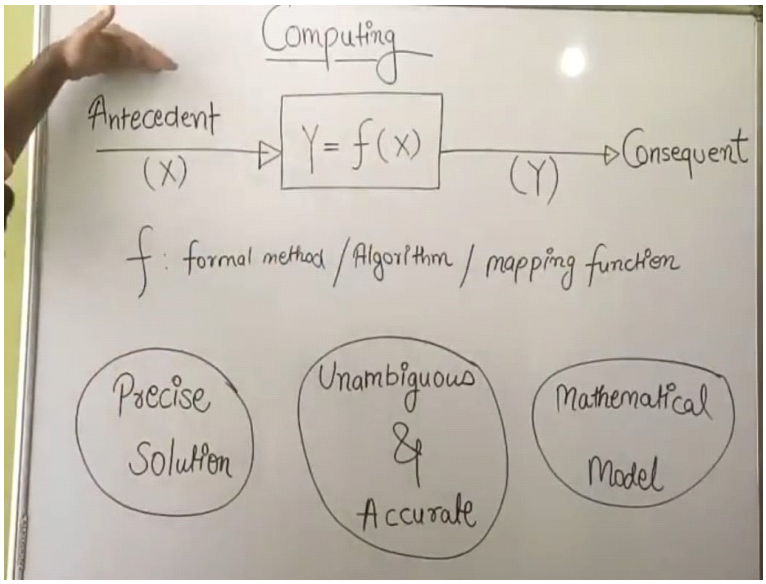
Application of Soft Computing (BCS056)

BTECH CSE – Vth Semester

Introduction to Soft Computing



Q : Compare and contrast Biological Neuron and Artificial Neuron with a suitable diagram

Q : Discuss about the name of a network that includes backward links from a given output to its inputs along with the hidden layers?

Ans : Recurrent neural network

Define Soft Computing. How is it different from conventional computing?

Write down the applications of genetic algorithm.

Why do we use bias function in neural network?

There are a Function which takes some input which we say Antecedent and giving input “x” and we are doing computing in Function x so it’ll give some output which stores in Y which we can say consequent

Here this is single function f(x)

Here function x is doing computing process which we can say this function F Formal Method / Algorithm / Mapping Function

Here in picture we can say that we get desired output in consequent which is done by control Action

Features of Soft Computing

- Precise Solution
- Unambiguous & Accurate
- Mathematical Model

Precise Solution

If we use computing and does problem solving then it should be precise manner

Unambiguous & Accurate

As we have mentioned computing unit which transforms data from function that is nothing but which is algorithm

And algorithms process the data by control action

So control action should be unambiguous and algorithm should not be two meaning

Mathematical

This is also algorithm to solve the problem

Soft Computing Technique

Computation is a process of converting the input of one form to some other desired output form using certain control actions. According to the concept of computation, the input is called an antecedent and the output is called the consequent. A mapping function converts the input of one form to another form of desired output using certain control actions. The computing concept is mainly applicable to computer science engineering. **There are two types of computing, hard computing, and soft computing.**

Hard Computing

- **Precise Results**

Hard computing is a process in which we program the computer to solve certain problems using mathematical algorithms that already exist, which provides a precise output value. One of the fundamental examples of hard computing is a numerical problem.

This will be precise manner because in Hard computing allows mathematical operation to solve algorithm which exists on the method

- **Control Action**
    - **Unambiguous Data**

    Control action allows the mathematical operation which performs on algorithm

So whereas mathematical operation exists there is need for hard computing

Quick sort

Bubble Sort

Merge sort

Insertion Sort

Heap sort

Selection

Computational Geometry Problem

In mathematics Graph x and y coordinate

I**n come cases there are uncertainty the output**

Imprecision

Dynamic -> The result may change in the output because what algorithm size is changes that is not adaptive

Uncertainty

When programmer runs any program then it is not the same data as user wants because user also apply implementation in different manner

Low solution cost

Because lack of service and when we use the hard computing that is High Solution Cost due precise and unambiguous method

LLM (Large Language Model)

LLM stands for Large Language Model, which is a type of **Artificial intelligence (AI)** that can understand and generate human language. **LLMs are a subset of deep learning**, a type of machine learning that uses algorithms to recognize complex patterns in large data sets

Some Basic Components of ANN Artificial Neural Network

- Neurons (Nodes)
- Layers
- Connections
- Activation Function

Now check the degree of fastness

Data is stored and it will calculating the data now it is creating Layets when its processing the data

Layers Input data

Data Stores hidden layer data is breaking and associated

Layers are multiple Layers

Connection

The above picture is a Neuron Network

What Are Activation Functions?

Activation functions are an integral building block of **neural networks** that enable them to learn complex patterns in data. They transform the input signal of a node in a neural network into an output signal that is then passed on to the next layer.

An activation function in a neural network **determines which neurons are activated as information moves through the network's layers.** It's a key component of neural networks that allows them to learn complex patterns in data.

Neural Network

A neural network is a machine learning technique that uses a network of interconnected nodes to process data and learn from mistakes,

For example Internet : Sharing the Data

If we talk about human brain there are high number of neurons "https://en.wikipedia.org/wiki/Neuron"1 HYPERLINK "https://en.wikipedia.org/wiki/Neuron"] or **nerve cell** which are connected

There are gazillions of Neurons in Human Brain which are connected with network and that connectivity provides the flow information and process it and parallel process

In Human Brain can learning algorithm from experience and train the data in itself and the capacity of training and learning make the human brain a very important organ

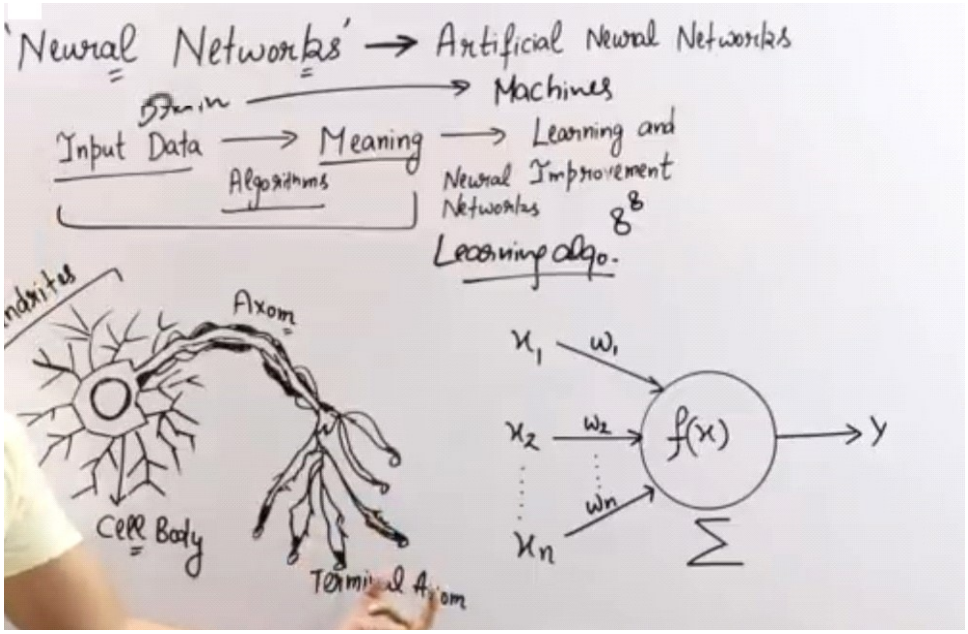
And we do in artificial machine we want to implement brain like human brain which we can say neuron network and that we can say Artificial Neural Network which works the learn the algorithm and train the data

For Example, First day you go to the office then you recognize the faces of people and staff on there and some have less hair shaved or trim detect the faces next again you go to office then you'll detect the face and recognize the faces so many people may have shaved or trimmed But you will recognize them.

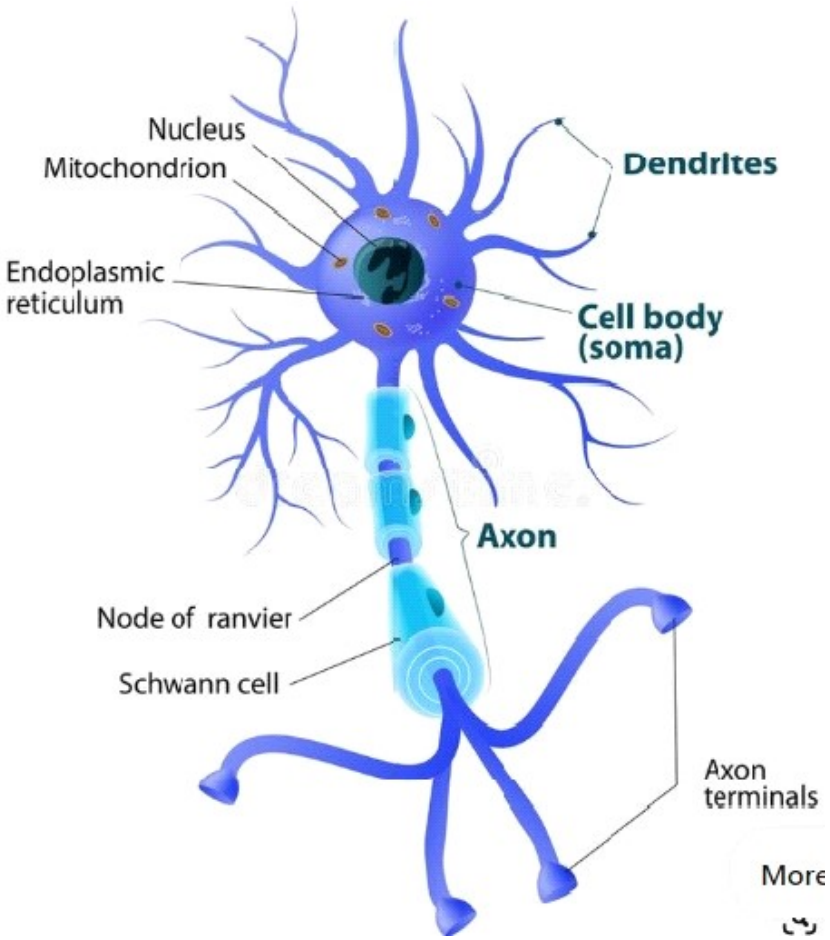
Human brain learns and trains But the same thing if we do in Normal Computer then it'll have to do all the scanning if there are 100 people. You'll have to scan their photos and give them to the computer. If you check again the next day

So the Normal computer can work only the base of algorithm which gives the output like recognize the face expression of any actor

If we implement this with the help of Neural Network, then it recognizes 90% so here the accuracy is increasing this is the power of Learning algorithm



Here this biological neuron we are implementing in Artificial Neurons in the Machine Learning where we train the data



1 billion = 100 Crore or 1 crore = 100 lakh

10 crore = 1000 Lakh

So 1 billion = 100 crore = 10000 Lakh

1 trillion = 1000 billion

1 trillion

Llama 3

Well, first of all, Meta's Llama 3 has a **15 trillion–token dataset** (enabling more efficient language encoding and better performance), which is 7x times larger than previous models

In ANN the LLM (large language model) used with the help of token and parameter

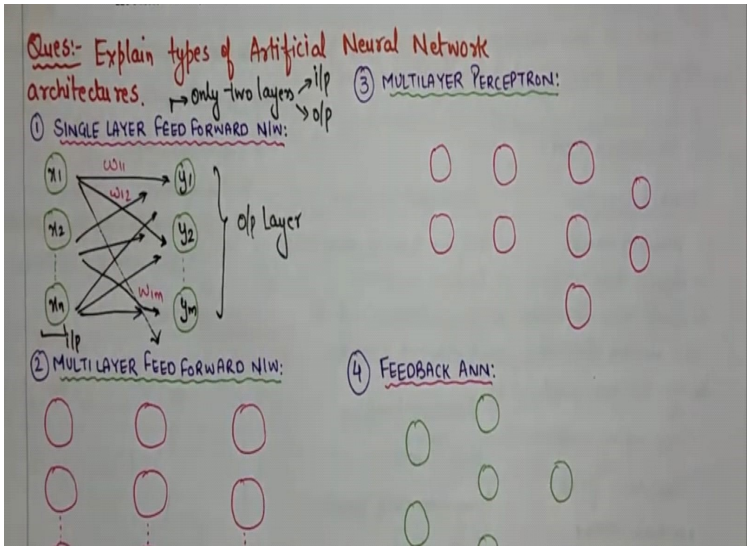
Token

Lets

Types of Artificial Neural Network

- **Single Layer Feed Forwarding Network**
- **Multi Layer Feed Forward Network**
- **Multi Layer Perceptron Network**
- **Feedback ANN**

- Single Layer Feed Forwarding Network1

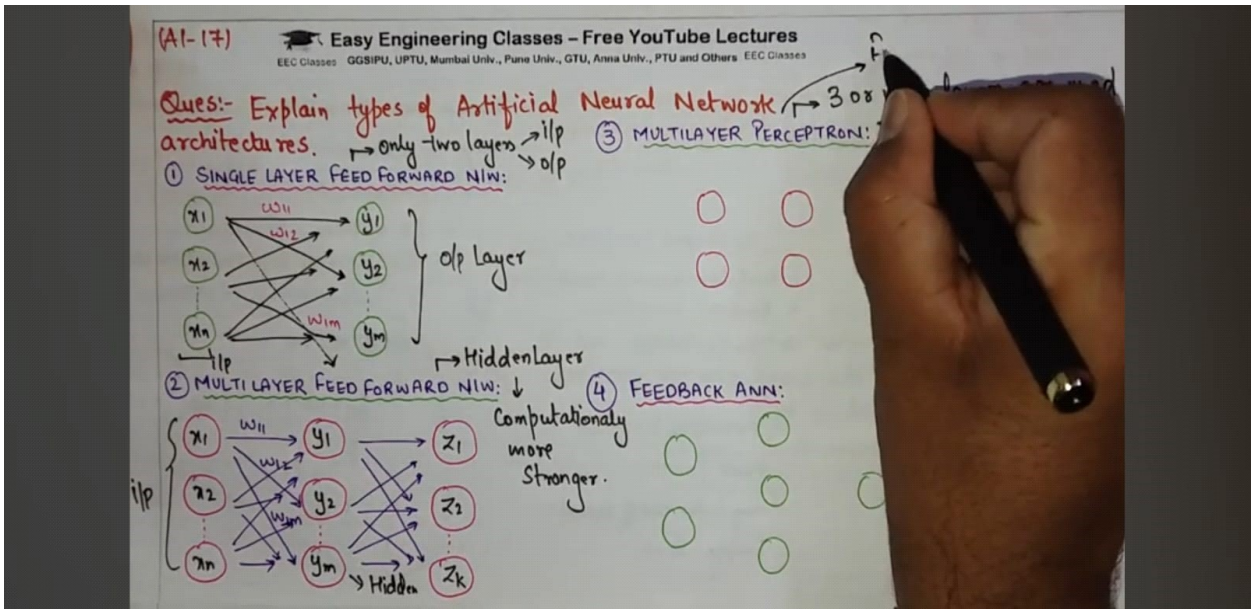


There are only two layer Input and Output

- **Multi Layer Feed Forward Network**

There is Hidden Layer in Multi Layer Feed Forward Network

Means here we can solve complex programming and their al



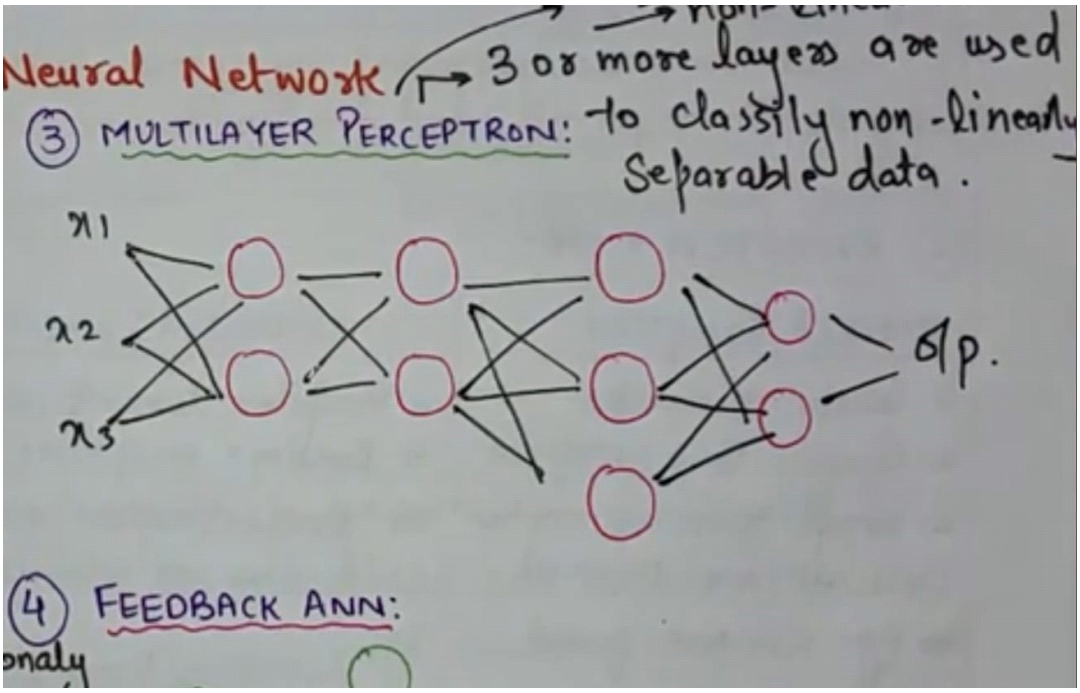
Here nodes are connected with some weight and other layer nodes are connected here with the help of activation function we can apply corresponding input get the output

