value of fitness is assigned rank 1, and other individuals are ranked accordingly.
2. The proportionate based selection scheme is then followed based on the assigned rank.

- Each individual in the population is assigned a numerical rank based on fitness and selection is based on these ranking rather than absolute differences in fitness.
- The advantage of this method is that it can prevent very fit individuals from gaining dominance early at the expense of less fit ones, which would reduce the population's genetic diversity and might hinder attempts to find an acceptable solution.

Q.32 Write short note on Routlette-Wheel Selection.

Ans. : • The commonly-used reproduction operator is the proportionate reproduction operator where a string is selected for the mating pool with a probability proportional to its fitness. Thus, the i^{th} string in the population is selected with a probability proportional to F_i.

- The population size is usually kept fixed in a simple GA, the sum of the probability of each string being selected for the mating pools must be one. Therefore, the probability for selecting the i^{th} string is

$$p_i = \frac{F_i}{\sum_{i=1}^{n} F_i}$$

where η is the population size.

- The expected value of an individual is individual's fitness divided by the actual fitness of the population.
- Each current string in the population has a slot assigned to it which is in proportion to it's fitness. We spin the weighted *roulette wheel* thus defined n times (where n is the total number of solutions).
- Each time Roulette Wheel stops, the string corresponding to that slot are created. Strings that

are fitter are assigned a larger slot and hence have a better chance of appearing in the new population.

- One way to implement this selection scheme is to imagine a roulette-wheel with it's circumference marked for each string proportionate to the string's fitness. The roulette-wheel is spun n times. Each time selecting an instance of the string chosen by the roulette-wheel pointer.

Example of Roulette Wheel Selection

Sr. No.	String	Fitness	% of Total
1.	01101	169	14.4
2.	11000	576	49.2
3.	01000	64	5.5
4.	10011	361	30.9
Total	-	1170	100.0

- Fig. Q.32.1 shows a roulette-wheel for each individual having different fitness values.

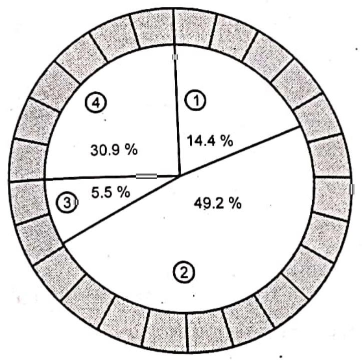


Fig. Q.32.1

4.4 : Generation Cycle and Convergence of GA

Q.33 Explain in detail generation cycle of genetic algorithm. [RGPV : Dec.-16, Marks 7]

OR Draw and explain generation cycle of GA ? [RGPV : June-16, Marks 3]

OR Explain the Genetic Algorithm cycle with example. [RGPV : June-15, Marks 7]

Ans. : • GA represents an iterative process. Each iteration is called a generation. A typical number of generations for a simple GA can range from 50 to over 500. The entire set of generations is called a **run**.

- Because GAs use a stochastic search method, the fitness of a population may remain stable for a number of generations before a superior chromosome appears.
- A common practice is to terminate a GA after a specified number of generations and then examine the best chromosomes in the population. If no satisfactory solution is found, the GA is restarted.

Convergence Analysis

- The convergence of a genetic algorithm arises when the genes of some high rated individuals quickly attain to dominate the population, constraining it to converge to a local optimum. In this case, the genetic operators cannot produce any more descendents better that the parents; the algorithm ability to continue the search for better solutions is therefore substantially reduced.
- The advantages of genetic algorithms first become apparent when a population of strings is observed.
- Let f be the function $x \rightarrow x^2$ which is to be maximized, in the interval [0, 1]. A population of N numbers in the interval [0, 1] is generated in 10-bit fixed-point coding.
- The function f is evaluated for each of the numbers $x_1, x_2, \ldots, x_N$, and the strings are then listed in descending order of their function values. Two strings from this list are always selected to generate a new member of a new population, whereby the probability of selection decreases monotonically in accordance with the ascending position in the sorted list.
- The computed list contains N strings which, for a sufficiently large N, should look like this :

1 0.1 *************

1 0.1 *************

...

...

1 0.0 *************

- The first positions in the list are occupied by strings in which the first bit after the point is a 1. The last positions are occupied by strings in which the first bit after the decimal point is a 0. The asterisk stands for any bit value from 0 to 1 and the zero in front of the point does not need to be coded in the strings.
- The upper strings are more likely to be selected for a recombination, so that the offspring is more likely to contain a 1 in the first bit than a 0. The new population is evaluated and a new fitness list is drawn up.
- Also the strings with a 0 in the first bit are placed at the end of the list and are less likely to be selected than the strings which begin with a 1. After several generations no more strings with a 0 in the first bit after the decimal point are contained in the population.
- The same process is repeated for the strings with a zero in the second bit. They are also pushed towards extinction. Gradually the whole population converges to the optimal string 0.1111111111.
- With this quadratic function the search operation is carried out within a very well-ordered framework. New points are defined at each crossover, but steps in the direction x = 1 are more likely than steps in the opposite direction. The step length is also reduced in each reproduction step in which the 0 bits are eliminated from a position.
- John Holland suggested the notion of schemata for the convergence analysis of genetic algorithms. Schemata are bit patterns which function as representatives of a set of binary strings. We already used such bit patterns in the example above : the bit patterns can contain each of the three symbols 0, 1 or *. The schema **00**, for example, is a representative of all strings of length 6 with two zeros in the central positions, such as : 100000, 110011, 010010, etc.
- Some authors have argued in favor of the building block hypothesis to explain why GAs do well in some circumstances. According to this hypothesis a GA finds building blocks which are then combined through the generations in order to reach the optimal solution. But the correlations between the optimization parameters sometimes preclude altogether the formation of these building blocks.
- Fig. Q.33.1 shows genetic algorithm cycle. Building blocks are combined together due to combined action of generic operators to form bigger and better building blocks and finally converge to the optimal solution.

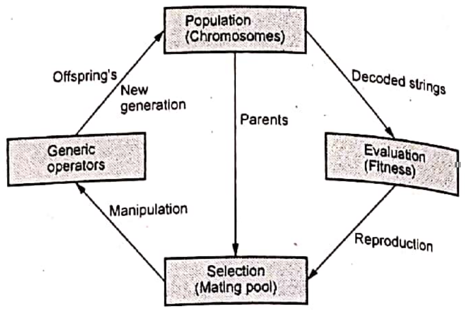


Fig. Q.33.1 Genetic algorithm cycle

Q.34 Write short note on convergence of G.A ☞[RGPV : June-15, Marks 4]

OR What is convergence of Genetic Algorithms ? ☞[RGPV : June-14, Marks 7]

Ans. : Refer Q.33.

