NEHRU COLLEGE OF ENGINEERING AND RESEARCH CENTRE
(NAAC Accredited)
(Approved by AICTE, Affiliated to APJ Abdul Kalam Technological University, Kerala)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

COURSE MATERIALS

CS 361 SOFT COMPUTING

VISION OF THE INSTITUTION

To mould true citizens who are millennium leaders and catalysts of change through excellence in education.

MISSION OF THE INSTITUTION

NCERC is committed to transform itself into a center of excellence in Learning and Research in Engineering and Frontier Technology and to impart quality education to mould technically competent citizens with moral integrity, social commitment and ethical values.

We intend to facilitate our students to assimilate the latest technological know-how and to imbibe discipline, culture and spiritually, and to mould them in to technological giants, dedicated research scientists and intellectual leaders of the country who can spread the beams of light and happiness among the poor and the underprivileged.

ABOUT DEPARTMENT

- Established in: 2002
- Course offered : B.Tech in Computer Science and Engineering

  M.Tech in Computer Science and Engineering

  M.Tech in Cyber Security
- Approved by AICTE New Delhi and Accredited by NAAC
- Affiliated to the University of A P J Abdul Kalam Technological University.

DEPARTMENT VISION

Producing Highly Competent, Innovative and Ethical Computer Science and Engineering Professionals to facilitate continuous technological advancement.

DEPARTMENT MISSION

1. To Impart Quality Education by creative Teaching Learning Process
2. To Promote cutting-edge Research and Development Process to solve real world problems with emerging technologies.
3. To Inculcate Entrepreneurship Skills among Students.
4. To cultivate Moral and Ethical Values in their Profession.

PROGRAMME EDUCATIONAL OBJECTIVES

PEO1: Graduates will be able to Work and Contribute in the domains of Computer Science and Engineering through lifelong learning.

PEO2: Graduates will be able to Analyse, design and development of novel Software Packages, Web Services, System Tools and Components as per needs and specifications.

PEO3: Graduates will be able to demonstrate their ability to adapt to a rapidly changing environment by learning and applying new technologies.

PEO4: Graduates will be able to adopt ethical attitudes, exhibit effective communication skills, Teamworkand leadership qualities.

PROGRAM OUTCOMES (POS)

Engineering Graduates will be able to:

1. **Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering

problems.

2. **Problem analysis**: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/development of solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems**: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society**: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and sustainability**: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and team work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. **Life-long learning**: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAM SPECIFIC OUTCOMES (PSO)

PSO1: Ability to Formulate and Simulate Innovative Ideas to provide software solutions for Real-time Problems and to investigate for its future scope.

PSO2: Ability to learn and apply various methodologies for facilitating development of high quality System Software Tools and Efficient Web Design Models with a focus on performance optimization.

PSO3: Ability to inculcate the Knowledge for developing Codes and integrating hardware/software products in the domains of Big Data Analytics, Web Applications and Mobile Apps to create innovative career path and for the socially relevant issues.

COURSE OUTCOMES

SUBJECT CODE: C306	
COURSE OUTCOMES	
C361.1	To acquire knowledge in fundamentals of artificial neural networks
C361.2	To analyze various neural network architectures
C361.3	To acquire knowledge in the usage of various operations on fuzzy systems
C361.4	To learn the implementation of Fuzzy membership functions
C361.5	To identify fuzzy rules and to illustrate the methods of fuzzy interference systems
C361.6	To learn the genetic algorithm concepts and their applications

MAPPING OF COURSE OUTCOMES WITH PROGRAM OUTCOMES

CO’S	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
C361.1	3	2		2							2	3
C361.2	3	2		2	2				2			3
C361.3	3	3		2	3				2		2	3
C361.4	2	2	2	3	2				2		2	3
C361.5	3	2		2	2	2			2			3
C361.6	2	3	2	2	2	2			2		2	3
C361	2.67	2.33	2	2.16	2.2	2			2		2	3

Note: H-Highly correlated=3, M-Medium correlated=2, L-Less correlated=1

PSO MAPPINGS

CO’S	PSO1	PSO2	PSO3
C361.1	3	2	
C361.2	3	2	
C361.3	3	2	2
C361.4	3	3	3
C361.5	3	3	3
C361.6	3	3	3
C361	3	2.5	2.75

SYLLABUS

Course code	Course Name	L-T-P Credits	Year of Introduction
CS361	SOFT COMPUTING	3-0-0-3	2016

Prerequisite: Nil

Course Objectives

- To introduce the concepts in Soft Computing such as Artificial Neural Networks, Fuzzy logic-based systems, genetic algorithm-based systems and their hybrids.

Syllabus

Introduction to Soft Computing, Artificial Neural Networks, Fuzzy Logic and Fuzzy systems, Genetic Algorithms, hybrid systems.

Expected Outcome

The Students will be able to

1. Learn soft computing techniques and their applications.
2. Analyze various neural network architectures.
3. Define the fuzzy systems.
4. Understand the genetic algorithm concepts and their applications.
5. Identify and select a suitable Soft Computing technology to solve the problem; construct a solution and implement a Soft Computing solution.

Text Books

1. S. N. Sivanandam and S. N.Deepa, Principles of soft computing – John Wiley & Sons, 2007.
2. Timothy J. Ross, Fuzzy Logic with engineering applications , John Wiley & Sons, 2016.

References

1. N. K. Sinha and M. M. Gupta, Soft Computing & Intelligent Systems: Theory & Applications-Academic Press /Elsevier. 2009.
2. Simon Haykin, Neural Network- A Comprehensive Foundation- Prentice Hall International, Inc.1998
3. R. Eberhart and Y. Shi, Computational Intelligence: Concepts to Implementation, Morgan Kaufman/Elsevier, 2007.
4. Driankov D., Hellendoorn H. and Reinfrank M., An Introduction to Fuzzy Control- Narosa Pub., 2001.
5. Bart Kosko, Neural Network and Fuzzy Systems- Prentice Hall, Inc., Englewood Cliffs, 1992
6. Goldberg D.E., Genetic Algorithms in Search, Optimization, and Machine Learning- Addison Wesley, 1989.

Course Plan

Module	Contents	Hours	End Sem. Exam Marks
I	Introduction to Soft Computing Artificial neural networks - biological neurons, Basic models of artificial neural networks – Connections, Learning, Activation Functions, McCulloch and Pitts Neuron, Hebb network.	07	15%
II	Perceptron networks – Learning rule – Training and testing algorithm, Adaptive Linear Neuron, Back propagation Network – Architecture, Training algorithm	07	15%
	FIRST INTERNAL EXAM		

III	Fuzzy logic - fuzzy sets - properties - operations on fuzzy sets, fuzzy relations - operations on fuzzy relations	07	15%
IV	Fuzzy membership functions, fuzzification, Methods of membership value assignments – intuition – inference – rank ordering, Lambda – cuts for fuzzy sets, Defuzzification methods	07	15%
SECOND INTERNAL EXAM			
V	Truth values and Tables in Fuzzy Logic, Fuzzy propositions, Formation of fuzzy rules - Decomposition of rules – Aggregation of rules, Fuzzy Inference Systems - Mamdani and Sugeno types, Neuro-fuzzy hybrid systems – characteristics - classification	07	20%
VI	Introduction to genetic algorithm, operators in genetic algorithm - coding - selection - cross over – mutation, Stopping condition for genetic algorithm flow, Genetic-neuro hybrid systems, Genetic-Fuzzy rule based system	07	20%
END SEMESTER EXAMINATION			

Question Paper Pattern

1. There will be *five* parts in the question paper – A, B, C, D, E
2. Part A
   a. Total marks : 12
   b. *Four*questions each having *3* marks, uniformly covering modules I and II; All*four* questions have to be answered.
3. Part B
   a. Total marks : 18
   b. *Three*questions each having *9* marks, uniformly covering modules I and II; Two questions have to be answered. Each question can have a maximum of three sub-parts
4. Part C
   a. Total marks : 12
   b. *Four* questions each having *3* marks, uniformly covering modules III and IV; All*four* questions have to be answered.
5. Part D
   a. Total marks : 18
   b. *Three*questions each having *9* marks, uniformly covering modules III and IV; *Two* questions have to be answered. Each question can have a maximum of three subparts
6. Part E
   a. Total Marks: 40
   b. *Six* questions each carrying 10 marks, uniformly covering modules V and VI; *four* questions have to be answered.
   c. A question can have a maximum of three sub-parts.
7. There should be at least 60% analytical/numerical/design questions.

QUESTION BANK

MODULE I

SL NO.	QUESTIONS	COS	KL	PAGE NO.
1.	Compare and contrast biological and artificial neuron	CO1	K4	17
2.	Explain the training algorithm for Hebb network	CO1	K3	32
3	Define artificial neural network and draw its mathematical model	CO1	K2	17
4.	Why Mc-collulloch network is widely used in logic functions	CO1	K3	31
5.	Implement AND function using Hebb network using bipolar inputs and targets	CO1	K6	32
6.	Implement OR function using Perceptron training algorithm using bipolar inputs and targets	CO1	K6	38
7.	Obtain the output of the neuron for a network with inputs are given as [x1, x2] = [0.7, 0.8] and the weights are [w1, w2] = [0.2, 0.3] with bias = 0.9. Use i) Binary sigmoidal activation function ii) Bipolar sigmoid activation function	CO1	K4	36
8.	Discuss the concept of MP neuron network	CO1	K2	31
9.	Differentiate between hard computing and soft computing	CO1	K4	13
10.	List any three activation function with their equation and graph	CO1	K1	27
11.	Implement NOR using MP neuron using binary inputs and targets	CO1	K6	37
12.	Implement AND function using MP neuron with binary inputs	CO1	K6	36
	MODULE II			
1.	Write the training algorithm of backpropagation network	CO2	K2	53
2.	Describe the concept Adaline	CO2	K5	46
3.	Write the testing algorithm of backpropagation	CO2	K2	57
4.	With the help of an example explain supervised, unsupervised and reinforcement learning	CO2	K3	24

5.	What is the role of Widrow-Hoff rule in Adaptive Linear neuron? Give appropriate equations.	CO2	K1	46
6.	Write the learning factors of backpropagation network	CO2	K2	50
7.	Draw the flowchart of perceptron learning rule training process	CO2	K3	39
8.	What is adaline	CO2	K1	46
9.	Write perceptron network testing algorithm	CO2	K1	43
10.	Write the training algorithm for backpropagation network	CO2	K1	50
11.	Using linear separability, draw the decision boundary for logical AND? Design and implement OR function with bipolar inputs and targets using Adaline network? Find total mean square error of 3 epochs?	CO2	K6	46
12.	Explain training algorithm used in adaptive linear neuron	CO2	K3	46
13.	Explain training algorithm used in perceptron network in single input class	CO2	K5	39
14.	Explain the testing algorithm of perceptron network	CO2	K3	38
	MODULE III			
1.	Define fuzzy set and write basic fuzzy set operations	CO3	K1	91
2.	For the given fuzzy set perform all fuzzy operations	CO3	K2	71
3.	Discuss fuzzy relation and list out its properties	CO3	K2	67
4.	For the given fuzzy set compute algebraic sum, algebraic product, bounded sum, bounded difference	CO3	K3	63
5.	Discuss fuzzy relation and list out its properties	CO3	K2	68
6.	List the stages involved in backpropagation algorithm	CO3	K1	50
7.	Discuss the properties of fuzzy set	CO3	K3	63
8.	For the given fuzzy membership function compute Cartesian product and compositions	CO3	K3	74
	Explain any two methods of composition technique on fuzzy sets with example.	CO3	K3	76
	Represent the standard fuzzy set operation using VENN diagram	CO3	K3	81
	Define fuzzy set and write basic fuzzy set operations	CO3	K1	61
	MODULE IV			
1.	Explain any two defuzzification method	CO4	K3	107
2.	Using your own intuition plot the fuzzy membership function	CO4	K2	101

3.	Using Zadehs notation express the fuzzy sets into lamda cut for the given fuzzy set	CO4	K2	104
4.	Using the inference approach find the membership values for the triangular shapes I,R,E,IR and T for the triangle with 45,55,80 degree	CO4	K4	103
5.	Explain the features of membership function	CO4	K3	96
6.	Give the canonical form of fuzzy rule based system. Give the syntax for the formation of fuzzy rule using i) Assignment statements ii) Conditional statements iii) Unconditional statements	CO4	K2	121
7.	State the relevance of fuzzification. Explain its types	CO4	K5	93
8..	Explain the various types of fuzzy types	CO4	K5	94
	MODULE V			
1.	Describe two methods used for the aggregation of fuzzy rules.	CO5	K2	123
2.	Explain the different classification of neuro hybrid system	CO5	K5	132
3.	Describe two methods used for the decomposition of fuzzy rules	CO5	K2	120
4.	Describe different types of FIS	CO5	K2	126
5.	Write the different steps of Mamdani FIS	CO5	K1	127
6.	Explain in detail about the FIS system with its block diagram	CO5	K5	125
7.	Explain about the Neuro fuzzy hybrid system and its characteristics	CO5	K5	131
	MODULE VI			
1.	List out the stopping condition for GA	C06	K1	165
2.	Explain different crossover method with example	C06	K3	155
3.	Explain the classification of NFS system	C06	K5	142
4.	Explain the steps of Genetic algorithm	C06	K5	148
5.	Explain about selection and mutation operator of GA	C06	K5	163
6.	Explain Genetic fuzzy rule based system	C06	K5	172
7.	Define the term Individual, Genes and Fitness function	C06	K1	170
	Explain in detail about the Genetic neuro hybrid system	C06	K3	167

APPENDIX 1

CONTENT BEYOND THE SYLLABUS

MODULE NOTES

Introduction to Soft Computing

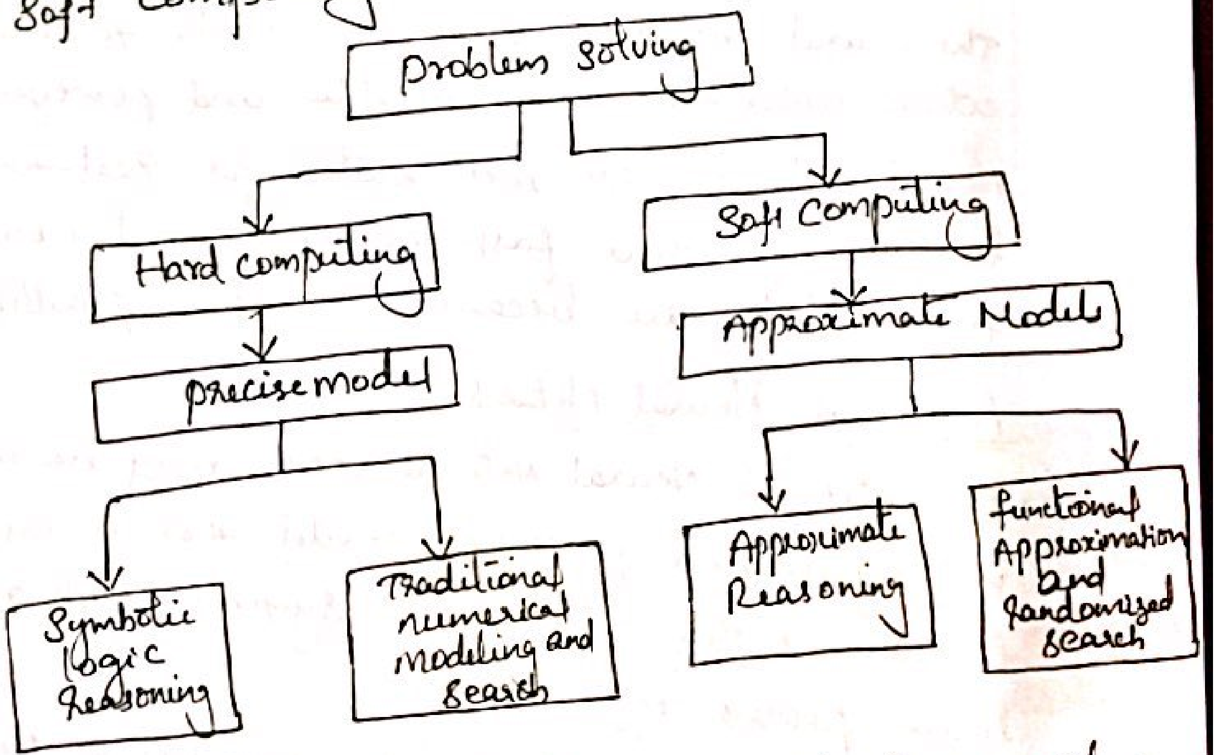
Soft computing was introduced with the objective of exploiting the tolerance for imprecision, uncertainty and partial truth to achieve tractability, robustness low solution cost and better rapport with reality.

The ultimate goal is to be able to emulate the human mind as closely as possible.

Soft computing performs a task of learning from environmental data and transforms them into analytical model.

The two major problem solving technologies include.

→ hard computing
→ Soft computing



Hard computing is based on mathematical approaches to get the accurate solution.

Hard Computing deals with precise model with precise data where accurate solutions are achieved quickly.
Soft computing is based on biological model and it deals with approximate data and gives solution.

Soft computing Consists of 3 Technologies

* Neural Networks
* Fuzzy Logic
* Genetic Algorithms

Neural Networks

A neural network is a processing device, either an algorithm or an actual hardware whose design was inspired by the design and functioning of animal brains and components.

The neural networks have the ability to learn by example which makes them very flexible and powerful.

Neural networks are well suited for real-time systems because of their fast response and computational times which are because of their parallel architecture.

Artificial Neural Network

An artificial Neural Network (ANN) may be defined as an information processing model that is inspired by the way biological nervous systems, such as the brain process information.

An ANN is composed of large number of highly interconnected processing elements working together to solve specific problems.

In ANN, a large number of highly interconnected processing elements are called nodes or neurons and Each neuron is connected with the Other by a connection link. Each connection link is associated with weights which contain information about the input signal. This information is used by the neuron net to solve a particular problem.

Each neuron has an internal state of its own is called the activation signal which is the function of the inputs the neuron receives. The activation signal of the neuron is transmitted to other neurons. Each neuron can send only one signal at a time which can be transmitted to several other neurons.

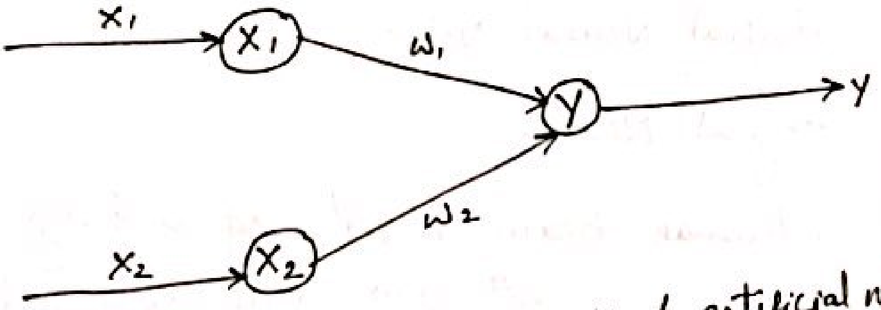


fig: Architecture of a simple artificial neuron net

Consider a set of neurons X_1 and X_2 transmitting signal to another neuron Y. Here X_1 and X_2 are input neurons, which transmit signals and Y is the output neurons which receives signals. Input neurons X_1 and X_2 are connected to the output neuron Y over a weighted interconnection link w_1 and w_2

For the above neuron net architecture, the net input has to be calculated in the following ways:

$$y_{in} = x_1 \omega_1 + x_2 \omega_2$$

where x_1 and x_2 are the activation of the input neurons X_1 and X_2 i.e the output of the ~~Ato~~ input Signals.

The output y of the output neuron Y can be obtained by ~~activ~~ applying activation over the net input i.e the function of the net input.

$$y = f(y_{in})$$

output = function (net calculated input)

The function to be applied over the net input is called activation function.

Biological Neural Network

Biological Neuron:

The human brain consists of a huge number of neurons approximately 10^{11} with numerous interconnection

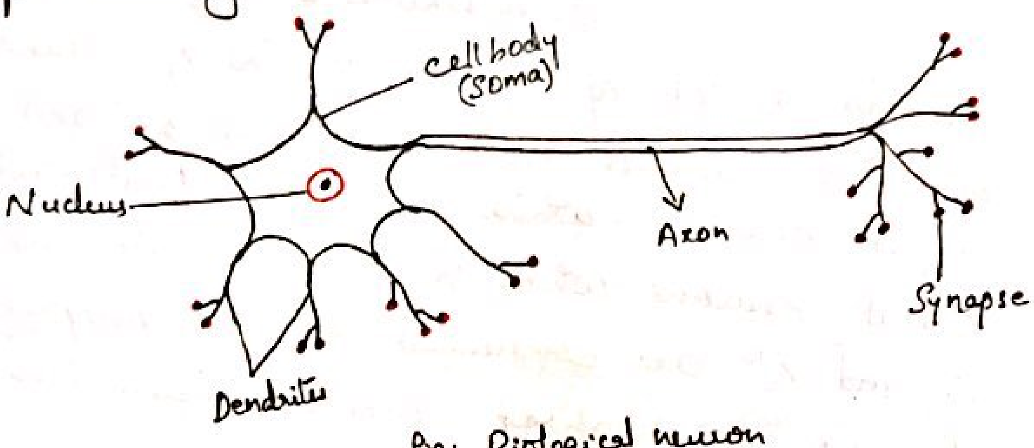


fig: Biological neuron

The biological neuron consists of three main parts :-

1. Soma or cell body - where the cell nucleus is located
2. Dendrites - where the nerve is connected to the cell body
3. Axon - which carries the impulses of the neuron

Dendrites are tree-like networks made of nerve fiber connected to the cell body.

An axon is a single, long connection extending from the cell body and carrying signals from the neuron. The end of the axon splits into fine strands and each strand terminates into small bulb like organ called synapse. It is through synapse that the neuron introduces its signal to other nearby neurons

Relationship between biological and artificial neurons

BNN	ANN.
SOMA	NODE
Dendrites	input
Synapse	weight/ interconnection
Axon	output.

BASIC model of Artificial Neural Network

The models of ANN are specified by the three basic entities namely.

* Connection :- The models synaptic interconnections
* Learning :- The training or learning rule adopted for updating and adjusting the connection weight
* Activation Function

Connections

The neurons should be visualized for their arrangements in layers. An ANN consists of a set of highly interconnected processing elements called neurons such that each processing element output is found to be connected through weights to the other processing elements or to itself. The arrangement of neurons to form layers and the connection pattern formed within and between layer is called the network architecture. There exist five basic types of neuron connection architectures. They are

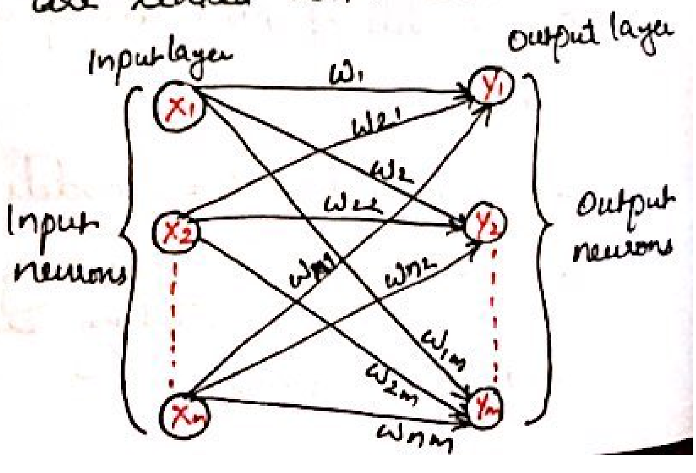
1. Single layer feed-forward network
2. Multilayer feed-forward network
3. Single node with its own feedback
4. Single-layer Recurrent network.
5. Multilayer Recurrent network.

Neural nets are classified into single-layer or multilayer neural nets. A layer is formed by taking a processing element and combining it with other processing elements.

Single layer feed forward Network

Layer is formed by taking processing elements and combining it with other processing elements. Input and output are linked with each other.

Input are connected to the processing node with various weights, resulting in series of output one per node.



Multilayer feed - forward Network

This network is formed by the interconnection of several layers.

The input layer receives the input and this layer has no function except buffering the input signal.

The output layer generates the output of the network.

The layer that is formed between the input and output layers is called hidden layer.

The hidden layer is internal to the network and has no direct contact with the external environment. There may be zero to several hidden layers in an ANN. If the complexity of the network is high then the number of hidden layers is more. and it will provide an efficient output response. Every output from one layer is connected to each and every node in the next layer

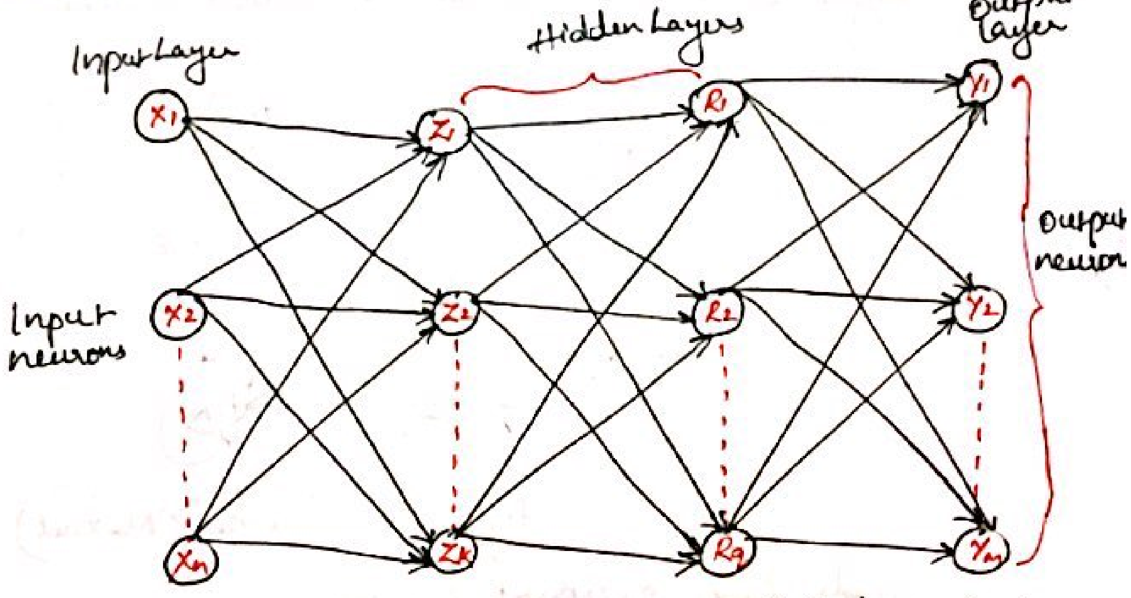
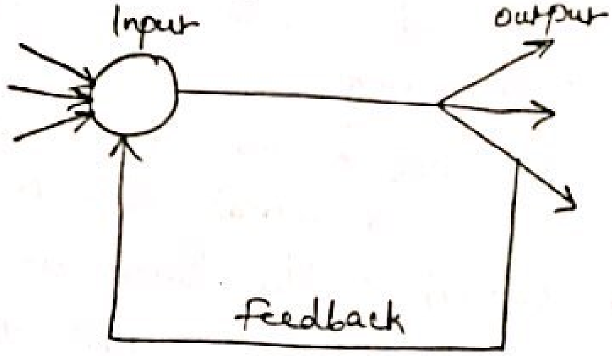


fig: Multilayer feed forward network

Feedback Network

A network is said to be a feed-forward network if no neuron in the output layer is an input to a node in the same layer or in the preceding layer

when outputs can be directed back as inputs to same or preceding layer nodes then it results in the formation of feedback networks.
If the feedback of the output of the processing element is directed back as input to the processing elements in the same layer then it is called lateral feedback



The above network is a single node with own feedback.
The following competitive interconnections having fixed weight of $-\varepsilon$. This net is called <u>Maxnet</u>

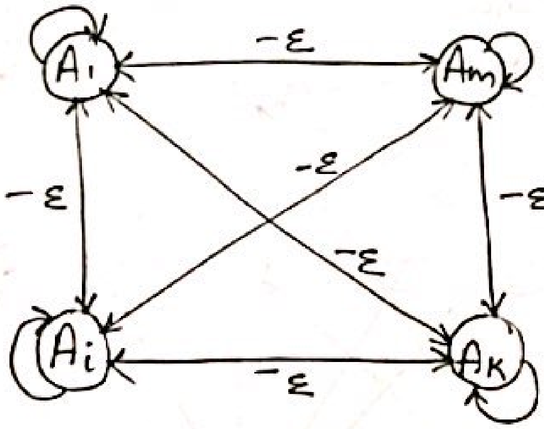


fig: Competitive net (Maxnet)

Single Layer Recurrent Networks

Recurrent networks are feedback networks with closed loop. The following figure shows a single layer network with a feedback connection in which a processing element's output can be directed back to the processing element itself or to the other processing element or to both. (feedback)

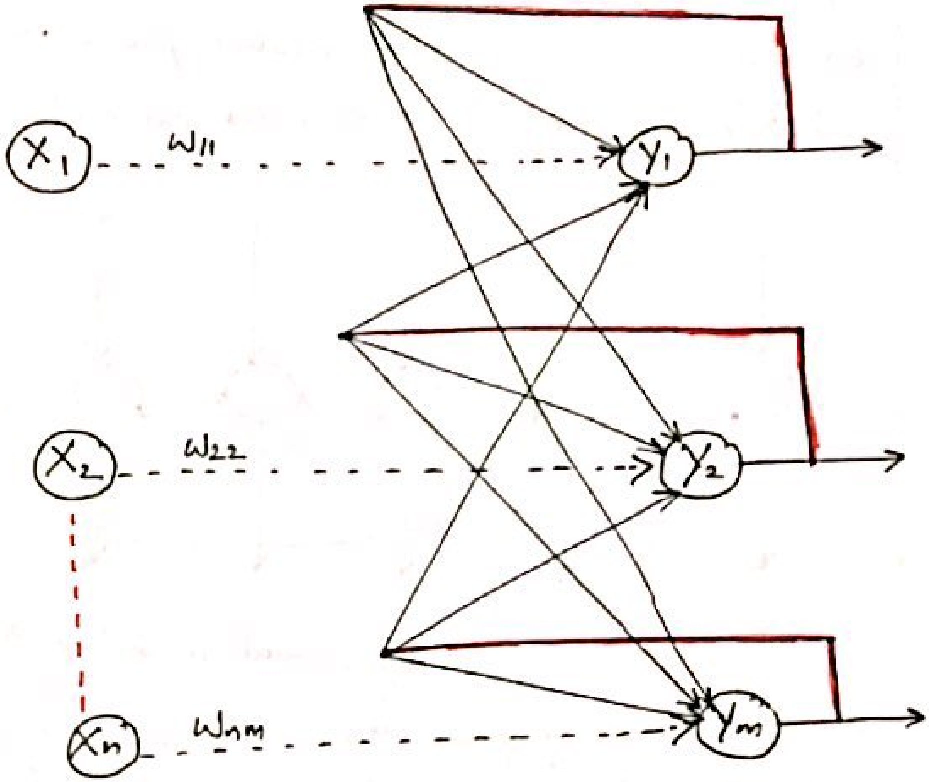
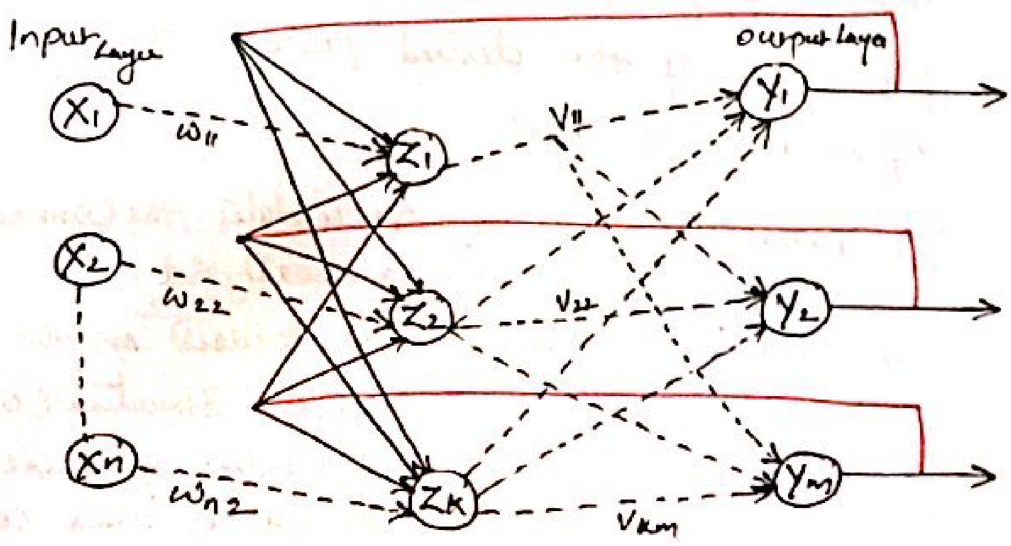


fig: Single Layer Recurrent network

Multilayer Recurrent Network

A processing element output can be directed back to the nodes in a preceding layer, forming a multilayer Recurrent network. In these networks, a processing element output can be directed back to the processing element itself and to other processing elements in the same layer



Lateral Inhibition Structure

In this structure each processing neuron receives two different classes of inputs -
excitatory - inputs from nearby processing elements
inhibitory - inputs from more distantly located processing elements

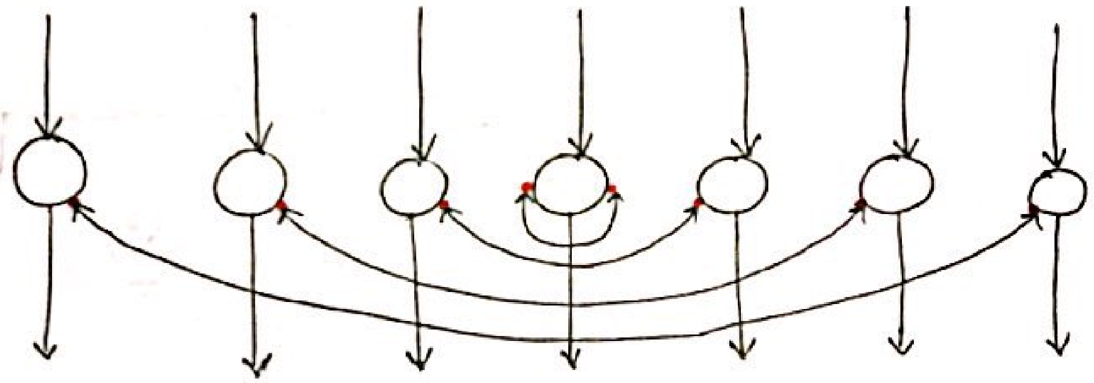


fig: Lateral Inhibition Structure

In the above figure the connection with open circles are excitatory connections and the links with solid connection circles are inhibitory connections.

Learning

The main property of ANN is its capability to learn. Learning or training is a process by means of which a neural network adapts itself to a stimulus by making proper parameter adjustments resulting in the production of the desired process. There are two kinds of learning in ANN.

1. parameter learning :- It updates the connecting weight in a neural net.
2. Structure learning :- It focuses on the change in network structure (which includes the number of processing elements as well as their connection type).

The above two types of learning can be performed Simultaneously or separately. Apart from these two Categories of learning, there are 3 categories of learning in ANN. They are

* Supervised Learning
* Unsupervised Learning
* Reinforcement Learning

Supervised Learning

In supervised learning method learning is performed with the help of a teacher or supervisor. All real time events involve supervised learning methodology. In ANNs supervised learning, each input vector requires a corresponding target vector, which represents the desired output. The input vector along with the target vector is called training pair.

The network is precisely informed about what should be emitted as output.

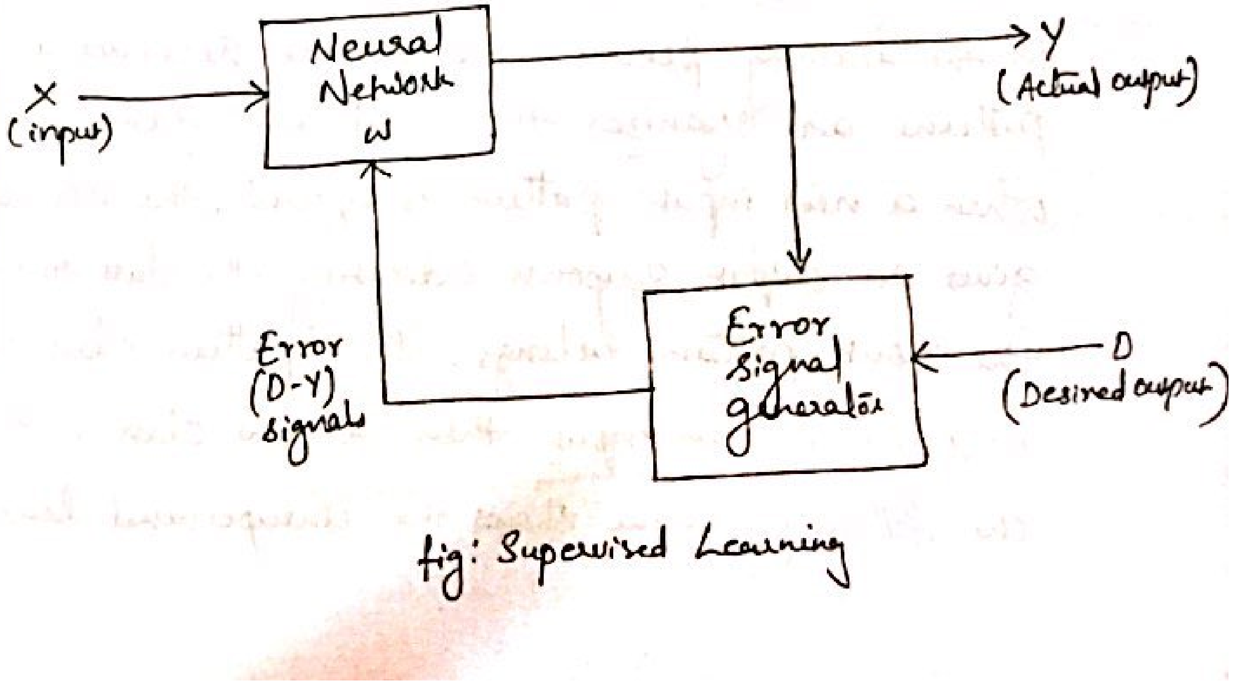


fig: Supervised Learning

During training phase, the input vector is presented to the network, which results in an output vector. This output vector is the actual output vector. Then this actual output vector is compared with the desired (target) output vector. If there exists a difference between the two output vectors then an error signal is generated by the network. This error signal is used for adjustment of weights until the actual output matches the desired (target) output. In supervised Learning it is assumed that the correct target output values are known for each input pattern.

Unsupervised Learning

In unsupervised Learning method, learning is performed without the help of a teacher or supervisor. In ANN, unsupervised learning, the input vectors of similar type are grouped without the use of training data to specify how a member of each group looks or to which group a number belongs.

In the training process, the network receives the input patterns and organizes these patterns to form clusters. When a new input pattern is applied, the neural network gives an output response indicating the class to which the input pattern belongs. If a pattern class cannot be found for an input then a new class is generated. The following figure shows the unsupervised learning

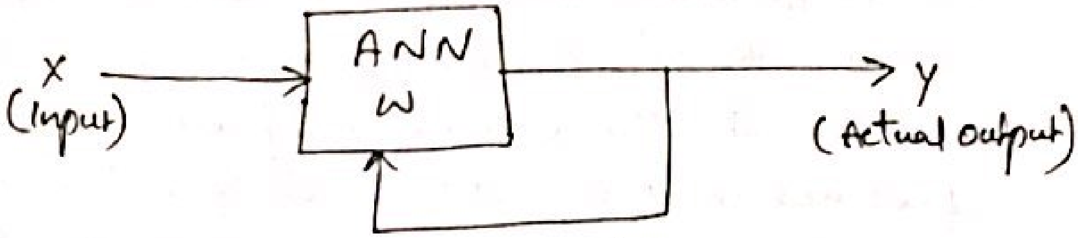


fig: Unsupervised Learning

In the above figure it is clear that no feedback from the environment is used to inform what the outputs should be or whether the outputs are correct. In this case the network must itself discover patterns, regularities features or categories from the input data and relations for the input data over the output. While discovering all these features, the network undergoes change in its parameters. This process is called self-organizing in which exact clusters will be formed by discovering similarities and dissimilarities among the objects.

Reinforcement Learning

This learning process is similar to Supervised learning. Less information might be available. The learning based on this critic information is called Reinforcement learning and the feedback sent is called Reinforcement Signal.

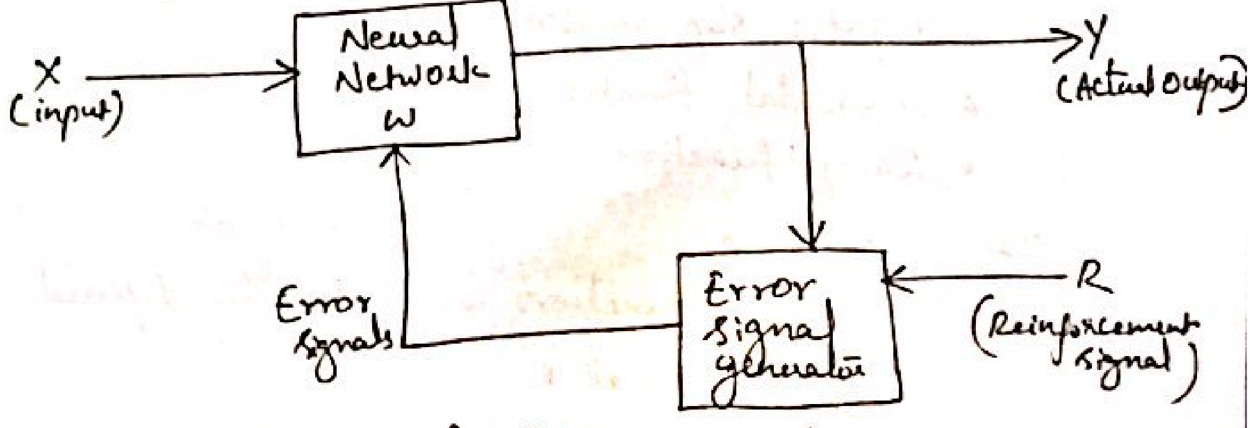


fig: Reinforcement Learning

The Reinforcement learning is a form of Supervised learning because the network receives some feedback from its environment. The external reinforcement signals are processed in the critic signal generator and the obtained critical signals are sent to the ANN for adjustment of weights so as to get better feedback in future.

Activation Functions

The activation functions helps in achieving the exact output. It is applied over the net input to calculate the output of an ANN.

The information processing of a processing element can be viewed as consisting of two major parts input and output. An integration function is associated with the input of a processing element. This function serves to combine activation, information or evidence from an external source or other processing elements into a net input to the processing element.

There are several activation function

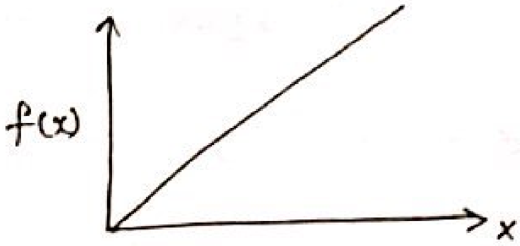
* Identity function
* Binary step function
* Bipolar step function
* Sigmoidal function
* Ramp function

Identity Function

It is a linear function and can be defined as

$$f(x) = x \quad \text{for all } x$$

The output here remains the same as input. The input layer uses the identity activation function.