- **Multi Layer Perceptron Network**

  In this three or more layers classified data are used to classify non-linearly separable data

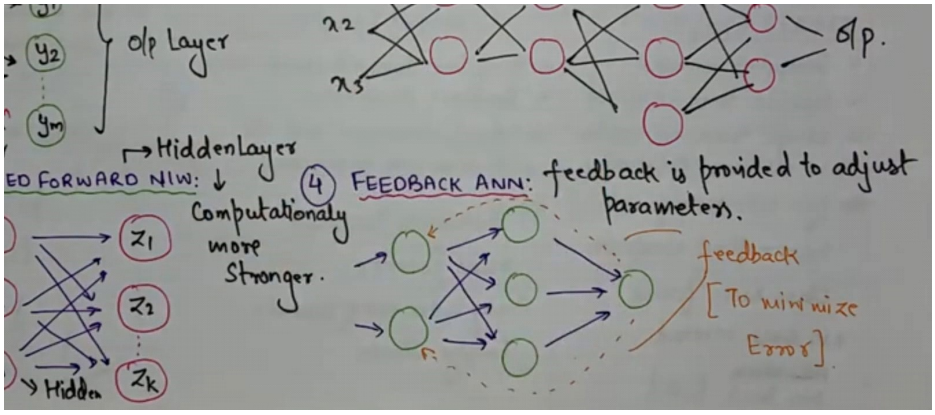
  This is fully connected node means each node connected to next layer node and usage non-linear activation function

  Let’s assume we have input x1, x2, x3 all will be connected and rest of the layer will be connection completely after that we get Output

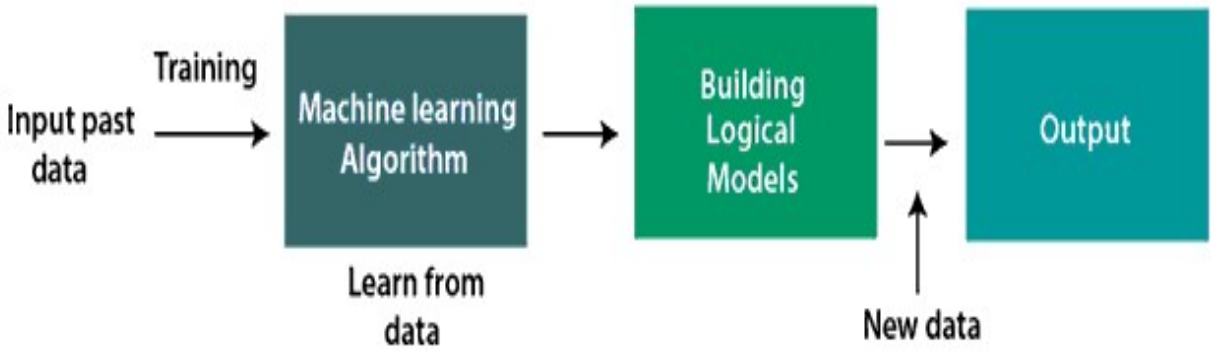


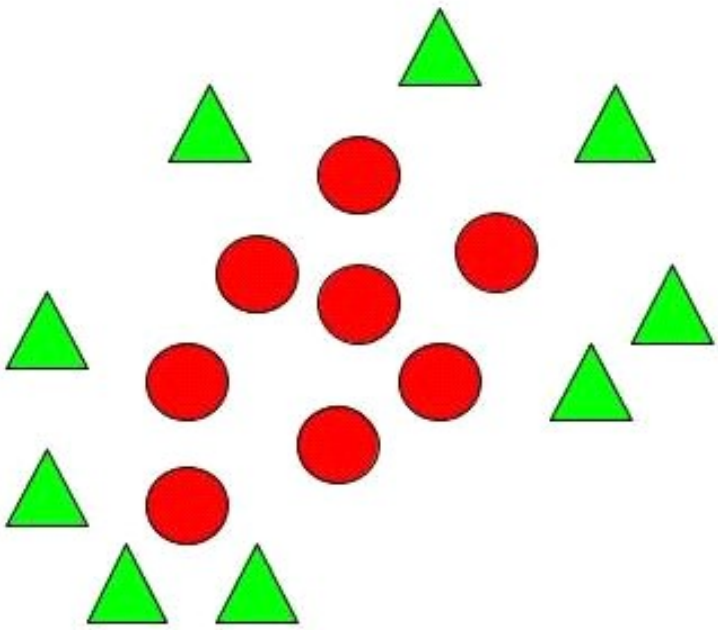
4 Feedback ANN

Feedback is provided to adjust parameters



Here parameter return back towards first layer if any error occurs then parameter can be updated and because to minimize





Non Linear Separable Data

Example 2

7, 7=> 1 (Pass),

2 hrs. sleep, 8 hrs. study => 0 (Fail)

Y =x1.w1 + x2.w2 + b

I want output come between 0 & 1 in y then use sigmoid function

Here b (bias) value will update as much as update then value will be fluctuating it.

So at initial level we have train the data giving the value of w1 and w2 and updated it so that loss should be minimal

$$S(x) = \frac{1}{1 + e^{-x}}$$ this is sigmoid function

w1=0.5

w2=-0.3

b1=0.1

Now check formula: Y =x1.w1 + x2.w2 + b1 let's take new data for the training the data

= 2*0.5 + 8*-0.3 + 0.1

1
-2.4
0.1
-1.3

Here -1.3 value comes after using activation function now put this in sigmoid function and value

1/1+(e^-(-1.3)) = 1/4.67 => 0.21 means it's the probability near fail

Lets say we take x1= 6 and x2 = 7 in activation function

Y =x1.w1 + x2.w2 + b1

= 6*0.5 + 7* -0.3 + 0.1 = 1

x1	x2		Total
6	7	0.1	
0.5	-0.3		
3	-2.1	0.1	1

0.9 0.1 1

1/1+(e^-(1)) = 1/1.37 => 0.72 which means its probability near Pass

Recurrent Neural Network (RNN)

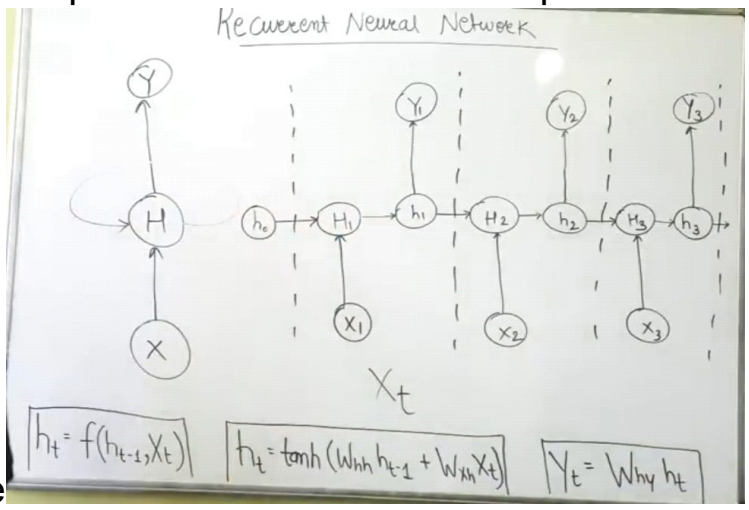
It is used in Feed Forward Neural Network stage and they cannot handle sequential data because they use only current input while we need previous state and memory element is absent

FFN RNN

Forward neural network which is going forward and it does not have any memory

It takes current input but here we require previous state which they do not consider it

To overcome this problem, we need RNN which have memory and previous data and current input

It also use loop method means when output comes again it goes input state 

It also consider previous state but suppose it has not previous state then it use0s at starting **Initial Hidden State**

Which is **H0** and **previous hidden state** and **X1** current input both process it uses with first Time Step then **output will be Current Hidden State**

$h_t = f(h_{t-1}, X_t)$

h_t = current state

h_{t-1} = Previous state

Activation Function

$h_t = \tanh(W_{hh} . h_{t-1} + W_{xh} . X_t)$

W_{hh} = Weight at Previous Hidden State

W_{xh} = Weight at Current Input State

$Y_t = W_{hy} . h_t$

W_{hy} = Weight at the output state

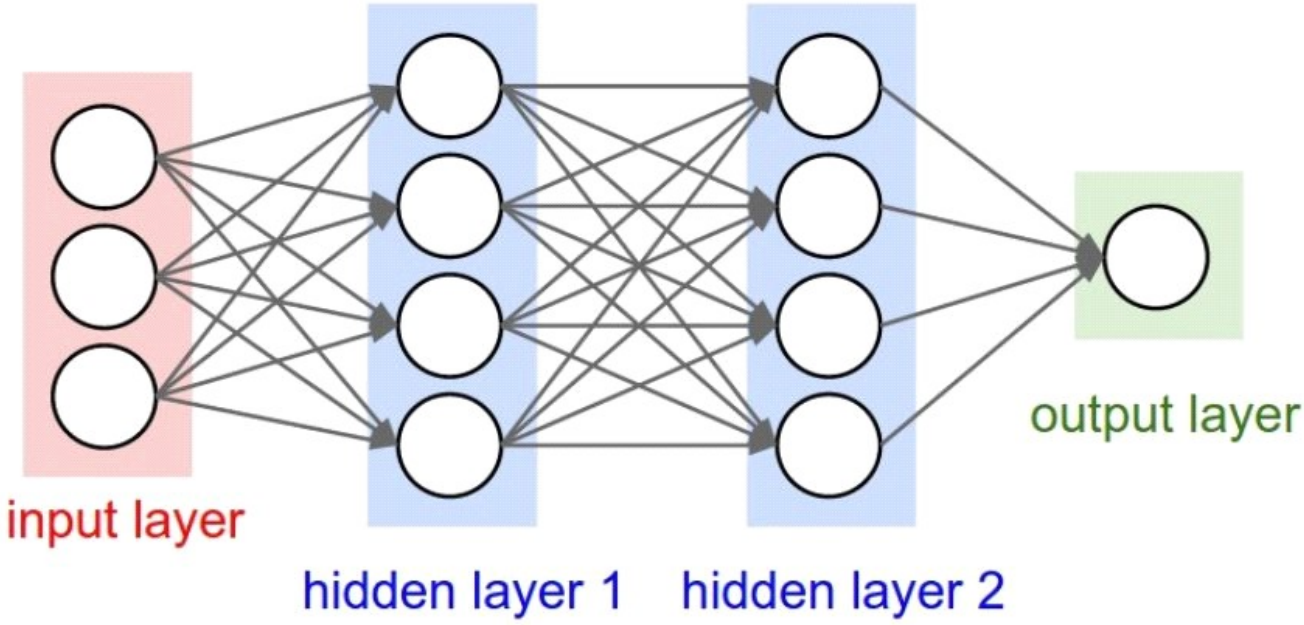
h_t = time step h_1 and h_2,h_3….h_n

A recurrent neural network (**RNN**) is a kind of artificial neural network mainly used in **speech recognition** and **natural language processing** (NLP). RNN is used in deep learning and in the development of models that imitate the activity of neurons in the human **brain**.

Recurrent Networks are designed to **recognize patterns** in data sequences, such as **text, genomes, handwriting, the spoken word,** and **numerical** time series data emanating from sensors, stock markets, and government agencies.

Genmo AI is a **tool** that uses artificial intelligence (AI) to create videos and images:

What is Deep Learning

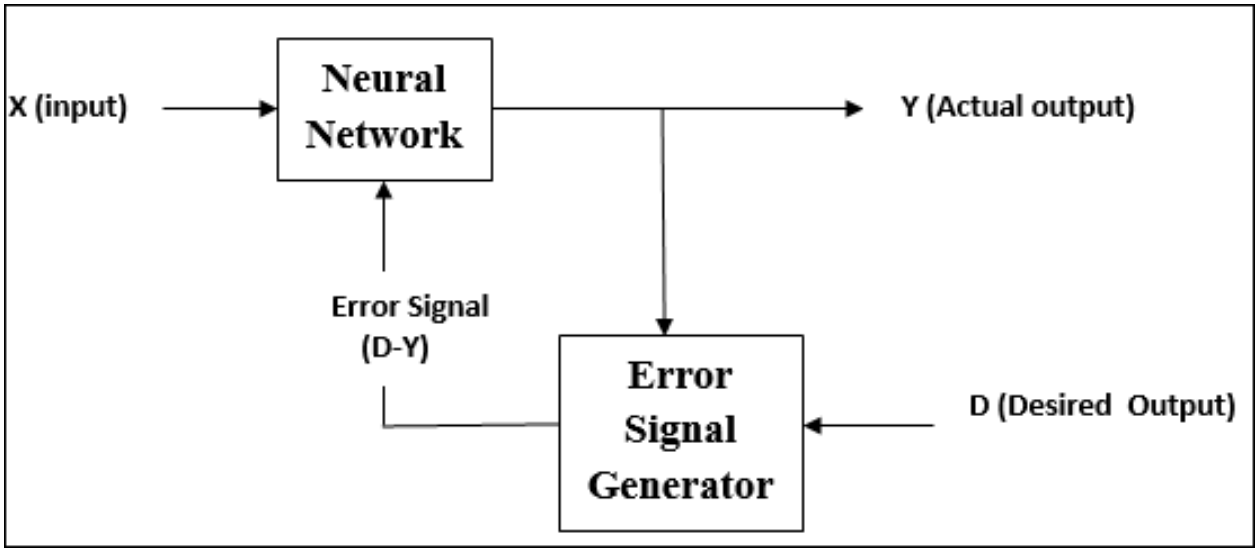


Deep Learning is a form of Machine Learning. It is known as 'Deep' Learning because it contains many layers of neurons. A neuron within a Deep Learning network is similar to a neuron of the human brain - another name for Deep Learning is 'Artificial *Neural* Networks'.

Various Learning Techniques of ANN

- **Supervised Learning**
- **Unsupervised Learning**
- **Reinforcement Learning** **f**

Supervised Learning



Our main is that how to make intelligent the machine in supervise learning e have training data means supervise means supervisor (teacher) and we have input data and output data which we create a model and that model we put new input and check valid output Example : Exit poll in Election

Now here learning algorithm works like **Naïve Bayes input your algorithm is learning and classifying means it will check the probability of wining or loosing so this is supervised learning**

As the name suggests, this type of learning is done under the supervision of a teacher. This learning process is dependent.