4.5 : Application Areas of GA

Q.35 Give a brief note on applications and advances in genetic algorithm. ☞[RGPV : Dec.-16, Marks 3]

Ans. : Applications in genetic algorithm :

1. Composite laminates

- The use of composite laminates finds many applications in mechanical structures, particularly inspired by the need of replacing metals in order to obtain lightweight structures.
- However, composite materials show very special features in comparison to metals because of their heterogeneity, and the architecture of their reinforcement structure can be tailored in order to obtain different types of anisotropy and couplings among different behaviours, which can find such applications as in smart and adaptive structures.

- Therefore, the concept of design and optimisation of the constitutive material is established as a fundamental step in the process of design and optimisation of composite structures.

Advances in genetic algorithm :

- Recent advances in the optimization of fixed time traffic signals have demonstrated a move toward the use of genetic algorithm optimization with traffic network performance evaluated via stochastic microscopic simulation models.
- Application of the cross-generational elitist selection, heterogeneous recombination, and cataclysmic mutation search algorithm with real crossover and mutation operators is found to offer improved optimization efficiency over the standard genetic algorithm with binary genetic operators

Q.36 Write a note on the following i) Advances in GA ii) Convergence of GA ☞[RGPV : June-16, Marks 7]

Ans. : Refer Q.33 and Q.37.

Q.37 Explain advanced in GA. ☞[RGPV : June-13, Marks 10]

Ans. : • Genetic Algorithm is an artificial intelligence procedure. It is based on the theory of natural selection and evolution. This search algorithm balances the need for :

1. **Exploitation** : Selection and crossover tend to converge on a good but sub-optimal solution.
2. **Exploration** : Selection and mutation create a parallel, noise-tolerant, hill climbing algorithm, preventing a premature convergence.

- Traditional methods of search and optimization are too slow in finding a solution in a very complex search space, even implemented in supercomputers. Genetic Algorithm is a robust search method requiring little information to search effectively in a large or poorly-understood search space.
- In particular a genetic search progress through a population of points in contrast to the single point of focus of most search algorithms. Moreover, it is useful in the very tricky area of nonlinear problems.
- Basically, Genetic Algorithm requires two elements for a given problem :

a. Encoding of candidate structures

b. Method of evaluating the relative performance of candidate structure, for identifying the better solutions

- Genetic Algorithm codes parameters of the search space as binary strings of fixed length. It employs a population of strings initialized at random, which evolve to the next generation by genetic operators such as selection, crossover and mutation.
- The fitness function evaluates the quality of solutions coded by strings. Selection allows strings with higher fitness to appear with higher probability in the next generation.
- Crossover combines two parents by exchanging parts of their strings, starting from a randomly chosen crossover point. This leads to new solutions inheriting desirable qualities from both parents.
- Mutation flips single bits in a string, which prevents the GA from premature convergence, by exploiting new regions in the search space. GA tends to take advantage of the fittest solutions by giving them greater weight and concentrating the search in the regions which lead to fitter structures and hence better solutions of the problem.
- Finding good parameter settings that work for a particular problem is not a trivial task. The critical factors are to determine robust parameter settings for population size, encoding, selection criteria, genetic operator probabilities and evaluation (fitness) normalization techniques.
- If the population is too small, the genetic algorithm will converge too quickly to a local optimal point and may not find the best solution. On the other hand, too many members in a population result in long waiting times for significant improvement.
- Coding the solutions is based on the principle of meaningful building blocks and the principle of minimal alphabets, by using the binary strings.
- The fitter member will have a greater chance of reproducing. The members with lower fitness are replaced by the offspring. Thus in successive generations, the members on average are fitter as solutions to the problem.
- Too high mutation introduces too much diversit and takes longer time to get the optimal solutio

Scanned with CamScanner

Too low mutation tends to miss some near-optimal points. Two point crossovers are quicker to get the same results and retain the solutions much longer than one point crossover.

- The fitness function links the Genetic Algorithm to the problem to be solved. The assigned fitness is used to calculate the selection probabilities for choosing parents for determining which member will be replaced by which child.
- Genetic Algorithm is applicable in many ways : State Assignment Problem, Economics, Scheduling and Computer-Aided Design.

Q.38 How traveling salesperson problem is solved using genetic algotithm ?

Ans. : • Suppose a salesperson has five cities to visit and then must return home.

- The goal of the problem is to find the shortest path for the salesperson to travel.

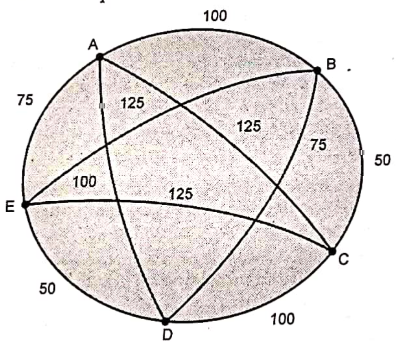


Fig. Q.38.1

- The traveling salesman must visit every city in his territory exactly once (possibly then return to the starting point). Given the cost of travel between all cities, how should he plan his itinerary for minimum total cost of the entire tour ?

Representation :

- Representation is an ordered list of city numbers known as an *order-based* GA.

1) Mumbai 3) Surat 5) Pune 7) Nagpur

2) Calcutta 4) Delhi 6) Bangalore 8) Chennai

CityList1 (3 5 7 2 1 6 4 8)

CityList2 (2 5 7 6 8 1 3 4)

- Crossover combines inversion and recombination :

Parent1	(3 5 <u>7 2 1 6 4</u> 8)	
Parent2	(2 5 7 6 8 1 3 4)	
Child	(5 8 <u>7 2 1 6</u> 3 4)	

This operator is called the ***Order1* crossover.**

- Mutation involves reordering of the list :

Before : (5 8 7 2 1 6 3 4)

After : (5 8 6 2 1 7 3 4)

Traveling Salesperson problem with 9 city

- Now, the problem is how to crossover ?

P1 = (192465783)

P2 = (459187623)

- First of all, select two cut point, indicate by a "|", which are randomly inserted into the same location of each parent.

P1 = (192 | 4657 | 83)

P2 = (459 | 1876 | 23)

Two children C1 and C2 are produced in the following way.