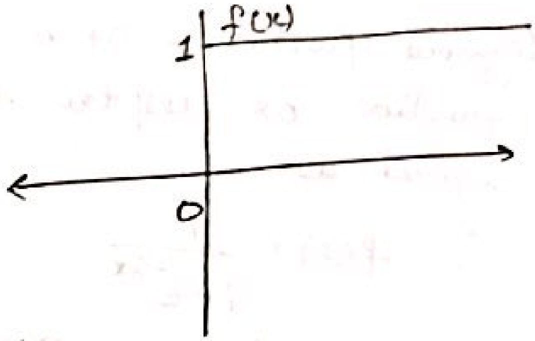


Binary step function

This function can be defined as

$$f(x) = \begin{cases} 1 & \text{if } x \geq \theta \\ 0 & \text{if } x < \theta \end{cases}$$

where θ represents the threshold value. This function is most widely used in single layer nets to convert the net input to an output that is a binary (1 or 0)



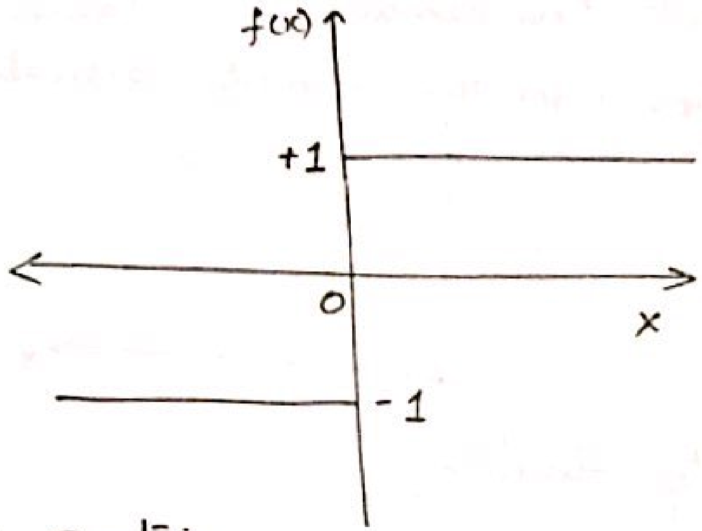
Bipolar step function

This function can be defined as

$$f(x) = \begin{cases} 1 & \text{if } x \geq \theta \\ -1 & \text{if } x < \theta \end{cases}$$

where θ represents the threshold value. This function is also used in single-layer nets to convert the net input to an output that is bipolar (+1 or −1).

Scanned with CamScanner



Sigmoidal Function

The sigmoidal functions are widely used in back-propagation nets because of the relationship between the value of the function at the point and the value of the derivative at that point which reduces the computational burden during training.

Sigmoidal functions are of two types.

→ Binary sigmoid function :- It is also termed as logistic sigmoid function or unipolar sigmoid function. It can be defined as

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

where λ is the steepness parameter. The range of the sigmoid function is from 0 to 1.

→ Bipolar sigmoid function :- This function is defined as

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

where λ is the steepness parameter and the sigmoid function range is between -1 and +1.

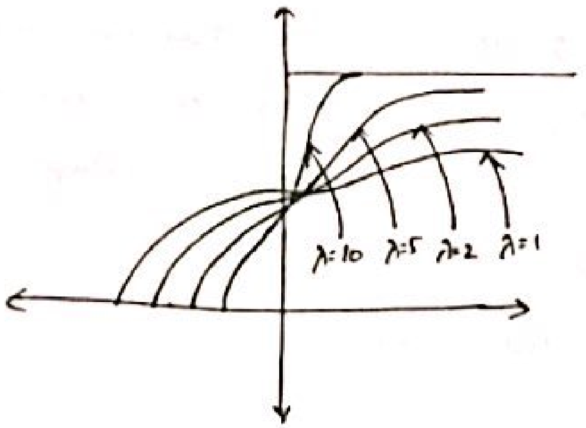


fig: Binary sigmoidal function

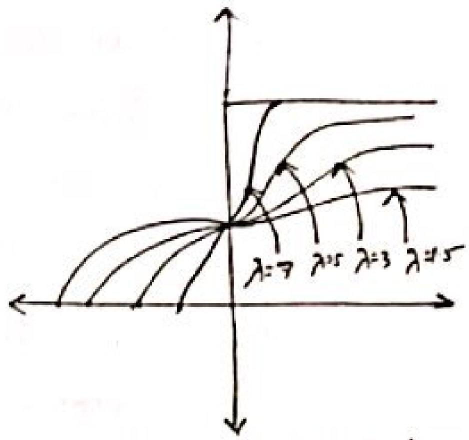


fig: Bipolar sigmoidal function

Ramp function

The ramp function is defined as

$$f(x) = \begin{cases} 1 & \text{if } x > 1 \\ x & \text{if } 0 \leq x \leq 1 \\ 0 & \text{if } x < 0 \end{cases}$$

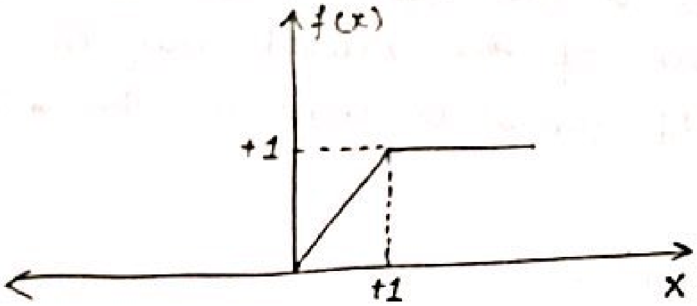


fig: Ramp function

Important Terminologies of ANNs

* Weights:-

In the architecture of ANN, each neuron is connected to other neurons by means of directed communication links and each communication link is associated with weights. The weights contain information about the input signal. This information is used by the net to solve a problem.

* Bias :-

The bias included in the network has its impact in Calculating the net input. The bias is considered like another weight. The bias plays a major role in determining the output of the network.

The bias can be of two types

→ positive bias

→ Negative bias

The positive bias helps in increasing the net input of the network and the Negative bias helps in decreasing the net input of the network.

As a result of the bias effect, the output of the network can be varied.

* Threshold :-

Threshold is a set value based upon which the final output of the network may be calculated. The threshold value is used in the activation function

* Learning Rate :-

The learning rate is denoted by 'α'. It is used to control the amount of weight adjustment at each step of training. The learning rate, ranging from 0 to 1. determines the rate of learning at each time step.

* Notations

w_{ij} : weight on connection from unit x_i to unit y_j

b_j : Bias acting on unit j.

θ_j : Threshold for activation of neuron

α : Learning rate.

McCulloch-pitts Neuron

McCulloch-pitts neuron was discovered in 1943. It is usually called as M-p neuron. The M-p neurons are connected by directed weighted paths. The activation of a M-p neuron is binary, that is at any time step the neuron may fire or may not fire.

The weights associated with the communication links may be excitatory (weight is positive) or inhibitory (weight is negative). The excitatory connected weights entering into a particular neuron will have same weights. The threshold plays a major role in M-p neuron. There is a fixed threshold for each neuron, and if the net input to the neuron is greater than the threshold then the neuron fires.

Architecture

The M-p neuron has both excitatory and inhibitory connections. It is excitatory with weight ($w>0$) or inhibitory weight $-p$ ($p<0$).

The firing of the output neuron is based upon the threshold, the activation function here is defined as

$$f(y_{in}) = \begin{Bmatrix} 1 & \text{if } y_{in} \geq \theta \\ 0 & \text{if } y_{in} < \theta \end{Bmatrix}$$

The threshold with the activation function should satisfy the following condition

$$\theta > nw - p.$$

The M-p neuron has no particular training algorithm. An analysis has to be performed to determine the values of the weights and the threshold. The M-P neurons are used as building blocks on which any function can be modeled.

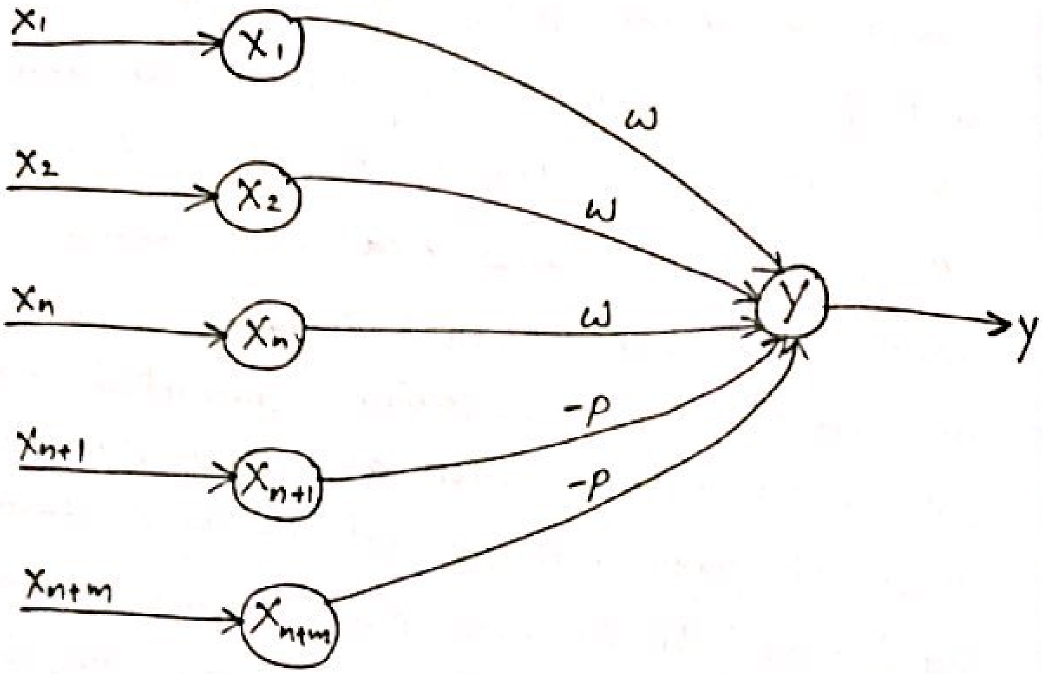


fig: McCulloch-pitts neuron Model.

Hebb Network

Hebb network was introduced in 1949 and the Hebb learning rule is a simple one.

According to Hebb rule, the weight vector is found to increase proportionately to the product of the input and the learning ~~rule~~ signal. Here the learning signal is equal to the neuron's output.

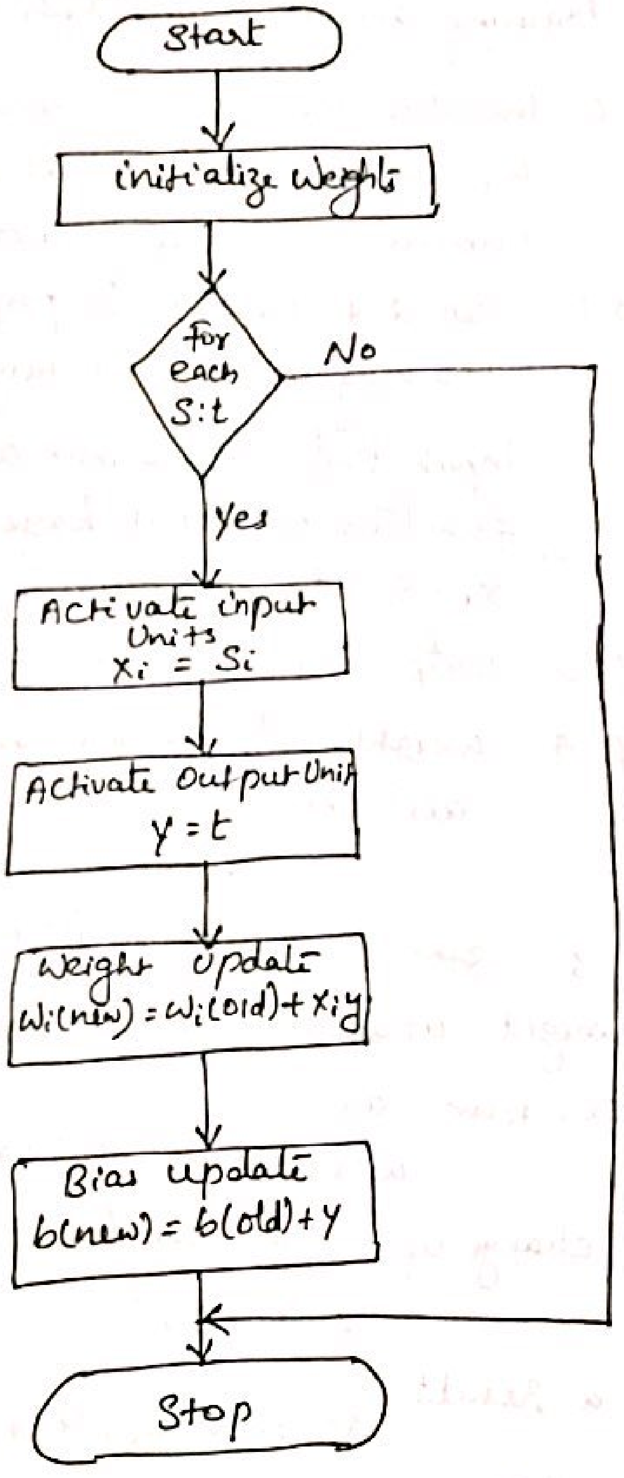
In Hebb learning, if two interconnected neurons are 'on' simultaneously then the weight associated with these neurons can be increased by the modification made in their synaptic gap. The weight update in Hebb rule is given by

$$\omega_i(new) = \omega_i(old) + x_i y$$

The Hebb rule is more suited for bipolar data than binary data.

Scanned with CamScanner

Flowchart of Training Algorithm



The training algorithm is used for the calculation and adjustment of weight. The flowchart for the training algorithm of Hebb network is given above.

Training Algorithm :-

The training algorithm of Hebb network is given below

Step 0: Initialize the weights and set to zero i.e. $w_i = 0$ for $i = 1$ to n where 'n' may be the total number of input neurons.

Step 1: Step 2-4 have to be performed for each input training vector and target output pair, $s:t$

Step 2: Input units activations are set. The activation function of input layer is identity function $x_i = s_i$ for $i = 1$ to n

Step 3: Output units activations are set: $y = t$

Step 4: Weight adjustments and bias adjustments are performed

$$w_i(new) = w_i(old) + x_i y$$

$$b(new) = b(old) + y$$

Step 5: Stop

The weight updation formula can also be given in vector form as

$$w(new) = w(old) + xy$$

The change in weight can be expressed as

$$\Delta w = xy$$

As a result

$$w(new) = w(old) + \Delta w.$$

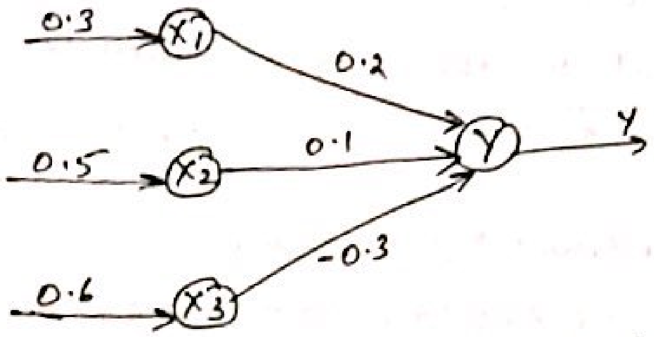
Note :-

The Hebb rule can be used for pattern association, pattern categorization, pattern classification and over a range of other areas.

Scanned with CamScanner

problems

1. For the network shown in the following figure. calculate the net input to the output neuron



The given neural net consists of three input neurons and one output neuron. The inputs and weights are

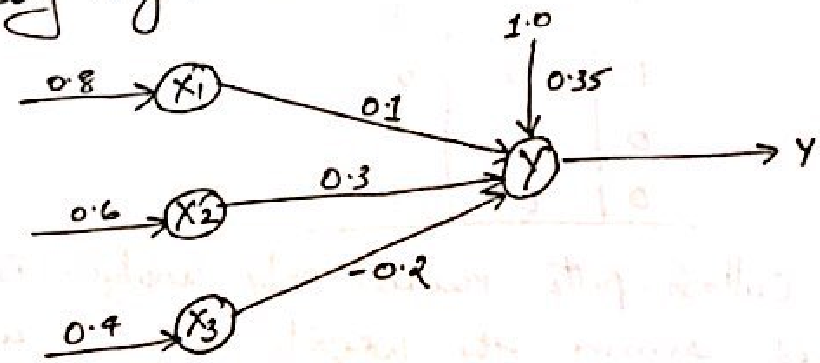
$[x_1, x_2, x_3] = [0.3, 0.5, 0.6]$

$[w_1, w_2, w_3] = [0.2, 0.1, -0.3]$

The net input can be calculated as

$$y_{in} = x_1 w_1 + x_2 w_2 + x_3 w_3$$
$$= 0.3 \times 0.2 + 0.5 \times 0.1 + 0.6 \times (-0.3)$$
$$= 0.06 + 0.05 - 0.18 = \underline{-0.07}.$$

2. Obtain the output of the neuron Y for the network shown in the following figure. using activation function. (i) binary sigmoidal and (ii) bipolar sigmoidal



The given network has three input neurons with bias and one output neuron. These form a single-layer network.

The inputs are given as $[x_1, x_2, x_3] = [0.8, 0.6, 0.4]$
The weights are given as $[w_1, w_2, w_3] = [0.1, 0.3, -0.2]$
Bias value $b = 0.35$
The net input to the output neuron is

$$y_{in} = b + \sum_{i=1}^{n} x_i w_i \qquad n = 3 \text{ (only 3 i/p neurons)}$$

$$= b + x_1 w_1 + x_2 w_2 + x_3 w_3$$

$$= 0.35 + 0.8 \times 0.1 + 0.6 \times 0.3 + 0.4 \times -0.2$$

$$= 0.35 + 0.08 + 0.18 - 0.08 = \underline{0.53}$$

(i) for binary sigmoidal activation function

$$y = f(y_{in}) = \frac{1}{1+e^{-y_{in}}} = \frac{1}{1+e^{-0.53}} = \underline{\underline{0.625}}$$

ii) for bipolar sigmoidal activation function

$$y = f(y_{in}) = \frac{2}{1+e^{-y_{in}}} - 1 = \frac{2}{1+e^{-0.53}} - 1 = \underline{\underline{0.259}}$$

3. Implement AND function using McCulloch-pitts neuron

Consider the truth table for AND function as.

x_1	x_2	y
1	1	1
1	0	0
0	1	0
0	0	0

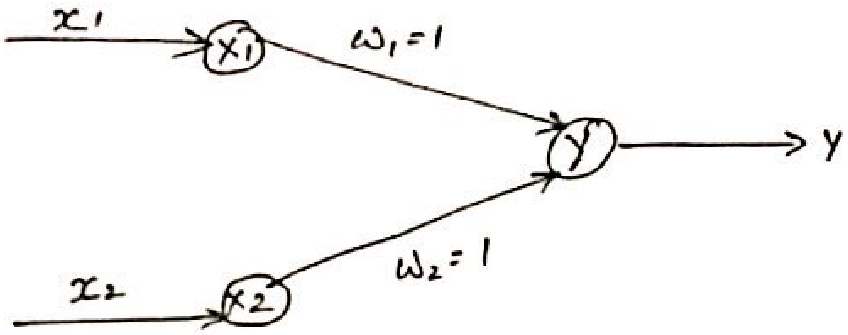
In McCulloch-pitts neuron only analysis is being performed. Assume the weights be $w_1 = 1$ and $w_2 = 1$. The net input is calculated for four inputs. For input

$(1,1)$, $y_{in} = x_1 w_1 + x_2 w_2 = 1 \times 1 + 1 \times 1 = 2$

$(1,0)$, $y_{in} = x_1 w_1 + x_2 w_2 = 1 \times 1 + 0 \times 1 = 1$

$(0,1)$, $y_{in} = x_1 w_1 + x_2 w_2 = 0 \times 1 + 1 \times 1 = 1$

$(0,0)$, $y_{in} = x_1 w_1 + x_2 w_2 = 0 \times 1 + 0 \times 1 = 0$



For an AND function, the output is high if both the inputs are high. For this condition, the net input is calculated as 2. Based on this input the threshold is set, If the threshold value is greater than or equal to 2 then the neuron fires else it does not fire. So the threshold value is set equal to 2 $(\theta = 2)$. This can be obtained by

$$\theta \geq nw - p$$

Here $n=2$, $w=1$ (excitatory weights) and $p=0$.

$$\theta \geq 2 \times 1 - 0 = \theta \geq 2$$

The output of neuron Y can be written as

$$y = f(y_{in}) = \begin{cases} 1 & \text{if } y_{in} \geq 2 \\ 0 & \text{if } y_{in} < 2 \end{cases}$$

4.

MOD-II

Perceptron Networks

Perceptron networks are single-layer feed-forward networks and are also called simple perceptron. Simple perceptron networks was discovered in 1962.

The key points to be noted in a perceptron network are

1. The perceptron network consists of three-units, namely Sensory unit (input unit), associator unit (hidden unit), Response unit (output unit)
2. The sensory units are connected to associator units with fixed weights having values 1, 0, or -1 which are assigned at Random
3. The binary activation function is used in sensory unit and associator unit.
4. The Response unit has an activation of 1, 0 or -1. The binary step with fixed threshold θ is used as activation for associator. The output signals that are sent from the associator unit to the response unit is binary.
5. The output of the perceptron network is given by

$$y = f(y_{in})$$

where $f(y_{in})$ is the activation function defined as

$$f(y_{in}) = \begin{cases} 1 & \text{if } y_{in} > \theta \\ 0 & \text{if } -\theta \le y_{in} \le \theta \\ -1 & \text{if } y_{in} < -\theta \end{cases}$$

6. The perceptron learning Rule is used in the weight updation between the associator unit and the response unit. For each training input the net will calculate the response and it determine whether or

not the error has occurred

7. The error calculation is based on the comparison of the values of targets with those of the calculated outputs

8. The weights on the connections from the units that send the nonzero signal will get adjusted

9. The weights will be adjusted on the basis of the learning rule if an error has occurred for a particular training pattern i.e.

$$w_i(new) = w_i(old) + \alpha t x_i$$
$$b(new) = b(old) + \alpha t$$

If no error occurs, no weight updation is done and the training process stopped.

In the above equation of weight and bias 'α' is the learning rate and 't' is the target value.

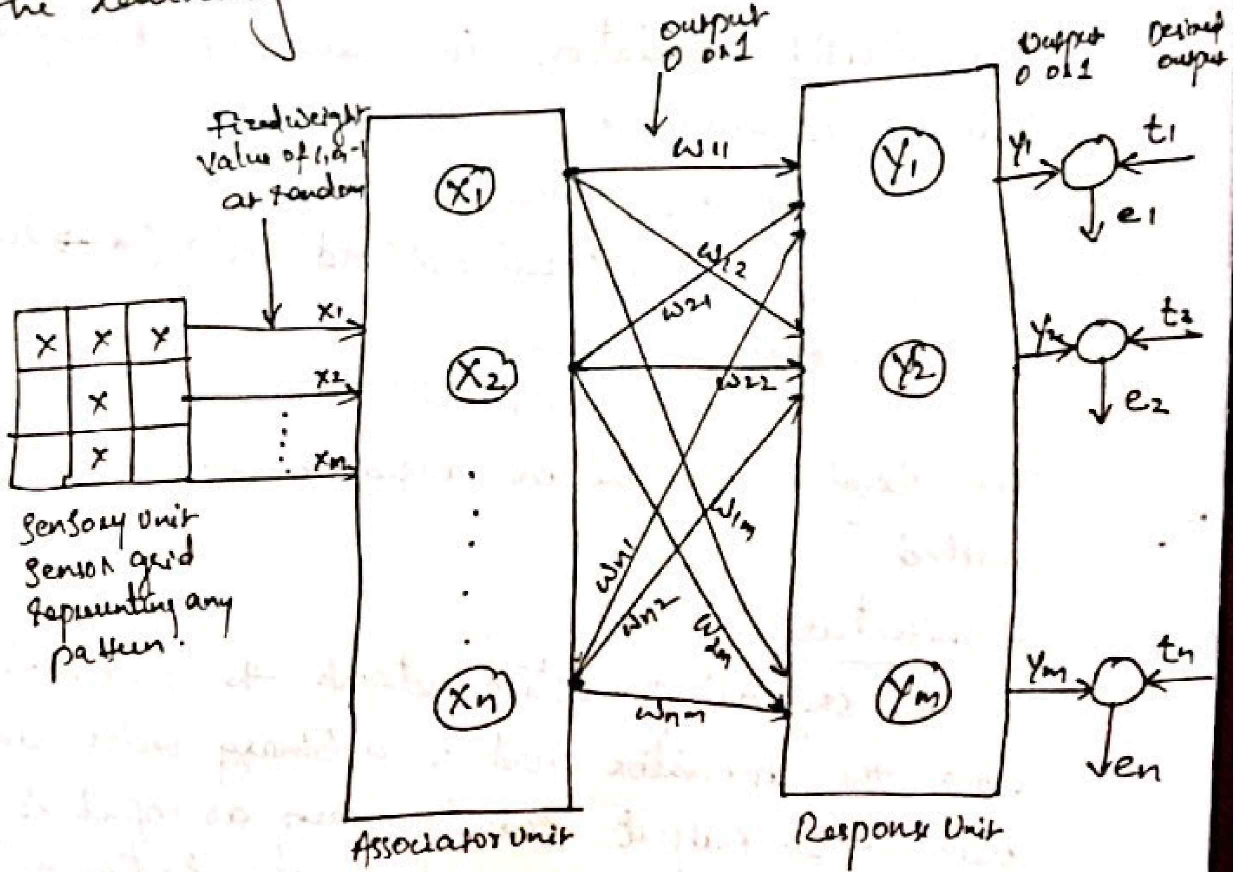


fig: perceptron Network

Perceptron Learning Rule.

In perceptron learning rule, the learning signal is the difference between the desired and actual response of a neuron. The perceptron learning rule is as follows.

Consider a finite 'n' number of input training vectors, with their associated target values $x(n)$ and $t(n)$. where 'n' ranges from 1 to N. The target is either +1 or -1. The output 'y' is obtained on the basis of the net input calculated and the activation function being applied over the net input.

$$y = f(y_{in}) = \begin{cases} 1 & \text{if } y_{in} > \theta \\ 0 & \text{if } -\theta \le y_{in} \le \theta \\ -1 & \text{if } y_{in} < -\theta \end{cases}$$

The weight updation in case of perceptron learning is given as.

If $y \neq t$ then

$$\omega(new) = \omega(old) + \alpha t x \quad (\alpha \rightarrow \text{learning rate})$$

else we have

$$\omega(new) = \omega(old).$$

The weights can be initialized at any values in this method.

Architecture

In the original perceptron network the output obtained from the associator unit is a binary vector and hence that output can be taken as input signal to the response unit and the classification can be performed

Only the weight between the associator and sensory units are adjusted and the weight between the sensory and associator units are fixed.

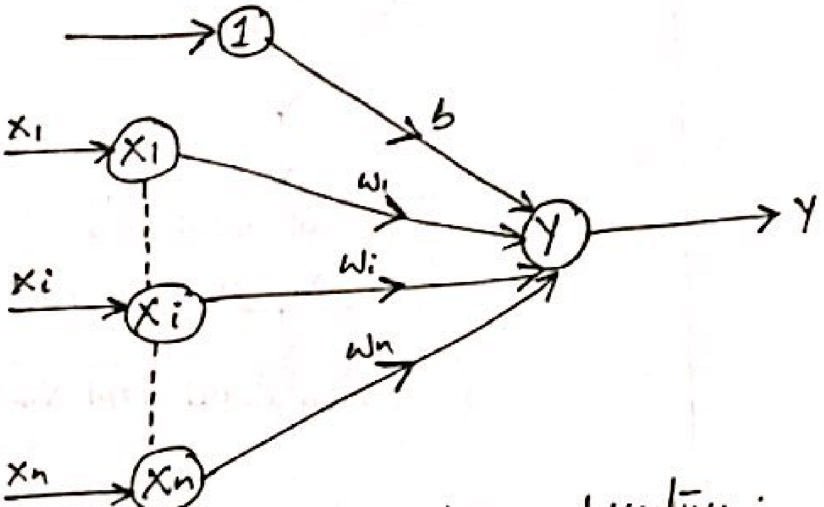
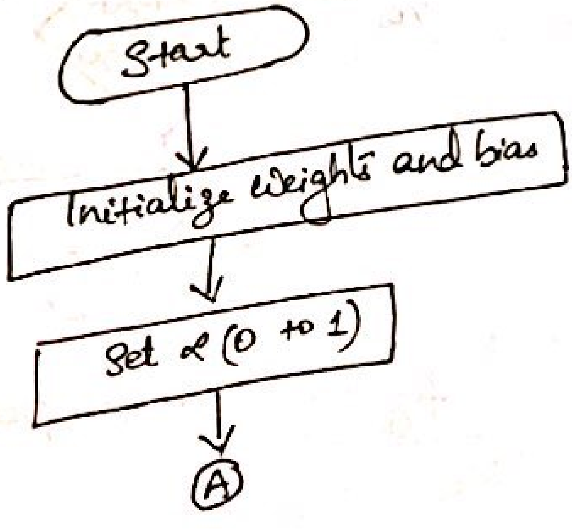


fig: Simple perceptron architecture.

In the above figure there are n input neurons, 1 output neuron and a bias. The input layer and the output layer neurons are connected through a directed communication link, which is associated with weights.

Flowchart for Training process

The flowchart for the perceptron network training is shown in the following figure. The network has to be suitably trained to obtain the response.



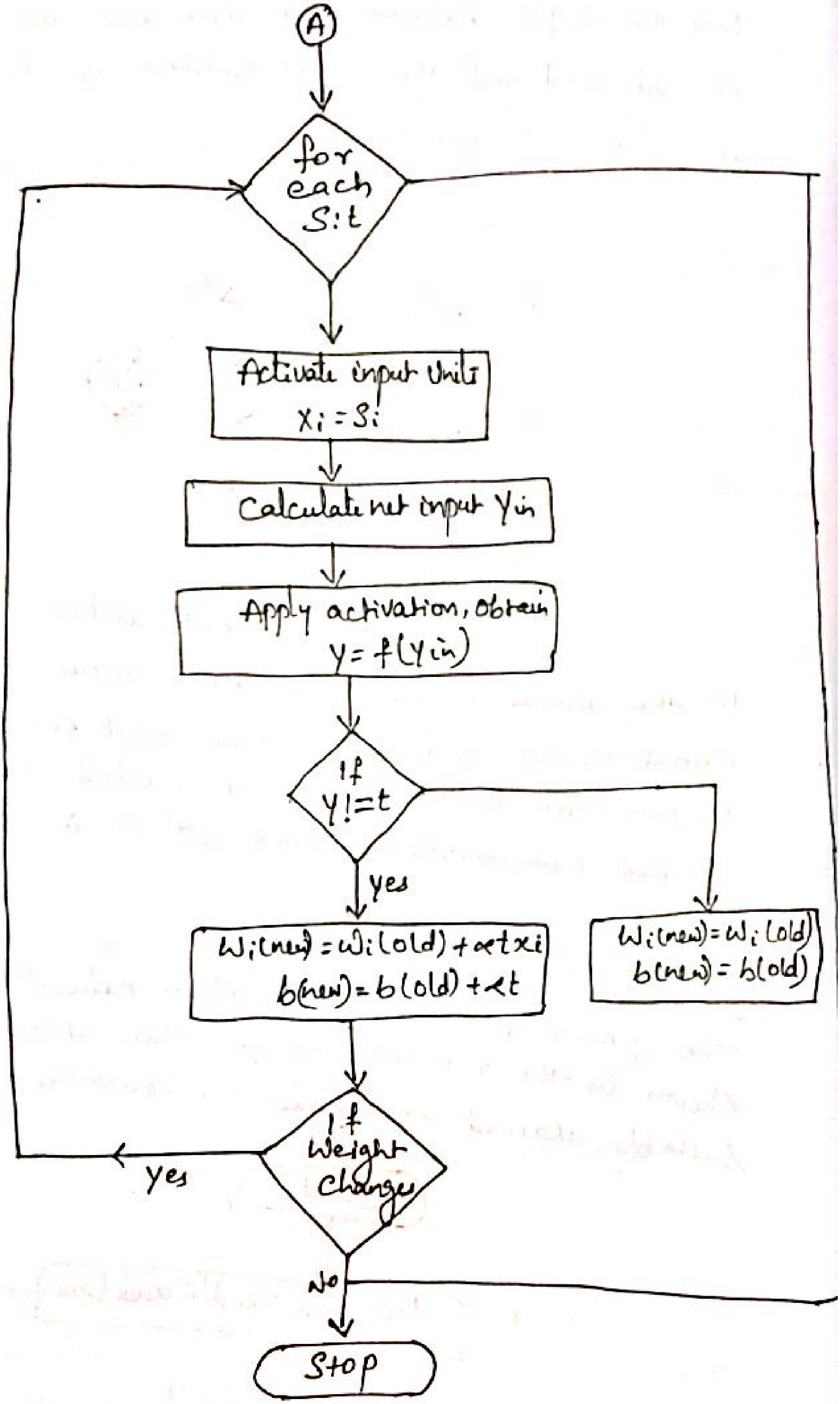


fig: flowchart for perceptron network with Single output.

Perceptron Training Algorithm for Single output class

The perceptron algorithm can be used for either binary or bipolar input vectors having bipolar targets, threshold being fixed and variables bias.

Step 0: Initialize the weight and the bias. Also initialize the learning rate α $(0 < \alpha \leq 1)$. Let $\alpha = 1$.

Step 1: Perform step 2-6 until the final stopping condition is false

Step 2: Perform step 3-5 for each training pair indicated by $s:t$.

Step 3: The input layer containing input units is applied with identity activation functions.

$$x_i = s_i$$

Step 4: Calculate the output of the network. To do so, first obtain the net input

$$y_{in} = b + \sum_{i=1}^{n} x_i w_i$$

where n is the number of input neurons in the input layer. Apply activation function over the calculated net input to obtain output

$$y = f(y_{in}) = \begin{cases} 1 & \text{if } y_{in} > \theta \\ 0 & \text{if } -\theta \leq y_{in} \leq \theta \\ -1 & \text{if } y_{in} \leq -\theta \end{cases}$$

Step 5: Weight and bias adjustment: compare the value of the actual output and the desired output

If $y \neq t$ then

$$w_i(new) = w_i(old) + \alpha t x_i$$
$$b(new) = b(old) + \alpha t$$

else

$$w_i(new) = w_i(old)$$
$$b(new) = b(old)$$

Step 6: Train the network until there is no weight change. If this condition is not met then start again from step 2.

Perceptron Training Algorithm for multiple output classes

Step 0: Initialize the weights, bias and learning rate

Step 1: Check for stopping condition. If it is false, perform step 2-6.

Step 2: Perform step 3-5 for each bipolar or binary training vector pair s:t

Step 3: Set activation of each input unit i=1 to n:

$$x_i = s_i$$

Step 4: Calculate output response for each output unit j=1 to m. First the net input is calculated as $y_{inj} = b_j + \sum_{i=1}^{n} x_i w_{ij}$

Then activation functions are applied over the net input to calculate the output response

$$y_j = f(y_{inj}) = \begin{cases} 1 & \text{if } y_{inj} > \theta \\ 0 & \text{if } -\theta \le y_{inj} \le \theta \\ -1 & \text{if } y_{inj} < -\theta \end{cases}$$

Step 5: Make adjustment in weight and bias for j=1 to m and i=1 to n

if $t_j \neq y_j$ then

$$w_{ij}(new) = w_{ij}(old) + \alpha t_j \cdot x_i$$
$$b_j(new) = b_j(old) + \alpha t_j$$

else

$$w_{ij}(new) = w_{ij}(old)$$
$$b_j(new) = b_j(old)$$

Step 6: If there is no change in weights then stop the training process else start again from step 2

The architecture of multiple output class perceptron network is given below.

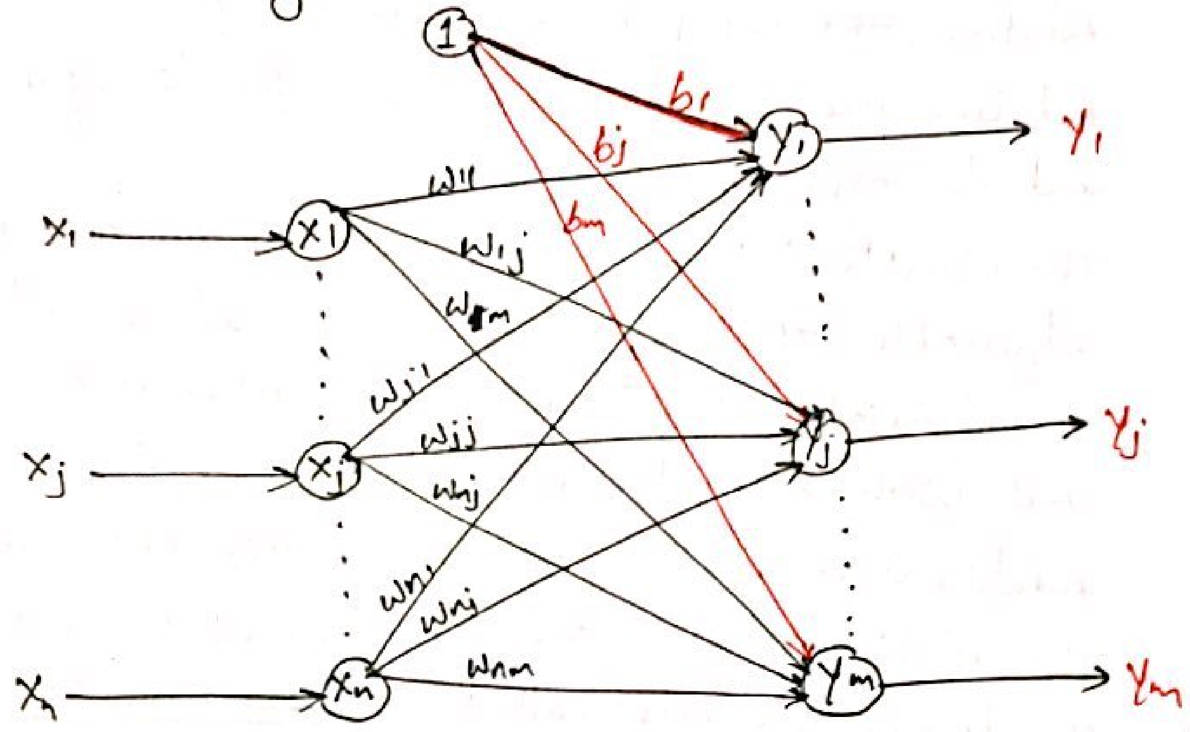


fig: perceptron N/w for several output class.

Perceptron Network Testing Algorithm.

Step 0: The initial weights to be used here are taken from the training algorithms

Step 1: For each input vector x to be classified perform step 2-3

Step 2: Set activations of the input unit

Step 3: Obtain the response of output unit.

$$y_{in} = \sum_{i=1}^{n} x_i w_i$$

$$y = f(y_{in}) = \begin{cases} 1 & \text{if } y_{in} > \theta \\ 0 & \text{if } -\theta \le y_{in} < \theta \\ -1 & \text{if } y_{in} < -\theta \end{cases}$$

Adaptive Linear Neuron (Adaline)

The Units With linear activation function are called linear Units. A network With a single linear Unit is called an Adaline (adaptive linear neuron). In an Adaline, the input-output relationship is linear. Adaline uses bipolar activation for its input signals and its target output.

The Weights between the input and the output are adjustable. The bias in Adaline in adaline acts like an adjustable weight, whose connection is from a Unit with activations being always 1.

Adaline is a network which has only one output Unit. The adaline network can be trained using delta rule. The delta rule is also called as least mean square (LMS)

Delta Rule for Single output Unit

The delta rule updates the weights between the connections so as to minimize the difference between the net input to the output Unit and the target Value. The major aim is to minimize the error over all training patterns. This is done by reducing the error for each pattern, one at a time

The delta rule for adjusting the weight of the i^{th} pattern is

$$\Delta W_i = \alpha(t - y_{in})x_i$$

where ΔW_i is the change in weight.

α is the learning rate

x is the vector of activation of input unit

y_{in} is the net input to the ~~unit~~ output unit

The delta rule in case of several output units for adjusting the weight from ith input unit to the jth output unit is

$$\Delta w_{ij} = \alpha (t_j - y_{inj}) x_i$$

Architecture

Adaline is a single unit neuron, which receives input from several units and also from one unit called bias. The Adaline model is shown in the following figure

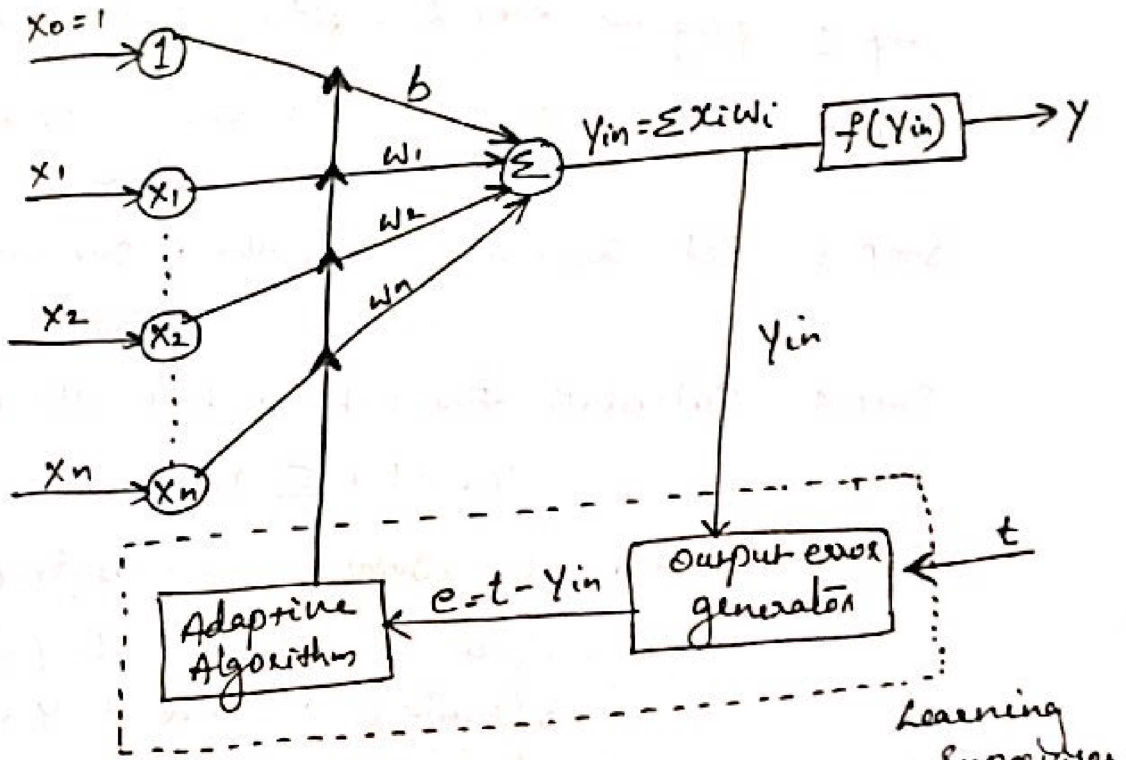


fig: Adaline Model.

The basic Adaline model consists of trainable weights. Inputs are either of the two values (+1 or -1) and the weights have sign (positive or negative). Initially

Random weights are assigned. The net input calculated is applied to a activation function that restores the output to +1 or −1. The adaline model compares the actual input with the target output and on the basis of training algorithm the weights are adjusted.

Training Algorithm

Adaline network training algorithm is as follows.

Step 0: Weights and bias are set to some random but non zero. Set the learning rate parameter α.

Step 1: perform step 2-6 when stopping condition is false.