During the training of ANN under supervised learning, the input vector is presented to the network, which will give an output vector. This output vector is compared with the desired output vector. **An error signal is generated, if there is a difference between the actual output and the desired output vector. On the basis of this error signal, the weights are adjusted until the actual output is matched with the desired output.**

Used Learning Algorithm Naïve Bayes Algorithm

Which apply input and output given and again given New Input actual counting in exit poll and prepare in Model

And find out result probability or output

Naïve Bayes theorem usage in Supervised Learning

Where we need to predict for any result for **example Email is spam or Not spam here we need to classify which requires Bays theorem**

Bag 1 Total Red Ball Probability in 1st Bag = 2/5

Bag 2 Total Red Ball Probability in 2nd Bag = 4/7

Now suppose we want to take a One Ball which should be Red from Both Bags. Then total probability will be

1/2 x 2/5 + 1/2 x 4/7

Now comes Bays Theorem **I have taken a Ball which is Red**

What is the probability if the Red ball is taken out from Bag1 -> (B1/R) so this is called cause property means Reverse method

Red ball is taken from Bag1 means Ball is Given Red from Bag1 -> 1/2 X 2/5-> B1/R

So formula is Bays Theorem P(Y/X) = P(X/Y) * P(Y) / P(X)

Here X is given that Red Ball and find out of Red is taken from Bag1

$$= \frac{R/B1 * B1}{R}$$

$$P(Y/X) = P(X/Y) * P(Y) = \frac{P(R/B1) * P(B1)}{P(X)} \qquad \frac{}{P(R)}$$

$$= 2/5 * 1/2$$

$$1/2 \times 2/5 + 1/2 \times 4/7$$

$$= \quad 1/5$$

$$1/5+2/7$$

Suppose this is Yes, Red ball taken from B1

$$P(Y/X_1,X_2\ldots\ldots.Xn) = P(X_1/Y)*P(X_2/Y)*P(X_3/Y)\ldots\ldots P(X_n/Y)*P(Y)$$

$$P(X_1) *P(X2)*P(X_3)\ldots\ldots P(X_n)$$

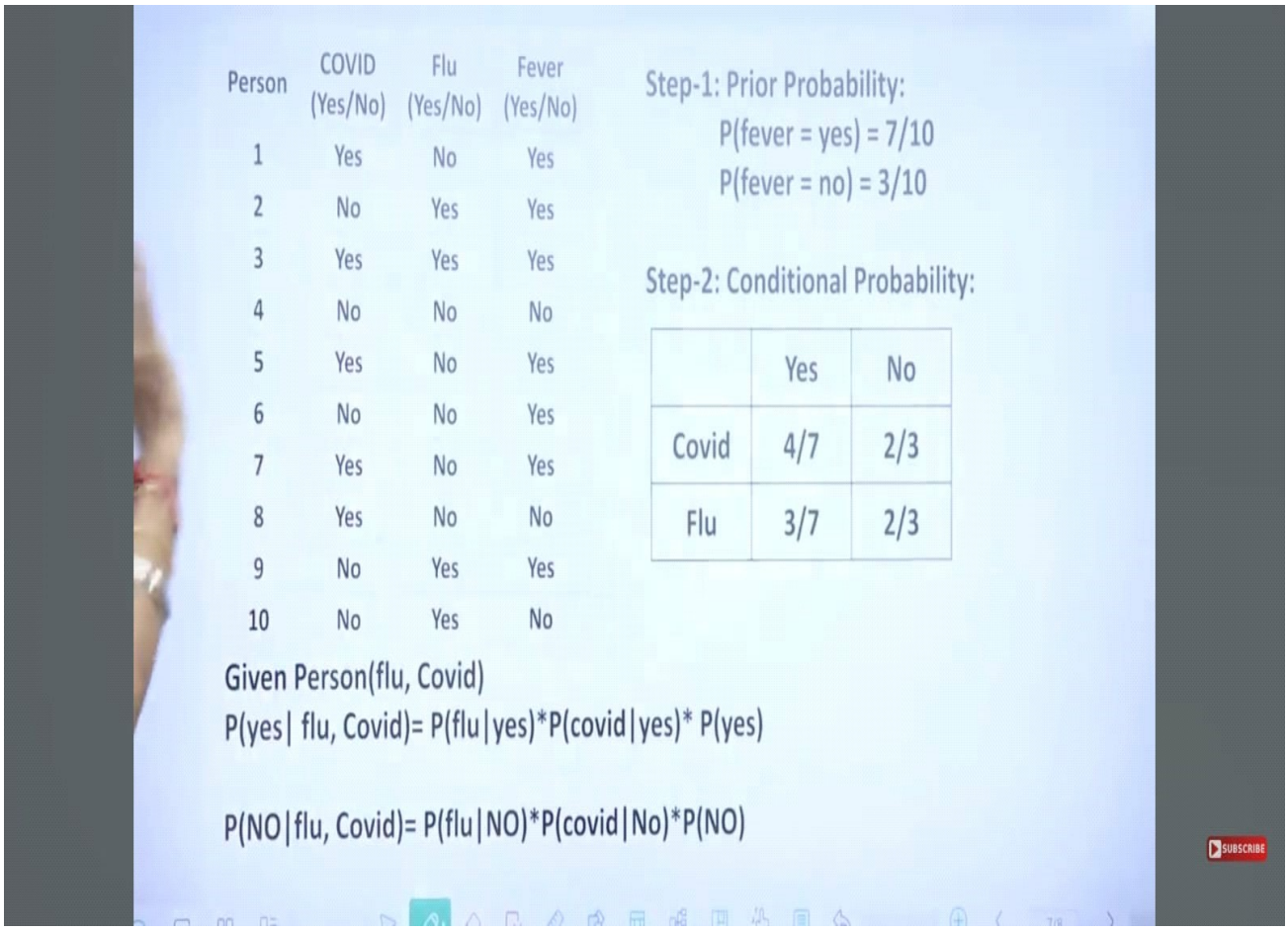
Again Suppose this is No, Red ball is not from B1

$$P(N/X)= P(X/N)*P(N)$$

$$P(X)$$

$$P(N/X_1,X_2\ldots\ldots.Xn) = P(X_1/N)*P(X_2/N)*P(X_3/N)\ldots\ldots P(X_n/N)*P(N)$$

$$P(X_1) *P(X2)*P(X_3)\ldots\ldots P(X_n)$$



Suppose Fever was Yes and We have Two Factor X1, X2

X1=Covid , X2=Flu

And Here person(Flu, Covid) -> X1,X2

$P(X_1/Y)*P(X_2/Y)$

$P(Y/X1,X2)= P(X_1/Y)*P(X_2/Y) * P(Y) / P(X_1) *P(X2)$……..n

=4/7*3/7* 7/10

=12/70

Unsupervised Learning

As the name suggests, this type of learning is done without the supervision of a teacher. This learning process is

independent. Here data is learning and we have only input which make some cluster and refine that based on the any category maximum machine are Unsupervised where we have to input to supervised technique

During the training of ANN under unsupervised learning, the input vectors of similar type are combined to form clusters. When a new input pattern is applied, then the neural network gives an output response indicating the class to which the input pattern belongs.

Reinforcement learning

It works in Reward and Policy

It is done by Action and changed the state get the reward

Suppose given Rs. 50 and change the state and again retrieve the reward then create any policy

Pk Movies takes currency behalf of Gandhi Note Ji and realize the penalty and learn the data

It becomes in Game that we can create in the for next Level

In Supervised Data we give input and give Output then we find our desired output

Continuous Data like we give similar input for weather forecasting morning (Humidity) afternoon (Hot) and Evening (Cloudy) then machine will learn and give the prediction which is Label Data

here we give the Label Data Means Input Output Pair this is
Supervised Learning has two type of Learning Algorithm

1. Regression Algorithm

What is Temperature going to be tomorrow

Temperature means Values is continuous change

Regression Algorithm are used if there is relationship between

Input Variable and Output Variable

Example Age, Salary, Prices etc. means continuous change

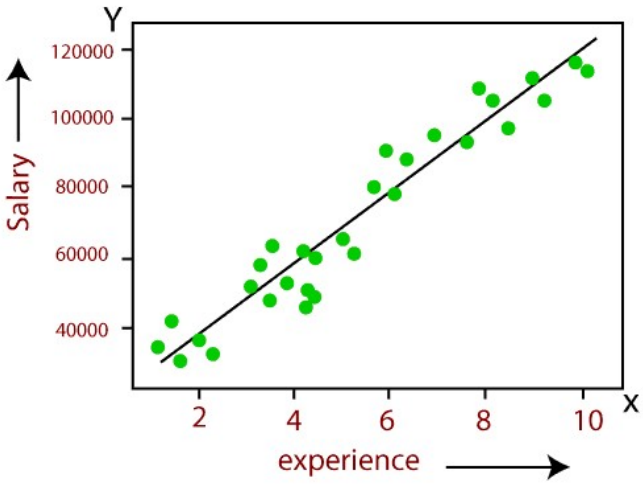


Figure: Regression Analysis

Regression Analysis Algorithm :

Linear Regression, Logistic Regression, Support Regression, Decision Tree Regression, Random Forest Regression

Linear Regression

Y=mX+b

Y represent the **dependent Variable** //to check score

X represent the **independent Variable**//Study Hrs.

m is the slope of the line (how much Y changes for a unit change in X)

b is the intercept (the value of Y when X is 0)

2. Classification Algorithm

Here we need to classify the data Either Hot or Cold

Will it be cold or Hot Tomorrow?

Means Yes/No, True/False, Male/Female

Example : Linear Regression

Project : Predicting Pizza Prizes

Step 1 : Data Collection

Diameter (X) in Inches	Price (Y) in Dollar	Mean (X)	Mean (Y)	Deviation (X)	Deviation (Y)	Product of Deviation	Sum of Product of Deviation	Square of Deviation for X
8'	10	10	13	-2	-3	6	12	4
10'	13			0	0	0		0
12'	16			2	3	6		4

X is Independent and Y is dependent

Mean (x) = 8+10+12=30/3=10

Mean (Y) = 10+13+16=39/3=13

Sum of Product of Deviation =6+0+6=12

Now calculate m i.e, slope

m=Sum of product deviation/sum of square of Deviation for x

=12/8=1.5

Means if I am changing x value 1 then Y will be change 1.5

Now suppose pizza X size is 0 then what will be charged of Pizza

$Y = mX+b$

Now Calculate b = Mean of Y – (m*Mean of X)

=13 – (1.5 * 10)

= 13 – 15 = -2

Suppose if we have 20’ size pizza then what will be Price

Y=mX+b

=1.5*20+(-2)

=30-2=28$

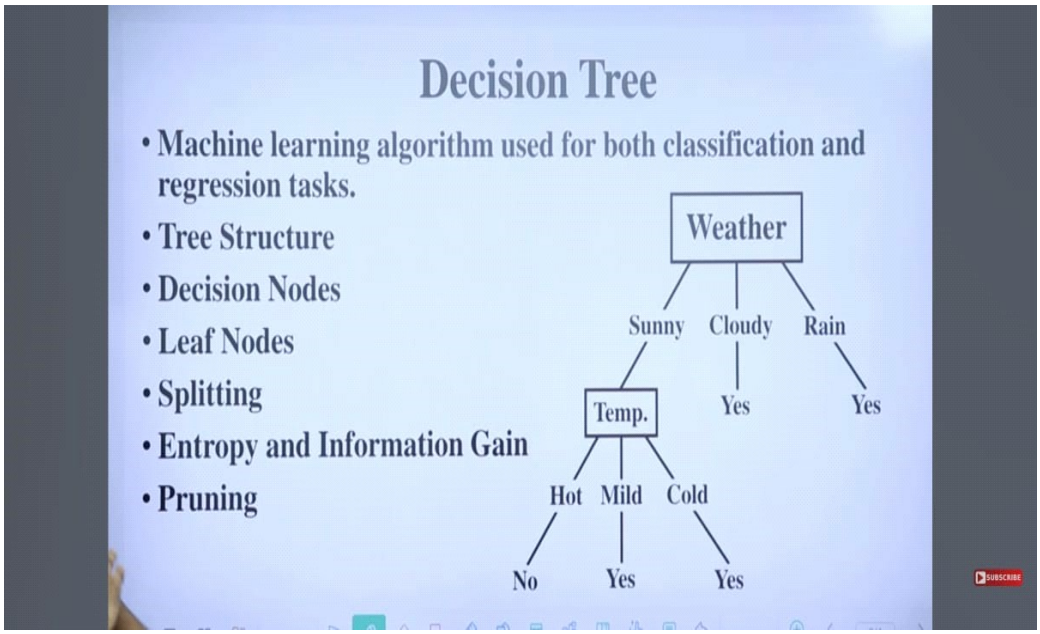
Regression analysis is a key part of predictive modeling and is used in many different applications of machine learning. For example, regression analysis can be used to:

- Predict house prices
- Forecast stock or share prices
- Map salary changes.

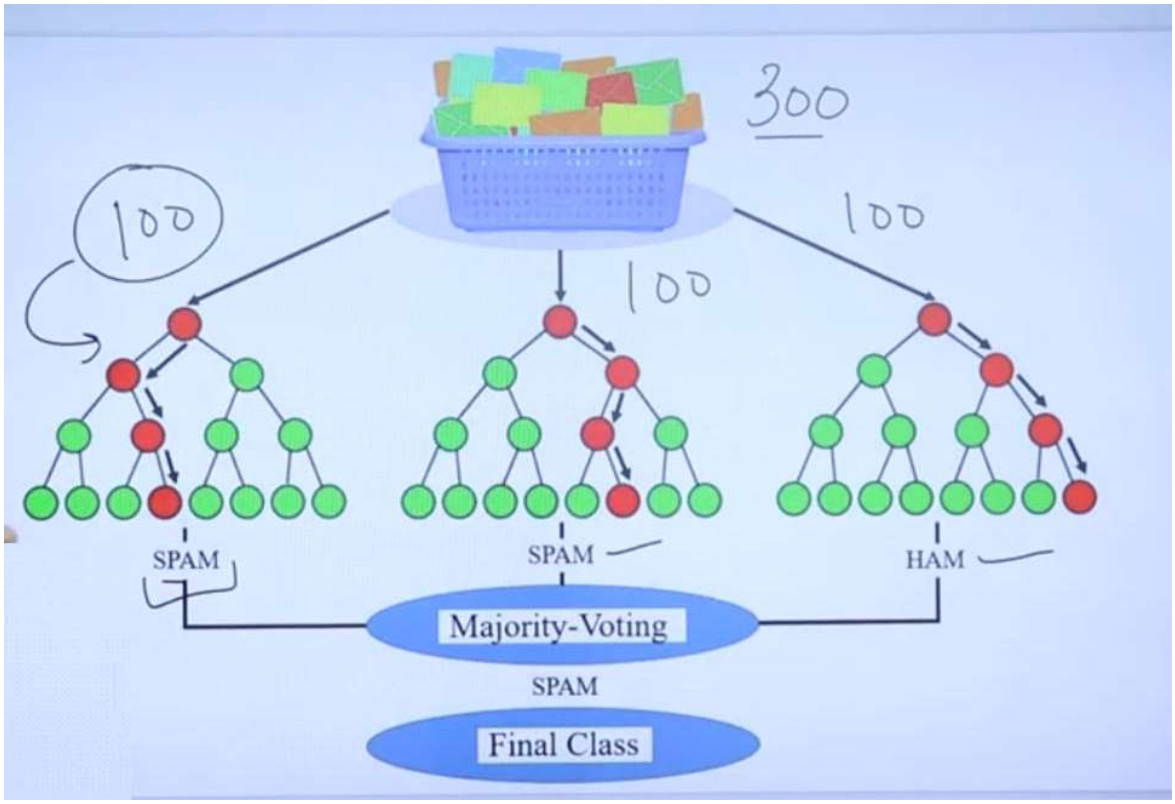
Different type of Classification Algorithm

- Decision Trees
- Random Forest
- Logistic Regression
- Support Vector Machine

Decision Trees



Random Forest



Here are multiple decision tree exist

Output is based on majority voting that email is either Spam or Not Spam

R F is an ensemble learning method

The Algorithm which implement the classification on a data set is known as a **classifier**

Random Decision forest correct for decision tree's habit of overfitting to their training set.

These are two type Classification

- **Binary Classifier**

  If the classification problem has only two possible outcomes then it is called as Binary Classifier

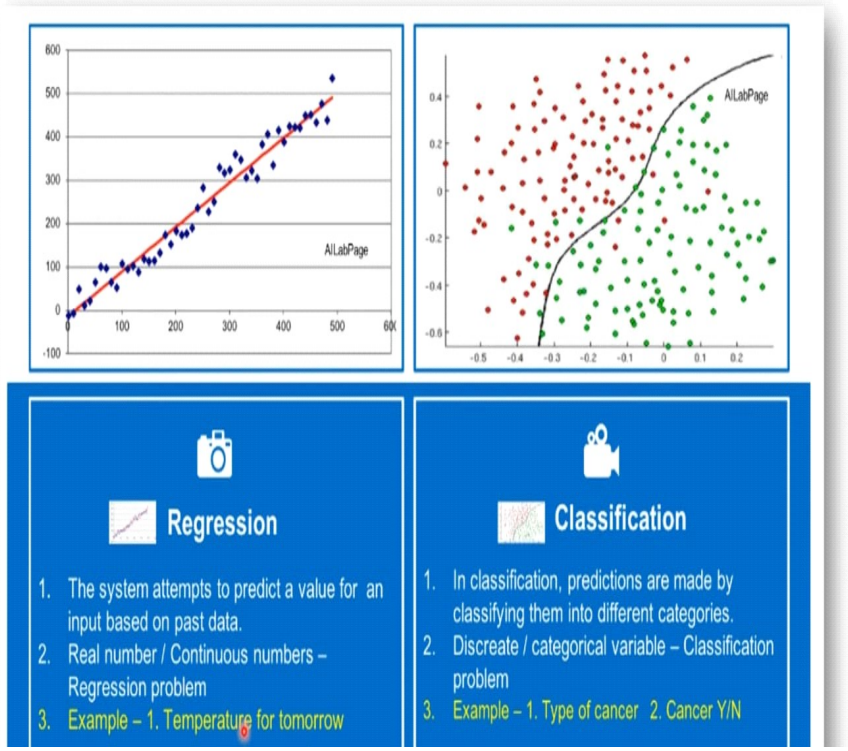
  Example : Yes/No, Male/Female, Spam/Not Spam, Cat/Dog, True/False

- **Multiclass Classifier :**

  if a classifier problem has more than two outcomes then it called as Multiclass Classifier

  Example : Classification of types of crops,

  Classification of types of Music

- 

  Non linear Regression

Support Vector Machine

Labeled data given in supervised algorithm for training data

If New data belongs to Square shape or Circle Shape during the testing for Prediction

If the New data is circle, then output belongs to circle class

Linear SVM: Linear SVM is used for linearly separable data, which means **if a dataset can be classified into two classes by using a single straight line**, then such data is termed as linearly separable data, and classifier is used called as Linear SVM classifier.