First, the segments between cut points are copied into the offspring :

C1 = (XXX | 4657 | XX)

C2 = (XXX | 1876 | XX)

Next, starting from the second cut point of one parent, the cities from the other parent are copied in the same order, omitting cities already present. When the end of the string is reached, continue on from the beginning.

Thus, the sequence of cities from P2 (459 | 1876 | 23) is 23 459 1876.

For C1= (XXX | 4657 | XX), once 4657 are removed from the sequence generated by P2, we get the sequence 23918.

Then we just use these numbers to fill in the XXX XX portion in order.

Thus, C1 = (239 | 4657 | 18)

Important Points to Remember

- A genetic algorithm is a search technique used in computing to find true or approximate solutions to optimization and search problems.
- Genetic programming is genetic algorithm wherein the population contains programs rather than bit strings.
- Crossover operator is used to recombine two strings to get a better string.
- A fitness function is a type of objective functions which summarizes the goodness of a solution with a single figure of merit.
- Encoding is the process of representing the solution in the form of a string that conveys the necessary information.
- Mutation operators are mutation and mutation rate.
- Mutation rate is the probability of mutation which is used to calculate number of bits to be muted.
- Tournament selection is a method of selecting an individual from a population of individuals in a genetic algorithm.

END...

Scanned with CamScanner

5 Advanced Soft Computing Techniques

5.1 Rough Set Theory

Q.1 What is set and rough set ?

Ans. : Set : A set of objects that possesses similar characteristics it is a fundamental part of mathematics. All the mathematical objects, such as relations, functions and numbers can be considered as a set.

- The various components of a set are known as elements and relationship between an element and a set is called of a pertinence relation. Cardinality is the way of measuring the number of elements of a set.

Rough set

- In the rough set theory, membership is not the primary concept. Rough sets represent a different mathematical approach to vagueness and uncertainty.
- The rough set methodology is based on the premise that lowering the degree of precision in the data makes the data pattern more visible.

Q.2 Define reducts.

Ans. : The process of reducing an information system such that the set of attributes of the reduced information system is independent and no attribute can be eliminated further without losing some information from the system, the result is known as reduct.

Q.3 What is rough approximation ? What is lower and upper approximation ? List the properties of approximation.

Ans. : • Rough set concept can be defined quite generally by means of topological operations, interior and closure, called **approximations.**

- Suppose we are given a set of objects U called the universe and an indiscernibility relation $R \subseteq U \times U$, representing our lack of knowledge about elements of U. For the sake of simplicity we assume that R is an equivalence relation.
- Let X is a subset of U. We want to characterize the set X with respect to R. To this end we will need the basic concepts of rough set theory given below.

1. The lower approximation of a set X with respect to R is the set of all objects, which can be for certain classified as X with respect to R (are certainly X with respect to R).
2. The upper approximation of a set X with respect to R is the set of all objects which can be possibly classified as X with respect to R (are possibly X in view of R).
3. The boundary region of a set X with respect to R is the set of all objects, which can be classified neither as X nor as not-X with respect to R.

- We are ready to give the definition of rough sets.

1. Set X is crisp (exact with respect to R), if the boundary region of X is empty.
2. Set X is rough (inexact with respect to R), if the boundary region of X is nonempty.

- *A set is said to be rough if its boundary region is non-empty, otherwise the set is crisp.*
- Formal definitions of approximations and the boundary region are as follows :
- R-lower approximation of X

$$R_*(x) = \bigcup_{x \in U} \{R(x) : R(x) \subseteq X\}$$

- R-upper apprixination of X

$$R^*(x) = \bigcup_{x \in U} \{R(x) : R(x) \cap X \neq \phi\}$$

- R-boundary region of X

$$RN_R(X) = R^*(x) - R_*(X)$$

- The **lower approximation** of a set is union of all granules which are entirely included in the set; the **upper approximation** is union of all granules which have non-empty intersection with the set; the **boundary region** of set is the difference between the upper and the lower approximation.

- Fig. Q.3.1 shows approximations.
- Approximations have the following properties :

1) $R_*(X) \subseteq X \subseteq R^*(X)$
2) $R_*(\phi) = R^*(\phi) = \phi.\ R_*(U) = R^*(U) = U$
3) $R^*(X \cup Y) = R^*(X) \cup R^*(Y)$
4) $R_*(X \cap Y) = R_*(X) \cap R_*(Y)$
5) $R_*(X \cup Y) \supseteq R_*(X) \cup R_*(Y)$
6) $R^*(X \cap Y) \subseteq R^*(X) \cap R^*(Y)$
7) $X \subseteq Y \rightarrow R_*(X) \subseteq R_*(Y) \ \&\ R^*(X) \subseteq R^*(Y)$
8) $R_*(-X) = -R^*(X)$
9) $R^*(-X) = R_*(X)$
10) $R_*R_*(X) = R^*R_*(X) = R_*(X)$
11) $R^*R^*(X) = R_*R^*(X) = R^*(X)$

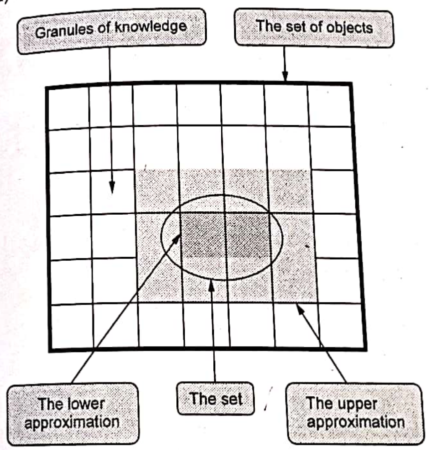


Fig. Q.3.1 Approximations

Q.4 Explain indiscernibility relation with example.

Ans. : • Indiscernibility relation is a central concept in rough set theory, and is considered as a relation between two objects or more, where all the values are identical in relation to a subset of considered attributes.

- A decision system (i.e. a decision table) expresses the entire model. This table may be unnecessarily large. The same or indiscernible objects may be represented several times.
- An information system or information table can be viewed as a table, consisting of objects (rows) and attributes (columns). It is used in the representation of data that will be utilized by rough set, where each object has a given amount of attributes.
- These objects are described in accordance with the format of the data table, in which rows are considered objects for analysis and columns as attributes. Below is shown an example of information.