Step 2: perform step 3-5 for each bipolar training pair s:t

Step 3: Set activations for the input units i=1 to n
$$x_i = s_i$$

Step 4: Calculate the net input to the output unit
$$y_{in} = b + \sum_{i=1}^{n} x_i w_i$$

Step 5: Update the weights and bias for i = 1 to n:
$$w_i(new) = w_i(old) + \alpha(t - y_{in}) x_i$$
$$b(new) = b(old) + \alpha(t - y_{in})$$

Step 6: If the highest weight change that occurred during training is smaller than a specified tolerance then stop the training process else continue.

Flowchart for Training process.

The flowchart for the training process is shown below.

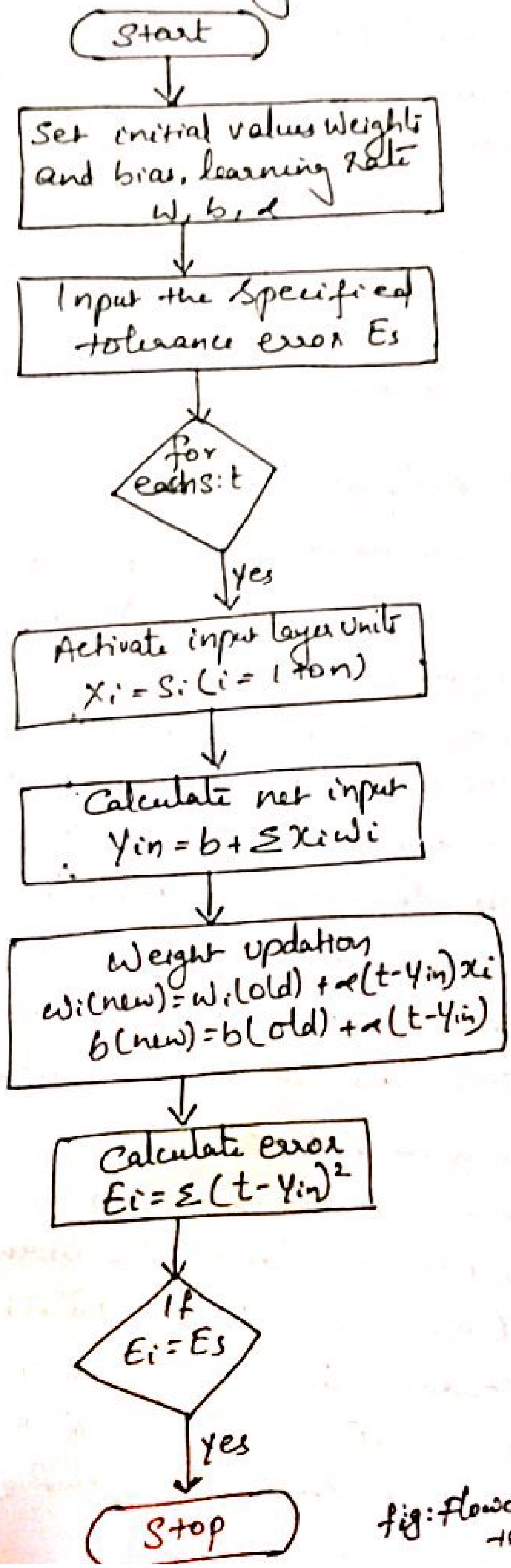


fig: Flowchart for Adaline training process

Testing Algorithm.

When training is completed, the Adaline can be used to classify input patterns. A step function is used to test the performance of the network. The testing procedure for the Adaline network is as follows.

Step 0: Initialize the weights (obtain weight from training algori

Step 1: Perform steps 2-4 for each bipolar input vector x

Step 2: Set the activations of the input units to x

Step 3: Calculate the net input to the output unit:

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

Step 4: Apply the activation function over the net input calculated:

$$y = \begin{cases} 1 & \text{if } y_{in} \geq 0 \\ -1 & \text{if } y_{in} < 0 \end{cases}$$

Backpropagation Network

The back propagation learning algorithm is one of the most important developments in neural networks. This learning algorithm is applied to multilayer feed forward networks consisting of processing elements with continuous differentiable activation functions.

The networks associated with back propagation learning algorithm are called back propagation networks.

For a given set of training input-output pair, this algorithm provides a procedure for changing the weight in a BPN to classify the given input patterns correctly.

The basic concept for this weight update algorithm is simply the gradient-descent method as used in the case of simple perceptron networks with differentiable units.

This is a method where the error is propagated back to the hidden unit.

The aim of the neural network is to train the net to achieve a balance between the net's ability to respond and its ability to give reasonable response to the input that is similar but not identical to the one that is used in training

The back propagation algorithm is different from other networks in respect to the process by which the weights are calculated during the learning period of the network.

The training of BPN is done in three stages.

- The feed forward of the input training pattern
- The calculation and back propagation of the error
- Updation of weights.

The testing of the BPN involves the computation of feed - forward phase only.

Architecture

A back-propagation neural network is a multilayer, feed - forward neural network consisting of an input layer, a hidden layer and an output layer.

The neurons present in the hidden and output layers have biases, which are the connections from the units whose activation is always 1. The bias term acts as weight.

During the back propagation phase of learning, signals are sent in the reverse direction.

The inputs are sent to the BPN and the output obtained from the net could be either binary (0,1) or bipolar (-1,+1)

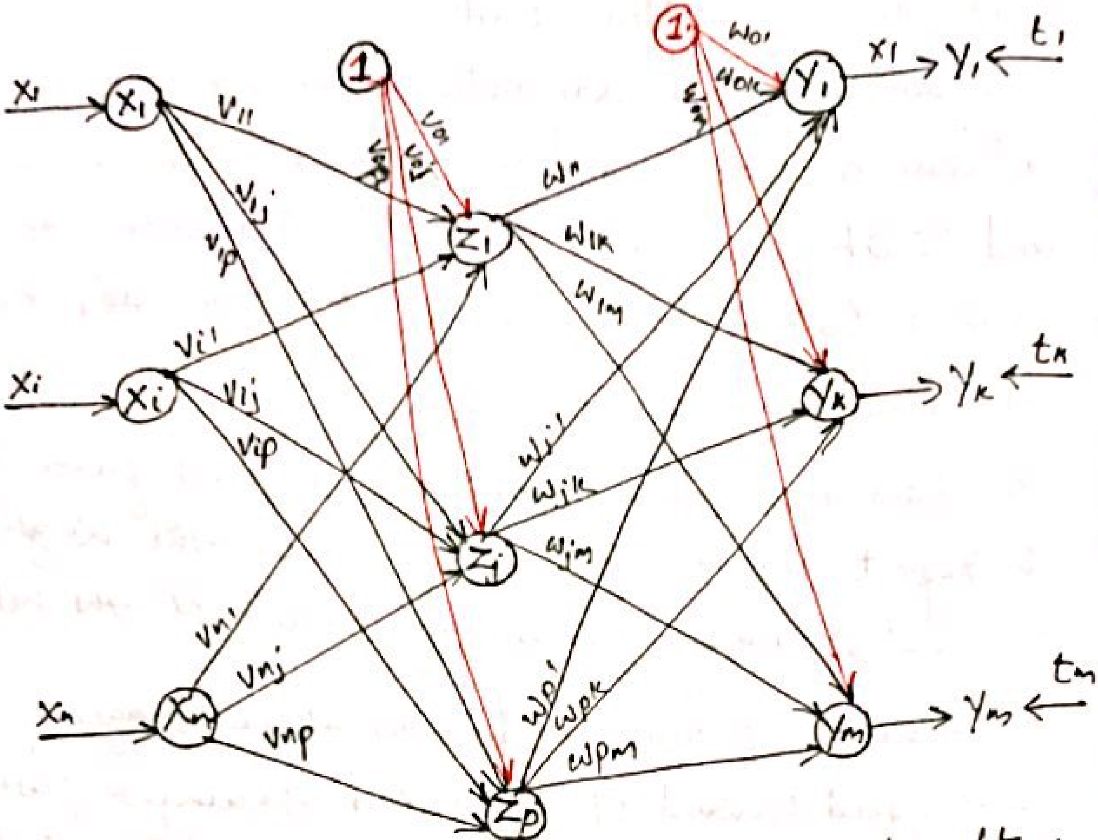


fig: Architecture of back propagation Network

$x \rightarrow$ input training vector $(x_1, \ldots x_i, \ldots x_n)$

$t \rightarrow$ target output vector $(t_1, \ldots t_k, \ldots t_m)$

$\alpha \rightarrow$ learning rate parameter

$x_i \rightarrow$ Input Unit

$v_{0j} \rightarrow$ bias on j^{th} hidden Unit

$w_{0k} \rightarrow$ bias on k^{th} output Unit

$Z_j \rightarrow$ hidden Unit j. The net input to Z_j is

$$Z_{inj} = v_{0j} + \sum_{i=1}^{n} x_i v_{ij}$$

The output is $Z_j = f(Z_{inj})$

y_k = output Unit k. The net input to y_k is

$$y_{ink} = w_{0k} + \sum_{j=1}^{p} z_j w_{jk}$$

And the output is $y_k = f(y_{ink})$

δ_k = error correction weight adjustment for w_{jk}

δ_j = error correction weight adjustment for v_{ij}

Training Algorithm.

Step 0: Initialize weights and learning rate

Step 1: perform step 2-9 when stopping condition is false

Step 2: perform 3-8 for each training pair

Feed forward phase (phase I) :-

Step 3: Each input unit receives input signal x_i and sends a to the hidden unit ($i = 1$ to n)

Step 4: Each hidden unit z_j ($j = 1$ to p) sums its weighted input signals to calculate net input:

$$z_{inj} = v_{0j} + \sum_{i=1}^{n} x_i v_{ij}$$

Output of the hidden unit is calculated by applying activation function over z_{inj}

$$z_j = f(z_{inj})$$

output signal is send from the hidden units to the input of output layer units.

Step 5: For each output unit y_k ($k = 1$ to m), calculate the net input as,

$$y_{ink} = w_{0k} + \sum_{j=1}^{p} z_j w_{jk}$$

Output signal is computed by applying activation function as $y_k = f(y_{ink})$

Back-propagation of Error (phase II) :-

Step 6: Each output unit y_k ($k = 1$ to m) receives a target pattern corresponding to the input training pattern and the error is computed as

$$\delta_k = (t_k - y_k) f'(y_{ink})$$

The updated change in weight and bias is calculated as $\Delta w_{jk} = \alpha \delta_k z_j$; $\Delta w_{0k} = \alpha \delta_k$

send δ_k back to the hidden layer

Step 7: Each hidden unit (z_j, $j = 1$ to p) sums its delta input from the output units:

$$\delta_{inj} = \sum_{k=1}^{m} \delta_k w_{jk}$$

The error term can be calculated as

$$\delta_j = \delta_{inj} f'(z_{inj})$$

On the basis of the calculated δ_j, update the change in weights and bias

$$\Delta v_{ij} = \alpha \delta_j x_i \; ; \; \Delta v_{oj} = \alpha \delta_j$$

Weight and bias updation (phase III) :-

Step 8: Each output unit (y_k, $k = 1$ to m) updates the bias and weights:

$$w_{jk}(new) = w_{jk}(old) + \Delta w_{jk}$$

$$w_{ok}(new) = w_{ok}(old) + \Delta w_{ok}$$

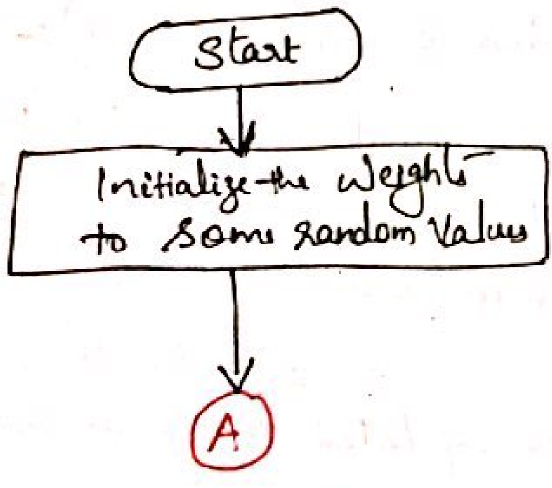
Each hidden unit (z_j, $j = 1$ to p) updates its bias and weights

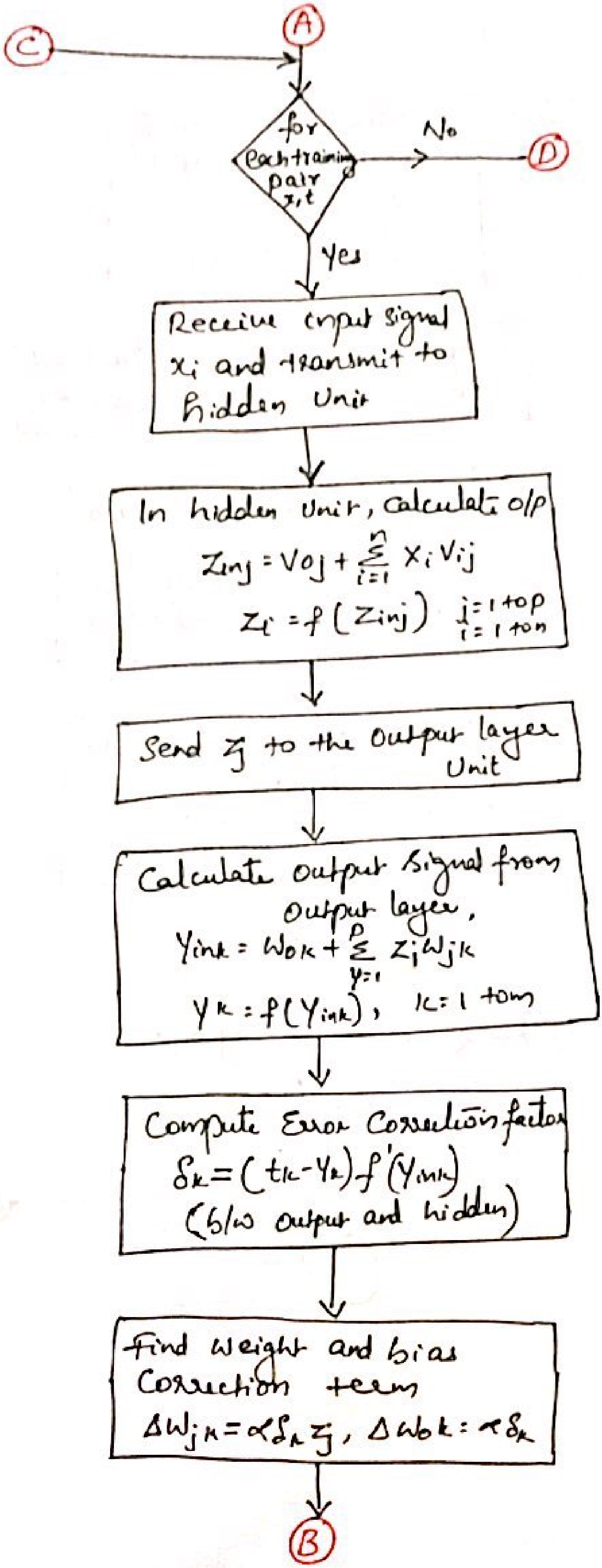
$$v_{ij}(new) = v_{ij}(old) + \Delta v_{ij}$$

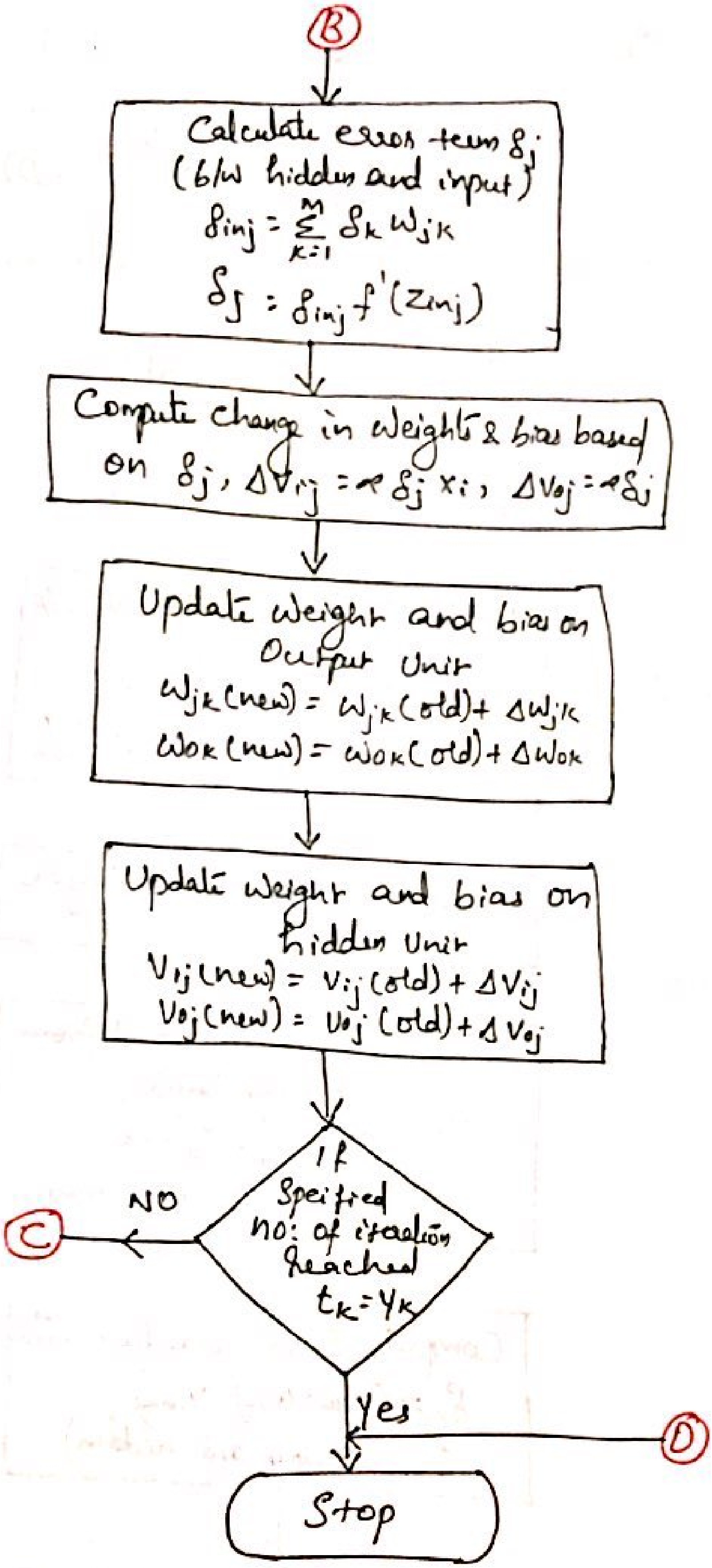
$$v_{oj}(new) = v_{oj}(old) + \Delta v_{oj}$$

Step 9: Check for the stopping condition. If actual output equals the target outputs

Flowchart for backpropagation




A
C
for each training pair x, t
No
D
Yes
Receive input signal x_i and transmit to hidden unit
In hidden unit, calculate o/p
$Z_{inj} = V_{0j} + \sum_{i=1}^{n} x_i V_{ij}$
$Z_i = f(Z_{inj})$ $j = 1$ to p, $i = 1$ to n
Send Z_j to the output layer unit
Calculate output signal from output layer,
$Y_{ink} = W_{0k} + \sum_{j=1}^{p} Z_j W_{jk}$
$Y_k = f(Y_{ink})$, $k = 1$ to m
Compute Error Correction factor
$\delta_k = (t_k - Y_k) f'(Y_{ink})$
(b/w output and hidden)
Find weight and bias correction term
$\Delta W_{jk} = \alpha \delta_k Z_j$, $\Delta W_{0k} = \alpha \delta_k$
B


B
Calculate error term δ_j (b/w hidden and input)
$\delta_{inj} = \sum_{k=1}^{m} \delta_k w_{jk}$
$\delta_j = \delta_{inj} f'(z_{inj})$
Compute change in weights & bias based on δ_j, $\Delta v_{ij} = \alpha \delta_j x_i$, $\Delta v_{oj} = \alpha \delta_j$
Update weight and bias on Output Unit
$w_{jk}(new) = w_{jk}(old) + \Delta w_{jk}$
$w_{ok}(new) = w_{ok}(old) + \Delta w_{ok}$
Update weight and bias on hidden Unit
$v_{ij}(new) = v_{ij}(old) + \Delta v_{ij}$
$v_{oj}(new) = v_{oj}(old) + \Delta v_{oj}$
If Speified no: of iteration reached $t_k = y_k$
NO
C
Yes
D
Stop

Testing Algorithm of Back propagation Network

The testing procedure of the BPN is as follows.

Step 0: Initialize the weights. The weights are taken from the training algorithm

Step 1: perform steps 2-4 for each input vector

Step 2: Set the activation of input unit for x_i ($i = 1$ to n)

Step 3: Calculate the net input to hidden unit x and its output. for $j = 1$ to p.

$$z_{inj} = v_{oj} + \sum_{i=1}^{n} x_i v_{ij}$$

$$z_j = f(z_{inj})$$

Step 4: Now compute the output of the output layer unit.

for $k = 1$ to m,

$$y_{ink} = w_{ok} + \sum_{j=1}^{p} z_j w_{jk}$$

$$y_k = f(y_{ink})$$

Sigmoidal function are used for calculating the output.

MOD-III

Fuzzy Logic

Fuzzy Logic approach is used to handle ambiguity and uncertainty existing in the complex problem.

Fuzzy Logic is a form of multivalued logic to deal with reasoning that is approximate rather than precise.

Fuzzy Logic variables may have a truth value that ranges between 0 and 1 and is not constrained to the two truth value of classic propositional Logic.

Fuzzy Logic provides an inference structure that enables appropriate human reasoning capabilities.

Imprecise and Vague data → Fuzzy Logic System → Decision

Fuzzy Logic provides a means to model the uncertainity associated with vagueness, imprecision and lack of information regarding a problem.

Fuzzy Logic operates on the concept of membership. The membership function lie over a range of real numbers from 0.0 to 1.0. The membership value is "1" if it belongs to the set and "0" if it is not a member of the set.

The membership in a set is found to be binary, that is either the element is a member of a set or not.

It can be indicated as

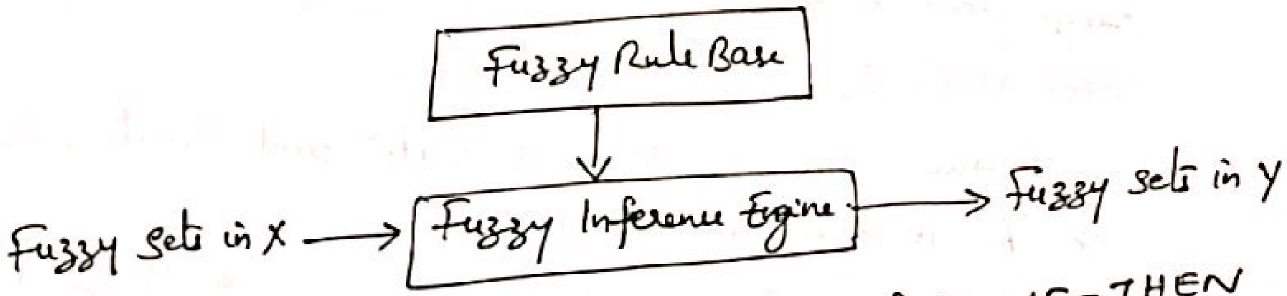
$$\chi_A(x) = \begin{cases} 1, & x \in A \\ 0, & x \notin A \end{cases}$$

Fuzzy logic consists of fuzzy inference engine or fuzzy rule base to perform approximate reasoning similar to that of the human brain.

Scanned with CamScanner

Fuzzy sets form the building blocks for fuzzy IF-THEN rules which have the general form "If x is A THEN y is B", where A and B are fuzzy sets.

A fuzzy system is a set of fuzzy rules that converts inputs to outputs. The basic configuration of a pure fuzzy system is shown in the following figure.



The fuzzy inference engine combines fuzzy IF-THEN rules into a mapping from fuzzy sets in the input space X to fuzzy sets in the output space Y based on fuzzy logic principles.

Fuzzy system are constructed from a collection of rules, and are nonlinear mapping of inputs to output.

Fuzzy sets:

Fuzzy set can be viewed as an extension of the basic concepts of crisp sets. An important property of fuzzy set is that it allows partial membership. A fuzzy set is a set having degree of membership between 0 and 1. The membership in a fuzzy set need not be complete i.e member of one fuzzy set can also be the member of other fuzzy set in the Universe.

A fuzzy set A in the Universe U can be defined as a set of ordered pairs and it is given by

$$A = \{ (x, \mu_A(x)) \mid x \in U \}$$

where $\mu_A(x)$ is the degree of membership of x in A and it indicates the degree that x belongs to A.

The degree of membership $\mu_A(x)$ assumes values in the range from 0 to 1. ie the membership is set to unit interval [0,1] or $\mu_A(x) \in [0,1]$.

When the Universe U is discrete and finite, fuzzy set A is given as follows.

$$A = \left\{ \frac{\mu_A(x_1)}{x_1} + \frac{\mu_A(x_2)}{x_2} + \frac{\mu_A(x_3)}{x_3} + \dots \right\} = \left\{ \sum_{i=1}^{n} \frac{\mu_A(x_i)}{x_i} \right\}$$

where n is a finite value.

When the Universe U is Continuous and infinite, fuzzy set A is given by

$$A = \left\{ \int \frac{\mu_A(x)}{x} \right\}$$

A fuzzy set is Universal fuzzy set if and only if the value of the membership function is 1 for all the membership under Consideration

Any fuzzy set A defined on a universe U is a Subset of that Universe. Two fuzzy sets A and B are said to be equal fuzzy sets if $\mu_A(x) = \mu_B(x)$ for all $x \in U$

A fuzzy Set A is said to be empty fuzzy set if and only if the value of the membership function is 0 for all possible members. The Universal fuzzy set can also be called whole fuzzy set.

Scanned with CamScanner

The collection of all fuzzy sets and fuzzy subsets on Universe U is called fuzzy power set $p(U)$. The cardinality of the fuzzy power set, $np(U)$ is infinite, ie $np(U) = \infty$

$$A \subseteq U \Rightarrow \mu_A(x) \leq \mu_U(x)$$

for all $x \in U$

$$\mu_\phi(x) = 0, \quad \mu_U(x) = 1$$

Fuzzy Set Operations

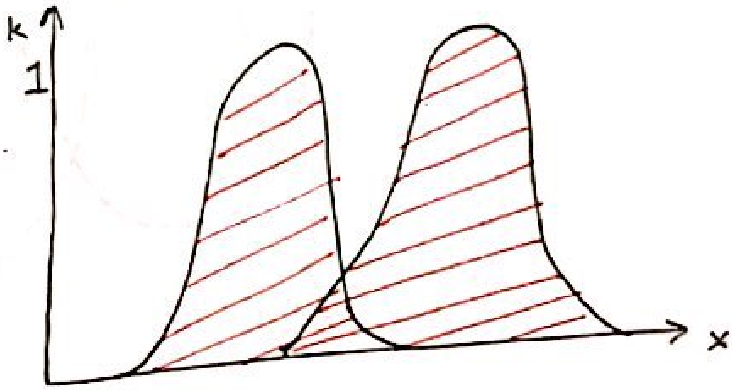
Let A and B be fuzzy sets in the universe of discourse U. For a given element x on the universe, the following function theoretic operations of union, intersection and complement are defined for fuzzy sets A and B on U.

Union.

The union of fuzzy sets $\underset{\sim}{A}$ and $\underset{\sim}{B}$ denoted by $\underset{\sim}{A} \cup \underset{\sim}{B}$ is defined as

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

where '$\vee$' indicates max operation. The venn diagram: Union operation is given

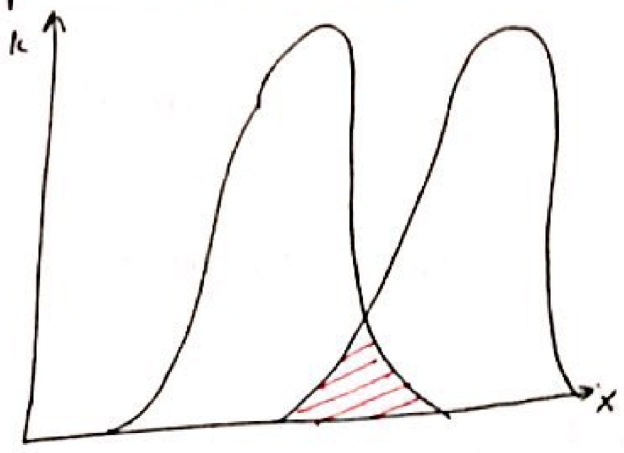


Intersection

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

$$\mu_{\underset{\sim}{A} \cap \underset{\sim}{B}}(x) = \min[\mu_{\underset{\sim}{A}}(x), \mu_{\underset{\sim}{B}}(x)] = \mu_{\underset{\sim}{A}}(x) \wedge \mu_{\underset{\sim}{B}}(x) \quad \forall x \in U$$

where $\wedge$ indicates min operator. The ven diagram for intersection operation is shown below

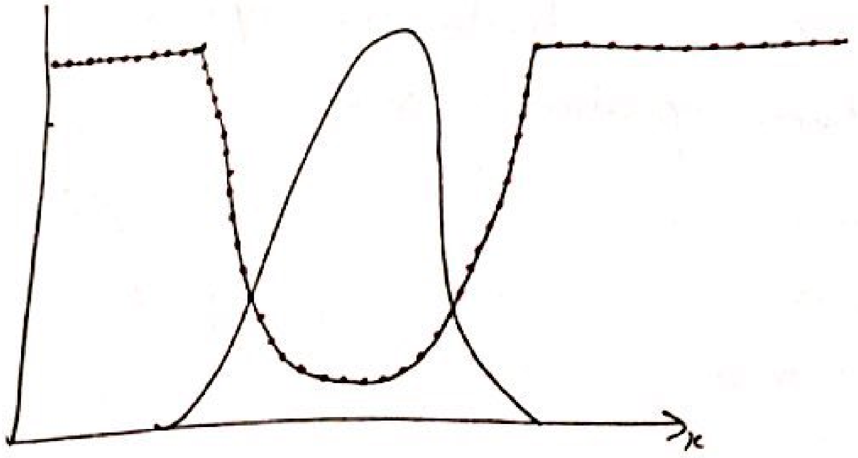


Complement.

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

$$\mu_{\bar{\underset{\sim}{A}}}(x) = 1 - \mu_{\underset{\sim}{A}}(x) \quad \forall x \in U$$

The ven diagram for complement operation of fuzzy set $\underset{\sim}{A}$ is shown below.



More operations on fuzzy set.

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

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

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

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

3. Bounded Sum; The bounded sum $(\underset{\sim}{A} \oplus \underset{\sim}{B})$ of two fuzzy sets $\underset{\sim}{A}$ and $\underset{\sim}{B}$ is defined as

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

4. Bounded difference: The bounded difference $(\underset{\sim}{A} \odot \underset{\sim}{B})$ of two fuzzy sets $\underset{\sim}{A}$ and $\underset{\sim}{B}$ is defined a

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

properties of fuzzy set

Frequently Used properties of fuzzy sets are given below.

1. Commutativity :.

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

2. Associativity :

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

3. Distributivity :

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

4. Idempotency :

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

5. Identity

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

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

6. Involution (double negation)

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

7. Transitivity

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

8. De Morgan's law

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

Fuzzy Relations

Fuzzy Relations relate elements of one Universe to those of another Universe through the Cartesian product of the two universe. These can also be referred to as fuzzy sets defined on universal sets, which are Cartesian products.

A fuzzy relation is based on the concept that everything is related to some extent or unrelated.

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

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

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

Let $X = \{x_1, x_2, \ldots x_n\}$ and $Y = \{y_1, y_2, \ldots y_m\}$ Fuzzy relation $R(X,Y)$ can be expressed by $n \times m$ matrix as follows.

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

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

A fuzzy graph is a graphical representation of a binary fuzzy relation. Each element in X and Y corresponds to a node in the fuzzy graph. The connection links are established between the nodes by the elements of $X \times Y$ with nonzero membership grades in $R(X,Y)$.

When $X \neq Y$ the link connecting the two nodes is an undirected binary graph called bipartite graph.

When $X = Y$ a node is connected to itself and directed links are used, in such a case, the fuzzy graph is called directed graph.

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

$$\mu_{domain\,R}(x) = \max_{y \in Y} \mu_R(x,y) \quad \forall x \in X.$$

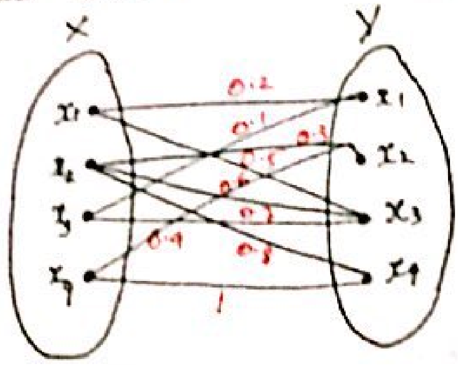
- The range of binary fuzzy relation $R(X,Y)$ is a fuzzy set, ran $R(X,Y)$, having the membership function as

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

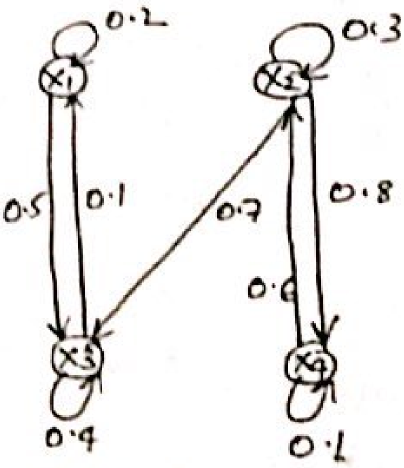
Consider a universe $X = \{x_1, x_2, x_3, x_4\}$ and binary fuzzy relation on X as

$$R(X,X) = \begin{array}{c} \\ x_1 \\ x_2 \\ x_3 \\ x_4 \end{array} \begin{array}{c} \begin{array}{cccc} x_1 & x_2 & x_3 & x_4 \end{array} \\ \begin{bmatrix} 0.2 & 0 & 0.5 & 0 \\ 0 & 0.3 & 0.7 & 0.8 \\ 0.1 & 0 & 0.4 & 0 \\ 0 & 0.6 & 0 & 1 \end{bmatrix} \end{array}$$

The bipartite graph and simple fuzzy graph of $R(X,X)$ is shown below.



Bipartite graph



Simple fuzzy graph

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

Let R be a Relation from X to Y given by

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

The corresponding fuzzy matrix for relation R is

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

The graph of the above relation $R = X \times Y$ is shown below

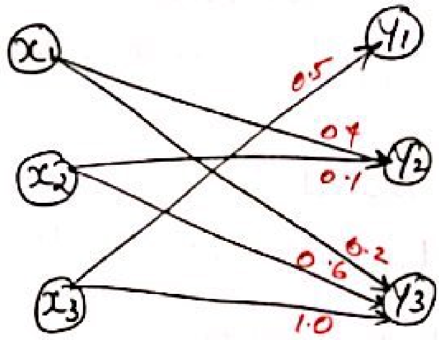


fig: Graph of fuzzy relation

Cardinality of fuzzy Relation

The cardinality of fuzzy sets on any universe is infinity. hence the cardinality of a fuzzy relation between two or more universe is also infinity.

operations on fuzzy Relation

The basic operations on fuzzy sets can also apply on fuzzy relations. Let R and S be fuzzy Relations on the Cartesian space $X \times Y$. The operation are as given below.

1. Union $\quad \mu_{R \cup S}(x,y) = \max[\mu_R(x,y), \mu_S(x,y)]$

Scanned with CamScanner

2. Intersection $\mu_{\underset{\sim}{R} \cap \underset{\sim}{S}}(x,y) = \min[\mu_R(x,y), \mu_S(x,y)]$

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

4. Containment $\underset{\sim}{R} \subset \underset{\sim}{S} \Rightarrow \mu_R(x,y) \leq \mu_S(x,y)$

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

6. Projection : For a fuzzy relation $R(X,Y)$, let $[R \downarrow Y]$ denote the projection of R onto Y. Then $[R \downarrow Y]$ is a fuzzy relation in Y whose membership function is defined by
$$\mu_{[R \downarrow Y]}(x,y) = \max_x \mu_R(x,y)$$

<u>Properties of fuzzy Relations</u>

Fuzzy Relations holds the property of

Commutativity

Associativity

Distributivity

Idempotency

Identity

$$\underset{\sim}{R} \cup \bar{\underset{\sim}{R}} \neq \underset{\sim}{E}$$

$$\underset{\sim}{R} \cap \bar{\underset{\sim}{R}} \neq \phi$$

<u>Fuzzy Composition</u>

Let A be a fuzzy set on Universe X and B be a fuzzy set on Universe Y. The Cartesian product over

over A and B results in fuzzy relation R and is contained within the entire Cartesian space. ie

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

where $\quad \underset{\sim}{R} \subseteq X \times Y$

The membership function of fuzzy relation is given by

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

There are two types of fuzzy composition techniques.

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

There also exist fuzzy min-max composition method, but the most commonly used technique is fuzzy max-min composition.

Let R be fuzzy relation on $X \times Y$ and be the fuzzy relation on $Y \times Z$

The max-min composition of $\underset{\sim}{R}(X,Y)$ and $\underset{\sim}{S}(Y,Z)$ denoted by $\underset{\sim}{R}(X,Y) \circ \underset{\sim}{S}(Y,Z)$ is defined as $\underset{\sim}{T}(X,Z)$ as

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

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

The min-max composition of $\underset{\sim}{R}(X,Y)$ and $\underset{\sim}{S}(Y,Z)$ dented a $\underset{\sim}{R}(X,Y) \circ \underset{\sim}{S}(Y,Z)$ is defined by $\underset{\sim}{T}(X,Z)$ as

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

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

The max-product composition of $\underset{\sim}{R}(X,Y)$ and $\underset{\sim}{S}(Y,Z)$ denoted as $\underset{\sim}{R}(X,Y) \cdot \underset{\sim}{S}(Y,Z)$ is defined as $\underset{\sim}{T}(X,Z)$ as

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

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

The properties of fuzzy composition can be given as

$$\underset{\sim}{R} \circ \underset{\sim}{S} \neq \underset{\sim}{S} \circ \underset{\sim}{R}$$

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

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

Problems

1. Find the power set and cardinality of the given set $X = \{2,4,6\}$. Also find the cardinality of power set.

Solution :-

Since set X contains three elements, so its cardinal number is $n_x = 3$

The power set of X is given by

$$P(X) = \{\phi, \{2\}, \{4\}, \{6\}, \{2,4\}, \{4,6\}, \{2,6\}, \{2,4,6\}\}$$

The cardinality of power set $P(X)$ denoted by $n\,P(X)$.

$$n\,P(X) = 2^{n_x} = 2^3 = \underline{8}$$

2. Consider two fuzzy sets.

$$\underset{\sim}{A} = \left\{ \frac{1}{2} + \frac{0.3}{4} + \frac{0.5}{6} + \frac{0.2}{8} \right\}$$

$$\underset{\sim}{B} = \left\{ \frac{0.5}{2} + \frac{0.4}{4} + \frac{0.1}{6} + \frac{1}{8} \right\}$$

Find $\underset{\sim}{A} \cup \underset{\sim}{B}$, $\underset{\sim}{A} \cap \underset{\sim}{B}$, $\bar{\underset{\sim}{A}}$, $\bar{\underset{\sim}{B}}$, $\underset{\sim}{A}/\underset{\sim}{B}$, $\underset{\sim}{B}/\underset{\sim}{A}$

$$\underset{\sim}{A} \cup \underset{\sim}{B} = \max\{\mu_A(x), \mu_B(x)\}$$

$$= \left\{\frac{1}{2} + \frac{0.4}{4} + \frac{0.5}{8} + \frac{1}{8}\right\}$$

$$\underset{\sim}{A} \cap \underset{\sim}{B} = \min\{\mu_A(x), \mu_B(x)\}$$

$$= \left\{\frac{0.5}{2} + \frac{0.3}{4} + \frac{0.1}{6} + \frac{0.2}{8}\right\}$$

$$\bar{\underset{\sim}{A}} = 1 - \mu_A(x) = \left\{\frac{0}{2} + \frac{0.7}{4} + \frac{0.5}{6} + \frac{0.8}{8}\right\}$$

$$\bar{\underset{\sim}{B}} = 1 - \mu_B(x) = \left\{\frac{0.5}{2} + \frac{0.6}{4} + \frac{0.9}{6} + \frac{0}{8}\right\}$$

$$\underset{\sim}{A} | \underset{\sim}{B} = \underset{\sim}{A} \cap \bar{\underset{\sim}{B}} = \left\{\frac{0.5}{2} + \frac{0.3}{4} + \frac{0.5}{6} + \frac{0}{8}\right\}$$

$$\underset{\sim}{B} | \underset{\sim}{A} = \underset{\sim}{B} \cap \bar{\underset{\sim}{A}} = \left\{\frac{0}{2} + \frac{0.4}{4} + \frac{0.1}{6} + \frac{0.8}{8}\right\}$$

3. For the following two fuzzy sets

$$\underset{\sim}{B_1} = \left\{\frac{1}{1.0} + \frac{0.75}{1.5} + \frac{0.3}{2.0} + \frac{0.15}{2.5} + \frac{0}{3.0}\right\}$$

$$\underset{\sim}{B_2} = \left\{\frac{1}{1.0} + \frac{0.6}{1.5} + \frac{0.2}{2.0} + \frac{0.1}{2.5} + \frac{0}{3.0}\right\}$$

Find $\underset{\sim}{B_1} \cup \underset{\sim}{B_2}$, $\underset{\sim}{B_1} \cap \underset{\sim}{B_2}$, $\bar{\underset{\sim}{B_1}}$, $\bar{\underset{\sim}{B_2}}$, $\underset{\sim}{B_1} | \underset{\sim}{B_2}$, $\overline{\underset{\sim}{B_1} \cup \underset{\sim}{B_2}}$, $\overline{\underset{\sim}{B_1} \cap \underset{\sim}{B_2}}$,
$\underset{\sim}{B_1} \cap \bar{\underset{\sim}{B_1}}$, $\underset{\sim}{B_1} \cup \bar{\underset{\sim}{B_2}}$, $\underset{\sim}{B_2} \cap \bar{\underset{\sim}{B_2}}$, $\underset{\sim}{B_2} \cup \bar{\underset{\sim}{B_2}}$

Solution :-

$$\underset{\sim}{B_1} \cup \underset{\sim}{B_2} = \left\{\frac{1}{1.0} + \frac{0.75}{1.5} + \frac{0.3}{2.0} + \frac{0.15}{2.5} + \frac{0}{3.0}\right\}$$

$$\underset{\sim}{B_1} \cap \underset{\sim}{B_2} = \left\{\frac{1}{1.0} + \frac{0.6}{1.5} + \frac{0.2}{2.0} + \frac{0.1}{2.5} + \frac{0}{3.0}\right\}$$

$$\bar{\underset{\sim}{B_1}} = \left\{\frac{0}{1.0} + \frac{0.25}{1.5} + \frac{0.7}{2.0} + \frac{0.85}{2.5} + \frac{1}{3.0}\right\}$$

$$\bar{\underset{\sim}{B_2}} = \left\{\frac{0}{1.0} + \frac{0.4}{1.5} + \frac{0.8}{2.0} + \frac{0.9}{2.5} + \frac{1}{3.0}\right\}$$

$$\underset{\sim}{B_1}/\underset{\sim}{B_2} = \underset{\sim}{B_1} \cap \overline{\underset{\sim}{B_2}}$$

$$= \left\{ \frac{0}{1.0} + \frac{0.4}{1.5} + \frac{0.3}{2.0} + \frac{0.15}{2.15} + \frac{0}{3.0} \right\}$$

$$\overline{\underset{\sim}{B_1} \cup \underset{\sim}{B_2}} = \left\{ \frac{0}{1.0} + \frac{0.25}{1.5} + \frac{0.7}{2.0} + \frac{0.85}{2.5} + \frac{1}{3.0} \right\}$$

$$\overline{\underset{\sim}{B_1} \cap \underset{\sim}{B_2}} = \left\{ \frac{0}{1.0} + \frac{0.4}{1.5} + \frac{0.8}{2.0} + \frac{0.9}{2.5} + \frac{1}{3.0} \right\}$$

$$\underset{\sim}{B_1} \cap \overline{\underset{\sim}{B_1}} = \left\{ \frac{0}{1.0} + \frac{0.25}{1.5} + \frac{0.3}{2.0} + \frac{0.15}{2.5} + \frac{1}{3.0} \right\}$$

$$\underset{\sim}{B_2} \cap \overline{\underset{\sim}{B_2}} = \left\{ \frac{0}{1.0} + \frac{0.4}{1.5} + \frac{0.2}{2.0} + \frac{0.1}{2.5} + \frac{0}{3.0} \right\}$$

$$\underset{\sim}{B_2} \cup \overline{\underset{\sim}{B_2}} = \left\{ \frac{1}{1.0} + \frac{0.6}{1.5} + \frac{0.8}{2.0} + \frac{0.9}{2.5} + \frac{1}{3.0} \right\}$$

4. For the following two fuzzy sets

$$\underset{\sim}{D_1} = \left\{ \frac{0}{0} + \frac{0.2}{10} + \frac{0.35}{20} + \frac{0.65}{30} + \frac{0.85}{40} + \frac{1}{50} \right\}$$

$$\underset{\sim}{D_2} = \left\{ \frac{0}{0} + \frac{0.35}{10} + \frac{0.25}{20} + \frac{0.8}{30} + \frac{0.95}{40} + \frac{1}{50} \right\}$$

Find the following membership function

(a) $\mu_{\underset{\sim}{D_1} \cup \underset{\sim}{D_2}}$ (b) $\mu_{\underset{\sim}{D_1} \cap \underset{\sim}{D_2}}(x)$ (c) $\mu_{\overline{\underset{\sim}{D_1}}}(x)$ (d) $\mu_{\overline{\underset{\sim}{D_2}}}(x)$

(e) $\mu_{\underset{\sim}{D_1} \cup \overline{\underset{\sim}{D_1}}}(x)$ (f) $\mu_{\underset{\sim}{D_1} \cap \overline{\underset{\sim}{D_1}}}(x)$ (g) $\mu_{\underset{\sim}{D_2} \cup \overline{\underset{\sim}{D_2}}}$ (h) $\mu_{\underset{\sim}{D_2} \cap \overline{\underset{\sim}{D_2}}}(x)$

(i) $\mu_{\underset{\sim}{D_1} / \underset{\sim}{D_2}}(x)$ (j) $\mu_{\underset{\sim}{D_2} / \underset{\sim}{D_1}}(x)$.