Non-linear SVM: Non-Linear SVM is used for non-linearly separated data, which means **if a dataset cannot be classified by using a straight line**, then such data is termed

as non-linear data and classifier used is called as Non-linear SVM classifier.

Unsupervised Learning

As the name suggests, this type of learning is done without the supervision of a teacher. This

learning process is independent.

Here we give only input in model here checks hidden information

And it checks in pattern and recognise it then it creates cluster

And Group

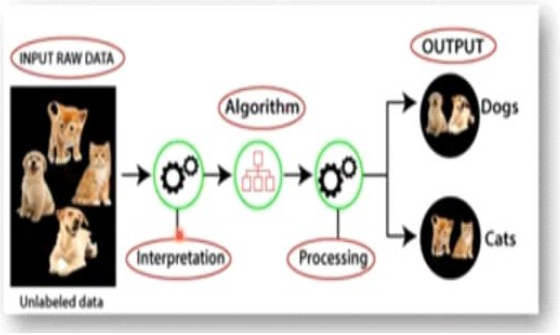
Age	Salary
22	10000
23	12000
24	14500
29	20000
31	22000
35	25000
42	34000
48	40000
54	48000
59	65000

we don't give label data I/O Here pattern are checked with cluster suppose we give different shape in input like triangle

which needs to recognize the pattern in unsupervised from other shapes **now when we give new shape** than it does not match with other shape then this rectangle shape puts into that group

Working of Unsupervised Learning Algorithm

- We have taken an **unlabelled input data**, which means it is **not categorized and corresponding outputs** are also not given.
- Now, this unlabelled input data is fed to the machine learning model in order to **train** it.
- **Firstly**, it will interpret **the raw data to find the hidden patterns** from the data and then will **apply suitable algorithms** such as k-means clustering, Decision tree, etc.
- Once it applies the suitable algorithm, the **algorithm divides the data objects into groups according to the similarities and difference between the objects.**



In the real world, we do not always input data with the corresponding output so to solve such cases we need unsupervised learning

in this learning use K-mean clustering

- **is your data labeled or unlabelled**? Supervised learning requires labeled datasets. You'll need to assess whether your organization has the time, resources, and expertise to validate and label data.
- **What are your goals?** It's important to consider the type of problem you're trying to solve and whether you are trying to **create a prediction model or looking to discover new insights or hidden patterns in data.**

  There are two type of Unsupervised Learning

  - **Clustering – making an group after separation in different objects**
  - **Association**

An Association rule is set an unsupervised learning method used for finding the relationship between variables in the large database

It determines the set of items that occur together the dataset

Example :

Market Basket Analysis :

Reinforcement Learning

Reinforcement Learning is a feedback-based machine learning Technique in which an agent learns to behave in an environment by performing an action and seeing the result. For a good action, the agent gets positive feedback, and for a bad action, the agent gets negative feedback or a penalty.

There is no labeled data. So the agent is bound to learn by its experience

Reinforcement Learning solves a specific type of problem where decision-making is sequential and the goal is long-term, such as game-play robotics.

Advantages of supervised

With the help of supervised learning the model can predict on the basis of prior experiance

In Supervised Learning, we can Have an exact idea about the class of the object

Supervised Learning model helps us to solve various real world problem such as fraud detection, spam filtering

Disadvantages

Supervised Learning models are not suitable for complex task

If the test data is different from the training data then it can not predict the correct output

Training required lots of computation times

We need enough knowledge about class of objects means images of Fruits and their label

Perceptron Rule

Understanding the Perceptron

A perceptron is a type of artificial neural network **invented in 1958 by Frank Rosenblatt**. It is the simplest form of a neural network, used for binary classification tasks.

Perceptron is a linear binary classifier used for supervised learning.

Here are the key components and concepts related to a perceptron:

Input Nodes: These are the features of the input data. Each input node is assigned a weight.

Weights: Each input is multiplied by a weight which can be adjusted during the learning process to minimize error**.**

Summation: The weighted inputs are summed together.

Activation Function: The sum of the weighted inputs is passed through an activation function. In the case of a perceptron, a

step function is often used. The output of this function is the prediction of the perceptron.

Bias: An additional parameter that allows the activation function to be shifted to the left or right, improving the model's fit.

Perceptron Rule is Supervised Learning algorithm because we have supervision of output mapping

Perceptron **are binary classifier which converts** the data in two classes

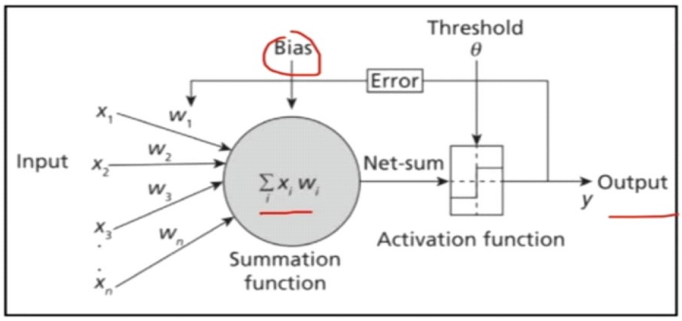
It is mapping of I/O in x, y

Perceptron Learning – Introduction

The perceptron model consists of 4 steps:

1. Inputs from other neurons
2. Weights and bias
3. Net sum
4. Activation function

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



Like, Share and Subscribe to Mahesh Huddar — Visit: vtupulse.com

X	Y

$$Y = X_1W_1+X_2W_2.....X_nW_n +b$$

Step function : Y > 0 -> 1

Y < 0 -> -1

So this is also activation function

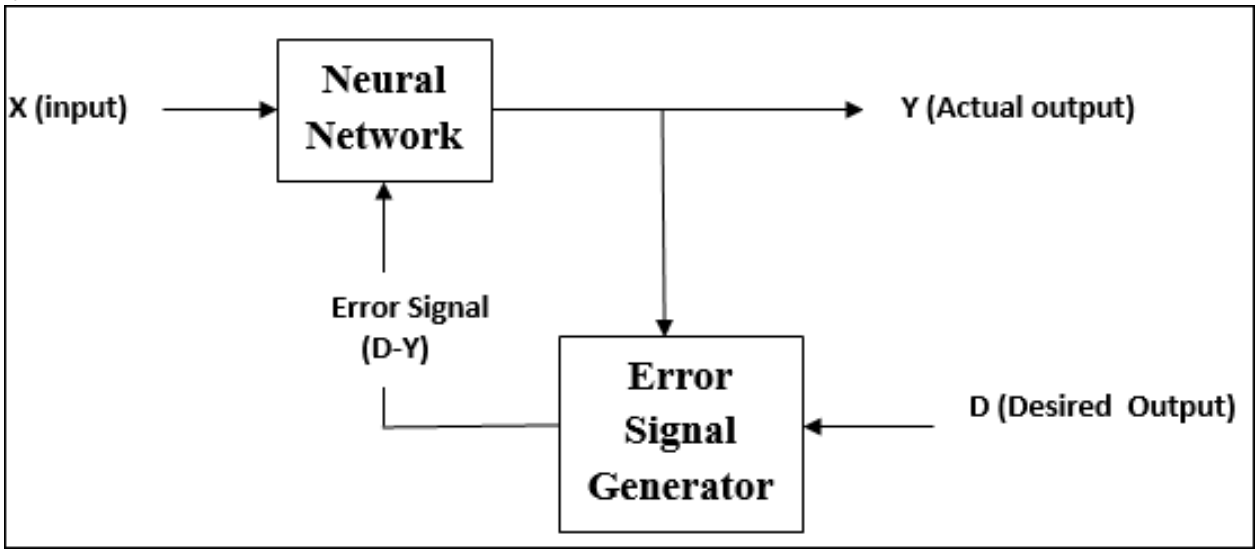
error e(t) = Y (desired o/p) – Y (Actual o/p)

Set the initial value is w1, w2……. Wn

In Supervised Learning, **we have input and also output which is labeled data or training data**

On the basis of training data, we create a Model and again give new input and check the output

q



On the basis of Label Data/Training (Input & Output) data, an **Error signal generated** if difference between Actual Output and Desired Output

On the basis of Error Signal weights are adjusted until both Actual output is matched with Desired Output

Linear Regression

Y=mX+b

Y represent the **dependent Variable** //to check score

X represent the **independent Variable**//Study Hrs.

Activation Functions in Neural Networks

Sigmoid, Hyperbolic Tangent Function (Tanh),

Softmax, softsign function

Need of bias

Now, Suppose if b was absent, then the graph will be formed like this: Due to absence of bias, model will train over point passing through origin only, which is not in accordance with real-world scenario

w1=0.5

w2=-0.3

b1=0.1

Why we use Activation functions with Neural Networks?

It is used to determine the output of neural network like yes or no. It maps the resulting values in between 0 to 1 or -1 to 1 etc. (depending upon the function).

The Activation Functions can be basically divided into 2 types-

1. **Linear Activation Function**
2. **Non-linear Activation Functions**

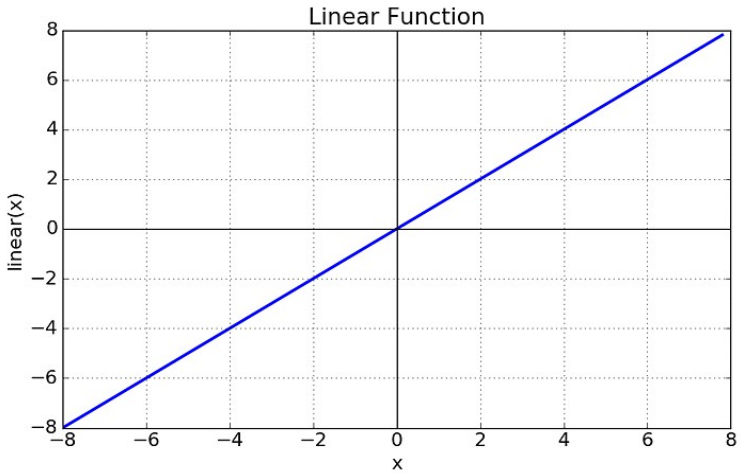


Fig: Linear Activation Function

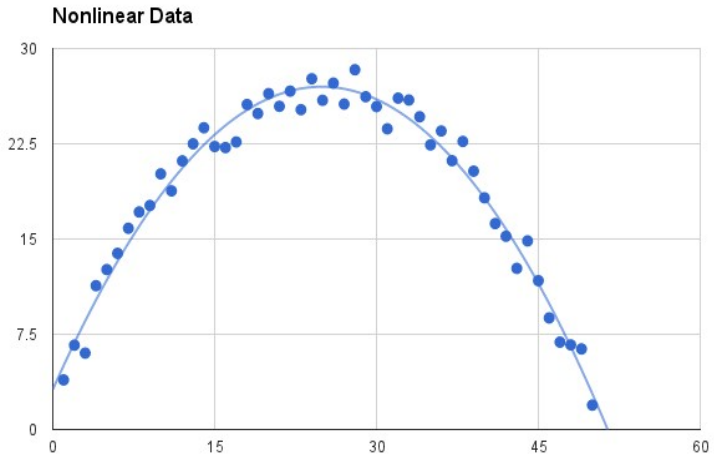
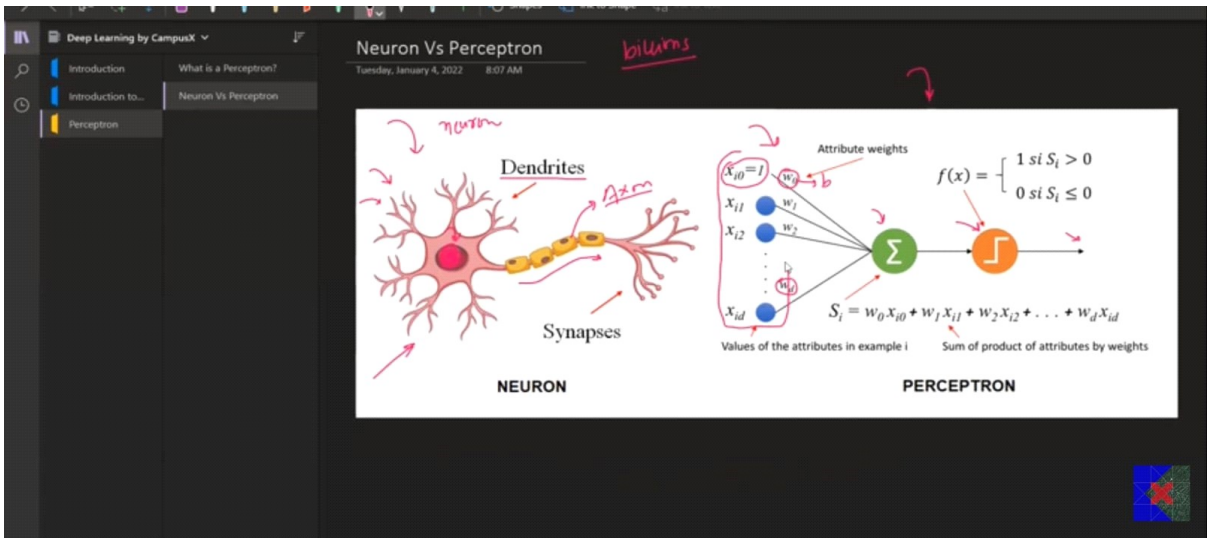


Fig : Non Linear Activation Function

nucleus in anatomy is a brain structure (plural = nuclei). It is a compact cluster of neurons.



Yes a perceptron (one fully connected unit) can be used for regression. It will just be a linear regression. **If you use no activatio**n function you

get a regression **and if you put a sigmoid activation you get a classifier**

Perceptron is a binary classifier

And Perceptron divides the data into two region means it is Line

And **Perceptron will classify the Linear data** means which is **separable Data through a Line**