Patient	Attributes			
	Headache	Vomiting	Temperature	Viral illness
#1	No	Yes	High	Yes
#2	Yes	No	High	Yes
#3	Yes	Yes	Very high	Yes
#4	No	Yes	Normal	No
#5	Yes	No	High	No
#6	No	Yes	Very high	Yes

Table Q.4.1

- It can be observed that the set is composed of attributes that are directly related to the patients' symptoms whether they be headache, vomiting and temperature.
- When Table Q.4.1 is broken down it can be seen that the set regarding {patient2, patient3, patient5} is indiscernible in terms of headache attribute.
- The set concerning {patient1, patient3, patient4} is indiscernible in terms of vomiting attribute. Patient2 has a viral illness, whereas patient5 does not, however they are indiscernible with respect to the attributes headache, vomiting and temperature. Therefore, patient2 and patient5 are the elements of patients' set with unconcluded symptoms.
- A **binary relation** $R \subseteq X \times X$ which is **reflexive** (i.e. an object is in relation with itself xRx), **symmetric** (if xRy then yRx) and **transitive** (if xRy and yRz then xRz) is called an **equivalence relation.**
- The **equivalence class** of an element $x \in X$ consists of all objects $y \in X$ such that **xRy.**

Scanned with CamScanner

- Let A = (U, A) be an information system then with any B ⊆ A there is associated an *equivalence relation* IND_A *(B) :*

$$INDA(B) = \{(x, x') \in U2 \mid a \in B\ a(x) = a(x')\}$$

- IND_A (B) is called the *B-indiscernibility relation.* If $(x, x') \in IND_A(B)$ then x and x' are indiscernible from each other by attributes from B. The equivalence classes of the *B-indiscernibility* relation are denoted $[x]_B$.

Q.5 Consider the following table .

U	Headache	Temp.	Flu
U1	Yes	Normal	No
U2	Yes	High	Yes
U3	Yes	Very-high	Yes
U4	No	Normal	No
U5	No	High	No
U6	No	Very-high	Yes
U7	No	High	Yes
U8	No	Very-high	No

Find indiscernibility classes.

Ans. : The indiscernibility classes defined by R = {Headache, Temp.}

= {u1}, {u2}, {u3}, {u4}, {u5, u7}, {u6, u8}.

X1 = {u | Flu(u) = yes} = {u2, u3, u6, u7}

Lower approximation

$\underline{R}X1$ = {u2, u3}

Upper approximation

$\overline{R}X1$ = {u2, u3, u6, u7, u8, u5}

Q.6 Explain advantages of rough set approach.

Ans. : • It does not need any preliminary or additional information about data - like probability in statistics, grade of membership in the fuzzy set theory.

- It provides efficient methods, algorithms and tools for finding hidden patterns in data.
- It allows to reduce original data, i.e. to find minimal sets of data with the same knowledge as in the original data.
- It allows to evaluate the significance of data.
- It allows to generate in automatic way the sets of decision rules from data.
- It is easy to understand.
- It offers straightforward interpretation of obtained results.
- It is suited for concurrent (parallel/distributed) processing.

5.2 Support Vector Machine (SVM)

Q.7 What is support vector machine ?

Ans. : • Support Vector Machines (SVMs) are a set of supervised learning methods which learn from the dataset and used for classification.

- An SVM is a kind of large-margin classifier : It is a vector space based machine learning method where the goal is to find a decision boundary between two classes that is maximally far from any point in the training data.
- Given a set of training examples, each marked as belonging to one of two classes, an SVM algorithm builds a model that predicts whether a new example falls into one class or the other.
- Simply speaking, we can think of an SVM model as representing the examples as points in space, mapped so that each of the examples of the separate classes are divided by a gap that is as wide as possible.

Q.8 What is two class problems ?

Ans. : • Consider two-input patterns (X_1, X_2) being classified into two classes as shown in Fig. Q.8.1. Each point with either symbol of x or 0 represents a pattern with a set of values (X_1, X_2).

- Linearly separable sets : There exist a hyperplane (here a line) that correctly classifies all points.
- Each pattern is classified into one of two classes. Notice that these classes can be separated with a single line L. They are known as linearly separable patterns.

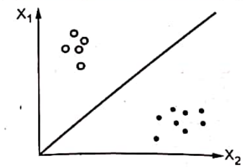


Fig. Q.8.1 Two class

- Linear separability refers to the fact that classes of patterns with n-dimensional vector $x = (x_1, x_2, \ldots x_n)$ can be separated with a single decision surface. In the case above, the line L represents the decision surface.
- If two classes of patterns can be separated by a decision boundary, represented by the linear equation then they are said to be linearly separable. The simple network can correctly classify any patterns.
- Decision boundary of linearly separable classes can be determined either by some learning procedures or by solving linear equation systems based on representative patterns of each class.
- If such a decision boundary does not exist, then the two classes are said to be linearly inseparable.
- Linearly inseparable problems cannot be solved by the simple network, more sophisticated architecture is needed.

Q.9 Explain empirical risk minimization.

Ans. : • The empirical risk is the average loss of an estimator for a finite set of data drawn from P.

- The idea of risk minimization is not only measure the performance of an estimator by its risk, but to actually search for the estimator that minimizes risk over distribution P.
- Because we don't know distribution P we instead minimize empirical risk over a training dataset drawn from P. This general learning technique is called **empirical risk minimization.**
- Fig. Q.9.1 shows empirical risk.

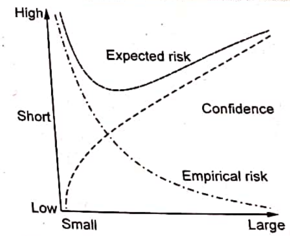


Fig. Q.9.1 Complexity of function set

Q.10 Explain key properties of SVM.

Ans. : 1. Use a single hyperplane which subdivides the space into two half-spaces, one which is occupied by Class 1 and the other by Class 2

2. They maximize the margin of the decision boundary using quadratic optimization techniques which find the optimal hyperplane.
3. Ability to handle large feature spaces.
4. Overfitting can be controlled by soft margin approach.
5. When used in practice, SVM approaches frequently map the examples to a higher dimensional space and find margin maximal hyperplanes in the mapped space, obtaining decision boundaries which are not hyperplanes in the original space.
6. The most popular versions of SVMs use non-linear kernel functions and map the attribute space into a higher dimensional space to facilitate finding "good" linear decision boundaries in the modified space.

Q.11 Explain slack variable, hard margin and soft margin.