Solution :-

(a) $\mu_{\underset{\sim}{D_1} \cup \underset{\sim}{D_2}}(x) = \max\{\mu_{\underset{\sim}{D_1}}(x), \mu_{\underset{\sim}{D_2}}(x)\}$

$$= \left\{ \frac{0}{0} + \frac{0.35}{10} + \frac{0.35}{20} + \frac{0.8}{30} + \frac{0.95}{40} + \frac{1}{50} \right\}$$

(b) $\mu_{\underset{\sim}{D_1} \cap \underset{\sim}{D_2}}(x) = \min\{\mu_{\underset{\sim}{D_1}}(x), \mu_{\underset{\sim}{D_2}}(x)\}$

$$= \left\{ \frac{0}{0} + \frac{0.2}{10} + \frac{0.25}{20} + \frac{0.65}{30} + \frac{0.85}{40} + \frac{1}{50} \right\}$$

(c) $\mu_{\overline{\underset{\sim}{D_1}}}(x) = 1 - \mu_{\underset{\sim}{D_1}}(x)$

$$= \left\{ \frac{1}{0} + \frac{0.8}{10} + \frac{0.65}{20} + \frac{0.35}{30} + \frac{0.15}{40} + \frac{0}{50} \right\}$$

(d) $\mu_{\overline{\underset{\sim}{D_2}}}(x) = 1 - \mu_{\underset{\sim}{D_2}}(x)$

$$= \left\{ \frac{1}{0} + \frac{0.65}{10} + \frac{0.75}{20} + \frac{0.2}{30} + \frac{0.05}{40} + \frac{0}{50} \right\}$$

(e) $\mu_{\underset{\sim}{D_1} \cup \overline{D_1}}(x) = \max\{\mu_{\underset{\sim}{D_1}}(x), \mu_{\overline{D_1}}(x)\}$

$$= \left\{ \frac{1}{0} + \frac{0.8}{10} + \frac{0.65}{20} + \frac{0.65}{30} + \frac{0.85}{40} + \frac{1}{50} \right\}$$

(f) $\mu_{\underset{\sim}{D_1} \cap \overline{D_1}}(x) = \min\{\mu_{\underset{\sim}{D_1}}(x), \mu_{\overline{D_1}}(x)\}$

$$= \left\{ \frac{0}{0} + \frac{0.2}{10} + \frac{0.35}{20} + \frac{0.35}{30} + \frac{0.15}{40} + \frac{0}{50} \right\}$$

(g) $\mu_{\underset{\sim}{D_2} \cup \overline{D_2}}(x) = \max\{\mu_{\underset{\sim}{D_2}}(x), \mu_{\overline{\underset{\sim}{D_2}}}(x)\}$

$$= \left\{ \frac{1}{0} + \frac{0.65}{10} + \frac{0.75}{20} + \frac{0.8}{30} + \frac{0.95}{40} + \frac{1}{50} \right\}$$

(h) $\mu_{\underset{\sim}{D_2} \cap \overline{\underset{\sim}{D_2}}}(x) = \min\{\mu_{\underset{\sim}{D_2}}(x), \mu_{\overline{D_2}}(x)\}$

$$= \left\{ \frac{0}{0} + \frac{0.35}{10} + \frac{0.25}{20} + \frac{0.2}{30} + \frac{0.05}{40} + \frac{0}{50} \right\}$$

(i) $\mu_{\underset{\sim}{D_1} | \underset{\sim}{D_2}}(x) = \mu_{\underset{\sim}{D_1} \cap \overline{\underset{\sim}{D_2}}}(x) = \min\{\mu_{\underset{\sim}{D_1}}(x), \mu_{\overline{\underset{\sim}{D_2}}}(x)\}$

$$= \left\{ \frac{0}{0} + \frac{0.2}{10} + \frac{0.35}{20} + \frac{0.2}{30} + \frac{0.05}{40} + \frac{0}{50} \right\}$$

(j) $\mu_{\underset{\sim}{D_2} / \underset{\sim}{D_1}}(x) = \mu_{\underset{\sim}{D_2} \cap \overline{\underset{\sim}{D_1}}}(x) = \min\{\mu_{\underset{\sim}{D_2}}(x), \mu_{\overline{\underset{\sim}{D_1}}}(x)\}$

$$= \frac{0}{0} + \frac{0.35}{10} + \frac{0.25}{20} + \frac{0.35}{30} + \frac{0.15}{40} + \frac{0}{50} \Big\}$$

5. Design a computer software to perform image processing to locate objects within a scene. The two fuzzy sets representing a plane and a train image are

$$\text{plane} = \left\{ \frac{0.2}{\text{train}} + \frac{0.5}{\text{bike}} + \frac{0.3}{\text{boat}} + \frac{0.8}{\text{plane}} + \frac{0.1}{\text{house}} \right\}$$

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

Find the following operations

(a) $\underset{\sim}{\text{plane}} \cup \underset{\sim}{\text{Train}} = \max\{\mu_{\text{plane}}(x), \mu_{\text{Train}}(x)\}$

$$= \left\{ \frac{1.0}{\text{train}} + \frac{0.5}{\text{bike}} + \frac{0.4}{\text{boat}} + \frac{0.8}{\text{plane}} + \frac{0.2}{\text{house}} \right\}$$

(b) $\underset{\sim}{\text{plane}} \cap \underset{\sim}{\text{Train}} = \min\{\mu_{\text{plane}}(x), \mu_{\text{Train}}(x)\}$

$$= \left\{ \frac{0.2}{\text{train}} + \frac{0.2}{\text{bike}} + \frac{0.3}{\text{boat}} + \frac{0.5}{\text{plane}} + \frac{0.1}{\text{house}} \right\}$$

(c) $\overline{\underset{\sim}{\text{plane}}} = 1 - \mu_{\underset{\sim}{\text{plane}}}(x)$

$$= \left\{ \frac{0.8}{\text{train}} + \frac{0.5}{\text{bike}} + \frac{0.7}{\text{boat}} + \frac{0.2}{\text{plane}} + \frac{0.9}{\text{house}} \right\}$$

(d) $\overline{\underset{\sim}{\text{Train}}} = 1 - \mu_{\underset{\sim}{\text{Train}}}(x)$

$$= \left\{ \frac{0}{\text{train}} + \frac{0.8}{\text{bike}} + \frac{0.6}{\text{boat}} + \frac{0.5}{\text{plane}} + \frac{0.8}{\text{house}} \right\}$$

(e) $\underset{\sim}{\text{plane}} \mid \underset{\sim}{\text{Train}} = \underset{\sim}{\text{plane}} \cap \overline{\underset{\sim}{\text{Train}}}$

$= \min\{\mu_{\text{plane}}(x), \mu$

(f) $\overline{\underset{\sim}{\text{plane}} \cup \underset{\sim}{\text{Train}}} = 1 - \max\{\mu_{\underset{\sim}{\text{plane}}}(x), \mu_{\underset{\sim}{\text{Train}}}(x)\}$

$$= \left\{ \frac{0}{\text{train}} + \frac{0.5}{\text{bike}} + \frac{0.6}{\text{boat}} + \frac{0.2}{\text{plane}} + \frac{0.8}{\text{house}} \right\}$$

(g) $\overline{\underset{\sim}{\text{plane}} \cap \underset{\sim}{\text{Train}}} = 1 - \min\{\mu_{\underset{\sim}{\text{plane}}}(x), \mu_{\underset{\sim}{\text{Train}}}(x)\}$

$$= \left\{ \frac{0.8}{\text{train}} + \frac{0.8}{\text{bike}} + \frac{0.7}{\text{boat}} + \frac{0.5}{\text{plane}} + \frac{0.9}{\text{house}} \right\}$$

(h) $\underset{\sim}{\text{plane}} \cup \overline{\underset{\sim}{\text{plane}}} = \max\{\mu_{\underset{\sim}{\text{plane}}}(x), \mu_{\overline{\underset{\sim}{\text{plane}}}}(x)\}$

$$= \left\{ \frac{0.8}{\text{train}} + \frac{0.5}{\text{bike}} + \frac{0.7}{\text{boat}} + \frac{0.8}{\text{plane}} + \frac{0.9}{\text{house}} \right\}$$

(i) $\underset{\sim}{\text{plane}} \cap \overline{\underset{\sim}{\text{plane}}} = \min\{\mu_{\underset{\sim}{\text{plane}}}(x), \mu_{\overline{\underset{\sim}{\text{plane}}}}(x)\}$

$$= \left\{ \frac{0.2}{\text{train}} + \frac{0.5}{\text{bike}} + \frac{0.3}{\text{boat}} + \frac{0.2}{\text{plane}} + \frac{0.9}{\text{house}} \right\}$$

(j) $\underset{\sim}{Train} \cup \overline{\underset{\sim}{Train}} = \max\{\mu_{Train}(x), \mu_{\overline{Train}}(x)\}$

$$= \left\{\frac{1.0}{train} + \frac{0.8}{bike} + \frac{0.6}{boat} + \frac{0.5}{plane} + \frac{0.8}{house}\right\}$$

(k) $\underset{\sim}{Train} \cap \overline{\underset{\sim}{Train}} = \min\{\mu_{Train}(x), \mu_{\overline{Train}}(x)\}$

$$= \left\{\frac{0}{train} + \frac{0.2}{bike} + \frac{0.4}{boat} + \frac{0.5}{plane} + \frac{0.2}{house}\right\}$$

6 For aircraft simulator data the termination of certain changes in its operating conditions is made on the basis of hard break points in the match region. We define two fuzzy sets $\underset{\sim}{A}$ and $\underset{\sim}{B}$ representing the condition of "near" a match number of 0.65 and "in the region" of a match number of 0.65 respectively, as follows.

$\underset{\sim}{A}$ = near mach 0.65

$$= \left\{\frac{0}{0.64} + \frac{0.75}{0.645} + \frac{1}{0.65} + \frac{0.5}{0.655} + \frac{0}{0.66}\right\}$$

$\underset{\sim}{B}$ = in the region of mach 0.65

$$= \left\{\frac{0}{0.64} + \frac{0.25}{0.645} + \frac{0.75}{0.65} + \frac{1}{0.655} + \frac{0.5}{0.66}\right\}$$

Find the following sets of operations

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

$$= \left\{\frac{0}{0.64} + \frac{0.75}{0.645} + \frac{1}{0.65} + \frac{1}{0.655} + \frac{0.5}{0.66}\right\}$$

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

$$= \left\{\frac{0}{0.64} + \frac{0.25}{0.645} + \frac{0.75}{0.65} + \frac{0.5}{0.655} + \frac{0}{0.66}\right\}$$

(c) $\bar{A} = 1 - \mu_{\underset{\sim}{A}}(x)$

$$= \left\{ \frac{1}{0.64} + \frac{0.25}{0.645} + \frac{0}{0.65} + \frac{0.5}{0.655} + \frac{1}{0.66} \right\}$$

(d) $\bar{B} = 1 - \mu_{\underset{\sim}{B}}(x)$.

$$= \left\{ \frac{1}{0.64} + \frac{0.75}{0.645} + \frac{0.25}{0.65} + \frac{0}{0.655} + \frac{0.5}{0.66} \right\}$$

(e) $\overline{\underset{\sim}{A} \cup \underset{\sim}{B}} = 1 - \max\{\mu_A(x), \mu_{\underset{\sim}{B}}(x)\}$

$$= \left\{ \frac{1}{0.64} + \frac{0.25}{0.645} + \frac{0}{0.65} + \frac{0}{0.655} + \frac{0.5}{0.66} \right\}$$

(f) $\overline{\underset{\sim}{A} \cap \underset{\sim}{B}} = 1 - \min\{\mu_{\underset{\sim}{A}}(x), \mu_{\underset{\sim}{B}}(x)\}$

$$= \left\{ \frac{1}{0.64} + \frac{0.75}{0.645} + \frac{0.25}{0.65} + \frac{0.5}{0.655} + \frac{1}{0.66} \right\}$$

7. For the two given fuzzy sets

$$\underset{\sim}{A} = \left\{ \frac{0.1}{0} + \frac{0.2}{1} + \frac{0.4}{2} + \frac{0.6}{3} + \frac{1}{4} \right\}$$

$$\underset{\sim}{B} = \left\{ \frac{1}{0} + \frac{0.5}{1} + \frac{0.7}{2} + \frac{0.3}{3} + \frac{0}{4} \right\}$$

Find the following

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

$$= \left\{ \frac{1}{0} + \frac{0.5}{1} + \frac{0.7}{2} + \frac{0.6}{3} + \frac{1}{4} \right\}$$

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

$$= \left\{ \frac{0.1}{0} + \frac{0.2}{1} + \frac{0.4}{2} + \frac{0.3}{3} + \frac{0}{4} \right\}$$

(c) $\bar{\underset{\sim}{A}} = 1 - \mu_{\underset{\sim}{A}}(x)$

$$= \left\{ \frac{0.9}{0} + \frac{0.8}{1} + \frac{0.6}{2} + \frac{0.4}{3} + \frac{0}{4} \right\}$$

(d) $\bar{\underset{\sim}{B}} = 1 - \mu_{\underset{\sim}{B}}(x)$

$$= \left\{ \frac{0}{0} + \frac{0.5}{1} + \frac{0.3}{2} + \frac{0.7}{3} + \frac{1}{4} \right\}$$

(e) $\underset{\sim}{A} \cup \bar{\underset{\sim}{A}} = \max\{\mu_{\underset{\sim}{A}}(x), \mu_{\bar{\underset{\sim}{A}}}(x)\}$

$$= \left\{ \frac{0.9}{0} + \frac{0.8}{1} + \frac{0.6}{2} + \frac{0.6}{3} + \frac{1}{4} \right\}$$

(f) $\underset{\sim}{A} \cap \bar{\underset{\sim}{A}} = \min\{\mu_{\underset{\sim}{A}}(x), \mu_{\bar{\underset{\sim}{A}}}(x)\}$

$$= \frac{0.1}{0} + \frac{0.2}{1} + \frac{0.4}{2} + \frac{0.4}{3} + \frac{0}{4} \}$$

(g) $\underset{\sim}{B} \cup \bar{\underset{\sim}{B}} = \max\{\mu_{\underset{\sim}{B}}(x), \mu_{\bar{\underset{\sim}{B}}}(x)\}$

$$= \left\{ \frac{1}{0} + \frac{0.5}{1} + \frac{0.7}{2} + \frac{0.7}{3} + \frac{1}{4} \right\}$$

(h) $\underset{\sim}{B} \cap \bar{\underset{\sim}{B}} = \min\{\mu_{\underset{\sim}{B}}(x), \mu_{\bar{\underset{\sim}{B}}}(x)\}$

$$= \left\{ \frac{0}{0} + \frac{0.5}{1} + \frac{0.3}{2} + \frac{0.3}{3} + \frac{0}{4} \right\}$$

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

$$= \left\{ \frac{0}{0} + \frac{0.2}{1} + \frac{0.3}{2} + \frac{0.6}{3} + \frac{1}{4} \right\}$$

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

$$= \left\{ \frac{0.1}{0} + \frac{0.5}{1} + \frac{0.4}{2} + \frac{0.7}{3} + \frac{1}{4} \right\}$$

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

$$= \left\{ \frac{0.9}{0} + \frac{0.5}{1} + \frac{0.6}{2} + \frac{0.3}{3} + \frac{0}{4} \right\}$$

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

$$= \left\{ \frac{1}{0} + \frac{0.8}{1} + \frac{0.7}{2} + \frac{0.4}{3} + \frac{0}{4} \right\}$$

(m) $\overline{\underset{\sim}{A} \cup \underset{\sim}{B}} = 1 - \max\{\mu_{\underset{\sim}{A}}(x), \mu_{\underset{\sim}{B}}(x)\}$

$$= \left\{ \frac{0}{0} + \frac{0.5}{1} + \frac{0.3}{2} + \frac{0.4}{3} + \frac{0}{4} \right\}$$

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

$$= \left\{\frac{0}{0} + \frac{0.5}{1} + \frac{0.3}{2} + \frac{0.4}{3} + \frac{0}{4}\right\}$$

8. Consider two fuzzy set.

$$\underset{\sim}{A} = \left\{\frac{0.2}{1} + \frac{0.3}{2} + \frac{0.4}{3} + \frac{0.5}{4}\right\}$$

$$\underset{\sim}{B} = \left\{\frac{0.1}{1} + \frac{0.2}{2} + \frac{0.2}{3} + \frac{1}{4}\right\}$$

Find the algebraic sum, algebraic product, bounded sum and bounded difference of the given fuzzy sets

(a) Algebraic sum

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

$$= \left\{\frac{0.3}{1} + \frac{0.5}{2} + \frac{0.6}{3} + \frac{0.5}{4}\right\} - \left\{\frac{0.02}{1} + \frac{0.06}{2} + \frac{0.08}{3} + \frac{0.5}{4}\right\}$$

$$= \left\{\frac{0.28}{1} + \frac{0.49}{2} + \frac{0.52}{3} + \frac{0}{4}\right\}$$

(b) Algebraic product

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

$$= \left\{\frac{0.02}{1} + \frac{0.06}{2} + \frac{0.08}{3} + \frac{0.5}{4}\right\}$$

(c) Bounded sum

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

$$= \min\left\{1, \left\{\frac{0.3}{1} + \frac{0.5}{2} + \frac{0.6}{3} + \frac{0.5}{4}\right\}\right\}$$

$$= \left\{\frac{0.3}{1} + \frac{0.5}{2} + \frac{0.6}{3} + \frac{0.5}{4}\right\}$$

(d) Bounded difference

$$\mu_{A \ominus B}(x) = \max[0, \mu_A(x) - \mu_B(x)]$$

$$= \max\left\{0, \left\{\frac{0.1}{1} + \frac{0.1}{2} + \frac{0.2}{3} + \frac{0.5}{4}\right\}\right\}$$

$$= \left\{\frac{0.1}{1} + \frac{0.1}{2} + \frac{0.2}{3} + \frac{0.5}{4}\right\}$$

The max-product composition of $\underset{\sim}{R}(X,Y)$ and $\underset{\sim}{S}(Y,Z)$ denoted as $\underset{\sim}{R}(X,Y) \cdot \underset{\sim}{S}(Y,Z)$ is defined as $\underset{\sim}{T}(X,Z)$ as

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

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

The properties of fuzzy composition can be given as

$$\underset{\sim}{R} \circ \underset{\sim}{S} \neq \underset{\sim}{S} \circ \underset{\sim}{R}$$

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

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

Problems

1. Find the power set and cardinality of the given set $X = \{2,4,6\}$. Also find the cardinality of power set.

Solution :-

Since set X contains three elements, so its cardinal number is $n_X = 3$

The power set of X is given by

$$P(X) = \{\phi, \{2\}, \{4\}, \{6\}, \{2,4\}, \{4,6\}, \{2,6\}, \{2,4,6\}\}$$

The cardinality of power set $P(X)$ denoted by $n_{P(X)}$.

$$n_{P(X)} = 2^{n_X} = 2^3 = \underline{8}$$

2. Consider two fuzzy sets.

$$\underset{\sim}{A} = \left\{\frac{1}{2} + \frac{0.3}{4} + \frac{0.5}{6} + \frac{0.2}{8}\right\}$$

$$\underset{\sim}{B} = \left\{\frac{0.5}{2} + \frac{0.4}{4} + \frac{0.1}{6} + \frac{1}{8}\right\}$$

Find $\underset{\sim}{A} \cup \underset{\sim}{B}$, $\underset{\sim}{A} \cap \underset{\sim}{B}$, $\bar{\underset{\sim}{A}}$, $\bar{\underset{\sim}{B}}$, $\underset{\sim}{A}/\underset{\sim}{B}$, $\underset{\sim}{B}/\underset{\sim}{A}$

$\underset{\sim}{A} \cup \underset{\sim}{B} = \max\{\mu_A(x), \mu_B(x)\}$

$= \left\{\frac{1}{2} + \frac{0.4}{4} + \frac{0.5}{8} + \frac{1}{8}\right\}$

$\underset{\sim}{A} \cap \underset{\sim}{B} = \min\{\mu_A(x), \mu_B(x)\}$

$= \left\{\frac{0.5}{2} + \frac{0.3}{4} + \frac{0.1}{6} + \frac{0.2}{8}\right\}$

$\overline{\underset{\sim}{A}} = 1 - \mu_A(x) = \left\{\frac{0}{2} + \frac{0.7}{4} + \frac{0.5}{6} + \frac{0.8}{8}\right\}$

$\overline{\underset{\sim}{B}} = 1 - \mu_B(x) = \left\{\frac{0.5}{2} + \frac{0.6}{4} + \frac{0.9}{6} + \frac{0}{8}\right\}$

$\underset{\sim}{A} / \underset{\sim}{B} = \underset{\sim}{A} \cap \overline{\underset{\sim}{B}} = \left\{\frac{0.5}{2} + \frac{0.3}{4} + \frac{0.5}{6} + \frac{0}{8}\right\}$

$\underset{\sim}{B} / \underset{\sim}{A} = \underset{\sim}{B} \cap \overline{\underset{\sim}{A}} = \left\{\frac{0}{2} + \frac{0.4}{4} + \frac{0.1}{6} + \frac{0.8}{8}\right\}$

2. For the following two fuzzy sets

$\underset{\sim}{B_1} = \left\{\frac{1}{1.0} + \frac{0.75}{1.5} + \frac{0.3}{2.0} + \frac{0.15}{2.5} + \frac{0}{3.0}\right\}$

$\underset{\sim}{B_2} = \left\{\frac{1}{1.0} + \frac{0.6}{1.5} + \frac{0.2}{2.0} + \frac{0.1}{2.5} + \frac{0}{3.0}\right\}$

Find $\underset{\sim}{B_1} \cup \underset{\sim}{B_2}$, $\underset{\sim}{B_1} \cap \underset{\sim}{B_2}$, $\overline{\underset{\sim}{B_1}}$, $\overline{\underset{\sim}{B_2}}$, $\underset{\sim}{B_1} / \underset{\sim}{B_2}$, $\overline{\underset{\sim}{B_1} \cup \underset{\sim}{B_2}}$, $\overline{\underset{\sim}{B_1} \cap \underset{\sim}{B_2}}$,
$\underset{\sim}{B_1} \cap \overline{\underset{\sim}{B_1}}$, $\underset{\sim}{B_1} \cup \overline{\underset{\sim}{B_2}}$, $\underset{\sim}{B_2} \cap \overline{\underset{\sim}{B_2}}$, $\underset{\sim}{B_2} \cup \overline{\underset{\sim}{B_2}}$

Solution :-

$\underset{\sim}{B_1} \cup \underset{\sim}{B_2} = \left\{\frac{1}{1.0} + \frac{0.75}{1.5} + \frac{0.3}{2.0} + \frac{0.15}{2.5} + \frac{0}{3.0}\right\}$

$\underset{\sim}{B_1} \cap \underset{\sim}{B_2} = \left\{\frac{1}{1.0} + \frac{0.6}{1.5} + \frac{0.2}{2.0} + \frac{0.1}{2.5} + \frac{0}{3.0}\right\}$

$\overline{\underset{\sim}{B_1}} = \left\{\frac{0}{1.0} + \frac{0.25}{1.5} + \frac{0.7}{2.0} + \frac{0.85}{2.5} + \frac{1}{3.0}\right\}$

$\overline{\underset{\sim}{B_2}} = \left\{\frac{0}{1.0} + \frac{0.4}{1.5} + \frac{0.8}{2.0} + \frac{0.9}{2.5} + \frac{1}{3.0}\right\}$

$\underset{\sim}{B_1}/\underset{\sim}{B_2} = \underset{\sim}{B_1} \cap \overline{\underset{\sim}{B_2}}$

$= \left\{ \frac{0}{1.0} + \frac{0.4}{1.5} + \frac{0.3}{2.0} + \frac{0.15}{2.15} + \frac{0}{3.0} \right\}$

$\overline{\underset{\sim}{B_1} \cup \underset{\sim}{B_2}} = \left\{ \frac{0}{1.0} + \frac{0.25}{1.5} + \frac{0.7}{2.0} + \frac{0.85}{2.5} + \frac{1}{3.0} \right\}$

$\overline{\underset{\sim}{B_1} \cap \underset{\sim}{B_2}} = \left\{ \frac{0}{1.0} + \frac{0.4}{1.5} + \frac{0.8}{2.0} + \frac{0.9}{2.5} + \frac{1}{3.0} \right\}$

$\underset{\sim}{B_1} \cap \overline{\underset{\sim}{B_1}} = \left\{ \frac{0}{1.0} + \frac{0.25}{1.5} + \frac{0.3}{2.0} + \frac{0.15}{2.5} + \frac{1}{3.0} \right\}$

$\underset{\sim}{B_2} \cap \overline{\underset{\sim}{B_2}} = \left\{ \frac{0}{1.0} + \frac{0.4}{1.5} + \frac{0.2}{2.0} + \frac{0.1}{2.5} + \frac{0}{3.0} \right\}$

$\underset{\sim}{B_2} \cup \overline{\underset{\sim}{B_2}} = \left\{ \frac{1}{1.0} + \frac{0.6}{1.5} + \frac{0.8}{2.0} + \frac{0.9}{2.5} + \frac{1}{3.0} \right\}$

4. For the following two fuzzy sets

$\underset{\sim}{D_1} = \left\{ \frac{0}{0} + \frac{0.2}{10} + \frac{0.35}{20} + \frac{0.65}{30} + \frac{0.85}{40} + \frac{1}{50} \right\}$

$\underset{\sim}{D_2} = \left\{ \frac{0}{0} + \frac{0.35}{10} + \frac{0.25}{20} + \frac{0.8}{30} + \frac{0.95}{40} + \frac{1}{50} \right\}$

Find the following membership function

(a) $\mu_{\underset{\sim}{D_1} \cup \underset{\sim}{D_2}}$ (b) $\mu_{\underset{\sim}{D_1} \cap \underset{\sim}{D_2}}(x)$ (c) $\mu_{\overline{\underset{\sim}{D_1}}}(x)$ (d) $\mu_{\overline{\underset{\sim}{D_2}}}(x)$

(e) $\mu_{\underset{\sim}{D_1} \cup \overline{\underset{\sim}{D_1}}}(x)$ (f) $\mu_{\underset{\sim}{D_1} \cap \overline{\underset{\sim}{D_1}}}(x)$ (g) $\mu_{\underset{\sim}{D_2} \cup \overline{\underset{\sim}{D_2}}}$ (h) $\mu_{\underset{\sim}{D_2} \cap \overline{\underset{\sim}{D_2}}}(x)$

(i) $\mu_{\underset{\sim}{D_1} / \underset{\sim}{D_2}}(x)$ (j) $\mu_{\underset{\sim}{D_2} / \underset{\sim}{D_1}}(x)$.

Solution :-

(a) $\mu_{\underset{\sim}{D_1} \cup \underset{\sim}{D_2}}(x) = \max\{\mu_{\underset{\sim}{D_1}}(x), \mu_{\underset{\sim}{D_2}}(x)\}$

$= \left\{ \frac{0}{0} + \frac{0.35}{10} + \frac{0.35}{20} + \frac{0.8}{30} + \frac{0.95}{40} + \frac{1}{50} \right\}$

(b) $\mu_{\underset{\sim}{D_1} \cap \underset{\sim}{D_2}}(x) = \min\{\mu_{\underset{\sim}{D_1}}(x), \mu_{\underset{\sim}{D_2}}(x)\}$

$= \left\{ \frac{0}{0} + \frac{0.2}{10} + \frac{0.25}{20} + \frac{0.65}{30} + \frac{0.85}{40} + \frac{1}{50} \right\}$

(c) $\mu_{\overline{\underset{\sim}{D_1}}}(x) = 1 - \mu_{\underset{\sim}{D_1}}(x)$

$= \left\{ \frac{1}{0} + \frac{0.8}{10} + \frac{0.65}{20} + \frac{0.35}{30} + \frac{0.15}{40} + \frac{0}{50} \right\}$

(d) $\mu_{\overline{\underset{\sim}{D_2}}}(x) = 1 - \mu_{\underset{\sim}{D_2}}(x)$

$$= \left\{ \frac{1}{0} + \frac{0.65}{10} + \frac{0.75}{20} + \frac{0.2}{30} + \frac{0.05}{40} + \frac{0}{50} \right\}$$

(e) $\mu_{\underset{\sim}{D_1} \cup \overline{D_1}}(x) = \max\{\mu_{\underset{\sim}{D_1}}(x), \mu_{\overline{\underset{\sim}{D_1}}}(x)\}$

$$= \left\{ \frac{1}{0} + \frac{0.8}{10} + \frac{0.65}{20} + \frac{0.65}{30} + \frac{0.85}{40} + \frac{1}{50} \right\}$$

(f) $\mu_{\underset{\sim}{D_1} \cap \overline{D_1}}(x) = \min\{\mu_{\underset{\sim}{D_1}}(x), \mu_{\overline{\underset{\sim}{D_1}}}(x)\}$

$$= \left\{ \frac{0}{0} + \frac{0.2}{10} + \frac{0.35}{20} + \frac{0.35}{30} + \frac{0.15}{40} + \frac{0}{50} \right\}$$

(g) $\mu_{\underset{\sim}{D_2} \cup \overline{D_2}}(x) = \max\{\mu_{\underset{\sim}{D_2}}(x), \mu_{\overline{\underset{\sim}{D_2}}}(x)\}$

$$= \left\{ \frac{1}{0} + \frac{0.65}{10} + \frac{0.75}{20} + \frac{0.8}{30} + \frac{0.95}{40} + \frac{1}{50} \right\}$$

(h) $\mu_{\underset{\sim}{D_2} \cap \overline{\underset{\sim}{D_2}}}(x) = \min\{\mu_{\underset{\sim}{D_2}}(x), \mu_{\overline{\underset{\sim}{D_2}}}(x)\}$

$$= \left\{ \frac{0}{0} + \frac{0.35}{10} + \frac{0.25}{20} + \frac{0.2}{30} + \frac{0.05}{40} + \frac{0}{50} \right\}$$

(i) $\mu_{\underset{\sim}{D_1} | \underset{\sim}{D_2}}(x) = \mu_{\underset{\sim}{D_1} \cap \overline{\underset{\sim}{D_2}}}(x) = \min\{\mu_{\underset{\sim}{D_1}}(x), \mu_{\overline{\underset{\sim}{D_2}}}(x)\}$

$$= \left\{ \frac{0}{0} + \frac{0.2}{10} + \frac{0.35}{20} + \frac{0.2}{30} + \frac{0.05}{40} + \frac{0}{50} \right\}$$

(j) $\mu_{\underset{\sim}{D_2} / \underset{\sim}{D_1}}(x) = \mu_{\underset{\sim}{D_2} \cap \overline{\underset{\sim}{D_1}}}(x) = \min\{\mu_{\underset{\sim}{D_2}}(x), \mu_{\overline{\underset{\sim}{D_1}}}(x)\}$

$$= \frac{0}{0} + \frac{0.35}{10} + \frac{0.25}{20} + \frac{0.35}{30} + \frac{0.15}{40} + \frac{0}{50} \}$$

5. Design a computer software to perform image processing to locate objects within a scene. The two fuzzy sets representing a plane and a train image are

$$\text{plane} = \left\{ \frac{0.2}{\text{train}} + \frac{0.5}{\text{bike}} + \frac{0.3}{\text{boat}} + \frac{0.8}{\text{plane}} + \frac{0.1}{\text{house}} \right\}$$

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

Find the following operations

(a) $\underset{\sim}{\text{plane}} \cup \underset{\sim}{\text{Train}} = \max\{\mu_{plane}(x), \mu_{Train}(x)\}$

$$= \left\{ \frac{1.0}{\text{train}} + \frac{0.5}{\text{bike}} + \frac{0.4}{\text{boat}} + \frac{0.8}{\text{plane}} + \frac{0.2}{\text{house}} \right\}$$

(b) $\underset{\sim}{\text{plane}} \cap \underset{\sim}{\text{Train}} = \min\{\mu_{plane}(x), \mu_{Train}(x)\}$

$$= \left\{ \frac{0.2}{\text{train}} + \frac{0.2}{\text{bike}} + \frac{0.3}{\text{boat}} + \frac{0.5}{\text{plane}} + \frac{0.1}{\text{house}} \right\}$$

(c) $\underset{\sim}{\text{plane}} = 1 - \mu_{\underset{\sim}{plane}}(x)$

$$= \left\{ \frac{0.8}{\text{train}} + \frac{0.5}{\text{bike}} + \frac{0.7}{\text{boat}} + \frac{0.2}{\text{plane}} + \frac{0.9}{\text{house}} \right\}$$

(d) $\overline{\underset{\sim}{\text{Train}}} = 1 - \mu_{Train}(x)$

$$= \left\{ \frac{0}{\text{train}} + \frac{0.8}{\text{bike}} + \frac{0.6}{\text{boat}} + \frac{0.5}{\text{plane}} + \frac{0.8}{\text{house}} \right\}$$

(e) $\underset{\sim}{\text{plane}} \mid \underset{\sim}{\text{Train}} = \underset{\sim}{\text{plane}} \cap \overline{\underset{\sim}{\text{Train}}}$

$= \min\{\mu_{plane}(x), \mu$

(f) $\overline{\underset{\sim}{\text{plane}} \cup \underset{\sim}{\text{Train}}} = 1 - \max\{\mu_{\underset{\sim}{plane}}(x), \mu_{\underset{\sim}{Train}}(x)\}$

$$= \left\{ \frac{0}{\text{train}} + \frac{0.5}{\text{bike}} + \frac{0.6}{\text{boat}} + \frac{0.2}{\text{plane}} + \frac{0.8}{\text{house}} \right\}$$

(g) $\overline{\underset{\sim}{\text{plane}} \cap \underset{\sim}{\text{Train}}} = 1 - \min\{\mu_{\underset{\sim}{plane}}(x), \mu_{\underset{\sim}{Train}}(x)\}$

$$= \left\{ \frac{0.8}{\text{train}} + \frac{0.8}{\text{bike}} + \frac{0.7}{\text{boat}} + \frac{0.5}{\text{plane}} + \frac{0.9}{\text{house}} \right\}$$

(h) $\underset{\sim}{\text{plane}} \cup \overline{\underset{\sim}{\text{plane}}} = \max\{\mu_{\underset{\sim}{plane}}(x), \mu_{\overline{\underset{\sim}{plane}}}(x)\}$

$$= \left\{ \frac{0.8}{\text{train}} + \frac{0.5}{\text{bike}} + \frac{0.7}{\text{boat}} + \frac{0.8}{\text{plane}} + \frac{0.9}{\text{house}} \right\}$$

(i) $\underset{\sim}{\text{plane}} \cap \overline{\underset{\sim}{\text{plane}}} = \min\{\mu_{\underset{\sim}{plane}}(x), \mu_{\overline{\underset{\sim}{plane}}}(x)\}$

$$= \left\{ \frac{0.2}{\text{train}} + \frac{0.5}{\text{bike}} + \frac{0.3}{\text{boat}} + \frac{0.2}{\text{plane}} + \frac{0.9}{\text{house}} \right\}$$

(j) $\underset{\sim}{Train} \cup \overline{\underset{\sim}{Train}} = \max\{\mu_{\underset{\sim}{Train}}(x), \mu_{\overline{\underset{\sim}{Train}}}(x)\}$

$$= \left\{\frac{1.0}{train} + \frac{0.8}{bike} + \frac{0.6}{boat} + \frac{0.5}{plane} + \frac{0.8}{house}\right\}$$

(k) $\underset{\sim}{Train} \cap \overline{\underset{\sim}{Train}} = \min\{\mu_{\underset{\sim}{Train}}(x), \mu_{\overline{\underset{\sim}{Train}}}(x)\}$

$$= \left\{\frac{0}{train} + \frac{0.2}{bike} + \frac{0.4}{boat} + \frac{0.5}{plane} + \frac{0.2}{house}\right\}$$

6 For aircraft simulator data the termination of certain changes in its operating conditions is made on the basis of hard break points in the match region. We define two fuzzy sets $\underset{\sim}{A}$ and $\underset{\sim}{B}$ representing the condition of "near" a match number of 0.65 and "in the region" of a match number of 0.65 respectively, as follows.