$Y = f(z)$

$Y = X1.w1 + X2.w2 + b$

$w1=A$, $w2=B$, $b=C$,

$X1=x$, $X2=y$

Here $AX + BY + C$ is line

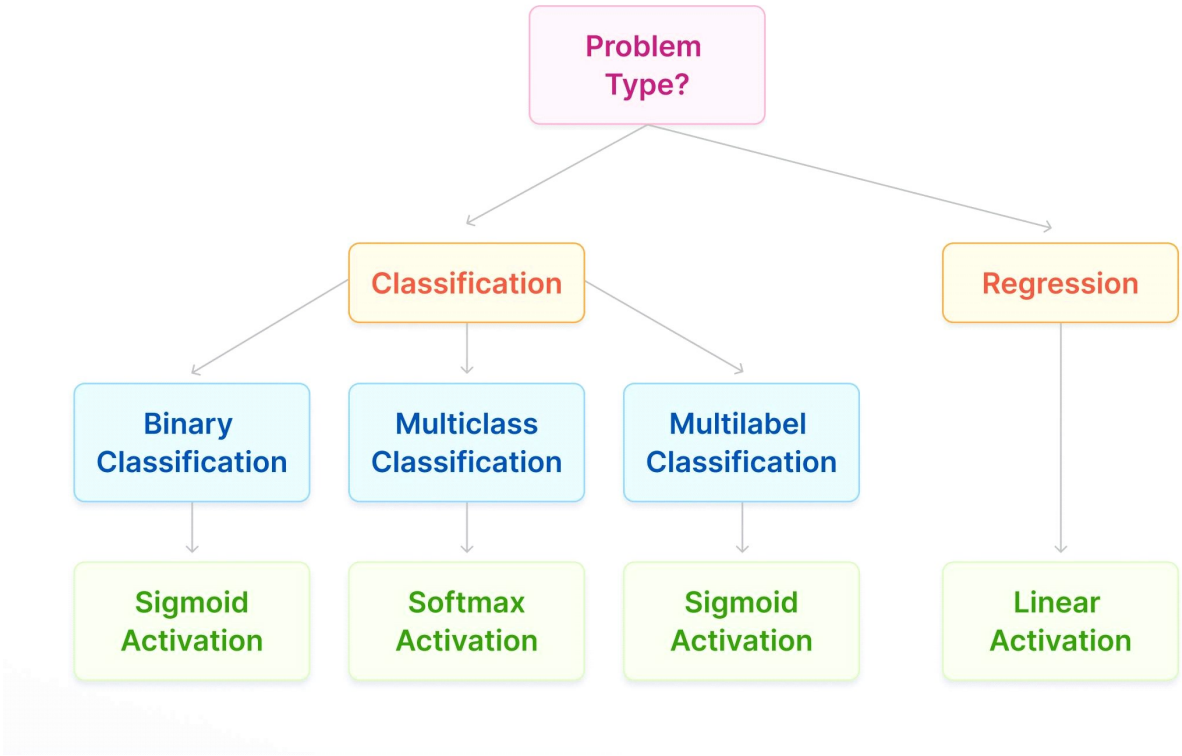
$Ax + BY + C > 0$ True then

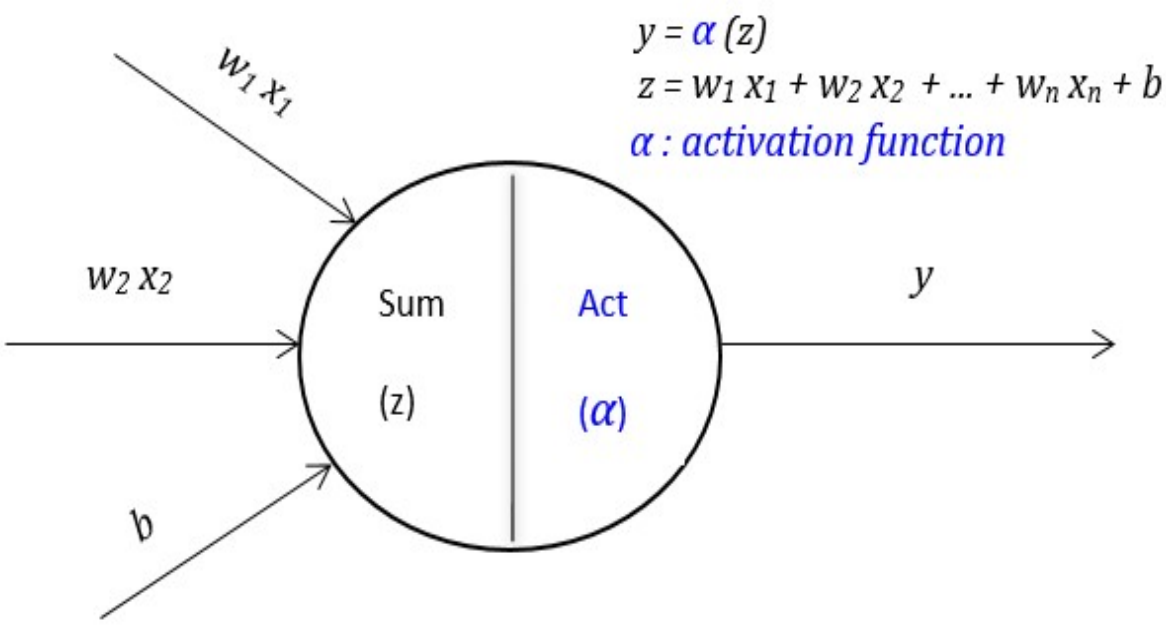
$f(z) = w1.x1+w2.x2+w3.x3 + b$

$ax+by+cz+b > 0$

Linear Regression

$Y=mX+b$


Problem Type?
Classification
Regression
Binary Classification
Multiclass Classification
Multilabel Classification
Sigmoid Activation
Softmax Activation
Sigmoid Activation
Linear Activation


$y = \alpha(z)$
$z = w_1 x_1 + w_2 x_2 + \ldots + w_n x_n + b$
α : activation function
$w_1 x_1$
$w_2 x_2$
Sum
(z)
Act
(α)
y
b

- Linear summation of inputs: In the above diagram, it has two inputs x1, x2 with weights w1, w2, and bias b. And the linear sum z = w1 x1 + w2 x2 + … + wn xn + b

  This Linear Activation Function

Regression Analysis Algorithm :

Linear Regression, Logistic Regression, Support Regression, Decision Tree Regression, Random Forest Regressy the data on

Logistic Regression : AX1+BX2+C

Suppose if we have extra line in graph. Then we can use AX1+BX2+CX3+D

In perceptron we use Line that we want to search line which could classify that is A, B, C means we pick random value for a line to get the maximum probability

which we take a random value

0 => placed student

X => Not placed student

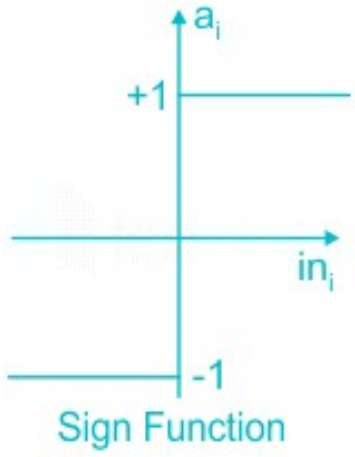
2D

Ax1+Bx2+C=0

3D

Ax1+BX2+Cx3+D=0

Perceptron Rule shows :



In Perceptron if X input and Weight vector are declared then

Perceptron is a type of linear classifier, while logistic regression is a classification algorithm that can predict probabilities

$P = X_1W_1+X_2W_2…..X_nW_n + b$

$P = \sum W_i * X_i$

P=1, if P > 0

P=-1 if P < 0

P=0, if P=0

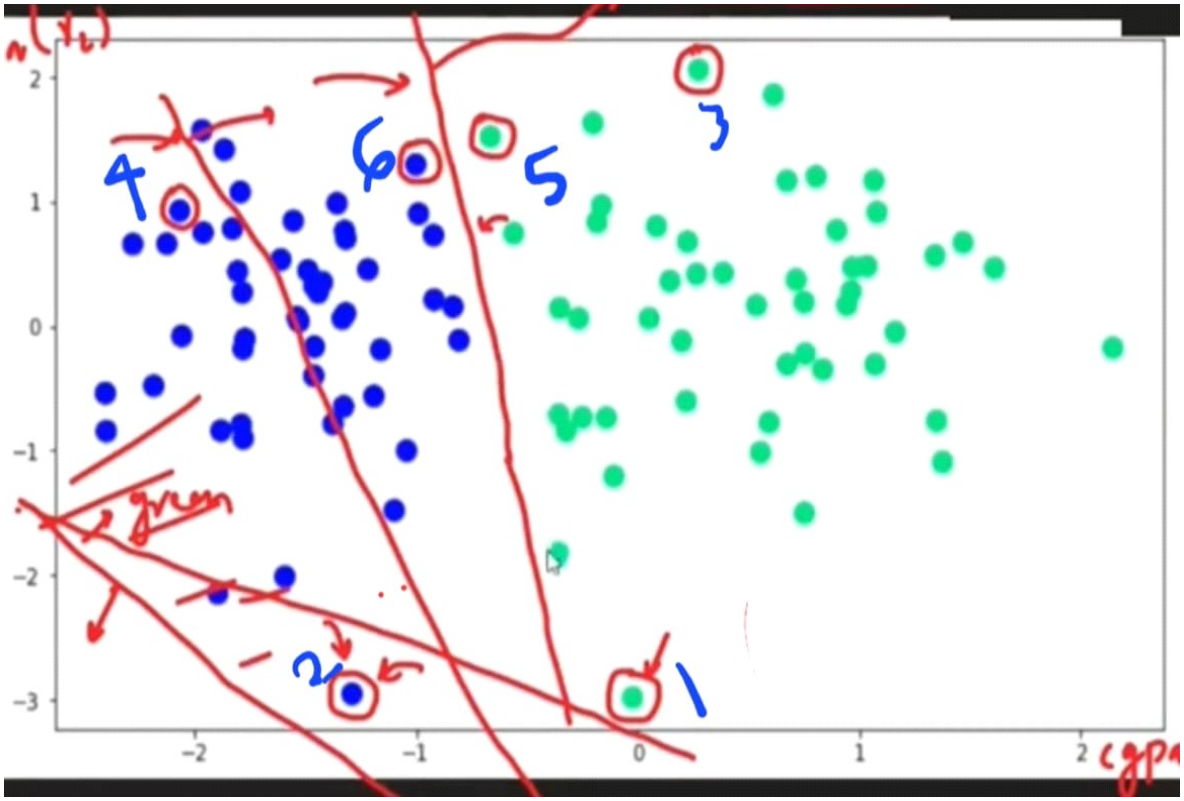
Perceptron Rule

in Perceptron we use Loop

Our goal is to find the line in the Graph means We have to find out ABC Value to classify the data

- We start A=1, B=1, C=0 from Random Value
- Run Loop and
- 3
- Select a point or student
- Check the point in the positive region or classified data
- Again check if it is a misclassification
- Change the ABC value and to make classification data

- Again repeat step no. 3,4 and 5 then go to step no. 6



Loop can be run in 1000 times or the second option till while convergence check how many points are misclassified until it becomes zero the loop will be executed after that it be terminated

And the convergence done successfully loop will be terminated

Here we have convergence and another option epoc means no. of

To classify the x data and 0 data use loop in perceptron trick

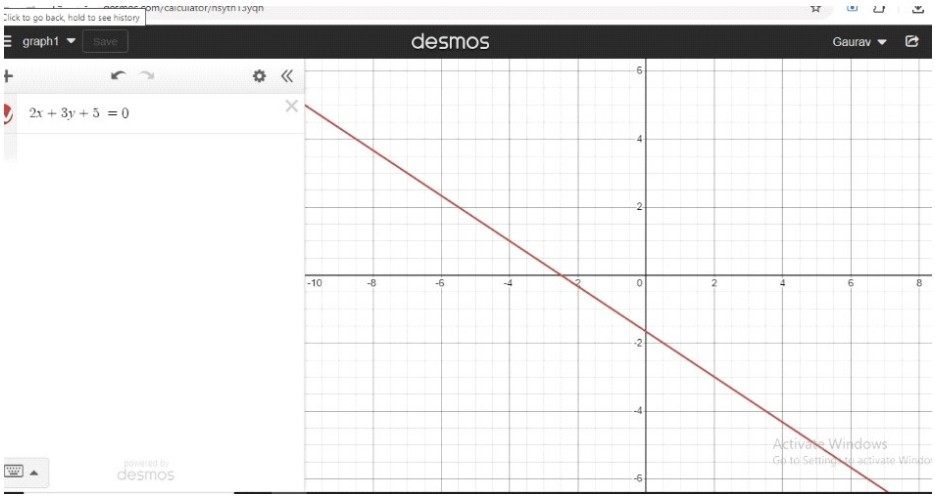
Until the data classify may be loop will be 1000 time either 10000 loop

In every loop we select random student that if the green student is in correct position according to line then no changes in A,B,C value

Now How to identify Positive Region and Negative Region

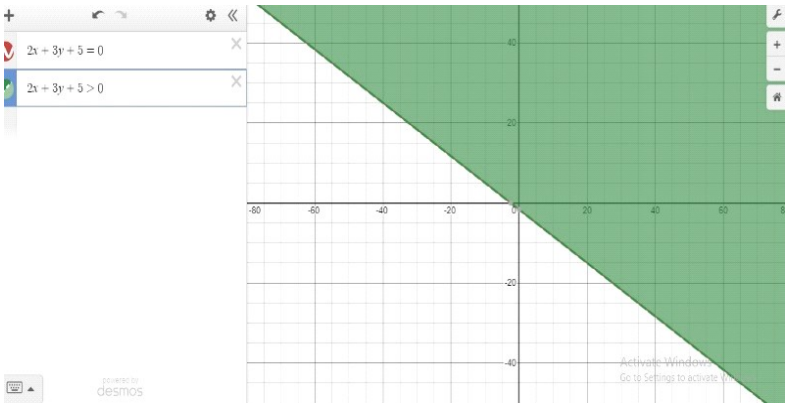
Now how to identify positive reason and region for this we take tool https://www.desmos.com/calculator

2x+3y+5 =0



Again given new line coordinate value

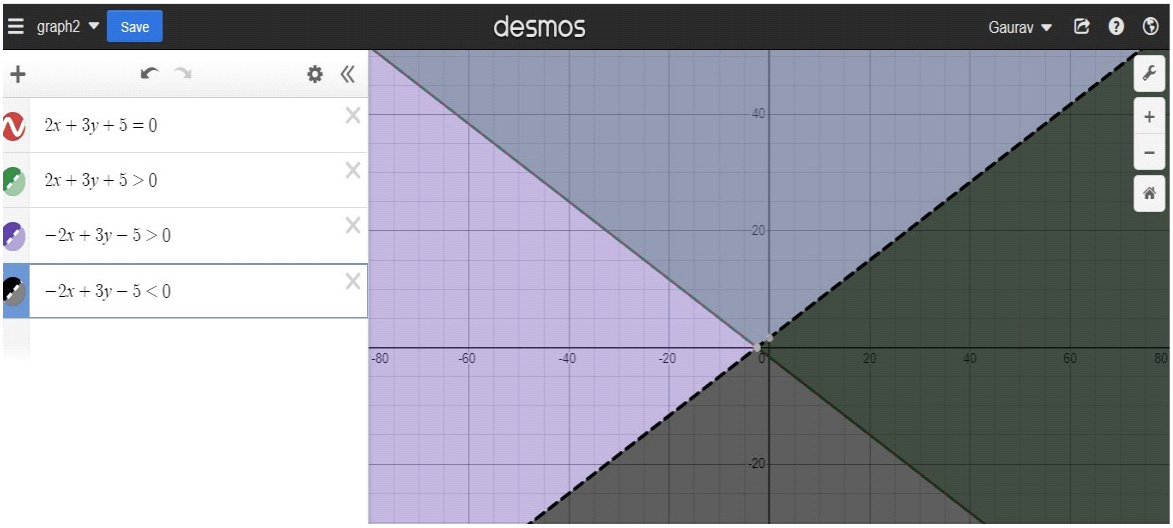
2X+3y+5>0 then it gives positive reason



Now u can check point whether it is positive reason or not

we know line equation : Ax+By+C=0 suppose these two points coordinates x1 and y1 then we put x1and y1 into equation

then eq: Ax1+By1+C if it is > 0 then point is in Positive reason else negative reason and it is = 0 then point is over the line

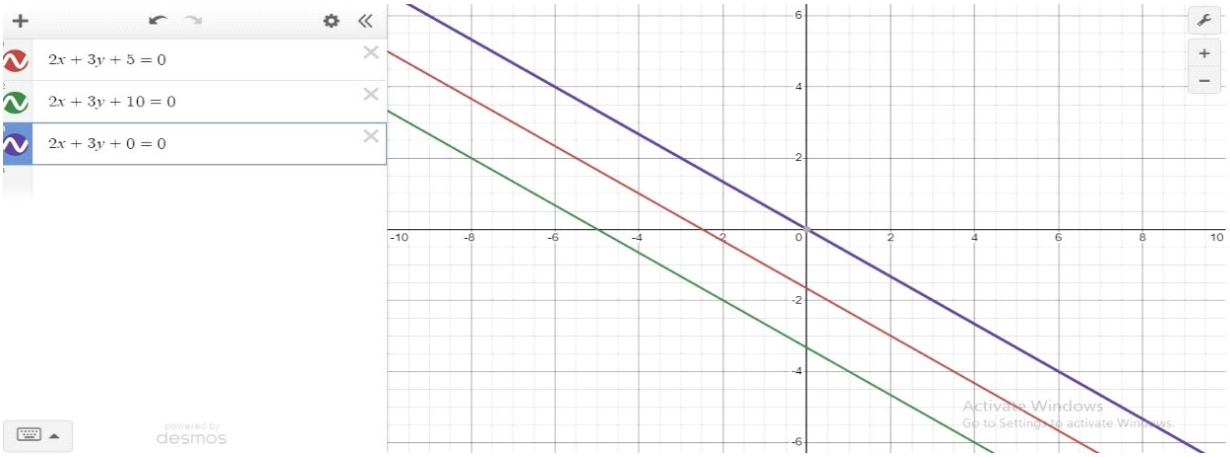


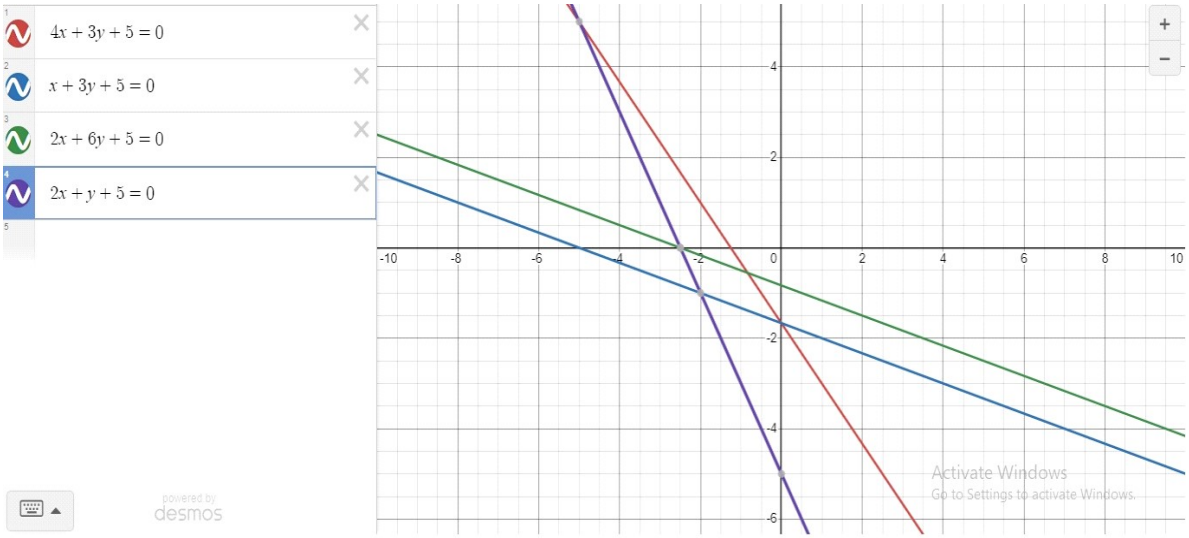
-2x+3y-5>0 then it is positive region and -2x+3y-5<0 then negative region