Ans. : • For the very high dimensional problems common in text classification, sometimes the data are linearly separable. But in the general case they are not, and even if they are, we might prefer a solution that better separates the bulk of the data while ignoring a few weird noise documents.

- What if the training set is not linearly separable ? *Slack variables* can be added to allow misclassification of difficult or noisy examples, resulting margin called *soft*.

Scanned with CamScanner

- A soft-margin allows a few variables to cross into the margin or over the hyperplane, allowing misclassification.
- We penalize the crossover by looking at the number and distance of the misclassifications. This is a trade off between the hyperplane violations and the margin size. The slack variables are bounded by some set cost. The farther they are from the soft margin, the less influence they have on the prediction.
- All observations have an associated slack variable
  1. Slack variable = 0 then all points on the margin.
  2. Slack variable > 0 then a point in the margin or on the wrong side of the hyperplane.
  3. C is the tradeoff between the slack variable penalty and the margin.

Q.12 From the following diagram, identify which data points (1, 2, 3, 4, 5) are support vectors (if any), slack variables on correct side of classifier (if any) and slack variables on wrong side of classifier (if any). Mention which point will have maximum penalty and why ?

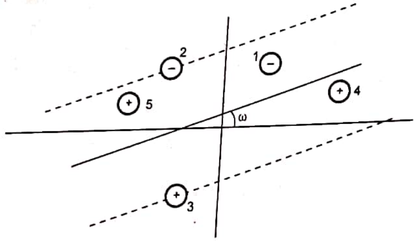


Fig. Q.12.1

Ans. : • Data points 1 and 5 will have maximum penalty.

- Margin (m) is the gap between data points and the classifier boundary. The margin is the minimum distance of any sample to the decision boundary. If this hyperplane is in the canonical form, the margin can be measured by the length of the weight vector.
- Maximal margin classifier : A classifier in the family F that maximizes the margin. Maximizing the margin is good according to intuition and PAC theory. Implies that only support vectors matter; other training examples are ignorable.
- What if the training set is not linearly separable ? Slack variables can be added to allow misclassification of difficult or noisy examples, resulting margin called soft.
- A soft-margin allows a few variables to cross into the margin or over the hyperplane, allowing misclassification.
- We penalize the crossover by looking at the number and distance of the misclassifications. This is a trade off between the hyperplane violations and the margin size. The slack variables are bounded by some set cost. The farther they are from the soft margin, the less influence they have on the prediction.
- All observations have an associated slack variable
  1. Slack variable = 0 then all points on the margin.
  2. Slack variable > 0 then a point in the margin or on the wrong side of the hyperplane.
  3. C is the tradeoff between the slack variable penalty and the margin.

Q.13 Compare SVM and NN.

Ans. :

Support vector machine	Neural network
Kernel maps to a very-high dimensional space.	Hidden Layers map to lower dimensional spaces.
Search space has a unique minimum.	Search space has multiple local minima.
Classification not efficient.	Classification extremely efficient.
Very good accuracy in typical domains.	Very good accuracy in typical domains.
Kernel and cost the two parameters to select.	Requires number of hidden units and layers.
Training is extremely efficient.	Training is expensive.

5.3 Introduction to Swarm Intelligence

Q.14 What is an agent ? What is a swarm ? Define swarm intelligence. Give an examples of swarms in nature.

Ans. : • An agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through effectors.

- Swarm is a loosely structured collection of interacting agents. Agents is an individuals that belong to a group. They contribute to and benefit from the group. They can recognize, communicate, and/or interact with each other.
- The instinctive perception of swarms is a group of agents in motion, but that does not always have to be the case.
- A swarm is better understood if thought of as agents exhibiting a collective behavior.
- Examples of Swarms in Nature :
  1. Classic Example : Swarm of Bees
  2. Can be extended to other similar systems : Ant colony (Agents : ants), Flock of birds(Agents : birds), Traffic(Agents : cars), Crowd(Agents : humans), Immune system(Agents : cells and molecules)
- Swarm Intelligence : Any attempt to design algorithms or distributed problem-solving devices inspired by the collective behavior of social insect colonies and other animal societies.
- Swarm intelligence (SI) is an artificial intelligence technique based around the study of collective behavior in decentralized, self-organized systems.

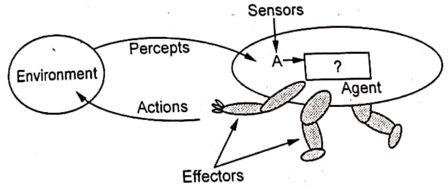


Fig. Q.14.1

Q.15 List the advantages of swarm intelligence.

Ans. : • The systems are scalable because the same control architecture can be applied to a couple of agents or thousands of agents.

- The systems are flexible because agents can be easily added or removed without influencing the structure.
- The systems are robust because agents are simple in design, the reliance on individual agents is small, and failure of a single agents has little impact on the system's performance.
- The systems are able to adapt to new situations easily.

5.4 Swarm Intelligence Techniques

Q.16 Write short note on particle swarm optimization.

Ans. : • Particle Swarm Optimization (PSO) is a population based stochastic optimization technique developed by Dr. Eberhart and Dr. Kennedy in 1995, inspired by social behavior of bird flocking or fish schooling.

- PSO system combines local search methods with global search methods, attempting to balance exploration and exploitation.
- Suppose a group of birds is searching food in an area. Only one piece of food is available. Birds do not have any knowledge about the location of the food. But they know how far the food is from their present location.
- **So what is the best strategy to locate the food ? The best strategy is to follow the bird nearest to the food.**
- A flying bird has a position and a velocity at any time. In search of food, the bird changes his position by adjusting the velocity. The velocity changes based on his past experience and also the feedbacks received from his neighbor.
- This searching process can be artificially simulated for solving non-linear optimization problem. So this is a population based stochastic optimization technique inspired by social behaviour of bird flocking or fish schooling.
- PSO shares many similarities with evolutionary computation techniques such as Genetic Algorithms. The system is initialized with a population of random solutions and searches for optima by updating generations.