$\underset{\sim}{A}$ = near mach 0.65

$$= \left\{\frac{0}{0.64} + \frac{0.75}{0.645} + \frac{1}{0.65} + \frac{0.5}{0.655} + \frac{0}{0.66}\right\}$$

$\underset{\sim}{B}$ = in the region of mach 0.65

$$= \left\{\frac{0}{0.64} + \frac{0.25}{0.645} + \frac{0.75}{0.65} + \frac{1}{0.655} + \frac{0.5}{0.66}\right\}$$

Find the following sets of operations

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

$$= \left\{\frac{0}{0.64} + \frac{0.75}{0.645} + \frac{1}{0.65} + \frac{1}{0.655} + \frac{0.5}{0.66}\right\}$$

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

$$= \left\{\frac{0}{0.64} + \frac{0.25}{0.645} + \frac{0.75}{0.65} + \frac{0.5}{0.655} + \frac{0}{0.66}\right\}$$

(c) $\bar{A} = 1 - \mu_{\underset{\sim}{A}}(x)$

$$= \left\{ \frac{1}{0.64} + \frac{0.25}{0.645} + \frac{0}{0.65} + \frac{0.5}{0.655} + \frac{1}{0.66} \right\}$$

(d) $\bar{B} = 1 - \mu_{\underset{\sim}{B}}(x)$.

$$= \left\{ \frac{1}{0.64} + \frac{0.75}{0.645} + \frac{0.25}{0.65} + \frac{0}{0.655} + \frac{0.5}{0.66} \right\}$$

(e) $\overline{\underset{\sim}{A} \cup \underset{\sim}{B}} = 1 - \max\left\{ \mu_A(x), \mu_{\underset{\sim}{B}}(x) \right\}$

$$= \left\{ \frac{1}{0.64} + \frac{0.25}{0.645} + \frac{0}{0.65} + \frac{0}{0.655} + \frac{0.5}{0.66} \right\}$$

(f) $\overline{\underset{\sim}{A} \cap \underset{\sim}{B}} = 1 - \min\left\{ \mu_{\underset{\sim}{A}}(x), \mu_{\underset{\sim}{B}}(x) \right\}$

$$= \left\{ \frac{1}{0.64} + \frac{0.75}{0.645} + \frac{0.25}{0.65} + \frac{0.5}{0.655} + \frac{1}{0.66} \right\}$$

7. For the two given fuzzy sets

$$\underset{\sim}{A} = \left\{ \frac{0.1}{0} + \frac{0.2}{1} + \frac{0.4}{2} + \frac{0.6}{3} + \frac{1}{4} \right\}$$

$$\underset{\sim}{B} = \left\{ \frac{1}{0} + \frac{0.5}{1} + \frac{0.7}{2} + \frac{0.3}{3} + \frac{0}{4} \right\}$$

Find the following

(a) $\underset{\sim}{A} \cup \underset{\sim}{B} = \max\left\{ \mu_{\underset{\sim}{A}}(x), \mu_{\underset{\sim}{B}}(x) \right\}$

$$= \left\{ \frac{1}{0} + \frac{0.5}{1} + \frac{0.7}{2} + \frac{0.6}{3} + \frac{1}{4} \right\}$$

(b) $\underset{\sim}{A} \cap \underset{\sim}{B} = \min\left\{ \mu_{\underset{\sim}{A}}(x), \mu_{\underset{\sim}{B}}(x) \right\}$

$$= \left\{ \frac{0.1}{0} + \frac{0.2}{1} + \frac{0.4}{2} + \frac{0.3}{3} + \frac{0}{4} \right\}$$

(c) $\overline{\underset{\sim}{A}} = 1 - \mu_{\underset{\sim}{A}}(x)$

$$= \left\{ \frac{0.9}{0} + \frac{0.8}{1} + \frac{0.6}{2} + \frac{0.4}{3} + \frac{0}{4} \right\}$$

(d) $\bar{\underset{\sim}{B}} = 1 - \mu_{\underset{\sim}{B}}(x)$

$$= \left\{ \frac{0}{0} + \frac{0.5}{1} + \frac{0.3}{2} + \frac{0.7}{3} + \frac{1}{4} \right\}$$

(e) $\underset{\sim}{A} \cup \bar{\underset{\sim}{A}} = \max\{\mu_{\underset{\sim}{A}}(x), \mu_{\bar{\underset{\sim}{A}}}(x)\}$

$$= \left\{ \frac{0.9}{0} + \frac{0.8}{1} + \frac{0.6}{2} + \frac{0.6}{3} + \frac{1}{4} \right\}$$

(f) $\underset{\sim}{A} \cap \bar{\underset{\sim}{A}} = \min\{\mu_{\underset{\sim}{A}}(x), \mu_{\bar{\underset{\sim}{A}}}(x)\}$

$$= \frac{0.1}{0} + \frac{0.2}{1} + \frac{0.4}{2} + \frac{0.4}{3} + \frac{0}{4} \}$$

(g) $\underset{\sim}{B} \cup \bar{\underset{\sim}{B}} = \max\{\mu_{\underset{\sim}{B}}(x), \mu_{\bar{\underset{\sim}{B}}}(x)\}$

$$= \left\{ \frac{1}{0} + \frac{0.5}{1} + \frac{0.7}{2} + \frac{0.7}{3} + \frac{1}{4} \right\}$$

(h) $\underset{\sim}{B} \cap \bar{\underset{\sim}{B}} = \min\{\mu_{\underset{\sim}{B}}(x), \mu_{\bar{\underset{\sim}{B}}}(x)\}$

$$= \left\{ \frac{0}{0} + \frac{0.5}{1} + \frac{0.7}{2} + \frac{0.7}{3} + \frac{0}{4} \right\}$$

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

$$= \left\{ \frac{0}{0} + \frac{0.2}{1} + \frac{0.3}{2} + \frac{0.6}{3} + \frac{1}{4} \right\}$$

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

$$= \left\{ \frac{0.1}{0} + \frac{0.5}{1} + \frac{0.4}{2} + \frac{0.7}{3} + \frac{1}{4} \right\}$$

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

$$= \left\{ \frac{0.9}{0} + \frac{0.5}{1} + \frac{0.6}{2} + \frac{0.3}{3} + \frac{0}{4} \right\}$$

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

$$= \left\{ \frac{1}{0} + \frac{0.8}{1} + \frac{0.7}{2} + \frac{0.4}{3} + \frac{0}{4} \right\}$$

(m) $\overline{\underset{\sim}{A} \cup \underset{\sim}{B}} = 1 - \max\{\mu_{\underset{\sim}{A}}(x), \mu_{\underset{\sim}{B}}(x)\}$

$$= \left\{ \frac{0}{0} + \frac{0.5}{1} + \frac{0.3}{2} + \frac{0.4}{3} + \frac{0}{4} \right\}$$

(n) $\bar{A} \cap \bar{B} = \min\{\mu_{\bar{A}}(x), \mu_{\bar{B}}(x)\}$

$$= \left\{\frac{0}{0} + \frac{0.5}{1} + \frac{0.3}{2} + \frac{0.4}{3} + \frac{0}{4}\right\}$$

8. Consider two fuzzy set.

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

$$B = \left\{\frac{0.1}{1} + \frac{0.2}{2} + \frac{0.2}{3} + \frac{1}{4}\right\}$$

Find the algebraic sum, algebraic product, bounded sum and bounded difference of the given fuzzy sets

(a) Algebraic sum

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

$$= \left\{\frac{0.3}{1} + \frac{0.5}{2} + \frac{0.6}{3} + \frac{0.5}{4}\right\} - \left\{\frac{0.02}{1} + \frac{0.06}{2} + \frac{0.08}{3} + \frac{0.5}{4}\right\}$$

$$= \left\{\frac{0.28}{1} + \frac{0.44}{2} + \frac{0.52}{3} + \frac{0}{4}\right\}$$

(b) Algebraic product

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

$$= \left\{\frac{0.02}{1} + \frac{0.06}{2} + \frac{0.08}{3} + \frac{0.5}{4}\right\}$$

(c) Bounded sum

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

$$= \min\left\{1, \left\{\frac{0.3}{1} + \frac{0.5}{2} + \frac{0.6}{3} + \frac{0.5}{4}\right\}\right\}$$

$$= \left\{\frac{0.3}{1} + \frac{0.5}{2} + \frac{0.6}{3} + \frac{0.5}{4}\right\}$$

(d) Bounded difference

$$\mu_{A \odot B}(x) = \max[0, \mu_A(x) - \mu_B(x)]$$

$$= \max\left\{0, \left\{\frac{0.1}{1} + \frac{0.1}{2} + \frac{0.2}{3} + \frac{0.5}{4}\right\}\right\}$$

$$= \left\{\frac{0.1}{1} + \frac{0.1}{2} + \frac{0.2}{3} + \frac{0.5}{4}\right\}$$

9. Let U be the Universe of military aircraft of interest as defined below.

$$U = \{a10, b52, c130, f2, f9\}$$

Let $\underset{\sim}{A}$ be the fuzzy set of bomber class aircraft

$$\underset{\sim}{A} = \left\{ \frac{0.3}{a10} + \frac{0.4}{b52} + \frac{0.2}{c130} + \frac{0.1}{f2} + \frac{1}{f9} \right\}$$

Let $\underset{\sim}{B}$ be the fuzzy set of fighter class aircraft

$$\underset{\sim}{B} = \left\{ \frac{0.1}{a10} + \frac{0.2}{b52} + \frac{0.8}{c130} + \frac{0.7}{f2} + \frac{0}{f9} \right\}$$

Find the following

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

$$= \left\{ \frac{0.3}{a10} + \frac{0.4}{b52} + \frac{0.8}{c130} + \frac{0.7}{f2} + \frac{1}{f9} \right\}$$

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

$$= \left\{ \frac{0.1}{a10} + \frac{0.2}{b52} + \frac{0.2}{c130} + \frac{0.1}{f2} + \frac{0}{f9} \right\}$$

(c) $\overline{\underset{\sim}{A}} = 1 - \mu_{\underset{\sim}{A}}(x)$

$$= \left\{ \frac{0.7}{a10} + \frac{0.6}{b52} + \frac{0.8}{c130} + \frac{0.9}{f2} + \frac{0}{f9} \right\}$$

(d) $\overline{\underset{\sim}{B}} = 1 - \mu_{\underset{\sim}{B}}(x)$

$$= \left\{ \frac{0.9}{a10} + \frac{0.8}{b52} + \frac{0.2}{c130} + \frac{0.3}{f2} + \frac{1}{f9} \right\}$$

(e) $\underset{\sim}{A} / \underset{\sim}{B} = \underset{\sim}{A} \cap \overline{\underset{\sim}{B}} = \min\{\mu_{\underset{\sim}{A}}(x), \mu_{\overline{\underset{\sim}{B}}}(x)\}$

$$= \left\{ \frac{0.3}{a10} + \frac{0.4}{b52} + \frac{0.2}{c130} + \frac{0.1}{f2} + \frac{1}{f9} \right\}$$

(f) $\underset{\sim}{B} / \underset{\sim}{A} = \underset{\sim}{B} \cap \overline{\underset{\sim}{A}} = \min\{\mu_{\underset{\sim}{B}}(x), \mu_{\overline{\underset{\sim}{A}}}(x)\}$

$$= \left\{ \frac{0.1}{a10} + \frac{0.2}{b52} + \frac{0.8}{c130} + \frac{0.7}{f2} + \frac{0}{f9} \right\}$$

(g) $\overline{\underset{\sim}{A} \cup \underset{\sim}{B}} = 1 - \max\{\mu_{\underset{\sim}{A}}(x), \mu_{\underset{\sim}{B}}(x)\}$

$$= \left\{\frac{0.7}{a10} + \frac{0.6}{b52} + \frac{0.2}{C130} + \frac{0.3}{f2} + \frac{0}{f9}\right\}$$

(h) $\overline{\underset{\sim}{A} \cap \underset{\sim}{B}} = 1 - \min\{\mu_{\underset{\sim}{A}}(x), \mu_{\underset{\sim}{B}}(x)\}$

$$= \left\{\frac{0.9}{a10} + \frac{0.8}{b52} + \frac{0.8}{C130} + \frac{0.9}{f2} + \frac{1}{f9}\right\}$$

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

$$= \left\{\frac{0.9}{a10} + \frac{0.8}{b52} + \frac{0.8}{C130} + \frac{0.9}{f2} + \frac{1}{f9}\right\}$$

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

$$= \left\{\frac{0.9}{a10} + \frac{0.8}{b52} + \frac{0.2}{C130} + \frac{0.3}{f2} + \frac{1}{f9}\right\}$$

10. The discretized membership functions for a transistor and a Registor are given below

$$\mu_{\underset{\sim}{T}} = \left\{\frac{0}{0} + \frac{0.2}{1} + \frac{0.7}{2} + \frac{0.8}{3} + \frac{0.9}{4} + \frac{1}{5}\right\}$$

$$\mu_{B} = \left\{\frac{0}{0} + \frac{0.1}{1} + \frac{0.3}{2} + \frac{0.2}{3} + \frac{0.4}{4} + \frac{0.5}{5}\right\}$$

Find the following

(a) Algebraic Sum.

$$\mu_{\underset{\sim}{T}+B}(x) = [\mu_T(x) + \mu_R(x)] - [\mu_T(x) \cdot \mu_R(x)]$$

$$= \left\{\frac{0}{0} + \frac{0.3}{1} + \frac{1.0}{2} + \frac{1.0}{3} + \frac{1.3}{4} + \frac{1.5}{5}\right\} -$$

$$\left\{\frac{0}{0} + \frac{0.02}{1} + \frac{0.21}{2} + \frac{0.16}{3} + \frac{0.36}{4} + \frac{0.5}{5}\right\}$$

$$= \left\{\frac{0}{0} + \frac{0.28}{1} + \frac{0.79}{2} + \frac{0.84}{3} + \frac{0.94}{4} + \frac{1}{5}\right\}$$

(b) Algebraic product.

$$\mu_{\underset{\sim}{I} \cdot \underset{\sim}{R}}(x) = \mu_{I}(x) \cdot \mu_{R}(x)$$

$$= \left\{ \frac{0}{0} + \frac{0.02}{1} + \frac{0.21}{2} + \frac{0.16}{3} + \frac{0.36}{4} + \frac{0.5}{5} \right\}$$

(c) Bounded sum

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

$$= \min\left\{1, \left\{\frac{0}{0} + \frac{0.3}{1} + \frac{1.0}{2} + \frac{1.0}{3} + \frac{1.3}{4} + \frac{1.5}{5}\right\}\right\}$$

$$= \left\{\frac{0}{0} + \frac{0.3}{1} + \frac{1.0}{2} + \frac{1.0}{3} + \frac{1.0}{4} + \frac{1.0}{5}\right\}$$

(d) Bounded difference

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

$$= \max\left\{0, \left\{\frac{0}{0} + \frac{0.1}{1} + \frac{0.4}{2} + \frac{0.6}{3} + \frac{0.5}{4} + \frac{0.5}{5}\right\}\right\}$$

$$= \left\{\frac{0}{0} + \frac{0.1}{1} + \frac{0.4}{2} + \frac{0.6}{3} + \frac{0.5}{4} + \frac{0.5}{5}\right\}$$

11. The elements in two sets A and B are given as

$A = \{2, 4\}$ and $B = \{a, b, c\}$.

Find the various Cartesian products of these two sets.

Solution:-

The various Cartesian products of these two given sets are

$A \times B = \{(2,a), (2,b), (2,c), (4,a), (4,b), (4,c)\}$

$B \times A = \{(a,2), (a,4), (b,2), (b,4), (c,2), (c,4)\}$

$A \times A = A^2 = \{(2,2), (2,4), (4,2), (4,4)\}$

$B \times B = B^2 = \{(a,a), (a,b), (a,c), (b,a), (b,b), (b,c),$
$(c,a), (c,b), (c,c)\}$

12. Consider the following two fuzzy sets.

$$\underset{\sim}{A} = \left\{\frac{0.3}{x_1} + \frac{0.7}{x_2} + \frac{1}{x_3}\right\} \text{ and } \underset{\sim}{B} = \left\{\frac{0.4}{y_1} + \frac{0.9}{y_2}\right\}$$

perform the Cartesian product over these fuzzy sets.

Solution: The fuzzy relation $\underset{\sim}{R}$ given by $\underset{\sim}{R} = \underset{\sim}{A} \times \underset{\sim}{B}$.

$$\underset{\sim}{R} = \begin{bmatrix} 0.3 & 0.3 \\ 0.4 & 0.7 \\ 0.4 & 0.9 \end{bmatrix}$$

The calculation for $\underset{\sim}{R}$ is as follows:

$$\mu_{\underset{\sim}{R}}(x_1, y_1) = \min[\mu_{\underset{\sim}{A}}(x_1), \mu_{\underset{\sim}{B}}(y_1)]$$
$$= \min(0.3, 0.4) = 0.3$$

$$\mu_{\underset{\sim}{R}}(x_1, y_2) = \min[\mu_{\underset{\sim}{A}}(x_1), \mu_{\underset{\sim}{B}}(y_2)]$$
$$= \min(0.3, 0.9) = 0.3$$

$$\mu_{\underset{\sim}{R}}(x_2, y_1) = \min[\mu_{\underset{\sim}{A}}(x_2), \mu_{\underset{\sim}{B}}(y_1)]$$
$$= \min(0.7, 0.4) = 0.4$$

$$\mu_{\underset{\sim}{R}}(x_2, y_2) = \min[\mu_{\underset{\sim}{A}}(x_2), \mu_{\underset{\sim}{B}}(y_2)]$$
$$= \min(0.7, 0.9) = 0.7$$

$$\mu_{\underset{\sim}{R}}(x_3, y_1) = \min[\mu_{\underset{\sim}{A}}(x_3), \mu_{\underset{\sim}{B}}(y_1)]$$
$$= \min(1, 0.4) = 0.4$$

$$\mu_{\underset{\sim}{R}}(x_3, y_2) = \min[\mu_{\underset{\sim}{A}}(x_3), \mu_{\underset{\sim}{B}}(y_2)]$$
$$= \min(1, 0.9) = 0.9$$

12. Two fuzzy relations are given by

$$R = \begin{array}{c c} & \begin{array}{cc} y_1 & y_2 \end{array} \\ \begin{array}{c} x_1 \\ x_2 \end{array} & \begin{bmatrix} 0.6 & 0.3 \\ 0.2 & 0.9 \end{bmatrix} \end{array} \quad \text{and} \quad S = \begin{array}{c c} & \begin{array}{ccc} z_1 & z_2 & z_3 \end{array} \\ \begin{array}{c} y_1 \\ y_2 \end{array} & \begin{bmatrix} 1 & 0.5 & 0.3 \\ 0.8 & 0.4 & 0.7 \end{bmatrix} \end{array}$$

obtain fuzzy relation T as a composition between the fuzzy relation.

(a) Max-Min composition.

$$T = R \circ S = \begin{array}{c c} & \begin{array}{ccc} z_1 & z_2 & z_3 \end{array} \\ \begin{array}{c} x_1 \\ x_2 \end{array} & \begin{bmatrix} 0.6 & 0.5 & 0.3 \\ 0.8 & 0.4 & 0.7 \end{bmatrix} \end{array}$$

The calculation for obtaining T as follows.

$\mu_{\underset{\sim}{T}}(x_1,z_1) = \max\{\min[\mu_{\underset{\sim}{R}}(x_1,y_1), \mu_{\underset{\sim}{S}}(y_1,z_1)],$
$\min[\mu_{\underset{\sim}{R}}(x_1,y_2), \mu_{\underset{\sim}{S}}(y_2,z_1)]\}$
$= \max[\min(0.6,1), \min(0.3,0.8)]$
$= \max(0.6,0.3) = \underline{\underline{0.6}}$

$\mu_{\underset{\sim}{T}}(x_1,z_2) = \max[\min(0.6,0.5), \min(0.3,0.4)]$
$= \max(0.5,0.3) = 0.5$

$\mu_{\underset{\sim}{T}}(x_1,z_3) = \max[\min(0.6,0.3), \min(0.3,0.7)]$
$= \max(0.3,0.3) = 0.3$

$\mu_{\underset{\sim}{T}}(x_2,z_1) = \max[\min(0.2,1), \min(0.9,0.8)]$
$= \max(0.2,0.8) = 0.8$

$\mu_{\underset{\sim}{T}}(x_2,z_2) = \max[\min(0.2,0.5), \min(0.9,0.4)]$
$= \max(0.2,0.4) = 0.4$

$\mu_{\underset{\sim}{T}}(x_2,z_3) = \max[\min(0.2,0.3), \min(0.9,0.7)]$
$= \max(0.2,0.7) = 0.7$

(b) Max-product composition

$\underset{\sim}{T} = \underset{\sim}{R} \cdot \underset{\sim}{S}$

$\mu_{\underset{\sim}{T}}(x_1,z_1) = \max\{[\mu_{\underset{\sim}{R}}(x_1,y_1) \cdot \mu_{\underset{\sim}{S}}(y_1,z_1)],$
$[\mu_{\underset{\sim}{R}}(x_1,y_2) \cdot \mu_{\underset{\sim}{S}}(y_2,z_1)]\}$
$= \max(0.6, 0.24) = 0.6$

$\mu_{\underset{\sim}{T}}(x_1,z_2) = \max[(0.6\times0.5), (0.3\times0.4)]$
$= \max(0.3, 0.12) = 0.3$

$\mu_{\underset{\sim}{T}}(x_1,z_3) = \max[(0.6\times0.3), (0.3\times0.7)]$
$= \max(0.18, 0.21) = 0.21$

$\mu_{\underset{\sim}{T}}(x_2,z_1) = \max[(0.2\times1), (0.9\times0.8)]$
$= \max(0.2, 0.72) = 0.72$

$\mu_{\underset{\sim}{T}}(x_2,z_2) = \max[(0.2\times0.5), (0.9\times0.4)]$
$= \max(0.1, 0.36) = 0.36$

$\mu_{\underset{\sim}{T}}(x_2,z_3) = \max[(0.2\times0.3), (0.9\times0.7)]$
$= \max(0.06, 0.63) = 0.63$

The fuzzy relation $\underset{\sim}{T}$ by max-product composition is given as

$$\underset{\sim}{T} = \begin{array}{c} \\ x_1 \\ x_2 \end{array}\begin{array}{c} \begin{array}{ccc} z_1 & z_2 & z_3 \end{array} \\ \begin{bmatrix} 0.6 & 0.3 & 0.21 \\ 0.72 & 0.36 & 0.63 \end{bmatrix} \end{array}$$

14. For a Speed Control of DC motor, the membership function of series resistance, armature current and speed are given as follows:

$$R_{se} = \left\{ \frac{0.4}{30} + \frac{0.6}{60} + \frac{1.0}{100} + \frac{0.1}{120} \right\}$$

$$\underset{\sim}{I_a} = \left\{ \frac{0.2}{20} + \frac{0.3}{40} + \frac{0.6}{60} + \frac{0.8}{80} + \frac{1.0}{100} + \frac{0.2}{120} \right\}$$

$$N = \left\{ \frac{0.35}{500} + \frac{0.67}{1000} + \frac{0.97}{1500} + \frac{0.25}{1800} \right\}$$

Compute relation T for relating series resistance to motor speed, i.e $\underset{\sim}{R_{se}}$ to $\underset{\sim}{N}$. perform max-min composition only

Solution: For relating series resistance to motor speed i.e R_{se} to $\underset{\sim}{N}$. we have to perform the following operation

$$\underset{\sim}{R} = \underset{\sim}{R_{se}} \times \underset{\sim}{I_a}$$

$$\underset{\sim}{S} = \underset{\sim}{I_a} \times \underset{\sim}{N}$$

$$\underset{\sim}{T} = \underset{\sim}{R} \circ \underset{\sim}{S}$$

$$\underset{\sim}{R} = \underset{\sim}{R_{se}} \times \underset{\sim}{I_a}$$

$$\begin{array}{c} \\ 30 \\ 60 \\ 100 \\ 120 \end{array}\begin{array}{c} \begin{array}{cccccc} 20 & 40 & 60 & 80 & 100 & 120 \end{array} \\ \begin{bmatrix} 0.2 & 0.3 & 0.4 & 0.4 & 0.4 & 0.2 \\ 0.2 & 0.3 & 0.6 & 0.6 & 0.6 & 0.2 \\ 0.2 & 0.3 & 0.6 & 0.8 & 1.0 & 0.2 \\ 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 \end{bmatrix} \end{array}$$

Relation $\underset{\sim}{S}$ is obtained as the Cartesian product of $\underset{\sim}{I_a}$ and $\underset{\sim}{N}$ i.e

$$\underset{\sim}{S} = \underset{\sim}{I}_a \times \underset{\sim}{N}. \quad \begin{array}{c|cccc} & 500 & 1000 & 1500 & 1800 \\ \hline 20 & 0.2 & 0.2 & 0.2 & 0.2 \\ 40 & 0.3 & 0.3 & 0.3 & 0.25 \\ 60 & 0.35 & 0.6 & 0.6 & 0.25 \\ 80 & 0.35 & 0.67 & 0.8 & 0.25 \\ 100 & 0.35 & 0.67 & 0.97 & 0.25 \\ 120 & 0.2 & 0.2 & 0.2 & 0.2 \end{array}$$

Relation T is obtained as the composition between relation R and S ie.

$$\underset{\sim}{T} = \underset{\sim}{R} \circ \underset{\sim}{S}. \quad \begin{array}{c|cccc} & 500 & 100 & 1500 & 1800 \\ \hline 30 & 0.35 & 0.4 & 0.4 & 0.25 \\ 60 & 0.35 & 0.6 & 0.6 & 0.25 \\ 100 & 0.35 & 0.67 & 0.97 & 0.25 \\ 120 & 0.1 & 0.1 & 0.1 & 0.1 \end{array}$$

15. Consider two fuzzy sets given by

$$\underset{\sim}{A} = \left\{ \frac{1}{low} + \frac{0.2}{medium} + \frac{0.5}{high} \right\}$$

$$\underset{\sim}{B} = \left\{ \frac{0.9}{positive} + \frac{0.4}{zero} + \frac{0.9}{negative} \right\}$$

(a) Find the fuzzy relation for the cartesian product of $\underset{\sim}{A}$ and $\underset{\sim}{B}$ ie $\underset{\sim}{R} = \underset{\sim}{A} \times \underset{\sim}{B}$

(b) Introduce fuzzy set $\underset{\sim}{C}$ given by

$$\underset{\sim}{C} = \left\{ \frac{0.1}{low} + \frac{0.2}{medium} + \frac{0.7}{high} \right\}$$

Find the relation between $\underset{\sim}{C}$ and $\underset{\sim}{B}$ using cartesian product. ie, find $\underset{\sim}{S} = \underset{\sim}{C} \times \underset{\sim}{B}$

(c) find $\underset{\sim}{C} \circ \underset{\sim}{B}$ using max-min composition

(d) find $\underset{\sim}{C} \circ \underset{\sim}{S}$ using max-min composition

Solution:-

(a) the cartesian product between $\underset{\sim}{A}$ and $\underset{\sim}{B}$ is obtained as

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

$$= \begin{array}{c} \\ \text{low} \\ \text{medium} \\ \text{high} \end{array} \begin{array}{c} \begin{array}{ccc} \text{positive} & \text{zero} & \text{negative} \end{array} \\ \begin{bmatrix} 0.9 & 0.4 & 0.9 \\ 0.2 & 0.2 & 0.2 \\ 0.5 & 0.4 & 0.5 \end{bmatrix} \end{array}$$

b) The new fuzzy set is

$$\underset{\sim}{C} = \left\{ \frac{0.1}{\text{low}} + \frac{0.2}{\text{medium}} + \frac{0.7}{\text{high}} \right\}$$

The Cartesian product between $\underset{\sim}{C}$ and $\underset{\sim}{B}$ is obtained as

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

$$= \begin{array}{c} \\ \text{low} \\ \text{medium} \\ \text{high} \end{array} \begin{array}{c} \begin{array}{ccc} \text{positive} & \text{zero} & \text{negative} \end{array} \\ \begin{bmatrix} 0.1 & 0.1 & 0.1 \\ 0.2 & 0.2 & 0.2 \\ 0.7 & 0.4 & 0.7 \end{bmatrix} \end{array}$$

(c)

$$\underset{\sim}{C} \circ \underset{\sim}{R} = [0.1 \;\; 0.2 \;\; 0.7] \begin{bmatrix} 0.9 & 0.4 & 0.9 \\ 0.2 & 0.2 & 0.2 \\ 0.5 & 0.4 & 0.5 \end{bmatrix}$$

$$= [0.5 \;\; 0.4 \;\; 0.5]$$

ie $\mu_{\underset{\sim}{C} \circ \underset{\sim}{R}}(x_1, y_1) = \max[\min(0.1, 0.9), \min(0.2, 0.2), \min(0.7, 0.5)]$

$$= \max(0.1, 0.2, 0.5) = 0.5$$

(d) $\underset{\sim}{C} \circ \underset{\sim}{S} = [0.1 \;\; 0.2 \;\; 0.7] \begin{bmatrix} 0.1 & 0.1 & 0.1 \\ 0.2 & 0.2 & 0.2 \\ 0.7 & 0.4 & 0.7 \end{bmatrix}$

$$= [0.7 \;\; 0.4 \;\; 0.7]$$

MOD-IV

Fuzzy Membership function

Membership function defines the fuzziness in a fuzzy set irrespective of the elements in the set, which are discrete or continuous.

The membership functions are generally represented in graphical form.

Membership function can be thought of as a technique to solve empirical problems on the basis of experience rather than knowledge.

Features of the membership functions

The membership function defines all the information contained in a fuzzy set.

A fuzzy set $\underset{\sim}{A}$ in the universe of discourse X can be defined as a set of ordered pairs:

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

where $\mu_{\underset{\sim}{A}}(\cdot)$ is called membership function of $\underset{\sim}{A}$. The membership function $\mu_{\underset{\sim}{A}}(\cdot)$ maps X to the membership space M, i.e., $\mu_{\underset{\sim}{A}} : X \to M$. The membership value range in the interval [0,1] i.e the range of the membership function is a subset of the non-negative real numbers

The three main basic features in membership function are the following.

1. Core :- The core of a membership function for some fuzzy set $\underset{\sim}{A}$ is defined as that region of universe that is characterized by complete

membership in the set $\underset{\sim}{A}$. The core has elements x of the Universe such that

$$\mu_{\underset{\sim}{A}}(x) = 1$$

The core of a fuzzy set may be an empty set.

2. Support :- The support of a membership function for a fuzzy set $\underset{\sim}{A}$ is defined as that region of Universe that is characterized by a nonzero membership in the set $\underset{\sim}{A}$. The support comprises elements x of the Universe such that

$$\mu_{\underset{\sim}{A}}(x) > 0$$

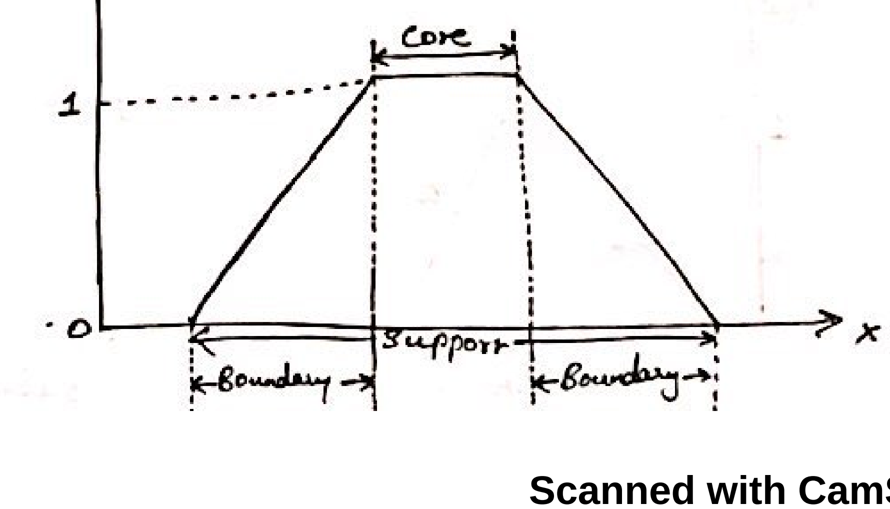
A fuzzy set whose support is a single element in X with $\mu_{\underset{\sim}{A}}(x) = 1$ is referred to as fuzzy singleton

3. Boundary :- The support of a membership function for a fuzzy set $\underset{\sim}{A}$ is defined as that region of Universe containing elements that have a nonzero but not complete membership. The boundary comprises those elements of x of the Universe such that

$$0 < \mu_{\underset{\sim}{A}}(x) < 1$$

The boundary elements are those which possess partial membership in the fuzzy set $\underset{\sim}{A}$.

The Core, Support and boundary are the 3 main features of a fuzzy set membership function.



There are various Types of fuzzy sets.

→ Normal fuzzy set :-

A fuzzy set whose membership function has at least one element x in the universe whose membership value is unity is called normal fuzzy set. The element for which the membership is equal to 1 is called prototypical element.

→ Subnormal fuzzy set :-

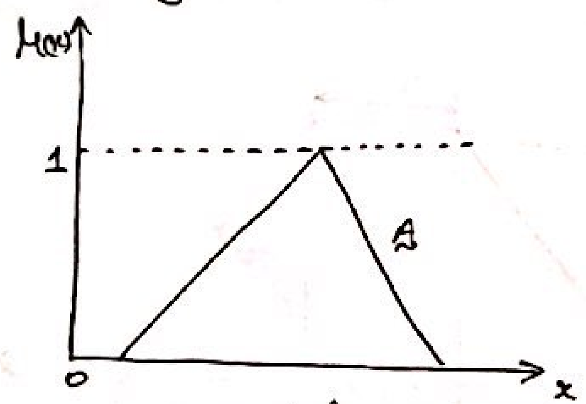
A fuzzy set wherein no membership function has its value equal to 1 is called subnormal fuzzy set.

→ Convex fuzzy set :-

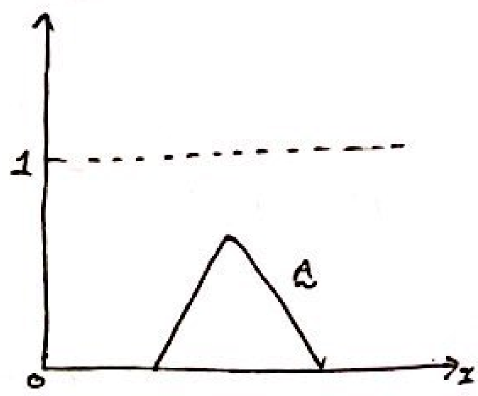
A convex fuzzy set has a membership function whose membership values are strictly monotonically increasing or strictly monotonically decreasing with increasing values for elements in the universe.

→ Non convex fuzzy set :-

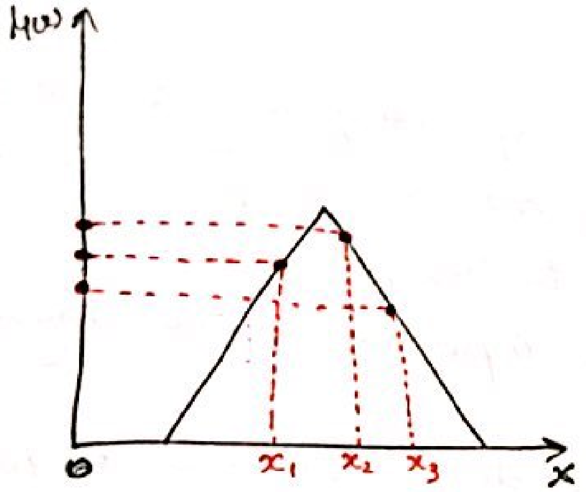
A fuzzy set possessing characteristics opposite to that of convex fuzzy set is called non convex fuzzy set, i.e. the membership values of the membership functions are not strictly monotonically increasing or decreasing or strictly monotonically increasing then decreasing.



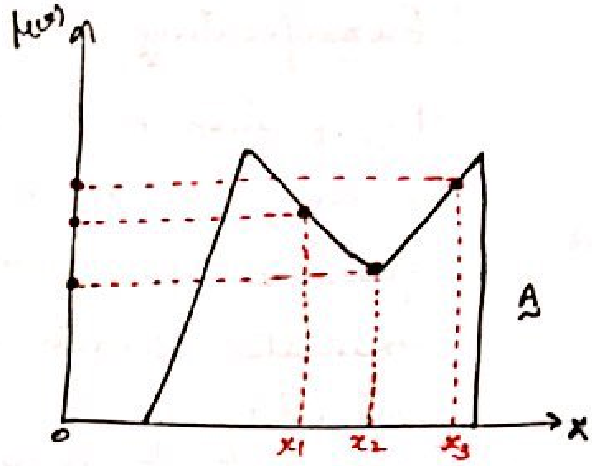
Normal fuzzy set



Subnormal fuzzy set.



Convex normal fuzzy set



Non convex normal fuzzy set.

The convex normal fuzzy set can be defined in the following way. For elements x_1, x_2 and x_3 in a fuzzy set $\underset{\sim}{A}$. If the following relation between x_1, x_2 and x_3 holds ie

$$\mu_{\underset{\sim}{A}}(x_2) \geq \min[\mu_{\underset{\sim}{A}}(x_1), \mu_{\underset{\sim}{A}}(x_3)]$$

then $\underset{\sim}{A}$ is said to be a convex fuzzy sets

The membership of the element x_2 should be greater than or equal to the membership of elements x_1 and x_3. for a non convex fuzzy set, the constraints is not satisfied

$$\mu_{\underset{\sim}{A}}(x_2) \not\geq \min[\mu_{\underset{\sim}{A}}(x_1), \mu_{\underset{\sim}{A}}(x_3)]$$

The maximum value of the membership function in a fuzzy set $\underset{\sim}{A}$ is called as the height of the fuzzy set. For a normal fuzzy set, the height is equal to 1, because the maximum value of the membership function allowed is 1. If the height of a fuzzy set is less than 1, then the fuzzy set is called subnormal fuzzy set.

Fuzzification

Fuzzification is the process of transforming a crisp set to a fuzzy set or a fuzzy set to a fuzzier set, i.e. crisp quantities are converted to fuzzy quantities. This operation translates accurate crisp input values into linguistic variables.

The uncertainty may arise due to vagueness, imprecision or uncertainty, in this case the variable is probably fuzzy and can be represented by a membership function.

eg: when the temperature is 9°C it is a crisp input value and is converted into linguistic variable such as cold or warm

For a fuzzy set $\underset{\sim}{A} = \{\mu_i / x_i / x_i \in X\}$, a common fuzzification algorithm is performed by keeping μ_i constant and x_i being transformed to a fuzzy set $Q(x_i)$ depicting the expression about x_i. The fuzzy set $\underset{\sim}{Q}(x_i)$ is referred to as the kernel of fuzzification. The fuzzified set $\underset{\sim}{A}$ can be expressed as

$$\underset{\sim}{A} = \mu_1 Q(x_1) + \mu_2 (Q(x_2)) + \dots + \mu_n Q(x_n)$$

where the symbol $\sim$ means fuzzified. This process of fuzzification is called support fuzzification. There is another method of fuzzification called grade fuzzification where x_i is kept constant and μ_i is expressed as a fuzzy set.

Methods of Membership Value Assignment.

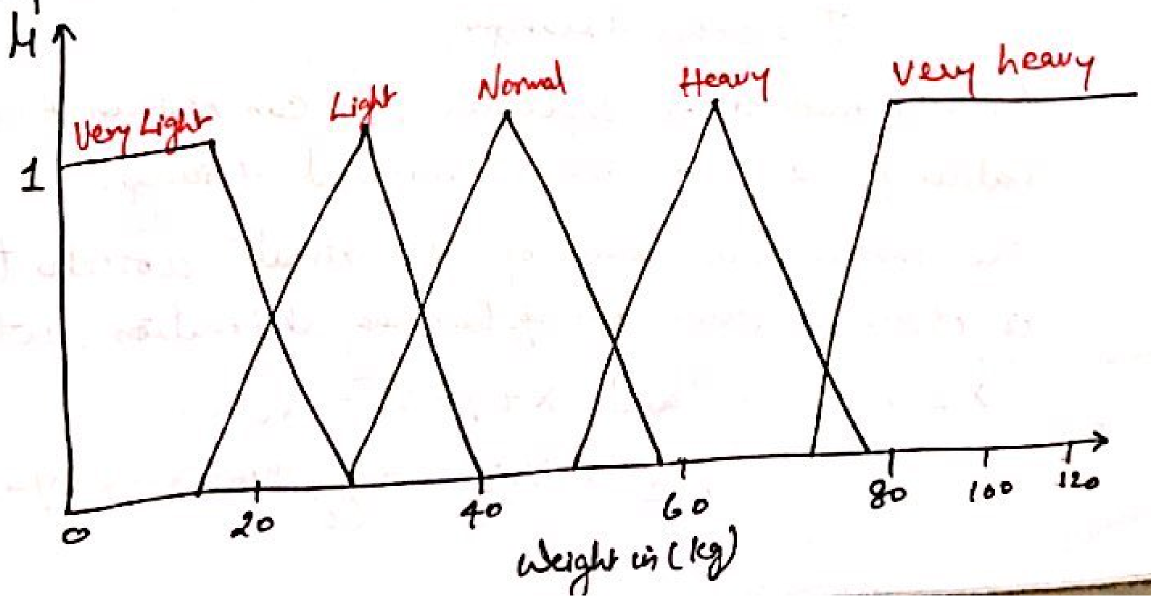
There are several ways to assign membership values to fuzzy variables in comparison with the probability density function to random variables. The process of membership value assignment may be by intuition, logical reasoning, procedural method or algorithmic approach. The methods of assigning membership values are as follows.

1. Intuition
2. Inference
3. Rank ordering
4. Angular fuzzy sets
5. Neural networks
6. Genetic algorithm
7. Induction Reasoning

Intuition

Intuition method is based upon the common Intelligence of human. It is the capacity of human to develop membership functions on the basis of their own intelligence and understanding capability. There should be an in-depth knowledge of the application to which membership value assignment has to be made.

The following figure shows various shapes of weights of people measured in kilogram in the universe.



Each curve is a membership function corresponding to various fuzzy variables, such as very light, light, normal, heavy and very heavy. The curves are based on context functions and the human developing them.

Inference.

The Inference method uses knowledge to perform deduction reasoning. Deduction achieves conclusion by means of forward inference. There are various methods for performing deductive reasoning. The membership function may be defined by various shapes triangular, trapezoidal, bell-shaped, Gaussian and so on.

Consider a triangle, where X, Y and Z are the angles, such that $X \geq Y \geq Z \geq 0$ and let U be the Universe of triangles i.e.

$$U = \{(X, Y, Z) \mid X \geq Y \geq Z \geq 0;\ X + Y + Z = 180\}$$

There are various types of triangles available.

$\underset{\sim}{I}$ = Isosceles triangle

$\underset{\sim}{E}$ = Equilateral triangle

$\underset{\sim}{R}$ = Right-angle triangle

$\underset{\sim}{IR}$ = Isosceles Right angle triangle

$\underset{\sim}{T}$ = Other triangles.

By the method of inference, we can obtain the membership values for all the above mentioned triangle.

The membership values of approximate isosceles triangle is obtained using the following definition, where $X \geq Y \geq Z \geq 0$ and $X + Y + Z = 180°$

$$\mu_I(X, Y, Z) = 1 - \frac{1}{60} \min(X - Y, Y - Z)$$

Scanned with CamScanner

If $x=y$ or $y=z$, the membership value of approximate isosceles triangle is equal to 1. On the other hand, If $x=120°$, $y=60°$ and $z=0°$, we get.

$$\mu_{I}(x,y,z) = 1-\frac{1}{60°}\min(120°-60°, 60°-0°)$$

$$= 1-\frac{1}{60°}\min(60°, 60°)$$

$$= 1-\frac{1}{60°}\times 60°$$

$$= 1-1=0$$

The membership value of approximate right-angle triangle is given by

$$\mu_{R}(x,y,z) = 1-\frac{1}{90°}|x-90°|$$

If $x=90°$, the membership value of a right-angle triangle is 1, and If $x=180°$, the membership value μ_R becomes 0:

$$x = 90° \Rightarrow \mu_R = 1$$
$$x = 180° \Rightarrow \mu_{\underset{\sim}{R}} = 0$$

The membership value of approximate isosceles right-angle triangle is obtained by taking the logical intersection of the approximate isosceles and approximate right-angle triangle membership function is

$$\underset{\sim}{IR} = \underset{\sim}{I} \cap \underset{\sim}{R}$$

and is given by

$$\mu_{\underset{\sim}{IR}}(x,y,z) = \min[\mu_{\underset{\sim}{I}}(x,y,z), \mu_{\underset{\sim}{R}}(x,y,z)]$$
$$= 1-\max\left[\frac{1}{60°}\min(x-y,y-z), \frac{1}{90°}|x-90°|\right]$$

The membership function for a fuzzy equilateral triangle is given by $\mu_{\underset{\sim}{E}}(x,y,z) = 1-\frac{1}{180°}|x-z|$

Scanned with CamScanner

The membership function of other triangles, denoted by $\underset{\sim}{I}$, is the complement of the logical union of $\underset{\sim}{I}$, $\underset{\sim}{R}$ and $\underset{\sim}{E}$ is

$$\underset{\sim}{I} = \overline{\underset{\sim}{I} \cup \underset{\sim}{R} \cup \underset{\sim}{E}}$$

By using De-Morgan's law, we get.

$$\underset{\sim}{I} = \overline{\underset{\sim}{I}} \cap \overline{\underset{\sim}{R}} \cap \overline{\underset{\sim}{E}}$$

The membership value can be obtained using the equation

$$\mu_{\underset{\sim}{I}}(x,y,z) = \min\{1-\mu_{\underset{\sim}{I}}(x,y,z), 1-\mu_{\underset{\sim}{E}}(x,y,z), 1-\mu_{R}(x,y,z)\}$$

$$= \frac{1}{180^\circ} \min\{3(x-y), 3(y-z), 2|x-90^\circ|, x-z\}$$

Rank Ordering

The formation of government is based on the polling concept, to identify a best student, ranking may be performed. All the above mentioned activities are carried out on the basis of the preferences made by an individual, a committee, a poll and other opinion methods.

This methodology can be adapted to assign membership value to a fuzzy variable. Pairwise comparisons enable us to determine preferences and this results in determining the order of the membership.

Lambda-Cuts for fuzzy sets (Alpha-cuts)

Consider a fuzzy set $\underset{\sim}{A}$. The set A_λ $(0 < \lambda < 1)$ called the lambda (λ)-cut (or alpha-cut) set, is a crisp set of the fuzzy set and is defined as follows.

$$A_\lambda = \{x \mid \mu_{\underset{\sim}{A}}(x) \geq \lambda\} \; ; \; \lambda \in [0,1]$$

The λ- cut sets A_λ is called a weak-lamda-cut set if it consists of all the elements of a fuzzy set whose membership function have values greater than or equal to a specified value.

On the other hand, the set A_λ is called a strong lambda-cut set if it consists of all the elements of a fuzzy set whose membership functions have values strictly greater than a specified value. A strong λ- cut set is given by

$$A_\lambda = \{x \mid \mu_{\underset{\sim}{A}}(x) > \lambda\} \; ; \; \lambda \in [0,1]$$

All the λ- cut sets form a family of crisp sets.

Any particular fuzzy set $\underset{\sim}{A}$ can be transformed into an infinite number of λ- cut sets, because there are infinite number of values λ can take in the interval $[0,1]$.