If the value is Increased -> Down line

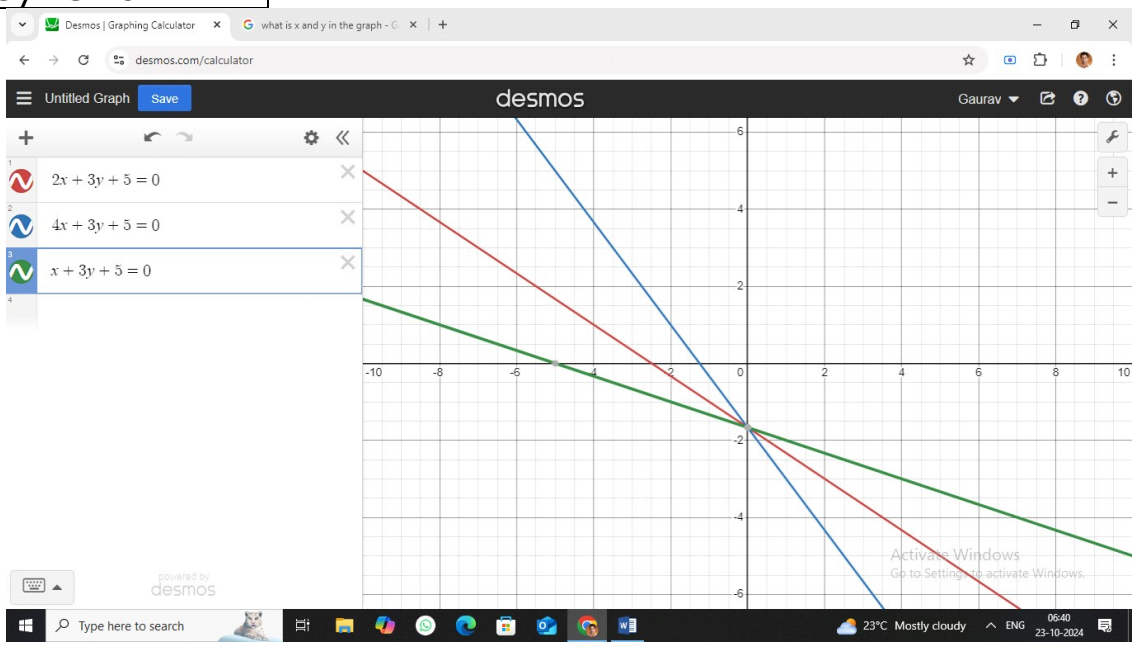
If the Value is Decreased -> Up line

Now here how the line is moving when **C value** is increasing then line is going down and decreasing then it is up side





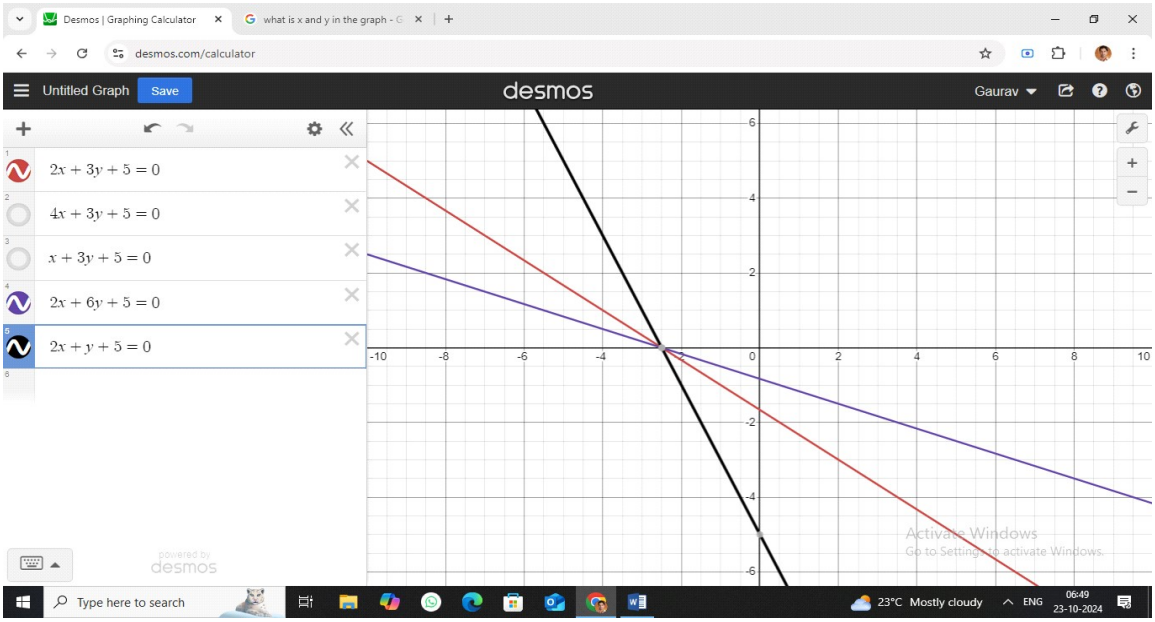
X Value
2x+3y+5=0
4x+3y+5=0
x+3y+5=0



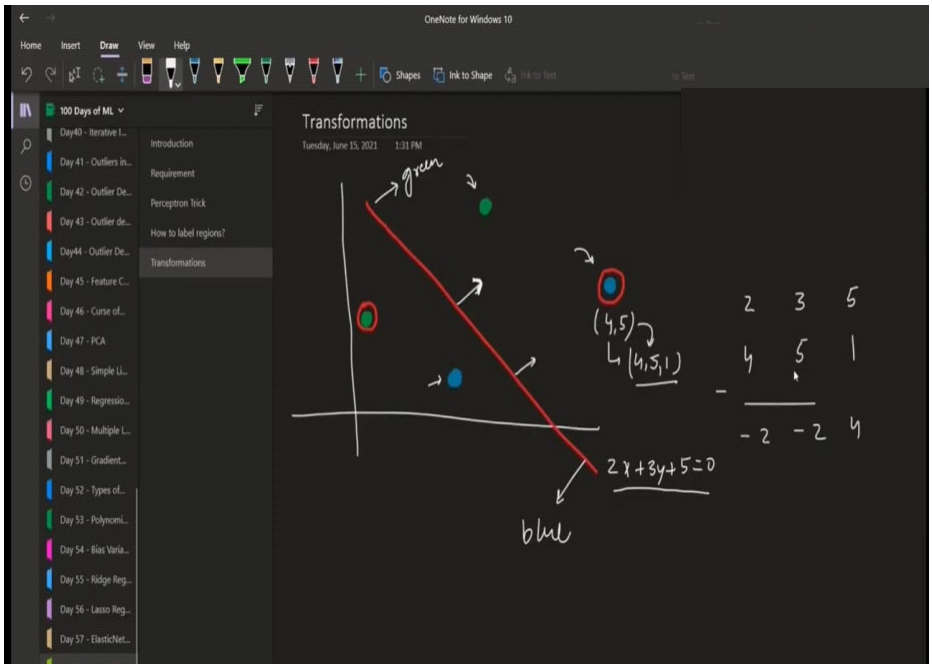
here x and y value is changed suppose your line is 2x+3y+5=0 and blue line coordinate value is (4,5) then add 1 -> 4,5,1

and subtract it

Y Value
2x+3y+5=0
2x+6y+5=0
2x+y+5=0

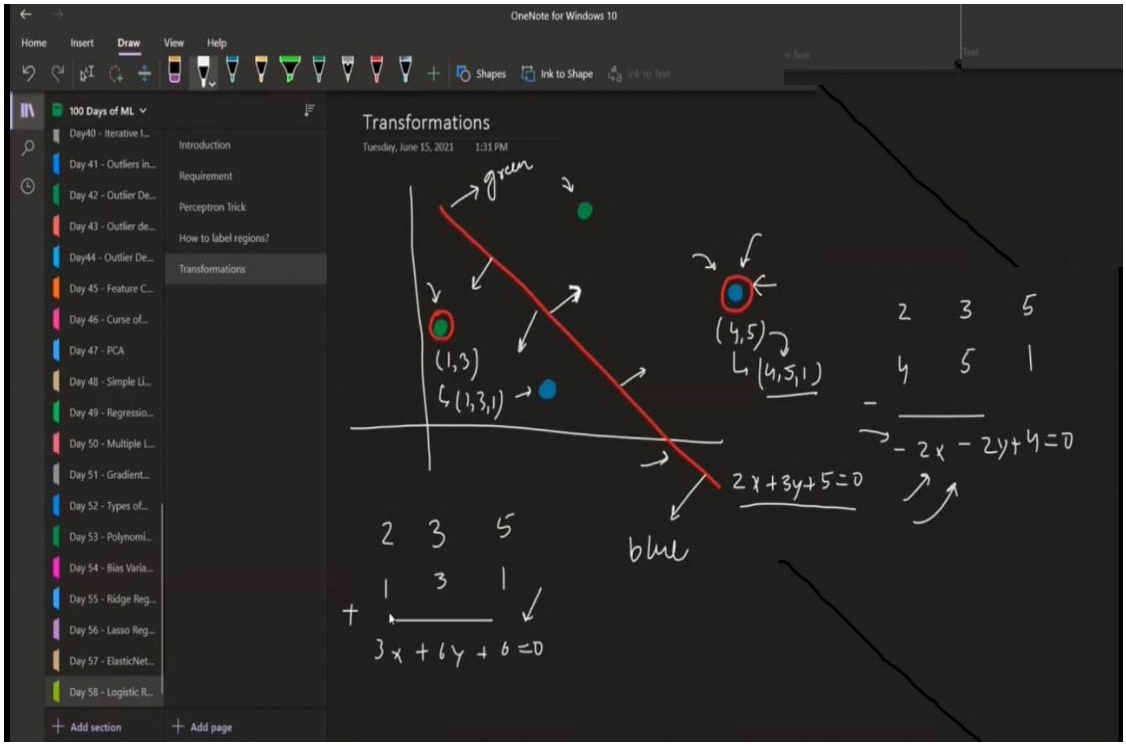


New right side blue point is wrong position



to correct this take coordinate value i.e, 4,5 and add 1 which is 4,5,1 and subtract from 2 coefficient 5 which -**2 -2 4 i.e,**

-2x-2y+4=0



Again check green point which is negative region so now apply 1,3,1 and add 2,3,5 =**3x+6y+6=0**

right side coordinate value 5,2 and left side -3,-2

-3x+y+4=0

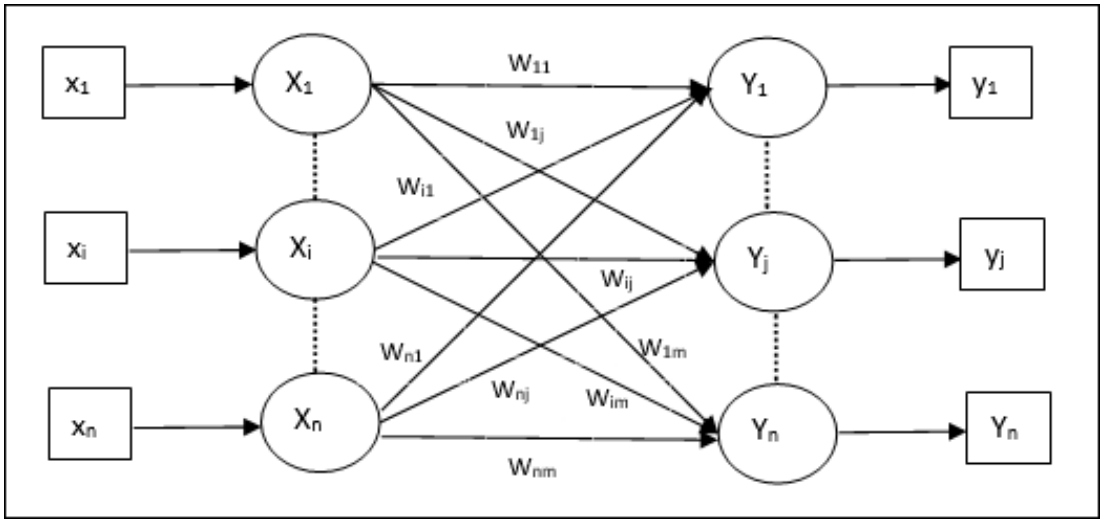
Auto Associative Memory Network: it is obtained by its content

If any content is missed, then associative memory completes this task

This is Single Layer architecture

Auto Associative Memory

This is a single layer neural network in which the input training vector and the output target vectors are the same. The weights are determined so that the network stores a set of patterns.



Associative memory is also known as content addressable memory (CAM) or associative storage or associative array.

Training algorithm

Input Data is Vector Input "S" and Output Data Vector Output "T"

Will be same output S=T

Input will be **partial** information and **Noisy** Information **Output** will be same

For training, this network is using the Hebb or Delta learning rule.

Step 1 – Initialize all the weights to zero as $w_{ij} = 0$, i=1 to n, j=1 to n
Step 2 – Perform steps 3-4 for each input vector.
Step 3 – Activate each input unit as follows –

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

Step 4 – Activate each output unit as follows –

$$y_j=s_j(j=1 \text{ to } n)$$

$$w_{ij}(new)=w_{ij}(old)+x_iy_j$$

Hebb rule :
Calculate the New Weight value
Activate each input unit as follows –
xi=si(i=1 to n) xi=si(i=1 to n)
Step 4 – Activate each output unit as follows –
yj=sj(j=1 to n) yj=sj(j=1 to n)
Step 5 – Adjust the weights as follows –
wij(new)=wij(old)+xiyj

Outer Product Rule

Wn = $[S]^t$ * [T]
t=transpose matrices, T=Target Matrics
Check the auto-associative network for input vector [1 1 -1] from the weight vector with no self connection. Test whether the net is able to recognize with one missing entry.
Input vector x = [1 1 -1]
And here no self connection in Weight -> No W_i Which is 0
Here St= target input and transpose of input

$$W = \sum S^T \times t = |1|$$

$$|1| [1\ 1\ -1]$$

|-1|

[1x1 1x1 1x-1]
[1x1 1x1 1x-1]
[-1x1 -1x1 -1x-1]

[1 1 -1] [0 1 -1]
[1 1 -1] = [1 0 -1]
[-1 -1 1] [-1 -1 0]

In weight No Self connection i.e, 0 so matrix will diagonal value zero

Now Test Input with one missing entry so X = [1 0 -1]

Y = x.W = [1 0 -1] [0 1 -1]
[1 0 -1]
[-1 -1 0]

[1x0+0x1+-1x-1 1x1+0x0+-1x-1 1x-1+0x-1+-1x0]

[0+0+1 1+0+1 -1+0+0]

[1 2 -1]

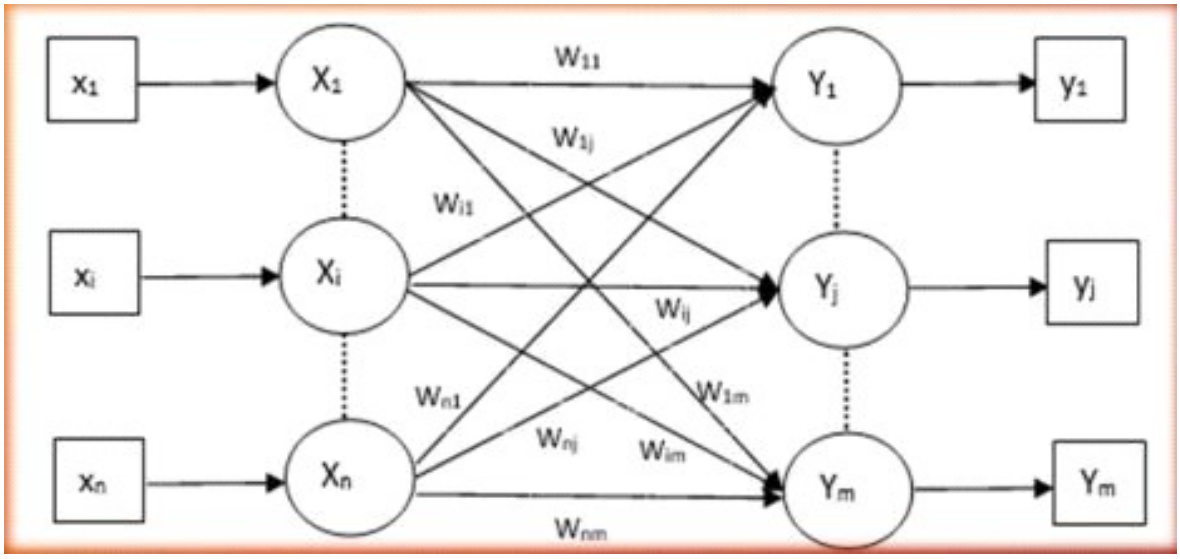
Hetro Associative Memory

It is capable of retrieving piece of data one category upon presentation of another category.