Scanned with CamScanner

- In PSO algorithms, the population $P = \{p_1,...,p_n\}$ of the feasible solutions is often called a **swarm.** The feasible solutions $p_1,...,p_n$ are called **particles.**
- However, unlike GA, PSO has no evolution operators such as crossover and mutation. In PSO, the potential solutions, called particles, fly through the problem space by following the current optimum particles.
- Each particle keeps track of its coordinates in the problem space which are associated with the best solution (fitness) it has achieved so far.
- The fitness value is also stored. This value is called pbest. Another "best" value that is tracked by the particle swarm optimizer is the best value, obtained so far by any particle in the neighbors of the particle. This location is called lbest. When a particle takes all the population as its topological neighbors, the best value is a global best and is called gbest.
- The particle swarm optimization concept consists of, at each time step, changing the velocity of (accelerating) each particle toward its pbest and lbest locations. Acceleration is weighted by a random term, with separate random numbers being generated for acceleration toward pbest and lbest locations.
- In past several years, PSO has been successfully applied in many research and application areas. It is demonstrated that PSO gets better results in a faster, cheaper way compared with other methods.

- **Advantages and disadvantages**

1. The fitness function can be non-differentiable
2. There is no general convergence theory applicable to practical, multidimensional problems.

Q.17 Compare Particle Swarm Optimization (PSO) with Genetic Algorithms (GA).

Ans. : • Unlike GAs, PSOs do not change the population from generation to generation, but keep the same population, iteratively updating the positions of the members of the population (i.e., particles).

- PSOs have no operators of "mutation", "recombination", and no notion of the "survival of the fittest".
- On the other hand, similarly to GAs, an important element of PSOs is that the members of the population "interact", or "influence" each other.
- GA usually converges towards a local optimum or even arbitrary points rather than the global optimum of the problem while as PSO tries to find the global optima.
- GA is discrete in nature, i.e. it changes the variables into binary 0's and 1's, and therefore it can easily handle discrete problems, and PSO is continuous and hence must be modified in order to handle discrete problems.

Q.18 Write an algorithm of particle swarm optimization.

Algorithm parameters :

A : Population of agents

p_i : Position of agent a_i in the solution space

f : Objective function

v_i : Velocity of agent's a_i

$V(a_i)$: Neighborhood of agent a_i (fixed)

- Particle update rule : $p = p + v$

With $v = v + c_1^* \text{ rand } *(pBest - p) + c_2^* \text{ rand } *(gBest - p)$

where

p : Particle's position

v : Path direction

c1 : Weight of local information

c2 : Weight of global information

pBest : Best position of the particle

gBest : Best position of the swarm

rand : Random variable

Algorithm :

1. Create a 'population' of agents (particles) uniformly distributed over X.
2. Evaluate each particle's position according to the objective function.
3. If a particle's current position is better than its previous best position, update it.
4. Determine the best particle (according to the particle's previous best positions).
5. Update particles' velocities :

$$\boxed{v_i^{t+1} = \underbrace{v_i^t}_{\text{inertia}} + \underbrace{c_1 U_1^t (pb_i^t - p_i^t)}_{\text{personal influence}} + \underbrace{c_2 U_2^t (gb^t - p_i^t)}_{\text{social inflence}}}$$

6. Move particles to their new positions :

$$\boxed{p_i^{t+1} = p_i^t + v_i^{t+1}}$$

7. Go to step 2 until stopping criteria are satisfied.

Q.19 What is Binary Particle Swarm Optimization (BPSO) ? How does it work ?

Ans. : • In BPSO, each solution in the population is a binary string. Each binary string is of dimension "n" which is evaluated to give parameter values.

- In the binary PSO, each binary string represents a particle. Strings are updated bit-by-bit based on its current value, the value of that bit in the best (fitness) of that particle to date, and the best value of that bit to date of its neighbors.
- For binary strings, neighbors can be selected in one of several ways. Some examples are : (for a neighbourhood of size k). Neighbours are the k binary strings whose Hamming distance is minimized. For equal Hamming distances, the choices are arbitrary.
- In BPSO, bit-by-bit updates are done probabilistically. In other words, for a chosen bit "d" in a chosen string "i" it is changed to a 1 with a probability "P" that is a function of its predisposition to be a 1, the best value of itself to date, and the best value of its neighbors.
- The (1 – P) is the probability of changing to a zero.Once P is determined, a random number "R" is generated. If R < P, then the bit becomes a 1; otherwise it becomes a zero.
- In BPSO, population has a set of particles. Each individual particle represents a binary decision.
- This decision can be represented by either YES/TRUE = 1 or NO/FALSE = 0.
- All particles represent their positions through binary values which are 0 or 1.Velocity is restricted within the range {0,1}
- The velocity vector equation and position vector equation are defined as :

1. Velocity vector equation :

$$v_i^n(t+1) = \frac{1}{1+e^{-v_i^n(t)}}$$

2. Position vector equation :

$$x_i^n(t+1) = \begin{cases} 1 & \text{if } \quad r < v_i^n \\ 0 & \text{otherwise} \end{cases},$$

where r is the random number selected from a uniform distribution in [0, 1].

Q.20 Explain flow diagram showing the particle swarm optimization algorithm with Pseudo code.

Ans. : Flow diagram : (See on next page)

Pseudo Code :

```
for each particle
{
    Initialize particle
}
Do until maximum iterations or minimum error criteria
{
    For each particle
    {
        Calculate Data fitness value
        If the fitness value is better than pBest
        {
            Set pBest = current fitness value
        }
        If pBest is better than gBest
        {
            Set gBest = pBest
        }
    }
    For each particle
    {
        Calculate particle Velocity
        Use gBest and Velocity to update particle Data
    }
```


Scanned with CamScanner

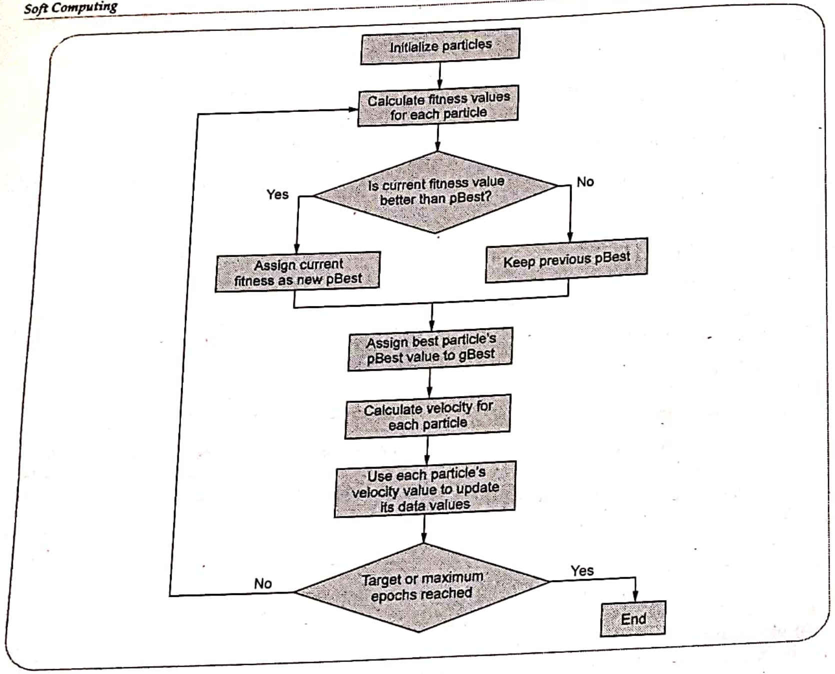


Fig. Q.20.1

Q.21 Discuss topology used in particle swarm optimization.