The properties of λ-cut sets are as follows:

1. $(\underset{\sim}{A} \cup \underset{\sim}{B})_\lambda = A_\lambda \cup B_\lambda$

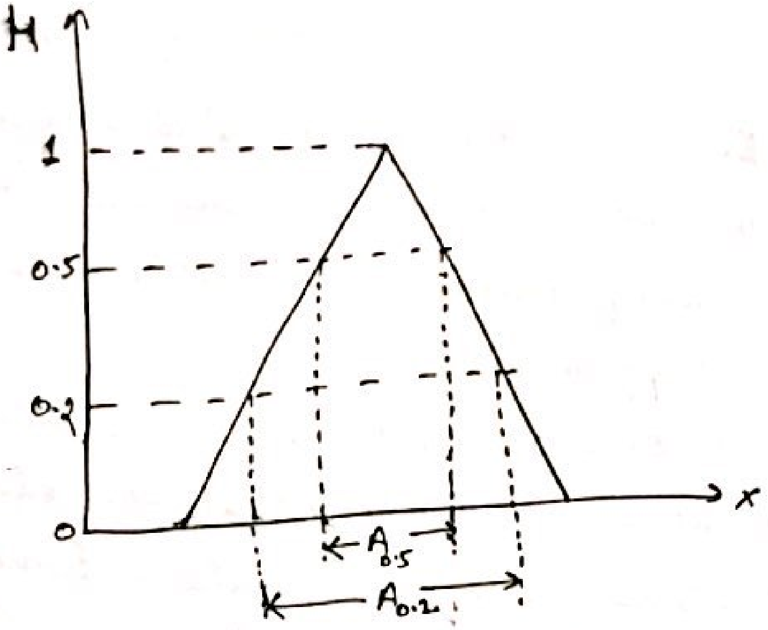
2. $(\underset{\sim}{A} \cap \underset{\sim}{B})_\lambda = A_\lambda \cap B_\lambda$

3. $(\bar{\underset{\sim}{A}})_\lambda \neq (\bar{A}_\lambda)$ except when $\lambda = 0.5$

4. For any $\lambda \leq \beta$, where $0 \leq \beta \leq 1$, it is true that

   $A_\beta \subseteq A_\lambda$ where $A_0 = X$

The following figure shows a continuous-valued fuzzy set with two λ-cut values.



The interval $[A_0^+, A_1]$ forms the boundaries of the fuzzy set $\underset{\sim}{A}$, i.e., the regions with the membership values between 0 and 1 i.e. for $\lambda = 0$ to 1

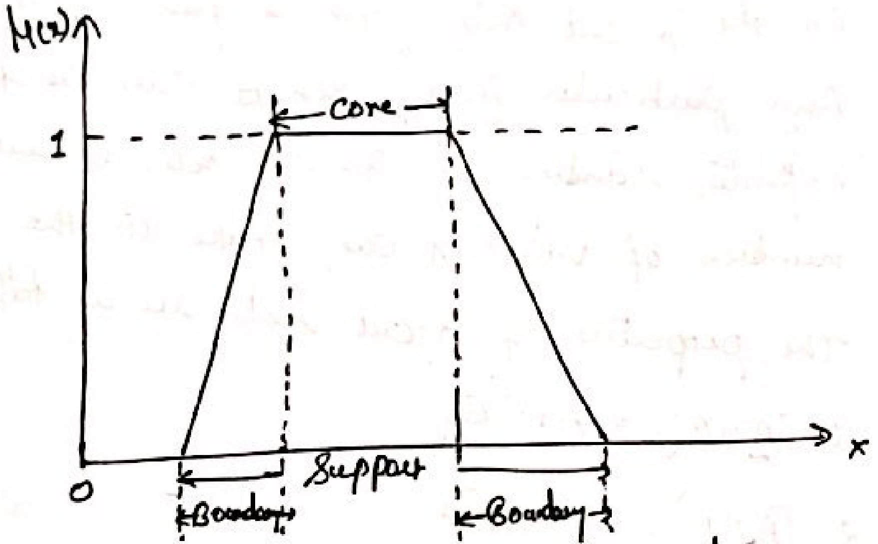


fig: features of membership function.

Lambda-cuts for fuzzy Relations

The λ-cut for fuzzy relation is similar to that for fuzzy sets. Let $\underset{\sim}{R}$ be a fuzzy relation where each row of the relational matrix is considered a fuzzy set.

For two fuzzy relations $\underset{\sim}{R}$ and $\underset{\sim}{S}$ the following properties

Should hold :

1. $(\underset{\sim}{R} \cup \underset{\sim}{S})_\lambda = R_\lambda \cup S_\lambda$

2. $(\underset{\sim}{R} \cap \underset{\sim}{S})_\lambda = R_\lambda \cap S_\lambda$

3. $(\overline{\underset{\sim}{R}})_\lambda \neq (\overline{R}_\lambda)$ except when $\lambda = 0.5$

4. for any $\lambda \leq \beta$, where $0 \leq \beta \leq 1$, it is true that $R_\beta \subseteq R_\lambda$

Defuzzification Methods.

Defuzzification is the process of conversion of a fuzzy quantity into a precise quantity. The output of a fuzzy process may be union of two or more fuzzy membership function defined on the universe of discourse of the output variable.

Defuzzification Methods include the following

1. Max-membership principle
2. Centroid method
3. Weighted average method
4. Mean-max membership
5. Centre of sums.
6. Centre of largest area
7. First of maxima, last of maxima

Max-membership principle

This method is also known as height method and is limited to peak output functions. This method is given by the algebraic expression

$$\mu_{\underset{\sim}{C}}(x^*) \geq \mu_{\underset{\sim}{C}}(x) \text{ for all } x \in X$$

This method is illustrated in the following figure.

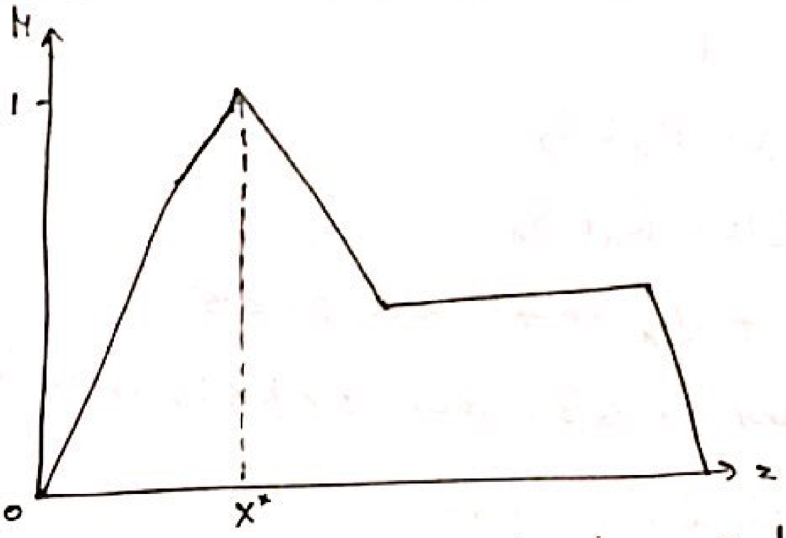
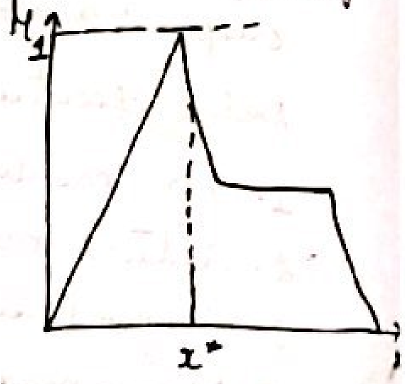


fig: Max-membership defuzzification method

Centroid Method

This method is known as center of mass, center of area or center of gravity method. It is the most commonly used defuzzification method. The defuzzified output x^* is defined as



$$x^* = \frac{\int \mu_{\underset{\sim}{c}}(x) \cdot x\,dx}{\int \mu_{\underset{\sim}{c}}(x)\,dx}$$

'$\int$' denotes an algebraic Integration

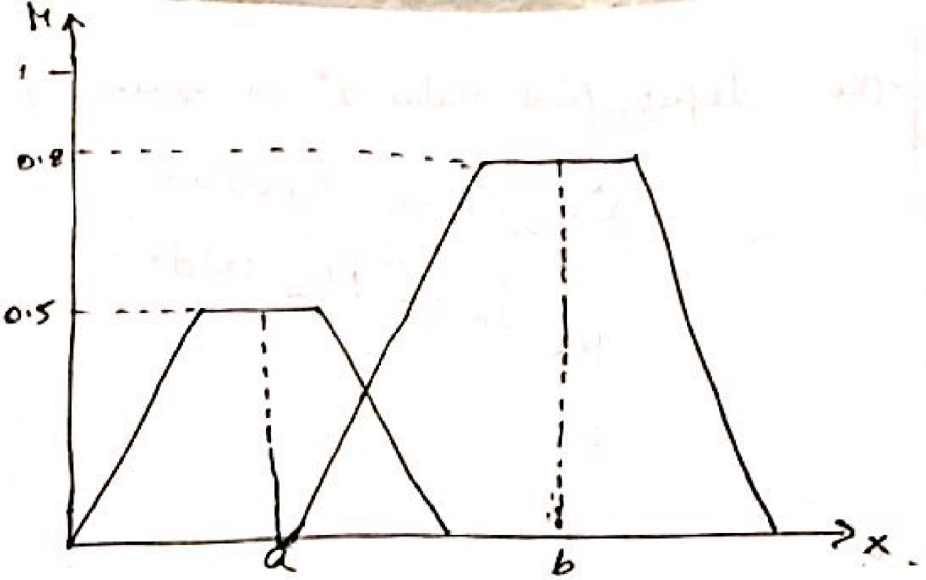
Weighted Average Method.

This method is valid for Symmetrical output membership functions only. Each membership function is weighed by its maximum membership value. The output in this case is given by

$$x^* = \frac{\sum \mu_{\underset{\sim}{c}}(\bar{x}_i) \cdot \bar{x}_i}{\sum \mu_{\underset{\sim}{c}}(\bar{x}_i)}$$

where Σ denotes algebraic sum and $\bar{x}_i$ is the maximum of the i^{th} membership function.

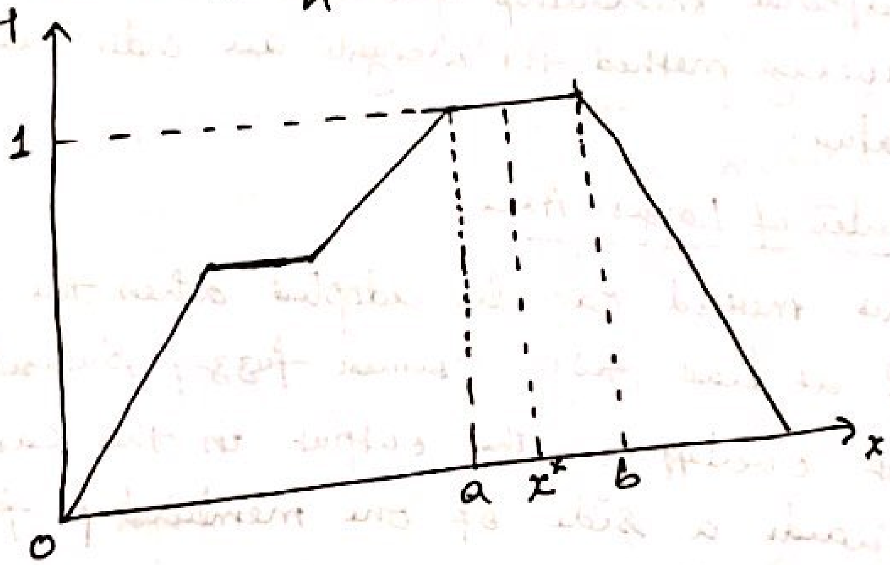
eg: $x^* = \frac{0.5a + 0.8b}{0.5 + 0.8}$



Mean - Max Membership

This method is also known as the middle of the maxima. This is closely related to max-membership method, except that the locations of the maximum membership can be non unique. The output here is given by

$$x^* = \frac{\sum_{i=1}^{n} \bar{x}_i}{n}$$



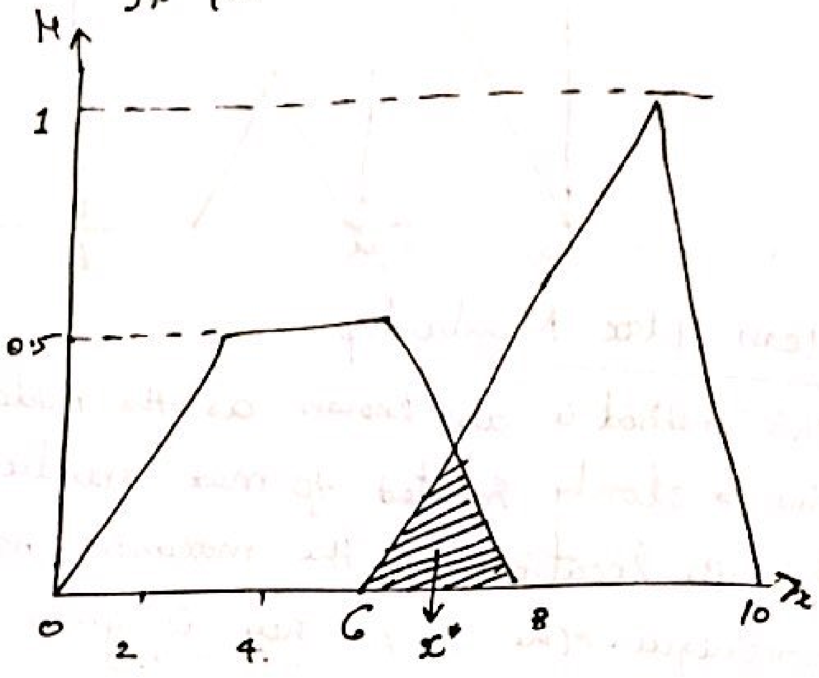
$$x^* = \frac{a+b}{2}$$

Center of Sums.

This method employs the algebraic sum of the individual fuzzy subsets instead of their union.

The defuzzified value x^* is given by

$$x^* = \frac{\int_x x \sum_{i=1}^{n} \mu_{C_i}(x)\,dx}{\int_x \sum_{i=1}^{n} \mu_{C_i}(x)\,dx}.$$



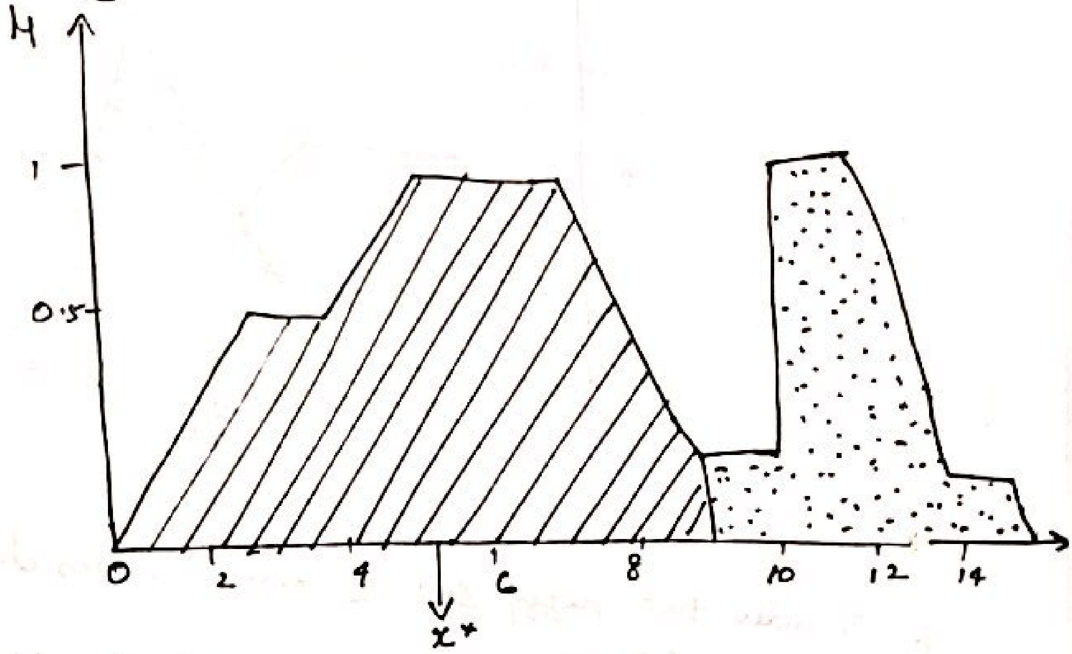
In center of sum method, the weights are the areas of the respective membership functions, whereas in the weighted average method the weights are individual membership value.

Center of Largest Area.

This method can be adopted when the output consist of at least two convex fuzzy subsets which are not overlapping. The output in this case is biased towards a side of one membership function. When output fuzzy set has at least two convex regions, the center of gravity of the convex fuzzy subregion having the largest area is used to obtain the defuzzified value x^*. The value is given by

$$x^* = \frac{\int \mu_{\underset{\sim}{C}i}(x) \cdot x \, dx}{\int \mu_{\underset{\sim}{C}i}(x) \, dx}$$

where $\underset{\sim}{C}i$ is the convex subregion that has the largest area making up $\underset{\sim}{C}j$.



First of Maxima (Last of Maxima)

This method uses the overall output or union of all individual output fuzzy set $\underset{\sim}{C}i$ for determining the smallest value of the domain with maximized membership in $\underset{\sim}{C}i$. The steps used for obtaining x^* are as follows:

1. Initially, the maximum height in the union is found:

$$hgt(\underset{\sim}{C}i) = \sup_{x \in X} \mu_{\underset{\sim}{C}i}(x)$$

where sup is supremum, i.e., the least upper bound

2. The first maxima is found:

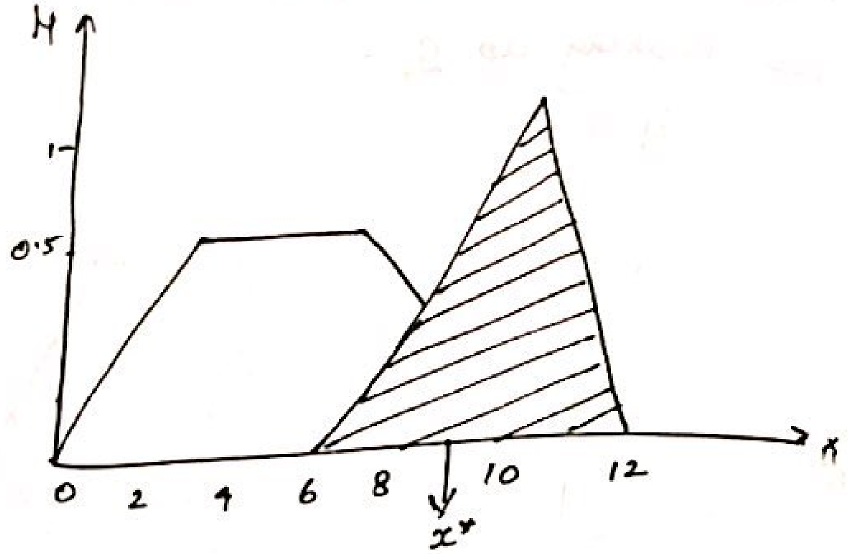
$$x^* = \inf_{x \in X} \{ x \in X \mid \mu_{\underset{\sim}{C}i}(x) = hgt(\underset{\sim}{C}i) \}$$

where inf is the infimum, i.e. the greatest lower bound.

Scanned with CamScanner

3. After this the last maxima is found:

$$x^* = \sup_{x \in X} \{x \in X \mid \mu_{\underset{\sim}{C}i}(x) = hgt(\underset{\sim}{C}i)\}$$



problems

1. Consider two fuzzy sets $\underset{\sim}{A}$ and $\underset{\sim}{B}$, both defined on x given as follows.

$\mu_{xi}(x)$	x_1	x_2	x_3	x_4	x_5
$\underset{\sim}{A}$	0.2	0.3	0.4	0.7	0.1
$\underset{\sim}{B}$	0.4	0.5	0.6	0.8	0.9

Express the following λ-cut sets using Zadeh's Notation

(a) $\bar{\underset{\sim}{A}}_{0.7}$ (b) $\underset{\sim}{B}_{0.2}$ (c) $(\underset{\sim}{A} \cup \underset{\sim}{B})_{0.6}$ (d) $(\underset{\sim}{A} \cap \underset{\sim}{B})_{0.5}$

(e) $(\underset{\sim}{A} \cup \bar{\underset{\sim}{A}})_{0.7}$ (f) $(B \cap \bar{\underset{\sim}{B}})_{0.3}$ (g) $(\overline{\underset{\sim}{A} \cap \underset{\sim}{B}})_{0.6}$ (h) $(\bar{\underset{\sim}{A}} \cap \bar{\underset{\sim}{B}})_{0.8}$

(a) $\bar{\underset{\sim}{A}}_{0.7} = 1 - \mu_A(x)$

$$= \left\{\frac{0.8}{x_1} + \frac{0.7}{x_2} + \frac{0.6}{x_3} + \frac{0.3}{x_4} + \frac{0.9}{x_5}\right\}$$

$\bar{\underset{\sim}{A}}_{0.7} = \{x_1, x_2, x_3\}$

(b) $\underset{\sim}{B} = \left\{ \frac{0.4}{x_1} + \frac{0.5}{x_2} + \frac{0.6}{x_3} + \frac{0.8}{x_4} + \frac{0.9}{x_5} \right\}$

$(B)_{0.2} = \{x_1, x_2, x_3, x_4, x_5\}$

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

$= \left\{ \frac{0.4}{x_1} + \frac{0.5}{x_2} + \frac{0.6}{x_3} + \frac{0.8}{x_4} + \frac{0.9}{x_5} \right\}$

$(A \cup B)_{0.6} = \{x_3, x_4, x_5\}$

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

$= \left\{ \frac{0.2}{x_1} + \frac{0.3}{x_2} + \frac{0.4}{x_3} + \frac{0.7}{x_4} + \frac{0.1}{x_5} \right\}$

$(A \cap B)_{0.5} = \{x_4\}$

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

$= \left\{ \frac{0.8}{x_1} + \frac{0.7}{x_2} + \frac{0.6}{x_3} + \frac{0.7}{x_4} + \frac{0.9}{x_5} \right\}$

$(A \cup \bar{A})_{0.7} = \{x_1, x_2, x_4, x_5\}$

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

$= \left\{ \frac{0.4}{x_1} + \frac{0.5}{x_2} + \frac{0.4}{x_3} + \frac{0.2}{x_4} + \frac{0.1}{x_5} \right\}$

$(B \cap \bar{B})_{0.3} = \{x_1, x_2, x_3\}$

(g) $(\overline{\underset{\sim}{A} \cap \underset{\sim}{B}}) = \{1 - \mu(\underset{\sim}{A} \cap \underset{\sim}{B})\}$

$= \left\{ \frac{0.8}{x_1} + \frac{0.7}{x_2} + \frac{0.6}{x_3} + \frac{0.3}{x_4} + \frac{0.9}{x_5} \right\}$

$(\overline{A \cap B})_{0.6} = \{x_1, x_2, x_3, x_5\}$

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

$$= \left\{ \frac{0.8}{x_1} + \frac{0.7}{x_2} + \frac{0.6}{x_3} + \frac{0.3}{x_4} + \frac{0.9}{x_5} \right\}$$

$(\bar{A} \cup \bar{B})_{0.8} = \{ x_1, x_5 \}$

2) Using Zadeh's notation, determine the λ-cut sets for the given fuzzy sets.

$$\underset{\sim}{S_1} = \left\{ \frac{0}{0} + \frac{0.5}{20} + \frac{0.65}{40} + \frac{0.85}{60} + \frac{1.0}{80} + \frac{1.0}{100} \right\}$$

$$\underset{\sim}{S_2} = \left\{ \frac{0}{0} + \frac{0.45}{20} + \frac{0.6}{40} + \frac{0.8}{60} + \frac{0.95}{80} + \frac{1.0}{100} \right\}$$

The λ-cut set is obtained using

$$A_\lambda = \{ x \mid \mu_A(x) \geq \lambda \}$$

Here $\lambda = 0.5$.

(a) $(\underset{\sim}{S_1} \cup \underset{\sim}{S_2}) = \max[\mu_{\underset{\sim}{S_1}}(x), \mu_{\underset{\sim}{S_2}}(x)]$

$$= \left\{ \frac{0}{0} + \frac{0.5}{20} + \frac{0.65}{40} + \frac{0.85}{60} + \frac{1.0}{80} + \frac{1.0}{100} \right\}$$

$(\underset{\sim}{S_1} \cup \underset{\sim}{S_2})_{0.5} = \{ 20, 40, 60, 80, 100 \}$

(b) $(\underset{\sim}{S_1} \cap \underset{\sim}{S_2}) = \min[\mu_{\underset{\sim}{S_1}}(x), \mu_{\underset{\sim}{S_2}}(x)]$

$$= \left\{ \frac{0}{0} + \frac{0.45}{20} + \frac{0.6}{40} + \frac{0.8}{60} + \frac{0.95}{80} + \frac{1.0}{100} \right\}$$

$(\underset{\sim}{S_1} \cap \underset{\sim}{S_2})_{0.5} = \{ 40, 60, 80, 100 \}$

(c) $\bar{\underset{\sim}{S_1}} = 1 - \mu_{\underset{\sim}{S_1}}(x)$

$$= \left\{ \frac{1}{0} + \frac{0.5}{20} + \frac{0.35}{40} + \frac{0.15}{60} + \frac{0}{80} + \frac{0}{100} \right\}$$

$\bar{S_1}_{0.5}$ $\{ 0, 20 \}$

(d) $\overline{\underset{\sim}{S_2}} = 1 - \mu_{\underset{\sim}{S_2}}(x)$

$$= \left\{ \frac{1}{0} + \frac{0.55}{20} + \frac{0.4}{40} + \frac{0.2}{60} + \frac{0.05}{80} + \frac{0}{100} \right\}$$

$(\overline{S_2})_{0.5} = \{0, 20\}$

(e) $(\overline{\underset{\sim}{S_1} \cup \underset{\sim}{S_2}}) = 1 - \mu_{\underset{\sim}{S_1} \cup S_2}(x)$

$$= \left\{ \frac{1}{0} + \frac{0.5}{20} + \frac{0.35}{40} + \frac{0.15}{60} + \frac{0}{80} + \frac{0}{100} \right\}$$

$(\overline{S_1 \cup S_2})_{0.5} = \{0, 20\}$

(f) $(\overline{\underset{\sim}{S_1} \cap \underset{\sim}{S_2}}) = 1 - \mu_{\underset{\sim}{S_1} \cap \underset{\sim}{S_2}}(x)$

$$= \left\{ \frac{1}{0} + \frac{0.55}{20} + \frac{0.4}{40} + \frac{0.2}{60} + \frac{0.05}{80} + \frac{0}{100} \right\}$$

$(\overline{S_1 \cap S_2})_{0.5} = \{0, 20\}$

Truth Values and Tables in Fuzzy Logic

Fuzzy logic uses linguistic variables. The value of linguistic variables are words or sentences in a natural or artificial language. eg: height is a linguistic variable if it takes values such as tall, medium, short etc.

The linguistic variable provides approximate characterization of a complex problem. A linguistic variable is characterized by

1. name of the variable (x);
2. term set of the variable t(x);
3. Syntactic rule for generating the values of x;
4. Semantic rule for associating each value of x with its meaning.

Apart from linguistic variables there exists linguistic hedges. eg: In the fuzzy set "very tall" the word "very" is a linguistic hedge. A few popular linguistic hedges include very, highly, slightly, moderately, plus, minus, fairly, rather

Reasoning has logic as its basis, whereas propositions are text sentences expressed in any language and are generally expressed in an canonical form as

$$z \text{ is } p.$$

where z is the symbol of the subject and p is the predicate designing the characteristics of the subject.

for eg: "London is in United Kingdom" is a proposition in which "London" is the subject and "is United Kingdom" is the predicate, which specifies a property of "London"

Truth tables define logic function of two proposition. Let X and Y be two propositions, either of which can be true or false. The basic logic operations performed over the proposition are the following

1. Conjunction ($\wedge$) : X AND Y
2. Disjunction ($\vee$) : X OR Y
3. Implication or Conditional ($\rightarrow$) : If X THEN Y
4. Bidirectional or Equivalence ($\Leftrightarrow$) : X If AND ONLY IF Y.

On the basis of these operations on proposition, Inference rules can be formulated. Few Inference rules are as follows:

$$[X \wedge (X \Rightarrow Y)] \Rightarrow Y$$

$$[\bar{Y} \wedge (X \Rightarrow Y)] \Rightarrow \bar{X}$$

$$[(X \Rightarrow Y) \wedge (Y \Rightarrow Z)] \Rightarrow (X \Rightarrow Z)$$

The above rule produce certain proposition that are always true irrespective of the truth values of proposition X and Y. Such proposition are called tautologies.

The true values of propositions in fuzzy logic are allowed to range over the Unit Interval [0,1]

The truth value of a proposition can be obtained from the logic operations of other propositions whose

Scanned with CamScanner

truth values are known. If $t_v(X)$ and $t_v(Y)$ are numerical truth values of propositions X and Y, respectively then

$$t_v(X \text{ AND } Y) = t_v(X) \wedge t_v(Y) = \min\{t_v(X), t_v(Y)\}$$

$$t_v(X \text{ OR } Y) = t_v(X) \vee t_v(Y) = \max\{t_v(X), t_v(Y)\}$$

$$t_v(\text{NOT } X) = 1 - t_v(X) \text{ (complement)}$$

$$t_v(X \Rightarrow Y) = t_v(X) \Rightarrow t_v(B) = \max\{1 - t_v(X), \min[t_v(X), t_v(Y)]\}$$

Fuzzy Propositions

For extending the reasoning capability fuzzy logic uses fuzzy predicates, fuzzy predicate modifiers, fuzzy quantifiers and fuzzy qualifiers in the fuzzy propositions. The fuzzy propositions make the fuzzy logic differ from classical logic. The fuzzy propositions are as follows.

1. Fuzzy predicates :- In fuzzy logic the predicates can be fuzzy. eg: tall, short, quick eg: "peter is tall"

2. Fuzzy predicate modifiers :- In fuzzy logic there exists a wide range of predicate modifiers that act as hedges. eg: very, fairly, moderately, rather etc. These predicate modifiers are necessary for generating the values of the linguistic variables eg: "Climate is moderately cool".

3. Fuzzy quantifiers :- The fuzzy quantifiers such as most, several, many, frequently are used in fuzzy Logic, eg: "Many people are educated". A fuzzy quantifier can be interpreted as a fuzzy number or a fuzzy proposition which provides an imprecise characterization of the cardinality of one or more fuzzy or non fuzzy sets. Fuzzy quantifiers can be used to represent the meaning of propositions containing probabilities as a result, they can be used to manipulate probabilities within the fuzzy logic.

4. fuzzy qualifiers :- There are four modes of ~~...~~ qualification in fuzzy logic, which are as follows.

- fuzzy truth qualification: It is expressed as "x is τ", in which τ is a fuzzy truth value. A fuzzy truth value claims the degree of truth of a fuzzy proposition. eg:

  (paul is Young) is NOT VERY True.

  Here the qualified proposition is (paul is Young) and the qualified fuzzy truth value is "NOT very True".

- fuzzy probability qualification :- It is denoted as "x is λ" where λ is fuzzy probability. In fuzzy logic, fuzzy probability is expressed by terms such as likely, very likely, unlikely, around etc.

  eg: (paul is young) is Likely.

  Here the qualifying fuzzy probability is "Likely".

• Fuzzy posibility qualification :- It is expressed as "X is π", where π is a fuzzy possibility and can be of the following form: possible, quite possible, almost impossible. eg: (paul is Young) is Almost Impossible. Here the qualifying fuzzy possibility is "Almost Impossible".

• Fuzzy usuality qualification: It is expressed as "usually (X) = usually (X is F)." in which the subject X is a variable taking values in the universe of discourse U and the predicate F is a fuzzy subset of U and interpreted as a usual value of X denoted by U(X) = F. The propositions that are usually true or the events that have high probability of occurrence are related by the concept of usuality qualification.

Formation of Rules.

The general way of representing human knowledge is by forming natural language expressions given by

IF antecedant THEN consequent

The above expression is referred to as the IF-THEN Rule-based form. There are three general forms that exist for any linguistic variable. They are

a) assignment statements
b) conditional statements
(c) unconditional statement.

1. Assignment statements: They are of the form

y = small

Orange color = Orange

a = s

Paul is not tall and not very short

climate = autumn

Outside temperature = normal

These statements utilize "=" for assignments.

2. Conditional statements: The following are some examples

IF y is very cool THEN stop

IF A is high THEN B is low ELSE B is not low

IF temperature is high THEN climate is hot.

The conditional statements use the "IF-THEN" rule based form.

3. Unconditional statements: They can be of the form

Goto Sum.

Stop

Divide by a.

Turn the pressure low.

The assignment statements limits the value of a variable to a specific quantity.

Both conditional as well as unconditional statements place some restrictions on the consequent of the rule-based process.

The restriction statements, irrespective of conditional or unconditional statements are usually connected by linguistic connectives such as "and", "or" or "else"

Decomposition of Rules (Compound Rules)

A Compound rule is a collection of many simple rules combined together. Any compound rule structure may be decomposed and reduced to a number of simple canonical rule forms. The rules are generally based on natural language representation. The following are the methods used for decomposition of compound linguistic rule into simple canonical rule.

1. Multiple conjunctive antecedents

$$IF\ x\ is\ \underset{\sim}{A_1}, \underset{\sim}{A_2}, \ldots \underset{\sim}{A_n}\ THEN\ y\ is\ B_m$$

Assume a new fuzzy subset $\underset{\sim}{A_m}$ defined as

$$\underset{\sim}{A_m} = \underset{\sim}{A_1} \cap \underset{\sim}{A_2} \cap \underset{\sim}{A_3} \cap \ldots \cap \underset{\sim}{A_n}$$

and expressed by means of membership function.

$$\mu_{\underset{\sim}{A_m}}(x) = \min[\mu_{\underset{\sim}{A_1}}(x), \mu_{\underset{\sim}{A_2}}(x) \ldots \mu_{\underset{\sim}{A_m}}(x)]$$

The compound rule may be rewritten as

$$IF\ \underset{\sim}{A_m}\ THEN\ \underset{\sim}{B_m}$$

2. Multiple disjunctive antecedents

$$IF\ x\ is\ A_1\ OR\ x\ is\ A_2, \ldots OR\ x\ is\ A_n\ THEN\ y\ is\ B_m.$$

This can be written as

$$IF\ x\ is\ A_n\ THEN\ y\ is\ B_m$$

where the fuzzy set A_m is defined as

$$\underset{\sim}{A_m} = \underset{\sim}{A_1} \cup \underset{\sim}{A_2} \cup \underset{\sim}{A_3} \cup \ldots \cup \underset{\sim}{A_n}$$

The membership function is given by

$$\mu_{\underset{\sim}{A_m}}(x) = \max[\mu_{\underset{\sim}{A_1}}(x), \mu_{\underset{\sim}{A_2}}(x), \ldots \mu_{\underset{\sim}{A_n}}(x)]$$

which is based on fuzzy union operation.

3. Conditional Statements (with ELSE and UNLESS):

Statements of the kind

IF $\underset{\sim}{A}_1$ THEN ($\underset{\sim}{B}_1$ ELSE $\underset{\sim}{B}_2$)

Can be decomposed into two simple canonical rule forms, connected by "OR"

IF $\underset{\sim}{A}_1$ THEN $\underset{\sim}{B}_1$

OR

IF NOT $\underset{\sim}{A}_1$ THEN $\underset{\sim}{B}_2$

Rule 1: IF $\underset{\sim}{A}_1$ (THEN $\underset{\sim}{B}_1$) UNLESS $\underset{\sim}{A}_2$

Can be decomposed as

IF $\underset{\sim}{A}_1$ THEN $\underset{\sim}{B}_1$

OR

IF $\underset{\sim}{A}_2$ THEN NOT $\underset{\sim}{B}_1$

Rule 2: IF $\underset{\sim}{A}_1$ THEN ($\underset{\sim}{B}_1$) ELSE IF $\underset{\sim}{A}_2$ THEN ($\underset{\sim}{B}_2$)

Can be decomposed into the form

IF $\underset{\sim}{A}_1$ THEN $\underset{\sim}{B}_1$

OR

IF NOT $\underset{\sim}{A}_1$ AND IF $\underset{\sim}{A}_2$ THEN $\underset{\sim}{B}_2$

4. Nested-IF-THEN Rules:

The rule "IF $\underset{\sim}{A}_1$ THEN [IF $\underset{\sim}{A}_2$ THEN ($\underset{\sim}{B}_1$)]" can be of the form IF $\underset{\sim}{A}_1$ AND $\underset{\sim}{A}_2$ THEN $\underset{\sim}{B}_1$

Aggregation of fuzzy Rules

The rule based system involves more than one rule. Aggregation of rules is the process of obtaining the overall consequences from the individual consequents provided by each rule. The following two methods are used for aggregation of fuzzy rules.

1. Conjunctive system of rules: For a system of rules to be jointly satisfied, the rules are connected by "and" connectives. Here the aggregated output y, is determined by the fuzzy intersection of all individual rule consequents, y_i where $i = 1$ to n, as

$$y = y_1 \text{ and } y_2 \text{ and } \dots \text{ and } y_n$$

or

$$y = y_1 \cap y_2 \cap y_3 \cap \dots \cap y_n$$

This aggregated output can be defined by the membership function

$$\mu_y(y) = \min[\mu_{y_1}(y), \mu_{y_2}(y), \dots, \mu_{y_n}(y)] \text{ for } y \in Y$$

2. Disjunctive System of rules: In this case the satisfaction of at least one rule is required. The rules are connected by "or" connectives. Here the fuzzy union of all individual rule contribution determines the aggregated output as

$$y = y_1 \text{ or } y_2 \text{ or } \dots \text{ or } y_n$$

or

$$y = y_1 \cup y_2 \cup y_3 \cup \dots \cup y_n$$

It can be defined by the membership function

$$\mu_y(y) = \max[\mu_{y_1}(y), \mu_{y_2}(y), \dots \mu_{y_n}(y)] \text{ for } y \in Y$$

Fuzzy Inference System (FIS)

Fuzzy rule based system, fuzzy models, and fuzzy expert systems are generally known as fuzzy inference system. The key unit of a fuzzy logic system is FIS. The primary work of this system is decision making. FIS uses "IF ... THEN" rules along with connectors "OR" or "AND" for making necessary decision rules.

The input to FIS may be fuzzy or crisp, but the output from FIS is always a fuzzy set. When FIS is used as a controller, it is necessary to have crisp output.

Construction and working principle of FIS

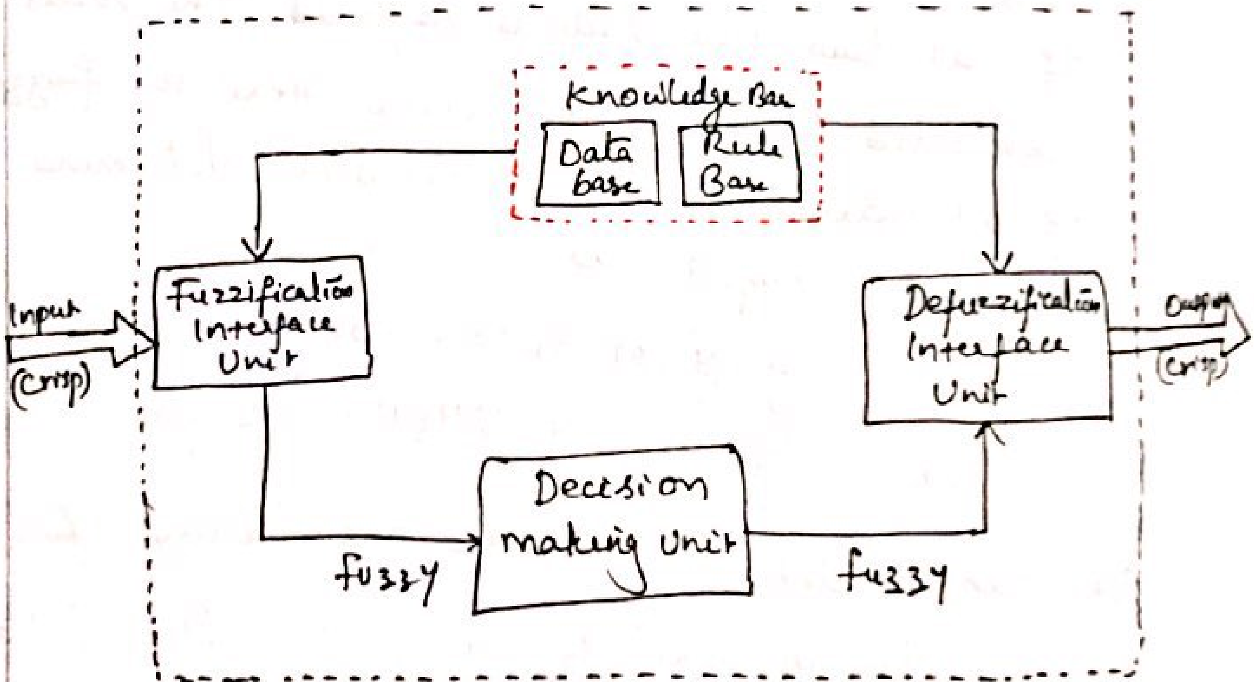


fig: Block diagram of FIS

A FIS is constructed of few functional blocks. They are

1. A rule base that contains numerous fuzzy IF-THEN rules
2. A database that defines the membership function of fuzzy sets used in fuzzy rules
3. Decision making unit that performs operation on the rules
4. Fuzzification Interface Unit that converts the fuzzy quantities into crisp quantities

The working methodology of FIS is as follows

Initially in the fuzzification unit, the crisp input is converted into a fuzzy input. Various fuzzification methods are employed for this.
After this process, rule base is formed. Database and rule base are collectively called the knowledge base. Finally defuzzification process is carried out to produce crisp output. Mainly the fuzzy rules are formed in the rule base and suitable decisions are made in the decision-making unit.

Methods of FIS

There are two important types of FIS. They are

1. Mamdani FIS (1975)
2. Sugeno FIS (1985)

The difference between the two methods lies in the consequent of fuzzy rules. Fuzzy sets are used as rule

rule consequents in Mamdani FIS and linear functions of input variables are used as rule consequents in Sugeno's method.

Mamdani FIS

Mamdani proposed this system in the year 1975 to control a steam engine and boiler combination by synthesizing a set of fuzzy rules obtained from people working on the system.
The output membership functions are expected to be fuzzy sets.
The following steps have to be followed to compute the output from this FIS

Step 1: Determine a set of fuzzy rules

Step 2: Make the inputs fuzzy using input membership function

Step 3: Combine the fuzzified inputs according to the fuzzy rules for establishing a rule strength

Step 4: Determine the consequent of the rule by combining the rule strength and the output membership function

Step 5: Combine all the consequents to get an output distribution

Step 6: Finally a defuzzified output distribution is obtained

The fuzzy rules are formed using "IF-THEN" statements and "AND/OR" connectives. The consequence of the rule can be obtained in two steps

1. by computing the rule strength completely using the fuzzified inputs from the fuzzy combination.
2. by clipping the output membership function at the rule strength.

The outputs of all the fuzzy rules are combined to obtain one fuzzy output distribution.