Here one category given input then it gives similar type output

Here Output Ym and input Xn

SO input set S not equal to Output T (Target data set)



Gradient Descent

Gradient Descent is an optimization algorithm which means to find out best result for the algorithm and it is a optimization technique if we give differentiable function then gradient function will give the minimal result

here i have taken data set suppose here cgpa lpa of four student

cgpa	lpa

Real world data affects the real world factor suppose student not place in desired lpa may be he was not suited or qualified interviewed which is called stochastic error

and due to stochastic error data becomes sort of linear

Backpropagation

It is a algorithm to train neural network

This is a supervised learning algorithm used for training a deep-learning model

This learning method is the most popular at the moment because it makes possible the use of powerful calculation with low computation time

Backpropagation Algo => Train NN

In backpropagation we check errors after the calculation of the Linear Activation Function

$$\mathbf{Y} = \sum Input * Weight + Bias(b)$$

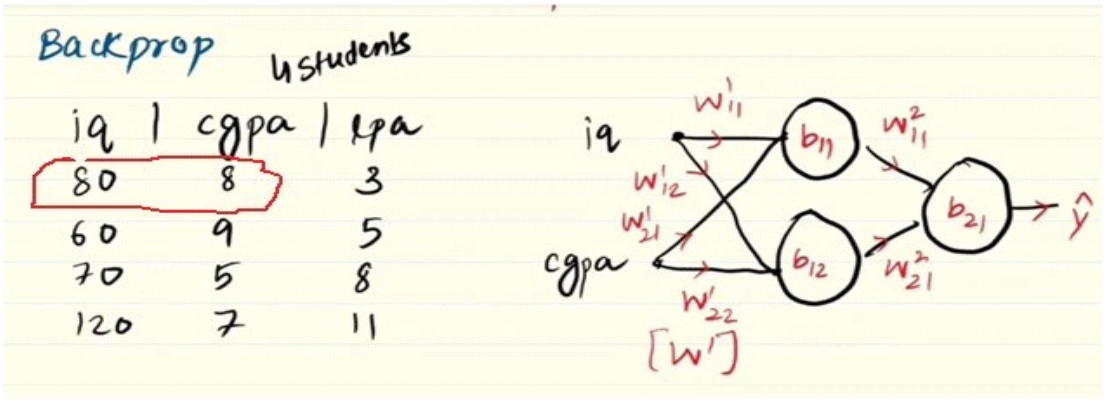
$Y = X_1W_1+X_2W_2…..X_nW_n +b$

In Neural Network has two thing :

Weight and Bias so which we have to calculate

Before calculation we should know two algorithm

- Gradient Descent
- Forward Propagation



Suppose we found 18 LPA for first student, after calculation of the Linear activation function

But It is given 3 LPA now we use the Loss function

Here X vector input data and Y Vector our data are not same.

Step 1 : Initialize W=1 and b=0 value

Step 2 : Select a point row or student

Step 3 : Choose a loss function and calculate mean squared error MSE which is given below :

Where y is actual data and is prediction data

$$L = (y - \hat{y})^2$$

$L = (3-18)^2$

$= 225$

Now we have to reduce error because Neural Network is 3 LPA here actual data output y i.e, 3 and $\hat{y}$ i.e, 18 which prediction of Neural Network

$\hat{y}$ Here is O_{21}

$O21 = W^2_{11} . O11 + W^2_{21} . O12 + b21$

Here ^Y i.e, O21 is dependent on 5 input

$\hat{y} =$

Similarly, b11 i.e, O11 is dependent on IQ, CGPA, $W^1_{11,}$ W^1_{21} and b11 on it self.

Now can these are complex hierarchy so If I have to minimize loss then we can change the value of bias and weight value but we cannot change cgpa and iq value because that is our data

So to minimize and decrease the loss then we go to previous or back step to change the value of weight and bias which is called backpropagation

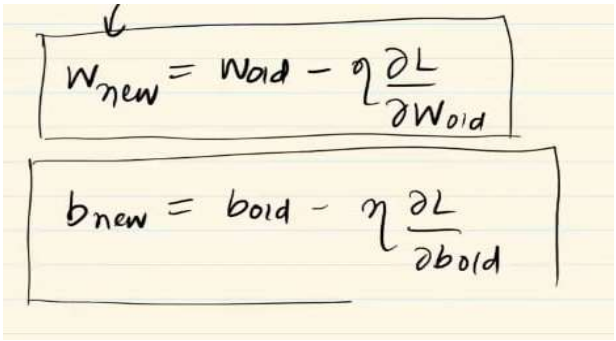
Backpropagation Error= Actual Output – Desired Output

Because we are approaching to solve this error after go to back step to update the value of Weight and Bias using with Gradient descent

Step 4 : Here update the weight and bias value using with gradient descent

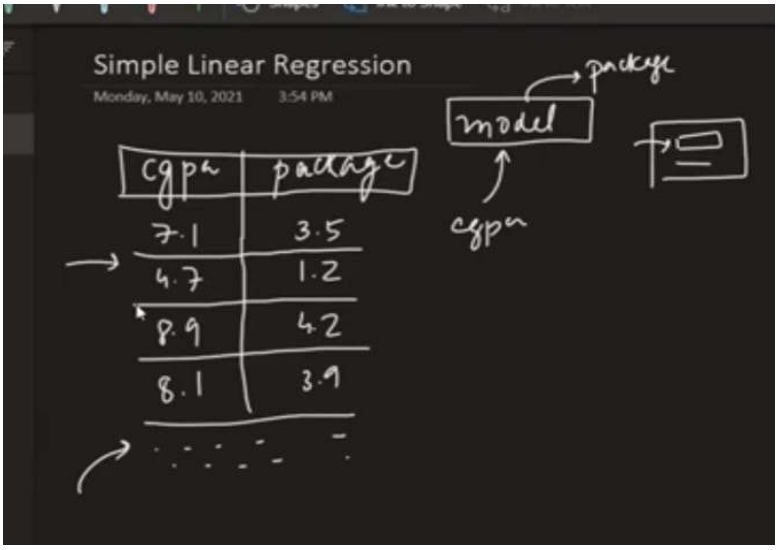
W new = W old – Learning rate times partial derivative of loss function with respect to weight

Similarly, it is applicable in b value



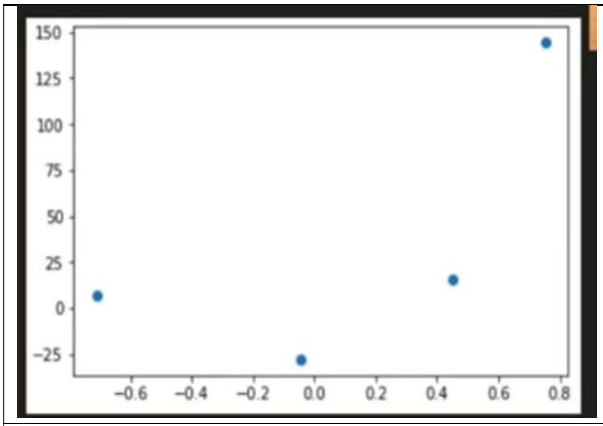
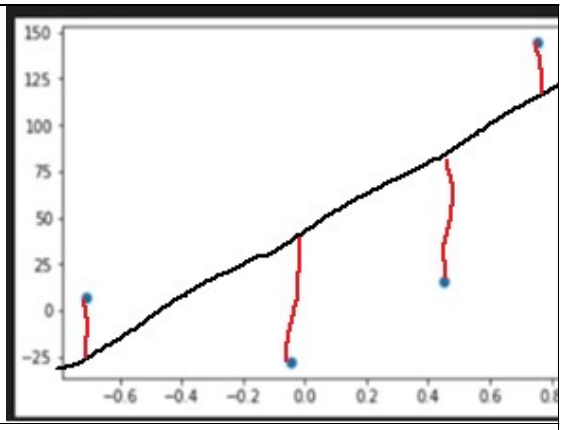
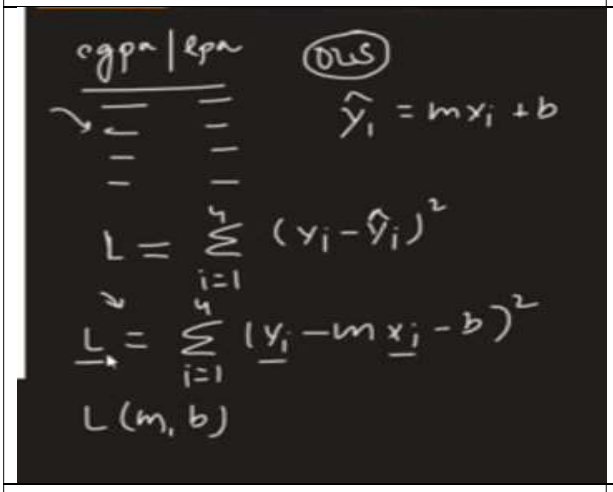
Gradient descent

Gradient descent is an optimization algorithm that's used when training a machine learning model.



We have data and we'll create best fit line

In a graph, a **residual** is the difference between the actual value of a data point and the predicted value of a data point:

	
	Create best-fit line for linear regression
	Yi= actual output n=4 student Here loss function is dependent on m and b If we change the m and b value then Suppose we also know m value m=78.35

Best fit line means which can minimize the error on y direction

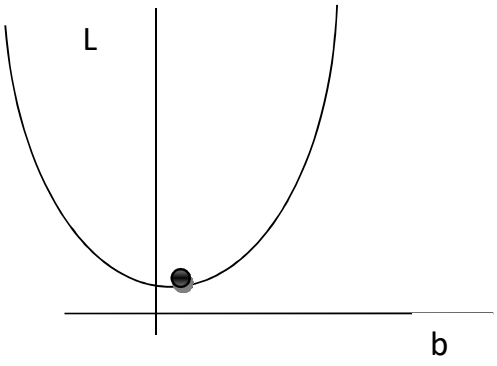
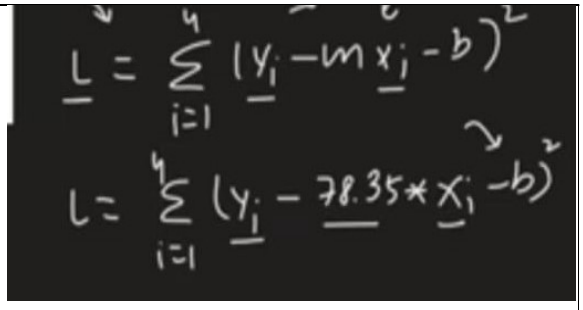
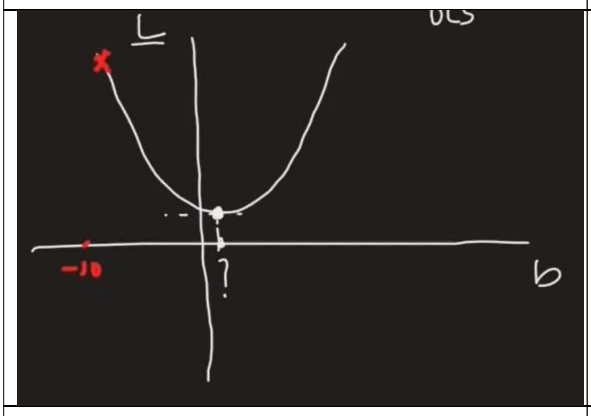
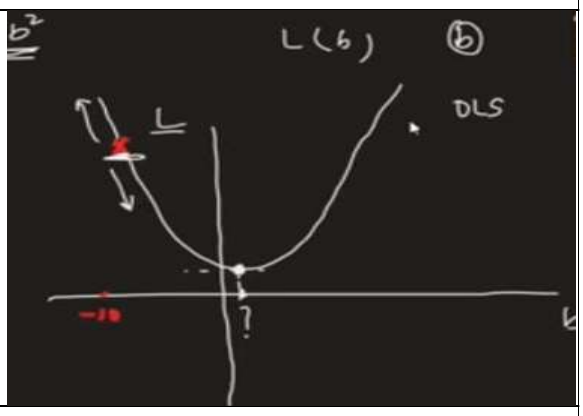
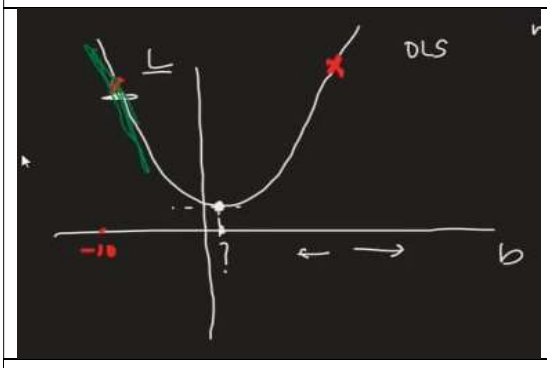
The sum square of Difference of Actual y value and predicted y value should be minimum

M is the slop in the line

Here we make a simple discussion and assumes m=78.35

$$L = \sum_{i}^{n} (yi - \hat{y})^2$$

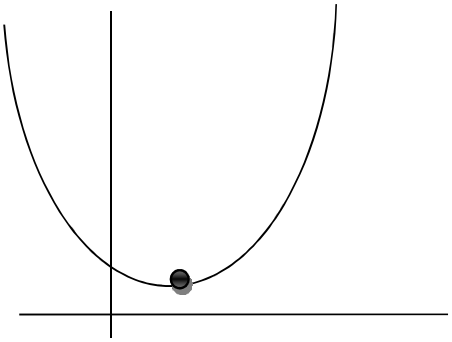
Here **we need b value which is minimum for L value** Here L and b dependent on square **L->b²** so we create a graph so it is parabolic graph

1 	
	
	Here we increase b then it close to L point And we decrease b value then it is far from minimum of L
	

Step 1 :**In Gradient Descent** select random value of b let assume b= -10 set the b value to reach minimum value at L →

How do we know to check L minimum after decreasing or increasing of b so we can find it from slope

How to find out a particular slope of a function let assume we have function



$y=x^2 + 2$ and $x=5$ given and what is slope then we differentiate $dy/dx= 2x$

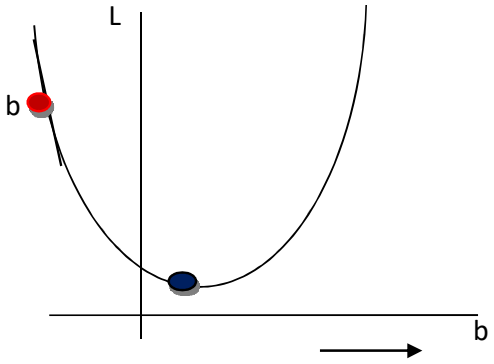
so $dy/dx = 10$ this function belongs to 10 on the slop

suppose when we find **slop and if the slope is negative then we move forward means increase the b value** and **if the slope is positive then go backward and decrease the b value** so if you want b new value

b new = b old – slope

this is gradient descent

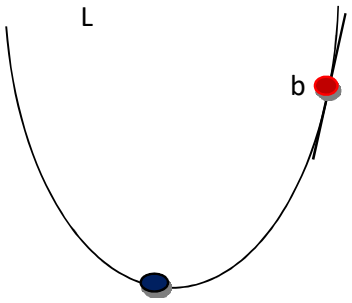
initially we had b=-10 and suppose slope = - 50
then b new = -10 –(-50)
b new = 40 which you move forward