Ans. : • In PSO, there have been two basic topologies used.

1. Ring Topology (neighborhood of 3)

2. Star Topology (global neighborhood)

Star Topology :

- Fig. Q.21.1 shows star topology.
- gbest model : Each particle is influenced by all the other particles.
- The fastest propagation of information in a population.
- Particles can get stuck easily in local minima.

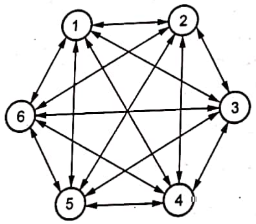


Fig. Q.21.1

- Star neighborhood topology has greater connectivity and interaction between particles compared with other topologies.
- Star neighborhood topology enables fast convergence of the algorithm.

Ring Topology :

- Fig. Q.21.2 shows ring topology.
- lbest model : Each particle is influenced only by particles in its own neighbourhood.
- The propagation of information is the slowest.
- Doesn't get stuck easily in local minima but might increase the computational cost.
- Note that, neighborhood topologies are usually defined by the particle index, and not by the particle location.

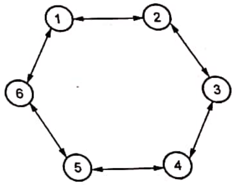


Fig. Q.21.2

Q.22 Explain global best (gbest) and local best (lbest) particle swarm optimization.

Ans. : Global best (gbest) PSO :

- For the global best (gbest) PSO, the neighborhood for each particle is the entire swarm. The social network employed by the gbest PSO reflects the star topology.
- gbest : Each particle is influenced by the best found from the entire swarm.
- Velocity update per dimension :

$$v_{ij}(t+1) = v_{ij}(t) + c_1 r_{1j}(t)[y_{ij}(t) - x_{ij}(t)] + c_2 r_{2j}(t)[\hat{y}_j(t) - x_{ij}(t)]$$

$v_{ij}(0) = 0$ (usually, but can be random) and c_1, c_2 are positive acceleration coefficients.

gbest PSO Algorithm :

```
Create and initialize an n_x-dimensional swarm, S;
repeat
    for each particle i = 1, . . . , S.n_s do
      if f(S·y_i) < f(S·y) then
        S·y_i = S·x_i ;
      end
      if f(S·y_i) < f(S·^y) then
          S·^y = S·y_i ;
      end
    end
  for each particle i = 1, . . . , S·n_s do
  update the velocity and then the position;
  end
until stopping condition is true ;
```

Local best PSO

- The local best (lbest) PSO, uses a ring social network topology where smaller neighborhoods are defined for each particle.
- The social component reflects information exchanged within the neighborhood of the particle, reflecting local knowledge of the environment.
- With reference to the velocity equation, the social contribution to particle velocity is proportional to the distance between a particle and the best position found by the neighborhood of particles.

Scanned with CamScanner

- Due to the larger particle interconnectivity of the gbest PSO, it converges faster than the lbest PSO. However, this faster convergence comes at the cost of less diversity than the lbest PSO.
- The lbest model is less vulnerable to the attraction of local optima but with a slower convergence speed than the gbest model.

lbest PSO algorithm :

```
Create and initialize and n_x-dimensional swarm;
  repeat
      for each particle i = 1 ... n_s do
          //set the personal best position
          if f(x_i) < f(y_i) then
              y_i = x_i;
          end
          //set the neighborhood best position
          if f(y_i) < f(ŷ_i) then
              ŷ = y_i;
      end
   end
   for each particle i = 1,...n_s do
        update the velocity
        update the position
   end
until stopping condition is true;
```

Q.23 Why used PSO ?

Ans. : • With a population of candidate solutions, a PSO algorithm can maintain useful information about characteristics of the environment.

- PSO, as characterized by its fast convergence behavior, has an in-built ability to adapt to a changing environment.
- Some early works on PSO have shown that PSO is effective for locating and tracking optima in both static and dynamic environments.

Q.24 Explain difference between gbest and lbest PSO.

gbest PSO	lbest PSO
Global best particle swarm optimization uses a "star" neighborhood topology where all the particles communicate with each other.	Local best particle swarm optimization can be used with different neighborhood topologies with the "ring" topology being the most common.
gbest PSO converges faster compared with local best particle swarm optimization.	lbest PSO converges is slower compared with global best particle swarm optimization.
For the global best PSO, the neighborhood for each particle is the entire swarm.	In lbest PSO, smaller neighborhoods are defined for each particle.
The gbest PSO performs best for unimodal problems.	In lbest, ring structure to provide better performance in terms of the quality of solutions found for multi-modal problems than the star structure.

Q.25 Write short note on ant colony optimization.

Ans. : • The way ants find their food in shortest path is interesting.

- Ants secrete pheromones to remember their path. These pheromones evaporate with time.

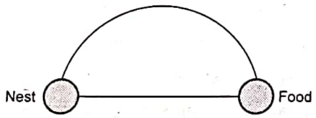


Fig. Q.25.1

- Whenever an ant finds food , it marks its return journey with pheromones. Pheromones evaporate faster on longer paths. Shorter paths serve as the way to food for most of the other ants.
- The shorter path will be reinforced by the pheromones further. Finally , the ants arrive at the shortest path.
- Ant Colony Optimization (ACO) is a paradigm for designing metaheuristic algorithms for combinatorial optimization problems.
- Metaheuristic algorithms are algorithms which, in order to escape from local optima, drive some basic heuristic : Either a constructive heuristic starting from a null solution and adding elements to build a good complete one, or a local search heuristic starting from a complete solution and iteratively modifying some of its elements in order to achieve a better one.
- ACO can be used to solve hard problems like TSP, Quadratic Assignment Problem (QAP).