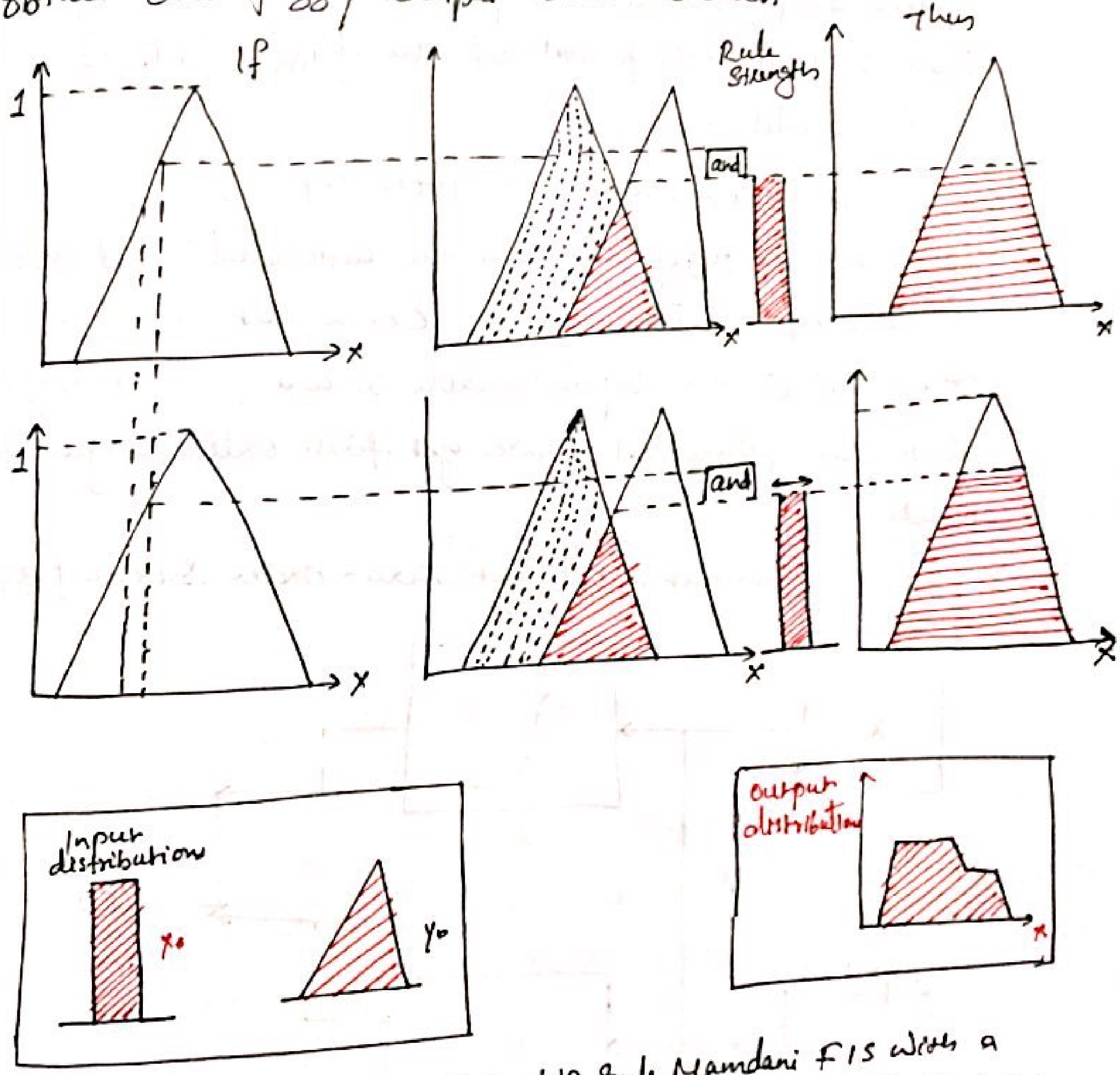


fig: A two-input, two rule Mamdani FIS with a Fuzzy input.

Consider a two input Mamdani FIS with two rules. The model fuzzifies the two inputs by finding the intersections of two crisp input value with the input membership function. The minimum operation is used to compute the fuzzy input "and" for combining the two fuzzified inputs to obtain the rule strength.

Takagi-Sugeno Fuzzy Model (TS Method)

Sugeno fuzzy method was proposed by Takagi, Sugeno and Kang in 1985. The format of the fuzzy rule of a Sugeno fuzzy model is given by

IF x is A and y is B THEN $z = f(x,y)$

where AB are fuzzy sets in the antecedents and $z = f(x,y)$ is a crisp function in the consequent. $f(x,y)$ is a polynomial in the input variable x and y. If $f(x,y)$ is a first order polynomial, we get first order Sugeno fuzzy model.

If f is constant, we get zero-order Sugeno fuzzy model

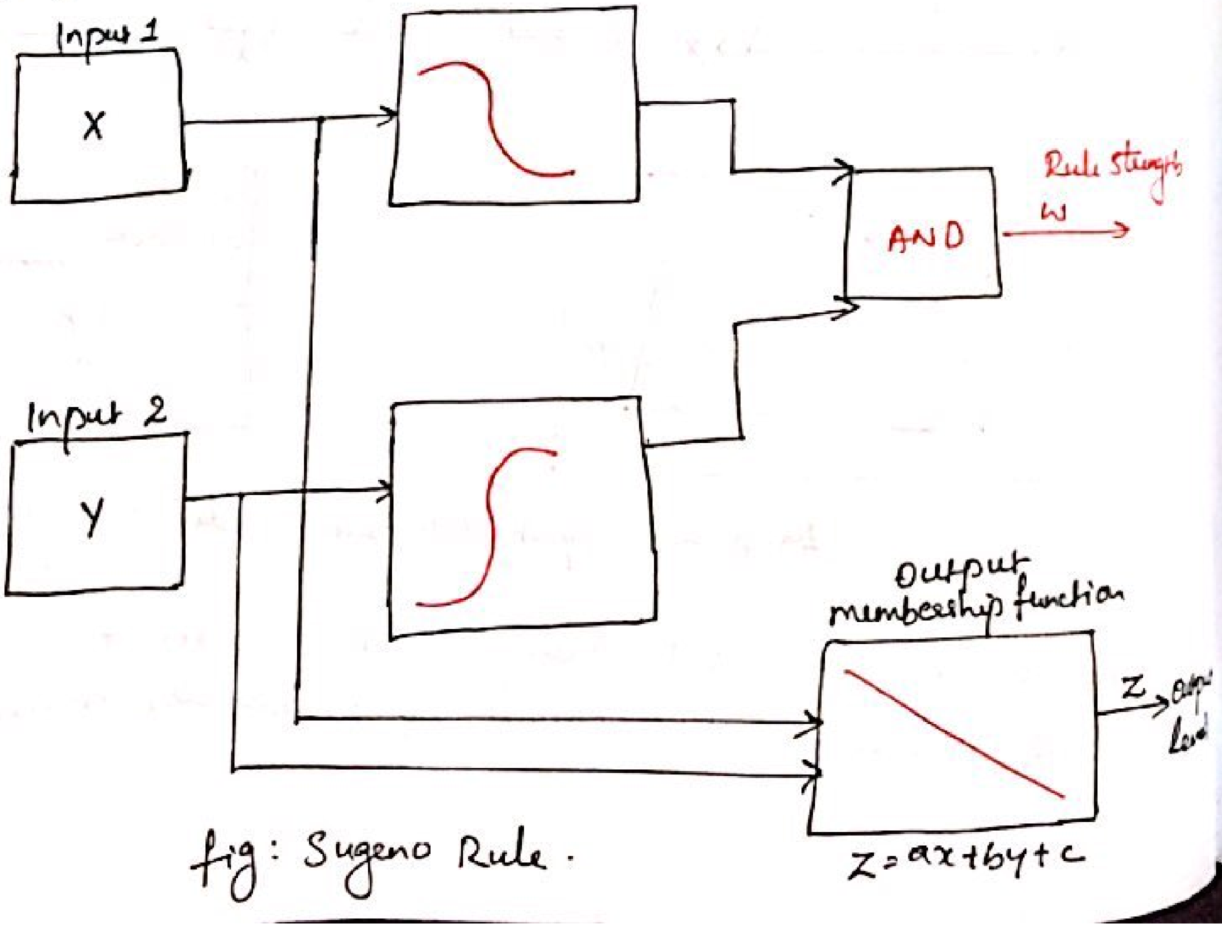


fig: Sugeno Rule.

The main steps of the fuzzy inference process namely
1. fuzzifying the inputs
2. applying the fuzzy operator
The main difference between Mamdani's and Sugeno's method is that Sugeno output membership functions are either linear or constant.
The rule format of sugeno form is given by
"IF 3 = x and 5 = y then output z = ax+by+c
For a Sugeno model of zero order, the output level z is a constant.

<u>Comparison between Mamdani and Sugeno Method</u>

The main difference between Mamdani and Sugeno method lies in the output membership function.
The Sugeno output membership function are either linear or constant.
The difference also lies in the consequents of their fuzzy rule and as a result their aggregation and defuzzification procedures differ suitably.
The configuration of Sugeno fuzzy system can be reduced and it becomes smaller than that of Mamdani fuzzy system if nontriangular or nontrapezoidal fuzzy input sets are used.
Sugeno controllers have more adjustable parameters in the rule consequent and the number of parameters grows exponentially with the increase of the number of input variable.

There exist several mathematical results for Sugeno fuzzy controllers than for Mamdani controllers.

Formation of Mamdani FIS is more easier than Sugeno FIS.

The main advantages of Mamdani method are

1. It has widespread acceptance
2. It is well suitable for human input
3. It is intuitive

The advantages of Sugeno method include.

1. It is computationally efficient
2. It is compact and works well with linear technique, optimization technique and adaptive technique.
3. It is best suited for mathematical analysis
4. It has a guaranteed continuity of the output surface.

Neuro Fuzzy Hybrid Systems

A neuro fuzzy hybrid system is a learning mechanism that utilizes the training and learning algorithms from neural networks to find parameters of a fuzzy system.
It can also be defined as a fuzzy system that determines its parameters by processing data samples by using a learning algorithm derived from or inspired by neural network theory.

It is hybrid intelligent system that fuses artifical neural networks and logic by combining the learning and connectionist structure of neural networks with human-like reasoning style of fuzzy systems.

The neuro-fuzzy is divided into two areas.

1. Linguistic fuzzy modeling focused on interpretability
2. Precise fuzzy modeling focused on accuracy

Comparison of fuzzy system with neural networks

Neural processing	Fuzzy processing
Mathematical model not necessary	Mathematical model not necessary
Learning can be done from scratch	A priori knowledge is needed.
There are several learning algorithms	Learning is not possible
Black-box behavior	Simple interpretation and Implementation

Characteristics of Neuro fuzzy System

The general architecture of neuro-fuzzy hybrid system is shown in the following figure.

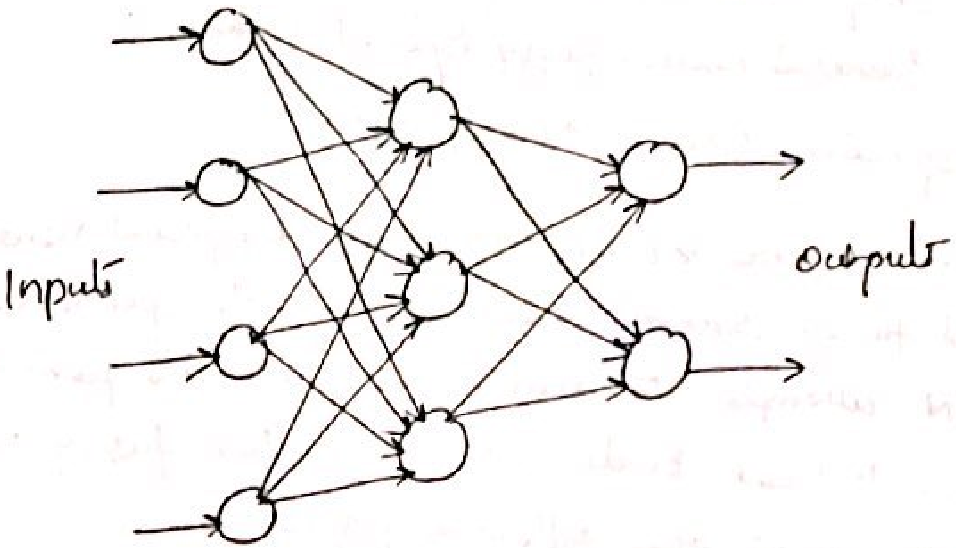


fig: Architecture of neuro-fuzzy hybrid System

A fuzzy system based NFS is trained by means of a data-driven learning method derived from neural network theory. This heuristic causes local changes in the fundamental fuzzy system.

An NFS is given by a three layer feedforward neural network model. It can also be observed that the first layer corresponds to the input variables, and the second and third layers correspond to the fuzzy rules and output variables respectively. The fuzzy sets are converted to connection weights.

NFS can be considered as a system of fuzzy rules wherein the system can be initialized in the form of fuzzy rules based on the prior knowledge available.

Classification of Neuro-Fuzzy Hybrid System

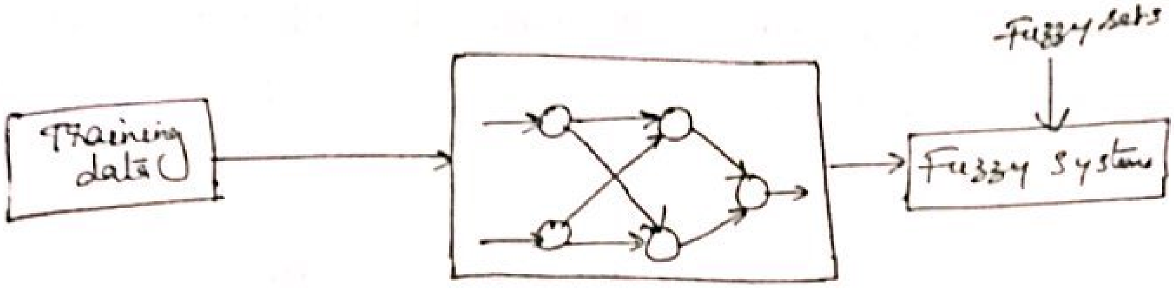
NFS can be classified into the following two systems.

1. Cooperative NFS
2. General neuro-fuzzy hybrid systems.

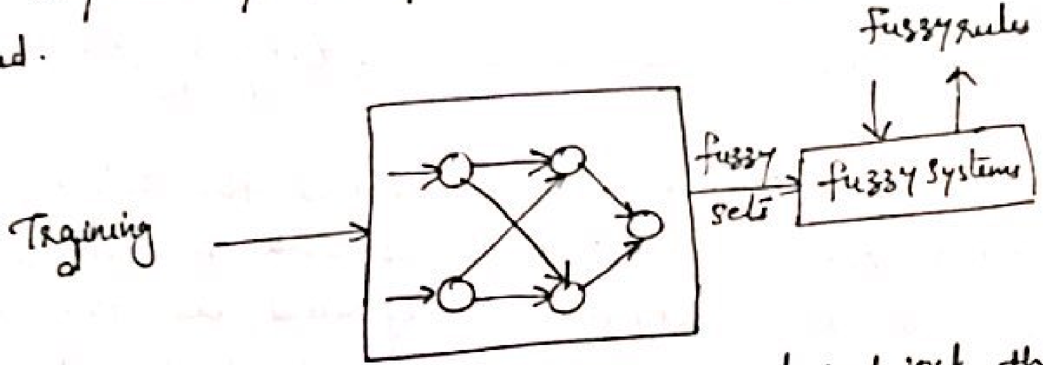
Cooperative Neural Fuzzy Systems.

In this type of system, both artificial neural network (ANN) and fuzzy system work independently from each other. The ANN attempts to learn the parameters from the fuzzy system. Four different kinds of cooperative fuzzy neural networks are shown in the following figure.

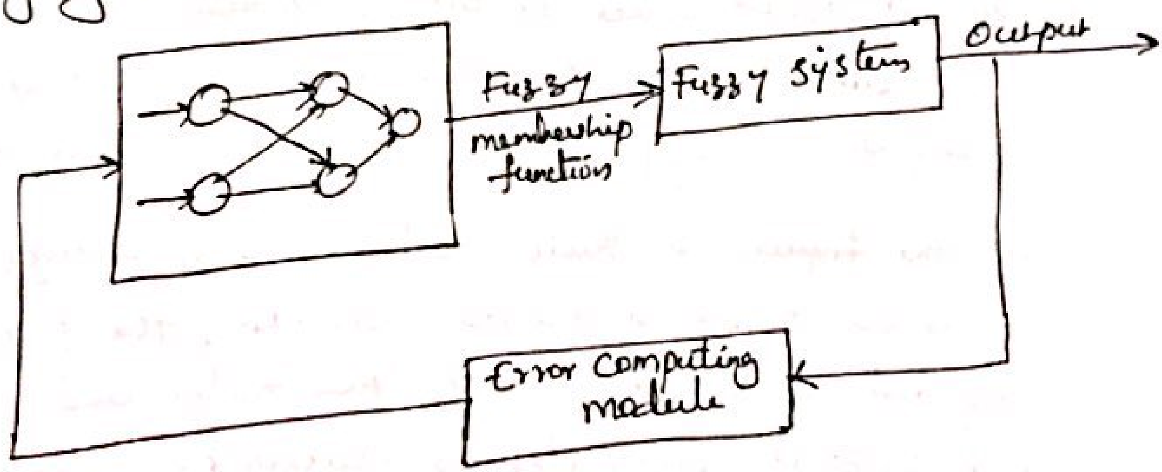
Scanned with CamScanner



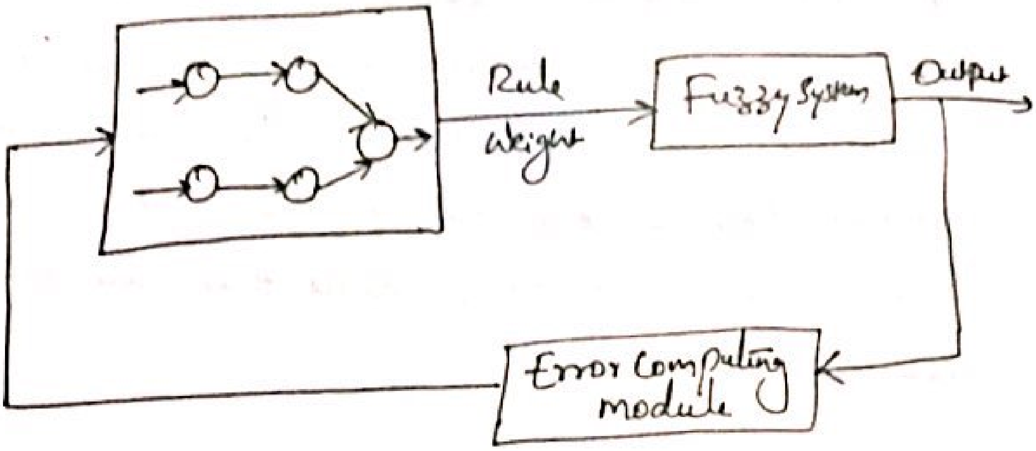
The fuzzy sets are learned from the given training data. This is done, usually by fitting membership function with a neural network, the fuzzy sets then being determined offline. This is followed by their utilization to form the fuzzy system by fuzzy rules that are given and not learned.



The above figure determines by a neural network, the fuzzy rules from the training data. Here again, the neural networks learn offline before the fuzzy system is initialized. The rule learning happens usually by clustering on self-organizing feature maps. There is also the possibility of applying fuzzy clustering methods to obtain rules.



In the above NFS the parameters of membership function are learnt online, while the fuzzy system is applied. Initially fuzzy rules and membership functions must be defined beforehand. The error has to be measured inorder to improve and guide the learning step.



The above NFS model determines the rule weights for all fuzzy rules by a neural network. A rule is determined by its rule weight - interpreted as the influence of a rule. They are then multiplied with the rule output.

General Neuro fuzzy Hybrid Systems (General NFHS)

General neuro fuzzy hybrid (NFHS) resemble neural networks where a fuzzy system is interpreted as a neural network of special kind. The architecture of general NFHS gives it an advantage because there is no communication between fuzzy system and neural network. The following figure illustrates the NFHS.

In the figure the rule rule base of a fuzzy system is assumed to be a neural network, the fuzzy sets are regarded as weights and the rules and the input and output variables as neurons.

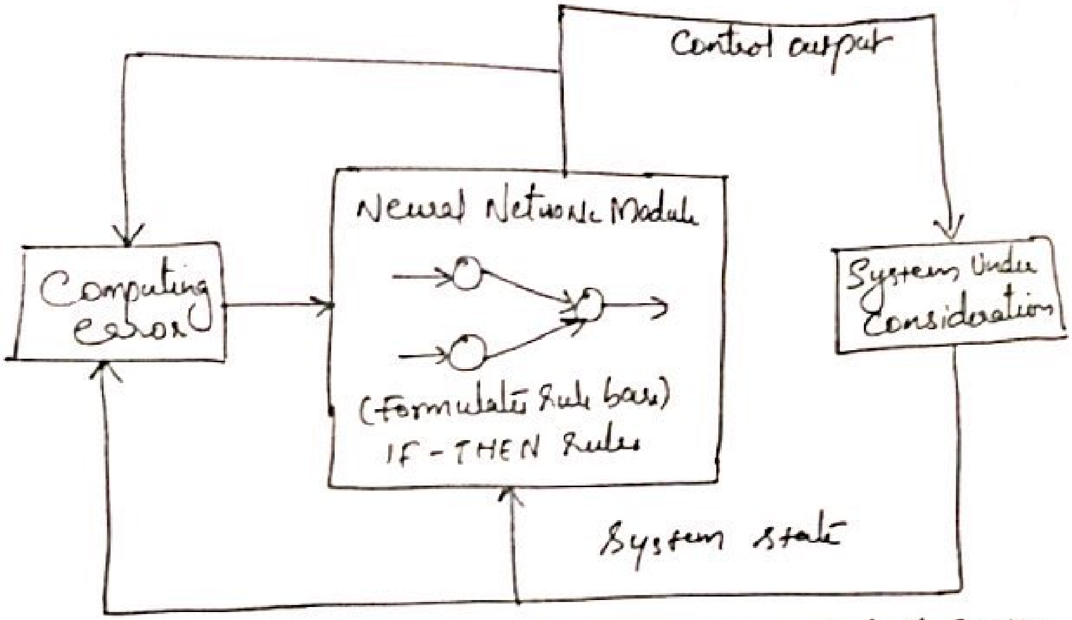


fig: General neuro Fuzzy hybrid System

The choice to include or discard neurons can be made in the learning step.

Using learning rule, the neural network must optimize the parameters by fixing a distinct shape of the membership function eg: triangular.

The neuro fuzzy hybrid system can also be modeled in another method. In this case, the training data is grouped into several clusters and each cluster is designed to represent a particular rule. These rules are defined by the crisp data points and are not defined linguistically. In this case a neural network might be applied to train the defined clusters.

The testing can be carried out by presenting a random testing sample to their trained neural network. Each and every output unit will return a degree which exceeds to fit to the antecedent of rule.

MOD-VI

Introduction to Genetic Algorithm

Genetic Algorithms (GA) are adaptive heuristic search algorithms based on the evolutionary ideas of natural selections and genetics.

GA is used to solve optimization problems, they exploit historical information to direct the search into the region of better performance within the search space.

The basic techniques of the GAs are designed to stimulate processes in natural systems necessary for evolution.

GA follows the principle "Survival of the fittest".

The science that deals with the mechanisms responsible for similarities and differences in a species called Genetics.

The science of genetics helps us to differentiate between heredity and variations and accounts for the resemblances and differences during the process of evolution.

The concept of GA are directly derived from natural evolution and heredity.

Every animal/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.

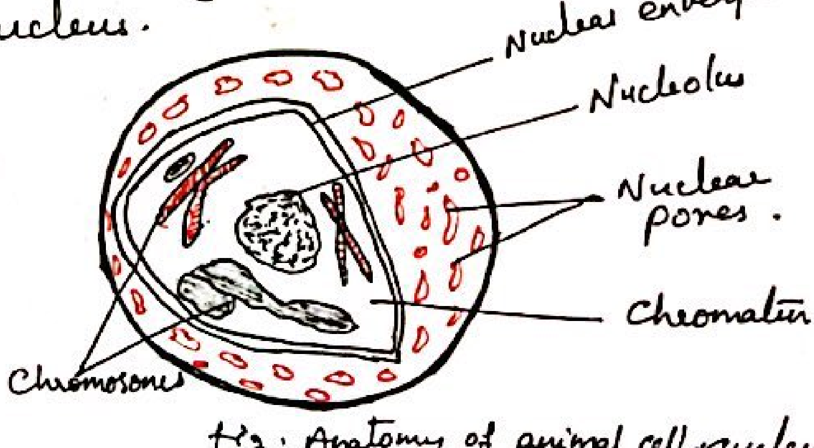


Fig: Anatomy of animal cell nucleus.

Scanned with CamScanner

Chromosomes

All the genetic information gets stored in the chromosomes. Each chromosome is build of deoxyribonucleic acid (DNA). In humans chromosomes exist in 23 pairs. The chromosome are divided into several parts called genes. Genes code the properties of species, ie the characteristics of an individual. The possibilities of combination of the genes for one property are called alleles, and a gene can take different alleles. eg:- there is a gene for eye color and the different possible alleles for eye are black, brown, blue and green.

The set of all possible alleles present in a particular population forms a gene pool. The gene pool can determine all the different possible variations for the future generations. The size of the gene pool helps in determining the diversity of the individuals in the population. The set of all the genes of a specific species is called genome. Each and every gene has a unique position on the genome called locus.

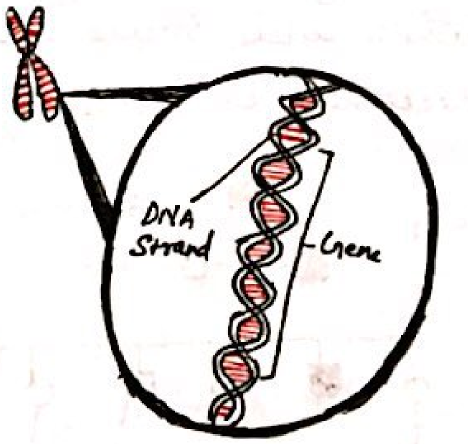


fig: Model of chromosome.

Genetics

For a particular individual, the entire combination of genes is called genotype. The phenotype describes the physical aspect of decoding the genotype to produce the phenotype.

The selection is always done on the pheonotype where the reproduction recombines genotype. Thus morphogenes plays a key role between selection and reproduction.

In highest life forms, chromosomes contain two set of genes. These are known as diploids. In the case of conflicts between two values of the same pair of genes the dominant one will determine the phenotype where as the other one called recessive will still be present and can be passed onto the offspring.

Diploidy allows a wider diversity of alleles. This provides a useful memory mechanism in changing or noisy environment.

Most of GA concentrate on haploid chromosome because they are much simple to construct. In haploid, only one set of each gene is stored, thus the process of determining which allele should be dominant and which one should be recessive is avoided.

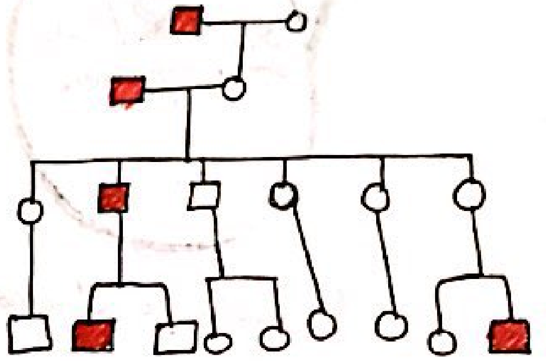


fig: Development of genotype to phenotype

Reproduction

Reproduction of species via genetic information is carried out by the following

1. Mitosis : In mitosis the same genetic information is copied to new offspring. There is no exchange of information. This is a normal way of growing of multicell structures such as organs. The following figure shows mitosis form of reproduction

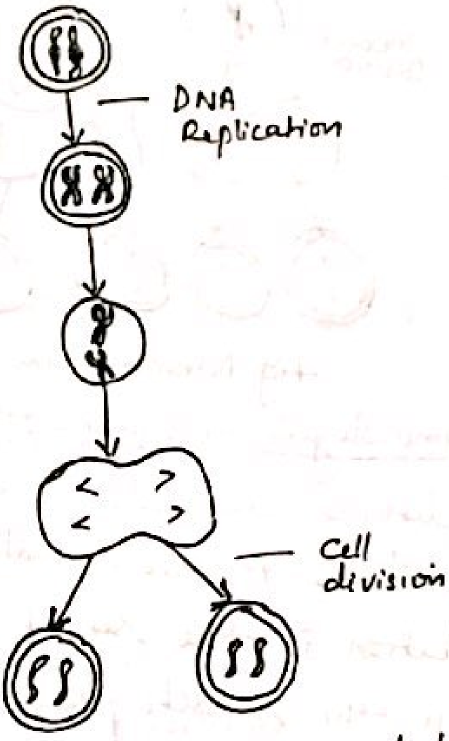


fig: Mitosis form of Reproduction

2. Meiosis : Meiosis forms the basis of sexual production when meiotic division takes place, two gametes appear in the process. When reproduction occurs then two gametes conjugate to a zygote which becomes the new individual. In this case genetic information is shared between the parents in order to create a new offspring. The following figure shows meiosis form of reproduction.

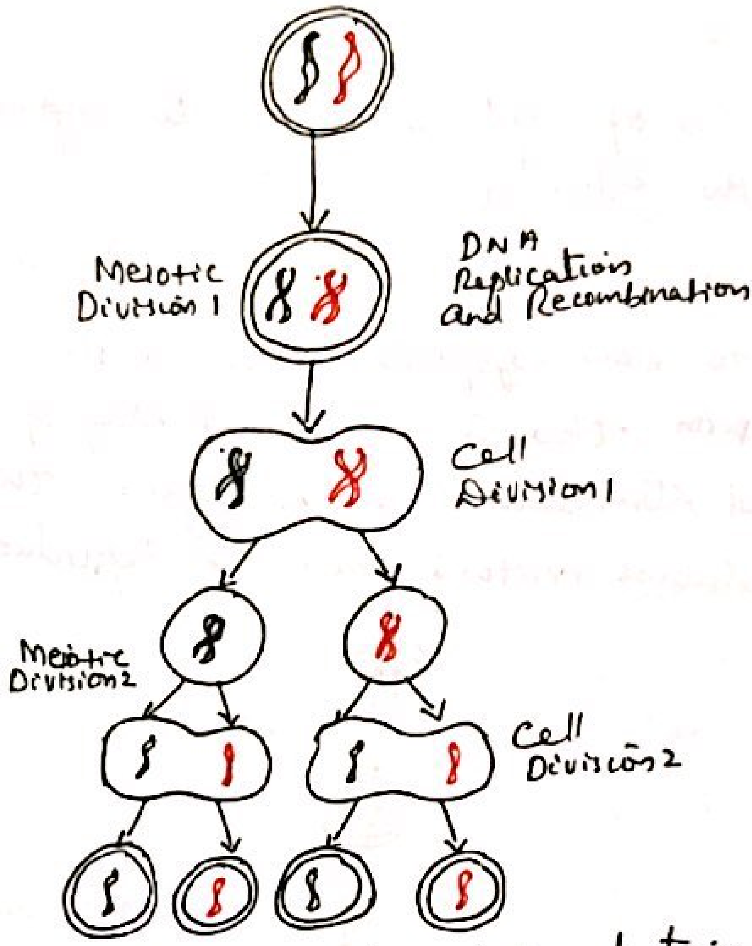


fig: Meiosis form of reproduction.

Basic Terminologies In Genetic Algorithm.

The two distinct elements in the GA are individuals and population. An individual is a single solution while the population is the set of individuals currently involved in the search process.

Individuals

An individual is a single solution. An individual group form two solutions.

1. The chromosome which is the raw genetic information that the GA deals.
2. The phenotype which is the expressive of the chromosome in the terms of the model.

A chromosome is subdivided into genes. A gene is the GA's representation of a single factor for a control factor.

Each factor in the solution set corresponds to a gene in the chromosome. The following figure shows the representation of a genotype

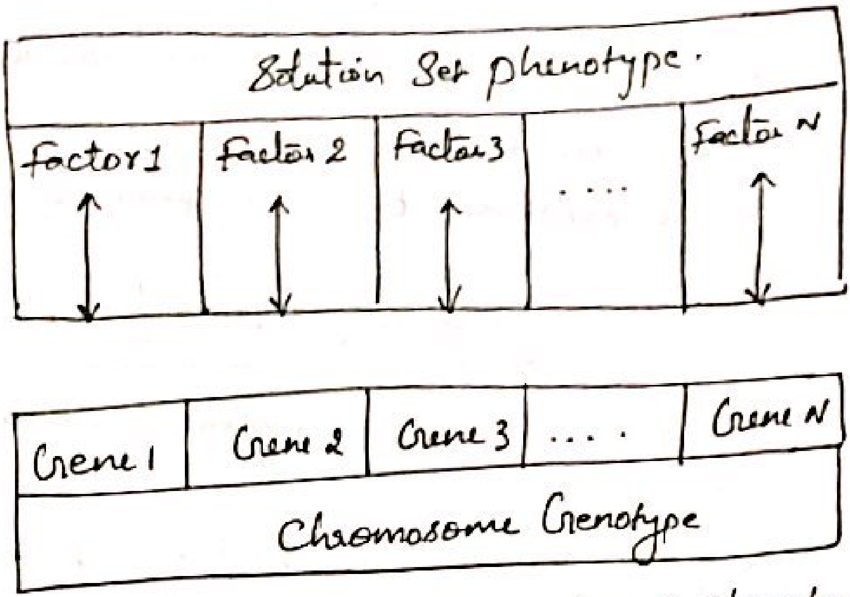


fig: Representation of genotype & phenotype.

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

fig: Representation of a chromosome.

A chromosome should in some way contain information about the solution that it represents. The morphogenesis function associates each genotype with its phenotype. Each chromosome must define one unique solution, but it does not mean that each solution is encoded by exactly one chromosome.

<u>Genes:</u>

Genes are the basic instructions for building a GA. A chromosome is a sequence of genes. Genes may describe a possible solution to a problem, without actually having the solution. A gene is a bit string of arbitrary lengths.

The bit string is a binary representation of number of intervals from a lower bound. A gene is the GA's representation of a single factor value for a control factor, where control factor must have an upper bound and a lower bound. This range can be divided into the number of intervals that can be expressed by the gene's bit string.

1010	1110	1111	0101

fig: Representation of a gene

Fitness

The fitness of an individual in a GA is the value of an objective function for its phenotype. For calculating fitness, the chromosome has to be first decoded and the objective function has to be evaluated. The fitness not only indicates how good the solution is, but also corresponds to how close the chromosome is to the optimal one.

Population

A population is a collection of individuals. A population consists of a number of individuals being tested, the phenotype parameters defining the individuals and some information about the search space. The two important aspects of population used in GAs are

1. The initial population generation
2. The population size.

For each and every problem, the population size will depend on the complexity of the problem. It is a random initialization of population. In this case of a binary coded chromosome this means that each bit is initialized to random 0 or 1.

population being combination of various chromosomes is represented in the following figure. It consist of 4 chromosome

population	Chromosome 1	11100010
	chromosome 2	01111011
	Chromosome 3	10101010
	chromosome 4	11001100

fig: population

Simple GA

GA handles a population of possible solutions. Each solution is represented through a chromosome which is just an abstract representation. Coding all the possible solutions into a chromosome is the first part, but it is not the straightforward solution of the GA.

Reproduction operators are applied directly on the chromosome, and are used to perform mutations and recombinations over solutions of the problem.

The simple form of GA is given by the following

1. Start with a randomly generated population
2. Calculate the fitness of each chromosomes in the population.

3. Repeat the following steps until n offsprings have been created.

- Select a pair of parent chromosomes from the current population
- With probability pc crossover the pair at a randomly chosen point to form two offsprings
- Mutate the two offsprings at each locus with probability pm

4. Replace the current population with the new population

5. Go to step 2.

Each iteration in the GA consists of the following steps

1. Selection : The first step consists in selecting individual for reproduction. This selection is done randomly with a probability depending on the relative fitness of the individuals so that best ones are often chosen for reproduction rather than poor ones.

2: Reproduction: In the second step, offspring are bred by selected individuals, for generating new chromosomes, the algorithm can use both recombination and mutation.

3. Evaluation : Then the fitness of the new chromosomes is evaluated

4. Replacement : During the last step, individuals from the old population are killed and replaced by the new ones.

The algorithm is stopped when the population converge towards the optimal solution.

```
BEGIN
Generate initial population
Compute fitness of each individual
WHILE NOT finished DO LOOP
BEGIN
  Select individuals from old generations
  for mating;
  Create offspring by applying
    recombination and/or mutation
    to the selected individuals.
  Compute fitness of the new individuals
  kill old individuals to make room for
    new chromosomes and insert
    offspring in the new generalization
  IF population has converged
     THEN finishes = TRUE;
  END
END.
```

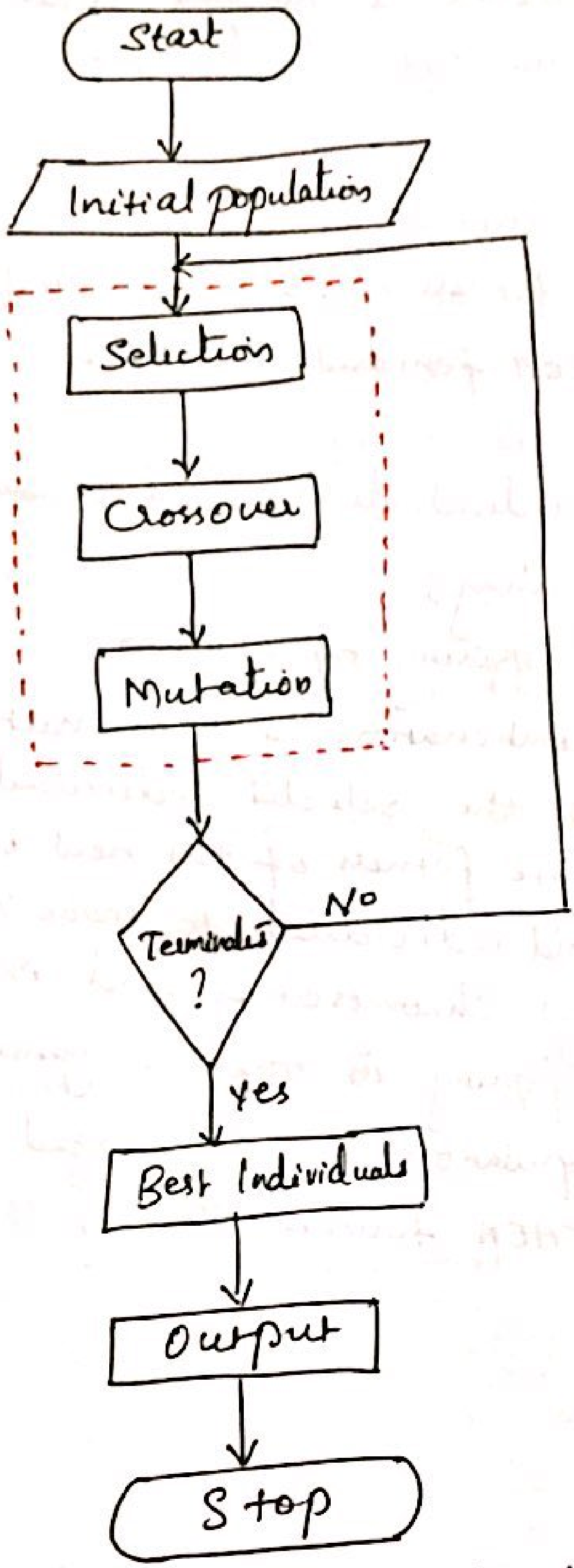


fig: flowchart of genetic Algorithm

General Genetic Algorithm

Step 1: Create a random initial state: An initial population is created from a random selection of solutions

Step 2: Evaluate fitness: A value for fitness is assigned to each solution depending on how close it actually is to solving the problem

Step 3: Reproduce (and Children mutate): Those chromosomes with a higher fitness value are more likely to reproduce offspring. The offspring is a product of the father and mother, whose composition consists of a combination of genes from the two.

Step 4: Next generation: If the new generation contains a solution that produces an output that is close enough or equal to the desired answer then the problem has been solved.

Operators in Genetic Algorithm

The basic operators in Genetic Algorithm include

- Encoding
- Selection
- Recombination and mutation

Encoding

Encoding is the process of representing individual genes. The process can be performed using bits, numbers, trees arrays. lists or any other objects. The encoding depends mainly on solving the problem. eg: One can encode directly real or integer numbers.

Binary Encoding

The most common way of encoding is a binary string, which would be represented as in the following figure.

Chromosome 1	1 1 0 1 0 0 0 1 1 0 1 0
Chromosome 2	0 1 1 1 1 1 1 1 1 1 0 0

Each chromosomes encodes a binary (bit) string. Each bit in the string can represent some characteristics of the solution. Every bit string therefore is a solution but not a best solution. Another possibility is that the whole string can represent a number.

Binary encoding gives many possible chromosomes with a smaller number of alleles. Binary coded strings with 1's and 0's are mostly used. The length of the string depends on the accuracy

In such coding

1. Integers are represented exactly
2. Finite number of real numbers can be represented
3. Number of real numbers represented increases with string length.

Octal Encoding

This encoding uses string made up of octal numbers (0-7)

Chromosome 1	03467216
Chromosome 2	15723314

Hexadecimal Encoding

This encoding uses String made up of hexadecimal numbers (0-9, A-F)

Chromosome 1	9CE7
Chromosome 2	3DBA

Permutation Encoding (Real Number Coding)

Every chromosome is a string of numbers, Represented in Sequence. Sometimes Corrections have to be done after genetic operation is complete. In permutation encoding, every chromosome is a string of integer/Real values, which represents number in a Sequence.

Permutation encoding is only useful for ordering problems.

Chromosome A	1 5 3 2 6 4 7 9 8
Chromosome B	8 5 6 7 2 3 1 4 9

Value Encoding

Every chromosome is a string of values and the values can be anything Connected to the problem. This encoding produces best results for some special problems. On the other hand, it is often necessary to develop new genetic operators specific to the problem.

Direct value encoding can be used in problems, where some Complicated values, Such as Real numbers are used.

In value encoding every chromosome is a string of some values. Values can be anything connected to problem form numbers, real numbers or characters to some complicated objects. Value encoding is very good for some special problems.

Chromosome A	1.2324 5.3243 0.4556 2.3293 2.4545
Chromosome B	ABDJEIFJDHDIERJFDLDFLFEGT
Chromosome C	(Back), (back), (right), (forward), (left)

fig: value encoding

Tree Encoding

It is mainly used for evolving program expressions for genetic programming. Every chromosome is a tree of some objects such as functions and commands of a programming language.

Selection

Selection is the process of choosing two parents from the population for crossing. After encoding step, next step is to decide how to perform selection i.e how to choose individuals in the population that will create offspring for the next generation

Chromosomes are selected from the initial population to be parents for reproduction. According to Darwin's theory of evolution the best ones survive to create new offspring.

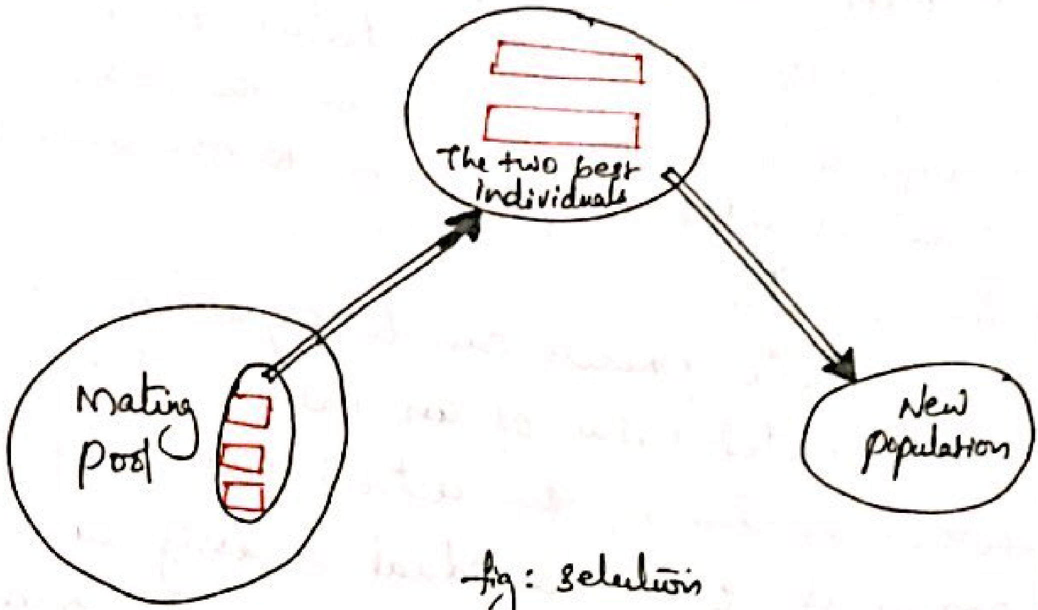


fig: selection

Selection is a method that randomly picks chromosomes out of the population according to their evaluation function. The higher the fitness function, the better chance that an individual will be selected.