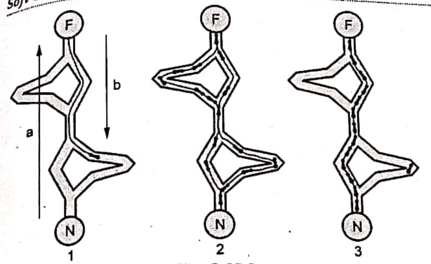


Fig. Q.25.2

- An ant will move from node i to node j with probability :

$$p_{i,j} = \frac{(\tau_{i,j}^{\alpha})(\eta_{i,j}^{\beta})}{\sum(\tau_{i,j}^{\alpha})(\eta_{i,j}^{\beta})}$$

where

$\tau_{i,j}$ is the amount of pheromone on edge i, j.

α is a parameter to control the influence of $\tau_{i,j}$.

$\eta_{i,j}$ is the desirability of edge i, j (typically $1/d_{i,j}$)

β is a parameter to control the influence of $\eta_{i,j}$.

- Amount of pheromone is updated according to the equation :

$$\tau_{i,j} = (1-\rho)\tau_{i,j} + \Delta\tau_{i,j}$$

where

$\tau_{i,j}$ is the amount of pheromone on a given edge i, j.

ρ is the rate of pheromone evaporation.

$\Delta\tau_{i,j}$ is the amount of pheromone deposited, typically given by,

$$\Delta\tau_{i,j}^{k} = \begin{cases} 1/L_k & \text{if ant k travels on edge i,j} \\ 0 & \text{otherwise} \end{cases}$$

- Where L_k is the cost of the k^{th} ant's tour (typically length).

Q.26 Explain various application of ACO and PSO.

Ans. : 1. Open shop scheduling :

- ACO can be combined with tree search methods with the aim to improve methods for solving combinatorial optimization problems.
- In combination of ACO and beam search is described. It has been applied to open shop scheduling, which is a NP-hard scheduling problem.
- The obtained results compared with two best approaches currently available Standard- show that the Beam-ACO performs better.

2. PSO algorithm in signal detection and blind extraction.

- In signal processing there are among others two important problems, namely multi-user detection and blind extraction of sources.
- The optimal multi-user detection is a NP-complete problem. Therefore the research effort has been focused on the development of suboptimum techniques with the aim to achieve a trade-off between complexity and performance.
- Binary PSO algorithm was used and reached better results than genetic algorithm. Similar results were achieved in blind source separation, which is an important issue in many science and engineering scenarios as well.

3. PSO for Solving Travelling Salesman Problem (TSP)

- TSP is a well-known NP-hard problem. Modification of PSO using fuzzy matrices to represent the position and velocity of the particles in PSO was developed.
- The algorithm can be used for resolving common routing problems and other combinatorial optimization problems.
- Discovering clusters in spatial data. In this application, each agent represents a simple task and the success of the method depends on the cooperative work of the agents.
- The algorithm combines a smart exploratory strategy based on the movements of a flock of birds with a shared nearest-neighbour clustering algorithm to discover clusters in parallel.
- Birds are used as agents with an exploring behaviour foraging for clusters. This strategy can be used as a data reduction technique to perform approximate clustering efficiently.

Scanned with CamScanner

4. Discovering clusters in biomedical signals

- In this application, individual parts of the signal are characterized by real valued features and the resulting set of samples is partitioned into groups using a clustering method based on PSO algorithm.
- The signal is segmented according to the resulting division into the corresponding groups, where each group represents certain type of segments, thus a certain cluster. The clustering method is applied on real data extracted from electrooculographic signals.

5. Combination of ants and cellular automata

- Application to clustering problem in data mining.
- Ant Sleeping Model combines advantages of the classical cellular automata model and the swarm intelligence.
- The ants have two states: sleeping state and active state.
- The ant's state is controlled by a function of the ant's fitness to the environment it locates and a probability for the ants becoming active.
- The state of an ant is determined only by its local information. By moving dynamically, the ants form different subgroups adaptively.
- The algorithm was applied to clustering problem in data mining. Results show that the algorithm is significantly better than other clustering methods in terms of both speed and quality. It is adaptive, robust and efficient, achieving high autonomy, simplicity and efficiency.

Q.27 List advantages and disadvantages of ant colony optimization.

Ans. : Advantages :

1. Inherent parallelism
2. Positive feedback accounts for rapid discovery of good solutions
3. Efficient for Traveling Salesman Problem and similar problems
4. It can be used in dynamic applications.

Disadvantages :

1. Theoretical analysis is difficult
2. Sequences of random decisions
3. Probability distribution changes by iteration
4. Research is experimental rather than theoretical
5. Time to convergence uncertain.

Q.28 What is bee colony optimization ?

Ans. : • The basic idea behind BCO is to build the multi agent system that will search for good solutions of various combinatorial optimization problems, exploring the principles used by honey bees during nectar collection process.

- Fig. Q.28.1 shows bee colony optimization techniques.

1. A bee initiates a search from a starting point.
2. The starting point is stored in a taboo list.
3. A search area defined by a set of searching points is determined from this point.
4. Each point is assigned to a bee (recruitment).

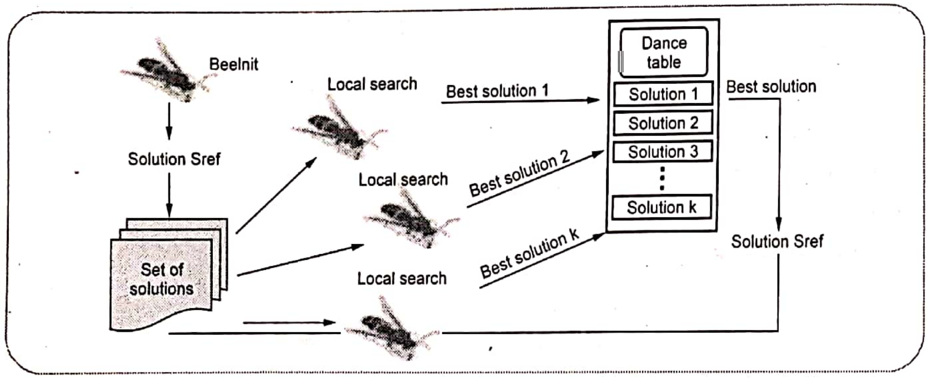


Fig. Q.28.1 Bee colony optimization techniques