The selection pressure is defined as the degree to which the better individuals are favored. The higher the selection pressured the more the better individuals are favored.

Types of Selection

1. proportionate based selection
2. Ordinal-based selection

Proportionate based selection picks out individuals based upon their fitness values relative to the fitness of the other individuals in the population

Ordinal based - selection schemes select individuals not upon their raw fitness, but upon the rank within the population.

Roulette wheel selection

The principle of Roulette selection is a linear search through a Roulette wheel with the slots in the wheel weighted in proportion to the individual fitness value.

The Roulette process can be explained as follows.
The expected value of an individual is individuals fitness divided by the actual fitness of the population. Each individual is assigned a slice of the Roulette wheel, the size of the slice being proportional to the individuals fitness. The wheel is spun N times, where N is the number of individuals in the population. On each spin, the individual under the wheels marker is selected to be in the pool of parents for the next generation.

1. Sum the total expected value of the individuals in the population. Let it be T
2. Repeat N times
   i. choose a random integer "r" between 0 and T
   ii. Loop through the individuals in the population summing the expected values, until the sum is greater than or equal to "r". The individual whose expected value puts the sum over the limit is the one selected.

Random Selection

This technique randomly selects a parent from the population. In terms of disruption of genetic codes, random selection is little more disruptive, on average, than Roulette wheel selection.

Rank Selection

Rank selection ranks the population and every chromosome receives fitness from the ranking. The worst has fitness 1 and the best has fitness N. There are many ways this rank selection can be achieved and two suggestions are.

1. Select a pair of individuals at random. Generate a random number R between 0 and 1. If R<r use the first individual as a parent. If the R≥r then use the second individual as the parent. This is repeated to select the second parent. The value of r is a parameter to this method.
2. Select two individuals at random. The individuals with the highest evaluation becomes the parent. Repeat to find the second partner

Tournament Selection

The best individual from the tournament is the one with the highest fitness, who is the winner of the tournament competitions and the winner are then inserted into the mating pool. The tournament competition is repeated until the mating pool for generating new offspring is filled. This method is more efficient and leads to an optimal solution

Boltzmann Selection

The probability that the best string is selected and introduced into the mating pool is very high. Elitism can be used to eliminate the chance of any undesired loss of information during the mutation stage.

Elitism :-
The first best chromosome or the few best chromosome are copied to the new population. Such individuals can be lost if they are not selected to reproduce or if crossover or mutation destroys them. This significantly improves the GA's performance.

Crossover (Recombination)

Crossover is the process of taking two parent solution and producing from them a child. After the selection process, the population is enriched with better individuals. Reproduction makes clones of good strings but does not create new ones. Crossover operator is applied to the mating pool with the hope that it creates a better offspring.

Crossover is a recombination operator that proceeds in three steps:

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.

Various types of crossovers are given as follows.

Single-point crossover

The traditional genetic algorithm uses single-point crossover, where the two mating chromosomes are cut once at corresponding points and the sections after the cuts are exchanged.

A cross site or crossover point is selected randomly along the length of the mated strings and bits next to the cross sites are exchanged. If appropriate site is chosen, better children can be obtained by combining good parents.

parent 1	1 0 1 1 0 ¦ 0 1 0
parent 2	1 0 1 0 1 ¦ 1 1 1

↓

child 1	1 0 1 1 0 ¦ 1 1 0
child 2	1 0 1 0 1 ¦ 0 1 1

fig: Single point crossover

Two-point Crossover.

In this crossover, many different crossover algorithms have been devised, often involving more than one cut point.

Note :-

Adding more crossover points ~~are chosen and the~~ reduces the performance of the GA.

In two point crossover, two crossover points are chosen and the contents between these points are exchanged between two mated parents.

In the following figure the dotted lines indicate the crossover points. Thus the contents between these points are exchanged between the parents to produce new children for mating in the next generation

parent 1	1 1 : 0 1 1 : 0 1 0
parent 2	0 1 : 1 0 1 : 1 0 0

Child 1	1 1 : 1 0 1 : 0 1 0
Child 2	0 1 : 0 1 1 : 1 0 0

fig: Two point Crossover

Multipoint crossover (N-point crossover)

There are two ways in this crossover. One is even number of cross site and the other odd number of cross site. In the case of even number of cross site, the cross sites are selected randomly around a circle and information is exchanged. In the case of odd number of cross sites a different cross point is always assumed at the string beginning

Uniform Crossover

Uniform crossover is different from the N-point crossover. Each gene in the offspring is created by copying the corresponding gene from one or the other parent chosen according to a random generated binary crossover mask of the same length as the chromosome. Where there is 1 in the crossover mask, the gene is copied from the first parent, and where there is 0 in the mask the gene is copied from the second parent. A new crossover mask is randomly generated for each pair of parents. offspring contains a mixture of genes from each parent.

parent 1	1 0 1 1 0 0 1 1
parent 2	0 0 0 1 1 0 1 0
Mask	$\boxed{1}$ $\boxed{1}$ 0 $\boxed{1}$ 0 $\boxed{1}$ $\boxed{1}$ 0
child 1	1 0 0 1 1 0 1 0
child 2	0 0 1 1 0 0 1 1

Three parent crossover.

In the crossover technique, three parents are randomly chosen. Each bit of the first parent is compared with the bit of the second parent. If both are the same, the bit is taken for the offspring. Otherwise the bit from the third parent is taken for the offspring.

parent 1	11010001
parent 2	01101001
parent 3	01101100
Child	01101001

Crossover With Reduced Surrogate.

The reduced surrogate operator constraints crossover to always produce new individuals wherever possible. This is implemented by restricting the location of crossover points such that crossover points only occur where gene values differ.

Shuffle crossover

Shuffle crossover is related to uniform crossover. A single crossover position is selected. Before the variables are exchanged, they are randomly shuffled in both parents. After recombination, the variables in the offspring are unshuffled. This removes positional bias as the variables are randomly reassigned each time crossover is performed.

Precedence Preservative Crossover

Precedence preservative crossover (PPX) was developed for vehicle routing problem. The operator passes on precedence relations of operations given in two parental permutation to one offspring at the same rate, while the no new precedence relations are introduced. The operators works as follows.

1. A vector of length Σ, sub $i = 1$ to m_i representing the number of operations involved in the problem, is randomly filled with elements of the set {1,2}
2. This vector defines the order in which the operations are successively drawn from parent 1 to parent 2.
3. The parents and offspring permutations can be considered as list, for which the operations "append" and "delete" are defined.
4. We start by initializing an empty offspring
5. The leftmost operation in one of the two parents is selected in accordance with the order of parents given in the vector
6. After an operation is selected, it is deleted in both parents.
7. Finally the selected operation is appended to the offspring.
8. Step 7 is repeated until both parents are empty and the offspring contains all operations involved

parent permutation 1	A B C D E F
parent permutation 2	C A B F D E
Select parent no.(1/2)	1 2 1 1 2 2
Offspring permutation	A C B D F E

fig: precedence preservative crossover

Ordered Crossover

Ordered two-point crossover is used when the problem is order based, Given two parent Chromosomes, two random crossover points are selected partitioning them into left, middle and right portion. The ordered two point crossover behaves in the following way.

Child 1 inherits its left and right section from parent 1, and its middle section is determined by the genes in the middle section of parent 1 in the order in which the values appear in parent 2. A similar process is applied to determine child 2.

parent 1 : 4 2 | 1 3 | 6 5 child 1 : 4 2 | 3 1 | 6 5

parent 2 : 2 3 | 1 4 | 5 6 child 2 : 2 3 | 4 1 | 5 6

fig: Ordered crossover.

partially Matched crossover (PMX)

PMX proceeds as follows.

1. The two chromosomes are aligned
2. Two crossing sites are selected uniformly at random along the strings, defining a matching section

3. The matching section is used to effect a cross through position-by-position exchange operation

4. Alleles are moved to their new positions in the offspring.

Name 9 8 4 . 5 6 7 . 1 3 2 10 Allele 101.001.1100

Name 8 7 1 . 2 3 10 . 9 5 4 6 Allele 111.011.1101

fig: Given string

In the above given string the dots mark the selected cross points. The matching section defines the position-wise exchanges that must take place in both parents to produce the offspring. The exchanges are read from the matching section of one chromosome to that of the other. In the given example, the numbers that exchange places are 5 and 2, 6 and 3, 7 and 10. The resulting offspring are shown in the following figure.

Name 9 8 4 . 2 3 10 . 1 6 5 7 Allele 101.010.1001

Name 8 10 1 . 5 6 7 . 9 2 4 3 Allele 111.111.1001

fig: partially matched crossover.

<u>Crossover probability</u>

The basic parameter in crossover technique is the crossover probability (Pc). Crossover probability is a parameter to describe how often crossover will be performed. If there is no crossover, offspring are exact copies of parents. If there is crossover, offspring are made from parts of both parents chromosome. If crossover probability is 100% then all offspring are made by crossover. If it is 0% whole new generation is made from exact copies of chromosomes from old population.

Mutation

After crossover the strings are subjected to the mutation. Mutation prevents the algorithm to be trapped in a local minimum. Mutation plays the role of recovering the lost genetic material as well as for randomly distributing genetic information. Mutation has be considered as a simple search operator.

There are many different forms of mutation for the different kinds of representation. For binary representation, a simple mutation can consist in inverting the value of each gene with a small probability. The probability is usually 1/L when L is the length of chromosome.

Following are the types of mutation

Flipping

Flipping is a bit involves changing 0 to 1 and 1 to 0 based on a mutation chromosome generated. In the following figure A parent is considered and a mutation chromosome is randomly generated for a 1 in mutation chromosome, the corresponding bit in parent chromosome is flipped (0 to 1 and 1 to 0) and the child chromosome is produced. In the given eg: 1 occurs at 3 places of mutation chromosome and the corresponding bits in parent chromosome are flipped and child is generated

parent	1 0 1 1 0 1 0 1
Mutation chromosome	1 0 0 0 1 0 0 1
Child	0 0 1 1 1 1 0 0

fig: Mutation flipping

<u>Interchanging</u>

Two random positions of the string are chosen and the bits corresponding to those positions are interchanged

parent	1 0 1 1 0 1 0 1
Child	1 1 1 1 0 0 0 1

<u>Reversing</u>

A random position is chosen and the bits next to that position are reversed and child chromosome is produced.

parent	1 0 1 1 0 ¦ 1 0 1
Child	1 0 1 1 0 ¦ 1 1 0

fig: Reversing.

<u>Mutation probability</u>

Mutation probability is an important parameter in mutation techniques. It decides how often parts of chromosome will be mutated. If there is no mutation, offspring are generated immediately after crossover without any change. If mutation is performed, one or more parts of the chromosome are changed. If mutation probability is 100% whole chromosome is changed. If it is 0% nothing is changed.

Stopping Condition for Genetic Algorithm flow

Various stopping conditions are listed as follows.

1. Maximum generation : The GA stops when the specified number of generation has evolved

2. Elapsed time : The genetic process will end when a specified time has elapsed.

Note :- If the maximum number of generation has been reached before the specified time has elapsed, the process will end.

3. No change in fitness : The genetic process will end if there is no change to the population best fitness for a specified number of generation

Note :- If the maximum number of generation has been reached before the specified number of generation with no changes has been reached, the process will end.

4. Stall Generation : The algorithm stops if there is no improvement in the objective function for a sequence of consecutive generation of length "stall generation"

5. Stall time limit : The algorithm stops if there is no improvement in the objective function during an interval of time in seconds equal to "Stall time limit".

Best Individual.

A best individual convergence criterion stops the search once the minimum fitness in the population drops below the convergence value. This brings the search to a faster conclusion, guaranteeing at least one good solution.

Worst Individual

Worst individual terminates the search when the least fit individuals in the population have fitness less than the convergence criteria. This guarantees the entire population to be of minimum standard. Although the best individual may not be significantly better than the worst.

Sum of fitness

In this termination scheme, the search is considered to have satisfaction converged when the sum of the fitness in the entire population is less than or equal to the convergence value in the population record. This guarantees that virtually all individuals in the population will be within a particular fitness range.

Median Fitness

Here at least half of the individuals will be better than or equal to the convergence value, which should give a good range of solutions to choose from.

Genetic Neuro Hybrid System

A neuro-genetic hybrid system or a genetic neuro hybrid system is one in which a neural network employs a genetic algorithm to optimize its structural parameters that defines its architecture.

Neural networks learn and execute different tasks using several examples, classify phenomena and model nonlinear relationships, that is neural networks solve problem by self-learning and self organizing. On the other hand, genetic algorithms present themselves as a potential solution for the optimization of parameters of neural networks.

Properties of Genetic Neuro hybrid System.

Certain properties of genetic neuro-hybrid systems are as follows.

1. The parameters of neural network are encoded by genetic algorithms as a string of properties of the network, that is chromosomes. A large population of chromosomes is generated, which represent the many possible parameter sets for the given neural network.
2. Genetic Algorithm - Neural Network or GANN has the ability to locate the neighbourhood of the optimal solution quickly, compared to other conventional search strategies

The following figure shows the block diagram of genetic neuro hybrid system.

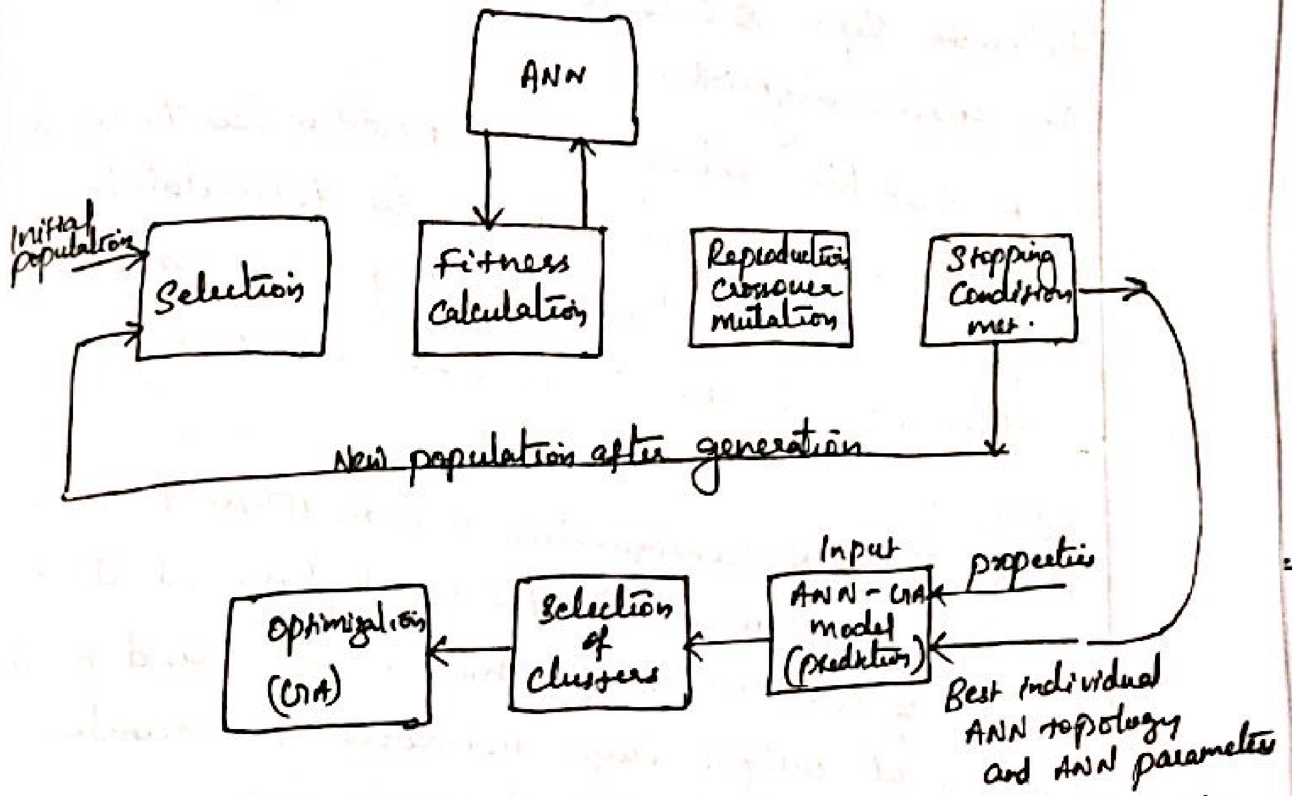


fig: Block diagram of genetic-neuro hybrids

Drawbacks.

* Large amount of memory required for handling and manipulation of chromosomes.
* Issue of Scalability as the size of networks becomes large.

Genetic Algorithm based Back-propagation Network (BPN)

BPN is a method of teaching multi layer neural networks how to perform a given task. Learning occurs during training phase.

The limitations of BPN are as follows.

1. BPN do not have the ability to recognize new patterns, they can recognize patterns similar to those they have learnt.
2. They must be sufficiently trained so that enough general features applicable to both seen and unseen instances can be extracted.

Following steps should be executed before the execution of · genetic algorithm.

1. A suitable coding for the problem has to be devised
2. A fitness function has to be formulated
3. parents have to be selected for reproduction and then crossed over to generate offspring.

Coding

Assume a BPN Configuration n-l-m where n is the number of neurons in the input layer, l is the number of neurons in the hidden layer and m is the number of output layer neurons. The number of weights to be determined is given by

$$(n+m)l$$

Each weight (gene) is a real number. Let d be the number of digits (gene length) in weight. Then a string S of decimal values having string length $(n+m)ld$ is randomly generated. It is a string that represents weight matrices of input hidden and the hidden output layers in a linear form arranged as row-major or column-major depending upon the style selected.

Weight Extraction

To determine the fitness values, weights are extracted from each chromosome. Let $a_1, a_2, \ldots a_d, \ldots a_l$ represent a chromosome and let $a_{pd+1}, a_{pd+2}, \ldots, a_{(p+1)d}$ represent p^{th} gene $(p \geq 0)$ in the chromosome

The actual weight W_p is given by

$$W_p = \begin{cases} \dfrac{a_{pd+2}10^{d-2} + a_{pd+3}10^{d-3} + \dots + a_{(p+1)d}}{10^{d-2}} & \text{if } 0 \le a_{pd+1} < 5 \\ \dfrac{+a_{pd+2}10^{d-2} + a_{pd+3}10^{d-3} + \dots + a_{(p+d)d}}{10^{d-2}} & \text{if } 5 \le a_{pd+1} \le 9 \end{cases}$$

Fitness Function

A fitness has to be formulated for each and every problem to be solved. Consider the matrix given by

$$\left\{ \begin{array}{l} (x_{11}, x_{21}, x_{31}, \dots, x_n)\ (y_{11}, y_{21}, y_{31}, \dots, y_{n1}) \\ (x_{12}, x_{22}, x_{32}, \dots, x_{n2})\ (y_{12}, y_{22}, y_{32}, \dots y_{n2}) \\ (x_{13}, x_{23}, x_{33}, \dots, x_{n3})\ (y_{13}, y_{23}, y_{33} \dots, y_{n3}) \\ \vdots \\ (x_{1m}, x_{2m}, x_{3m}, \dots, x_{nm})\ (y_{1m}, y_{2m}, y_{3m}, \dots y_{nm}) \end{array} \right\}.$$

where x and y are the inputs and targets. Compute initial population I_0 of size 'j'. Let $O_{10}, O_{20}, \dots O_{j0}$ represent 'j' chromosomes of the initial population I_0. Let the weights extracted for each of the chromosomes upto j^{th} chromosomes be $w_{10}, w_{20}, w_{30} \dots w_{j0}$. For n number of inputs and m number of outputs, let the calculated output of the considered BPN be

$$\left\{ \begin{array}{l} c_{11}, c_{21}, c_{31}, \dots, c_{n1} \\ c_{12}, c_{22}, c_{32}, \dots, c_{n2} \\ c_{13}, c_{23}, c_{33}, \dots, c_{n3} \\ \vdots \\ c_{1m}, c_{2m}, c_{3m}, \dots c_{nm} \end{array} \right\}$$

As a result the error is calculated as

$$ER_1 = (y_{11} - c_{11})^2 + (y_{21} - c_{21})^2 + (y_{31} - c_{31})^2 + \cdots + (y_{n1} - c_{n1})^2$$

$$ER_2 = (y_{12} - c_{12})^2 + (y_{22} - c_{22})^2 + (y_{32} - c_{32})^2 + \cdots + (y_{n2} - c_{n2})^2$$

$$\cdots$$

$$ER_m = (y_{1m} - c_{1m})^2 + (y_{2m} - c_{2m})^2 + (y_{3m} - c_{3m})^2 + \cdots + (y_{nm} - c_{nm})^2$$

The fitness function is further derived from this root mean square error given by

$$Ffn = \frac{1}{E}$$

The process has to be carried out for all the total number of chromosomes

Reproduction of offspring

A mating pool is formulated before the parents produce the offspring with better fitness. This is accomplished by neglecting the chromosome with minimum fitness and replacing it with a chromosome having maximum fitness. The fittest individuals among the chromosomes will be given more chances to participate in the generations and the worst individuals will be eliminated. Once the mating pool is formulated, parent pairs are selected randomly and the chromosomes of respective pairs are combined using crossover technique to reproduce offspring.

Convergence:

The convergence for genetic algorithm is the number of generations with which the fitness value increases towards the global optimum. Convergence is the progression towards increasing uniformity.

Advantages of neuro genetic Hybrid

The various advantages of neuro genetic hybrid as follows

- GA performs optimization of neural network parameters with simplicity, ease of operation, minimal requirements and global perspective.
- GA helps to find out complex structure of ANN for given input and the output data set by using its learning rule as a fitness function.
- Improve the predicrability of the system under construction.

Application of hybrid approach.

* load forecasting
* stock forecasting
* Cost optimization in texile industries
* medical diagnosis
* face recognition
* multi processor scheduling
* Job shop scheduling

Genetic fuzzy rule Based System

Fuzzy rule based system are identified for modeling complex systems.

The following figure shows genetic fuzzy system.

The main objective of optimization in fuzzy rule based system are as follows.

1. The task of finding an appropriate knowledge base for a particular problem. This is equivalent to parameterizing the fuzzy KB.

2. To find those parameter values that are optimal with respect to the design criteria.

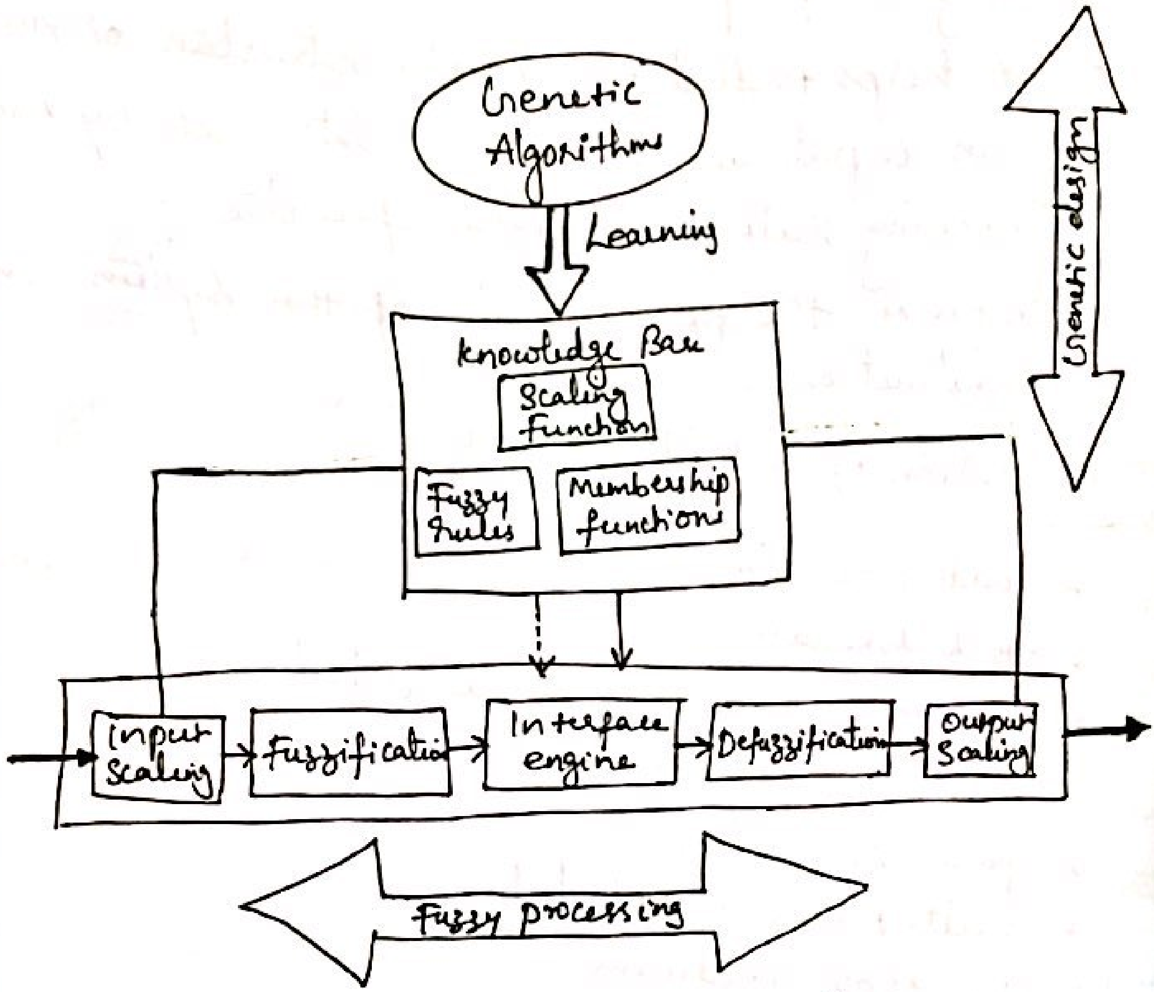


fig: Block diagram of genetic fuzzy system.

Considering a Genetic fuzzy rule based System GFRBS one has to decide which part of the knowledge base (KB) are subject to optimization by the GA. The KB of a fuzzy system is the union of qualitatively different components and not a homogenous structure.

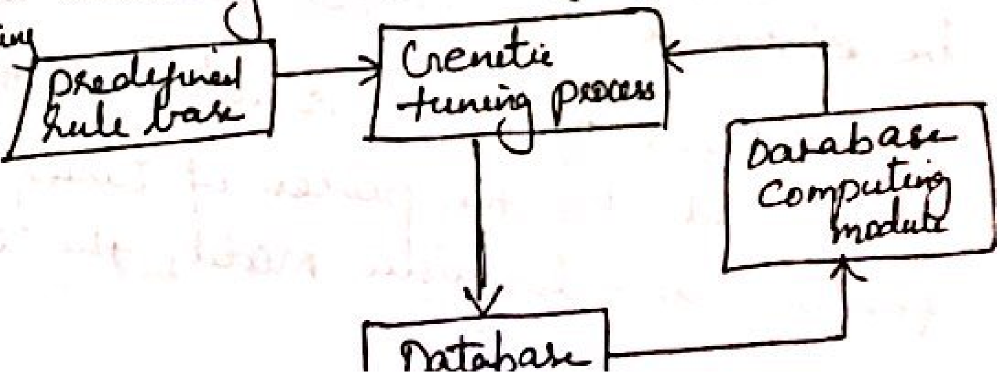
The following table distinguishes the Tuning and Learning problem.

Tuning	Learning problem.
It is Concerned with optimization of an existing FRBS	It constitutes an automated design method for fuzzy rule sets that start from scratch
Tuning processes assume a predefined RB and have the objective to find a set of optimal parameters for the membership or the scaling functions, DB parameters	Learning processes perform a more elaborated search in the space of possible RBs or whole KB and do not depend on a predefined set of rules.

Genetic Tuning process

The task of tuning the scaling functions and fuzzy membership functions is important in FRBS design. The adoption of parameterized scaling functions and membership functions by the GA is based on the fitness function that specifies the design criteria

The responsibility of finding a set of optimal parameters for the membership or the scaling functions rests with the tuning processes which assume a predefined rule base. The tuning process can be performed a priori genetic DB learning. The following figure illustrate the process of genetic tuning



Tuning scaling Function

Fuzzy membership functions are normalized by scaling functions applied to the input and output variables of FRBSs. In case of linear scaling, the scaling functions are parameterized by a single scaling factor or either by specifying a lower and upper bound. In case of non linear scaling, the scaling functions are parameterized by one or several contraction/dilation parameters.

In these kind of processes the approach is to adapt one to four parameters per variable, one when using a scaling factor, two for linear scaling and three to four for non-linear scaling.

Tuning Membership Functions

During the tuning of membership functions, an individual represents the entire DB. This is because its chromosome encodes the parameterized membership functions associated to the linguistic terms in every fuzzy partition considered by the fuzzy rule based system. The number of parameters per membership function can vary from one to four and each parameter can be either binary or real coded.

For FRBSs type the structure of the chromosome is different. In the process of tuning the membership functions in a linguistic model, the entire fuzzy partition are encoded into the chromosome and in order to maintain the global semantic in the RB, it is globally adapted. These approaches usually consider a predefined number of linguistic terms for each variable - with no requirement to be the same for each of them - which leads to a code of fixed length of membership functions.

In Descriptive fuzzy system the number of parameters to code is reduced to the one defining the core regions of the fuzzy sets.

Tuning the membership functions of a model working with fuzzy variables is a particular instance of knowledge base learning. This is because, instead of referring to linguistic terms in the DB, the rules are defined completely by their own membership function.

Genetic Learning of Rule Bases.

Genetic learning of rule bases assumes a predefined set of fuzzy membership functions in the DB to which the rules refer, by means of linguistic labels. When considering a rule based system and focusing on learning rules, there are three main approaches that have been applied in the architecture

1. pittsburgh approach
2. Michigan approach
3. Iterative rule learning approach

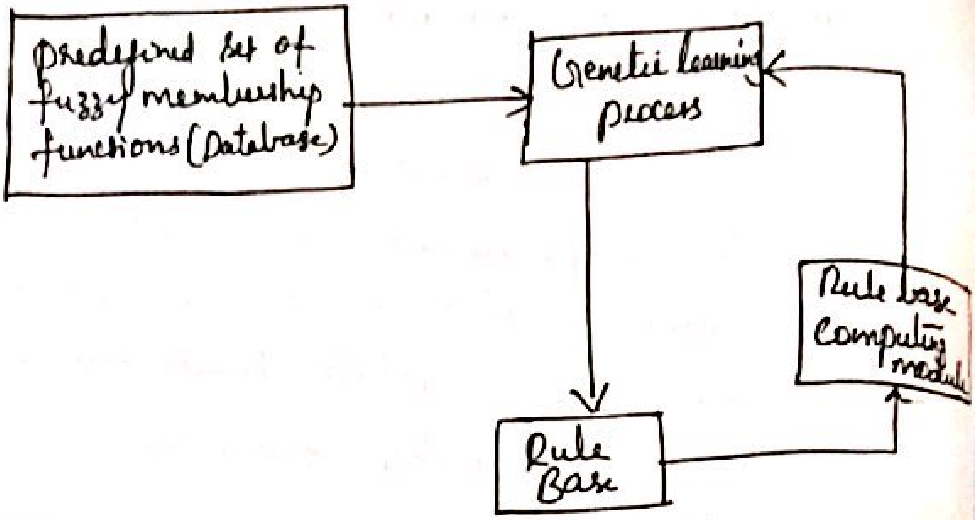


fig: Genetic learning of rule base.

The pittsburg approach is characterized by representing an entire rule set as a genetic code maintaining a population of candidate rule sets and using selection and genetic operators to produce new generation of rule sets.

The Michigan approach considers a different model where the members of the population are individual rules and the rule set is represented by the entire population.

In the iterative approach, chromosome code individual rules, and a new rule is adapted and added to the rule set, in an iterative fashion, in every run of the genetic algorithm

<u>Genetic Learning of knowledge Base.</u>

Genetic learning of a KB includes different genetic representations such as variable length chromosomes, multi chromosome genomes and chromosomes encoding single rules instead of a whole KB as it deals with heterogenous search space. As the complexity of the search space increase, the computational cost of the genetic search also grows. To solve this issue individual rules are encoded rather than entire KB. In this manner one can maintain a flexible, complex rule space in which the search for a solution remains feasible and efficient.

The three learning approaches used in case of rule base can also be considered in the learning of KB (Michigan, Pittsburg and iterative rule learning approach. Following figure illustrates the genetic learning of KB.

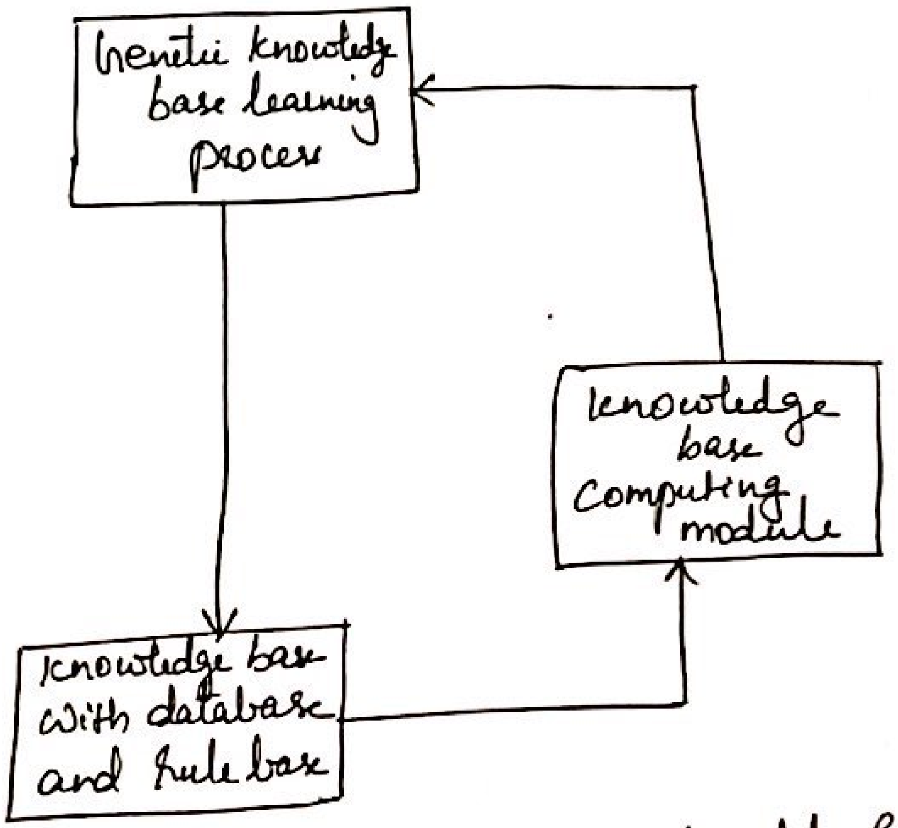


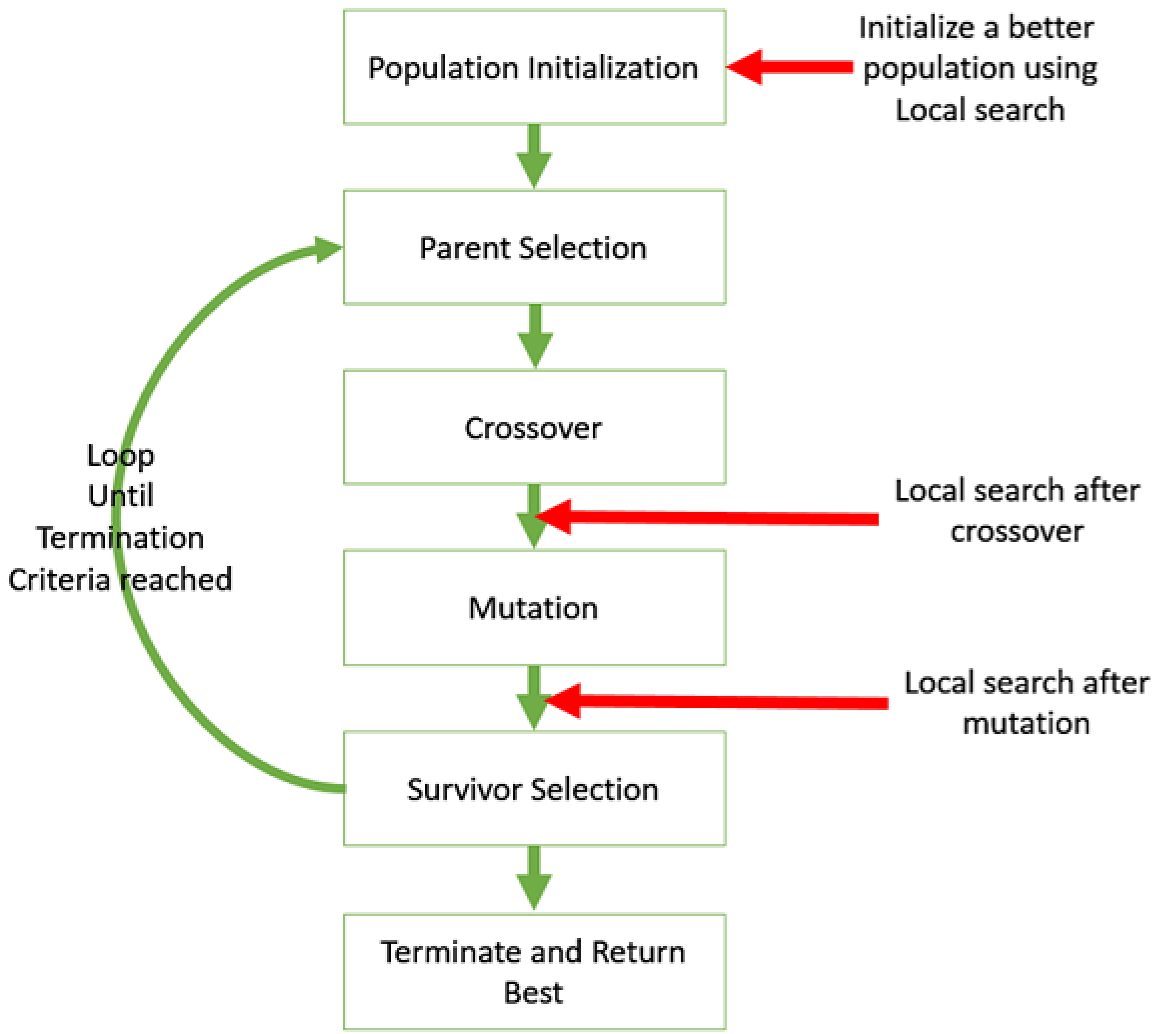
fig: Genetic learning of knowledge Base

Content beyond syllabus

CS361: Soft computing

Hybridize GA with Local Search

It may be sometimes useful to hybridize the GA with local search. The following image shows the various places in which local search can be introduced in a GA.

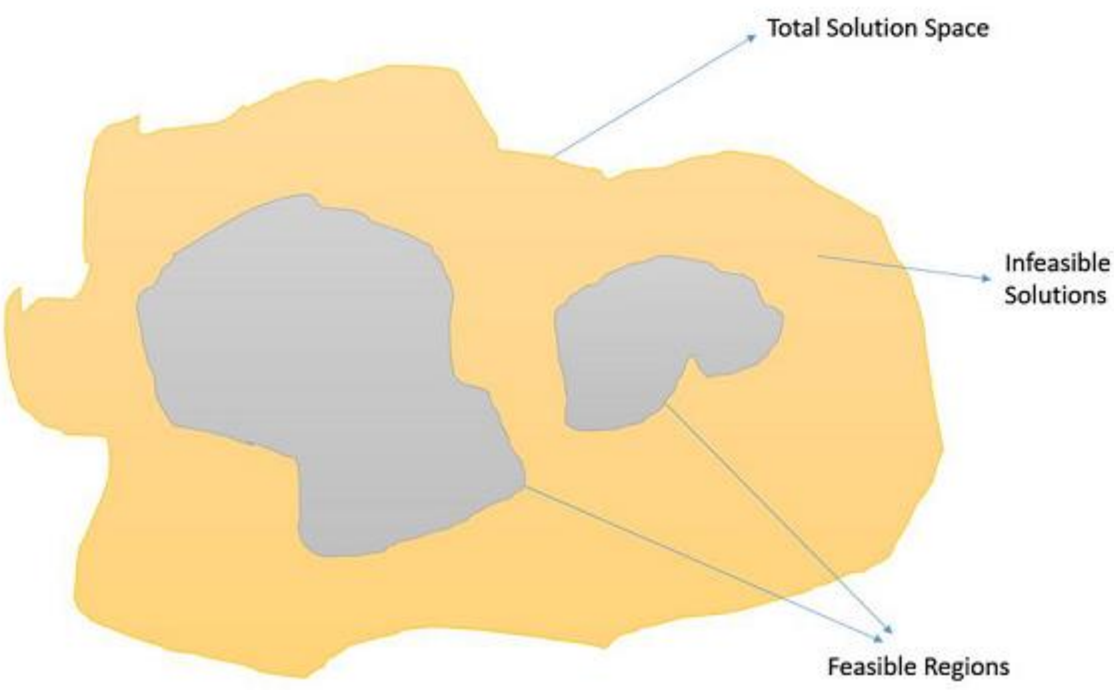


Variation of parameters and techniques

In genetic algorithms, there is no “one size fits all” or a magic formula which works for all problems. Even after the initial GA is ready, it takes a lot of time and effort to play around with the parameters like population size, mutation and crossover probability etc. to find the ones which suit the particular problem.

Constrained Optimization Problems

Constrained Optimization Problems are those optimization problems in which we have to maximize or minimize a given objective function value that is subject to certain constraints. Therefore, not all results in the solution space are feasible, and the solution space contains feasible regions as shown in the following image.



In such a scenario, crossover and mutation operators might give us solutions which are infeasible. Therefore, additional mechanisms have to be employed in the GA when dealing with constrained Optimization Problems.

Some of the most common methods are −

- Using **penalty functions** which reduces the fitness of infeasible solutions, preferably so that the fitness is reduced in proportion with the number of constraints violated or the distance from the feasible region.
- Using **repair functions** which take an infeasible solution and modify it so that the violated constraints get satisfied.
- **Not allowing infeasible solutions** to enter into the population at all.
- Use a **special representation or decoder functions** that ensures feasibility of the solutions.

Schema Theorem

The basic terminology to know are as follows −

- A **Schema** is a "template". Formally, it is a string over the alphabet = {0,1,*},

  where * is don't care and can take any value.

  Therefore, *10*1 could mean 01001, 01011, 11001, or 11011

  Geometrically, a schema is a hyper-plane in the solution search space.
- **Order** of a schema is the number of specified fixed positions in a gene.

Schema	Order
***	0
101	3
*11	2
1**	1

Defining length is the distance between the two furthest fixed symbols in the gene.

Schema	Defining Length
****	0
11	1
1*0*	2
1111	3

GA Based Machine Learning

Genetic Algorithms also find application in Machine Learning. **Classifier systems** are a form of **genetics-based machine learning** (GBML) system that are frequently used in the field of machine learning. GBML methods are a niche approach to machine learning.

There are two categories of GBML systems −

- **The Pittsburg Approach** − In this approach, one chromosome encoded one solution, and so fitness is assigned to solutions.
- **The Michigan Approach** − one solution is typically represented by many chromosomes and so fitness is assigned to partial solutions.

It should be kept in mind that the standard issue like crossover, mutation, Lamarckian or Darwinian, etc. are also present in the GBML systems.