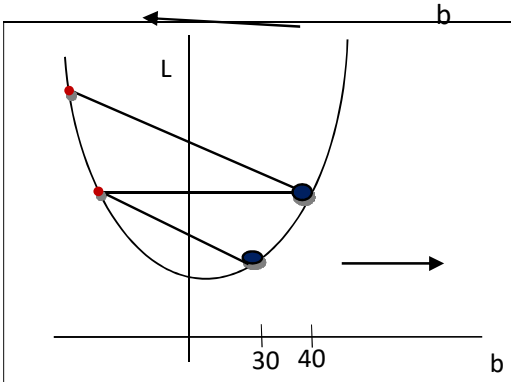


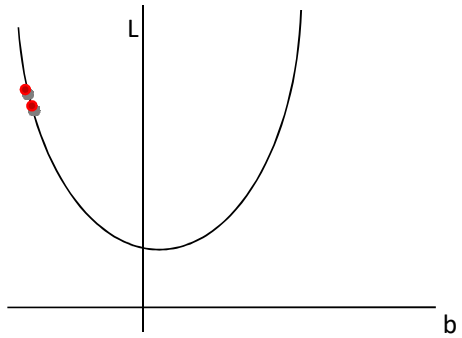
again we take b value suppose b=b10 and slop = 50

b new = b old – slope

= 10 – 50 = -40





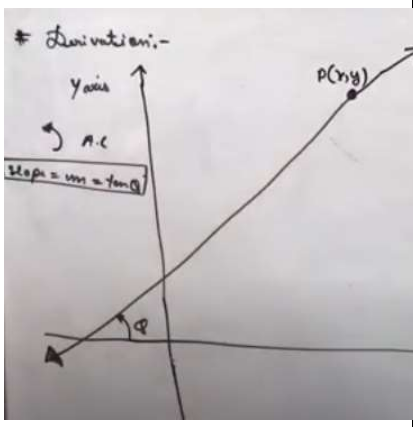


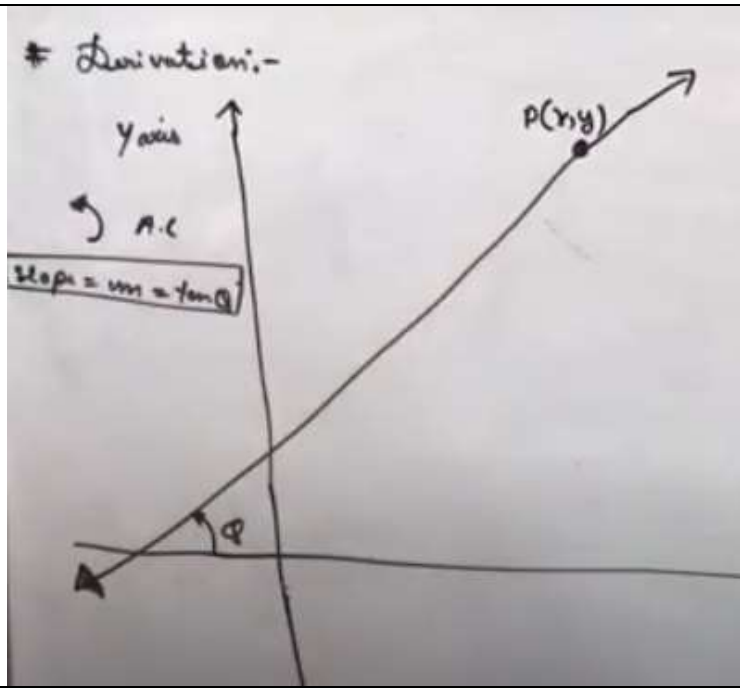
b new = b old – slope b new = -10-(-50) =40 Here it makes large jump so use now create the transform equation with Learning Rate η	b new = b old - η slope b new = -10 – (0.01*-50) = -10 + 0.5 = -9.5 Again iterate = -9.5 –(0.01* -40) = -9.5 + 0.4 = -9.1

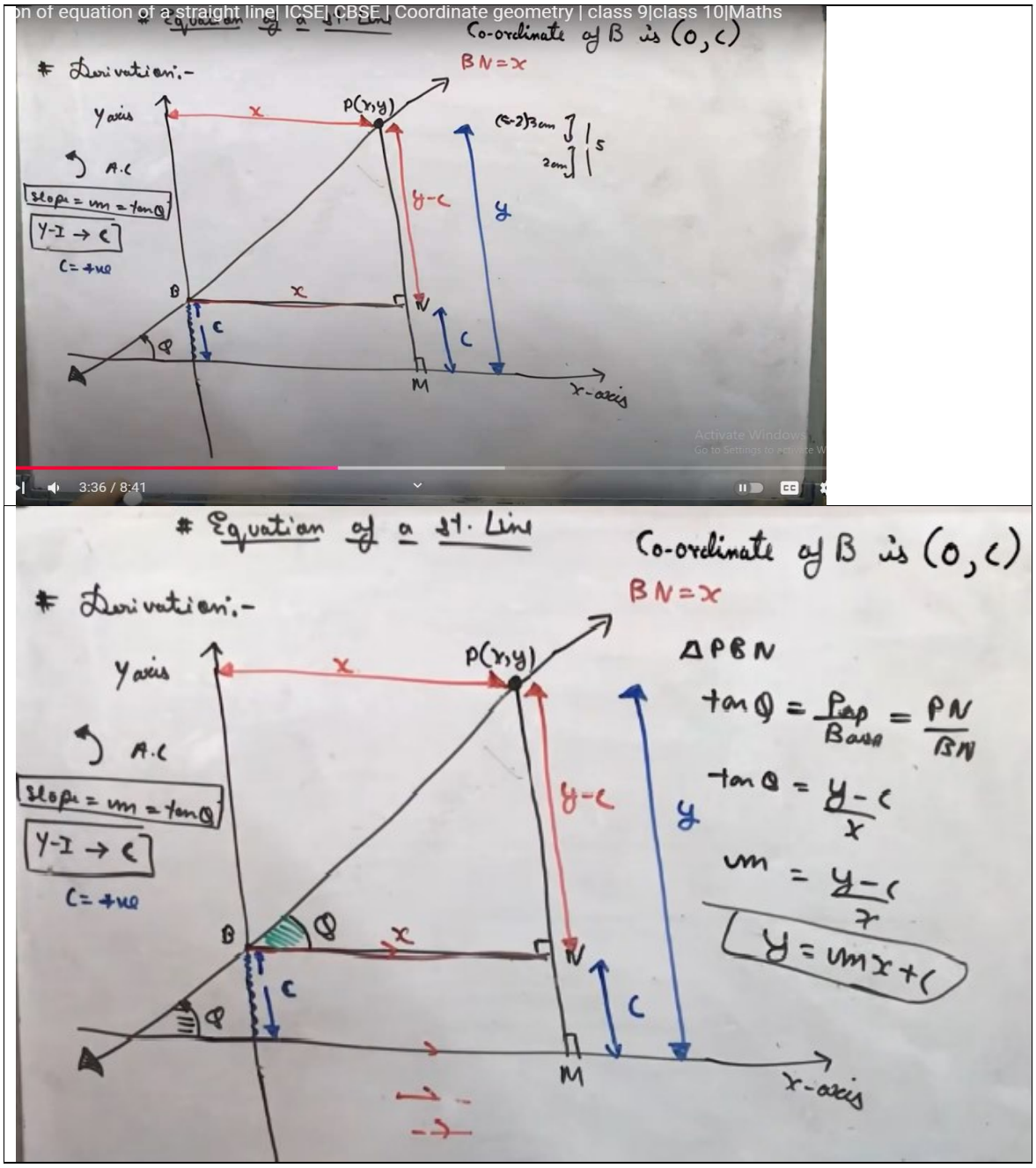
move backward

Linear Regression

Slope -> m= tanθ







Simple Linear Regression

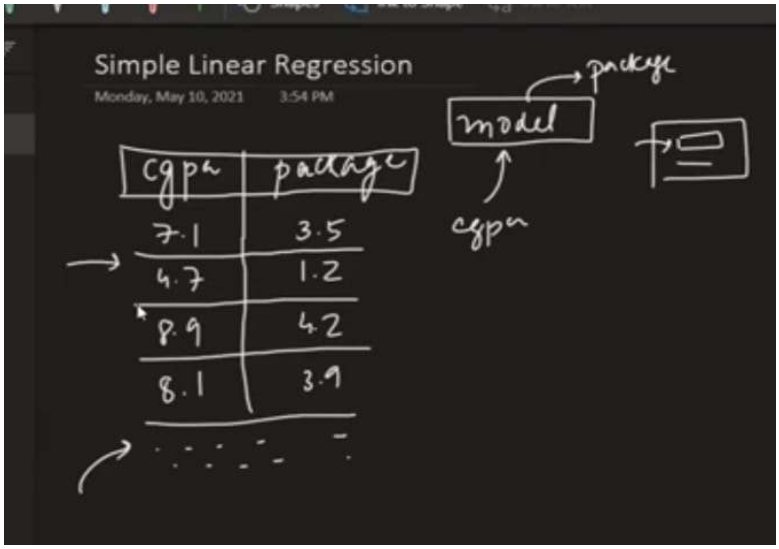
This is the simplest form of linear regression, and it involves only one independent variable and one dependent variable. The equation for simple linear regression is:

$y=mX+\beta$

where:

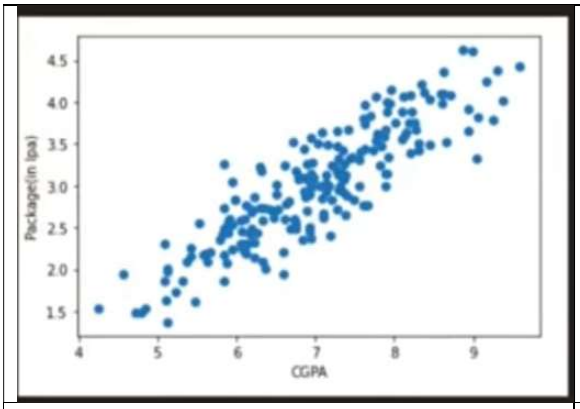
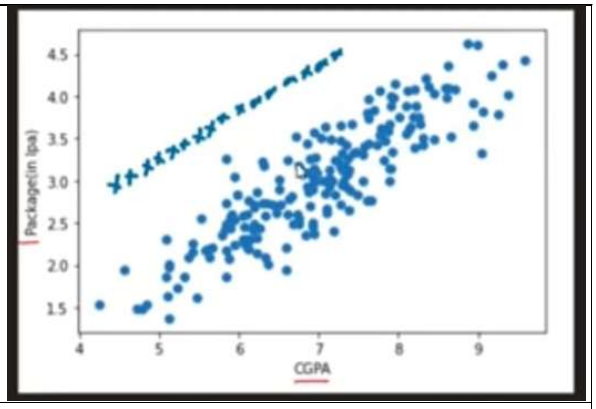
- **Y is the dependent variable**

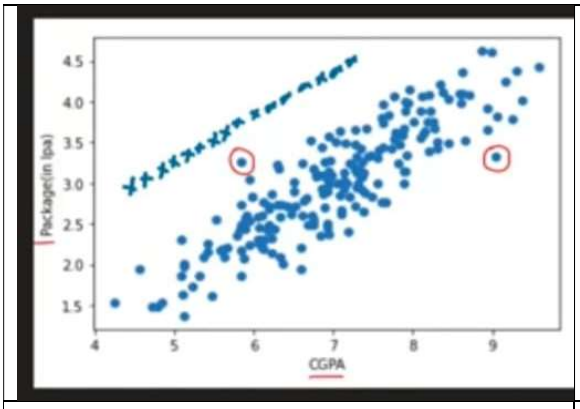
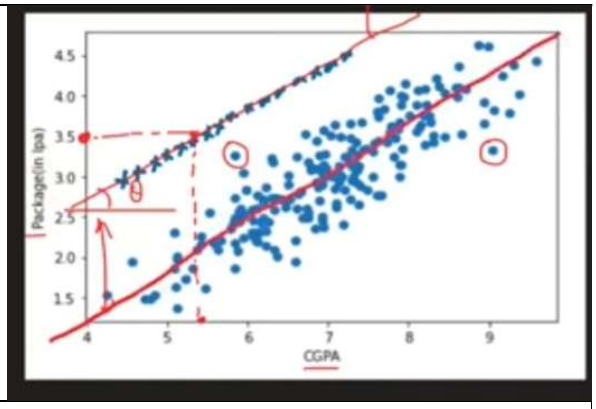
- **X is the independent variable**
- **β is the intercept**
- **m is the slope**



Suppose we have data x and y

X= cgpa and y= lpa

	
1. This is sort of linear	2. This is completely linear

	
3.	4. put best fit line

Step 1 : Initialize W=1 and b=0 value
Step 2 : Select a point row or student
Step 3 : Choose a loss function and calculate mean squared error MSE which is given below :
L=(yi-ŷ)^2
Step 4 : Here update the weight and bias value using with gradient descent

Types of Gradient Discent

- Batch Gradient Discent
- Stochastic Gradient Discent
- Mini Batch Gradient Discent

Suppose a student could not qualify enough lpa after having maximum cgpa what is reason then why does it happen **because real world data affects from real factors which u can not understand by mathematically what was reason for this lpa which you can not qyantify so this called stochastic error**

And due to stochastic error, which we can not determine and data becomes sort of linear

3. **Suppose if we have sort of linear data then we put perfect line and touch to the point which is best fit line**

Now we'll check m and b in next topic

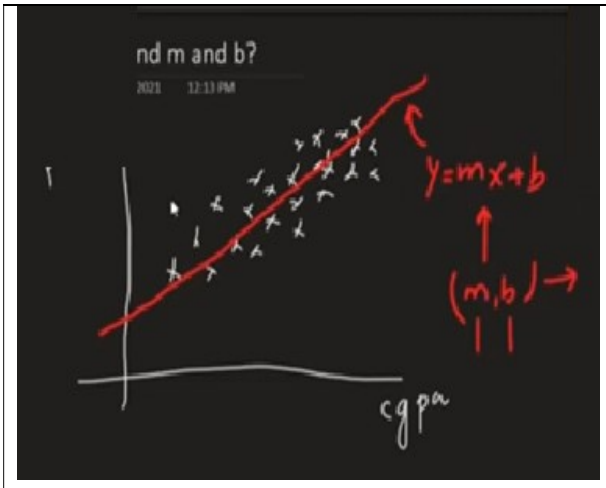
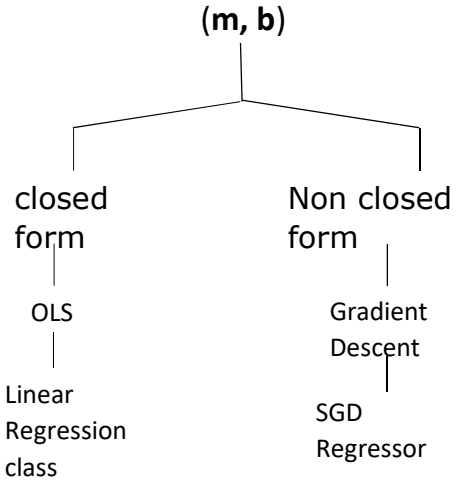


Figure 1



In Figure 1 Here need the correct value to balance **m** and **b** which cover minimum error on this line

Closed form means a mathematical formula i.e,

ols $b = \bar{y} - m\bar{x}$

$$m = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n} (x_i - \bar{x})^2}$$

Here $\bar{x}$, $\bar{y}$ is mean value x_i current cgpa and y_i is current row package

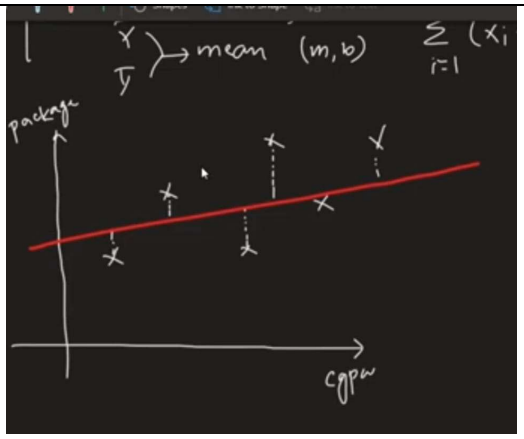
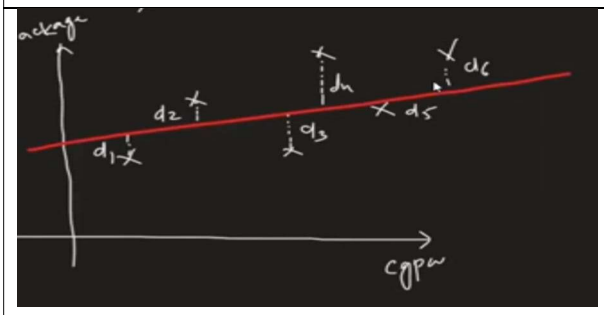


Figure 2

SGD : (Stochastic Gradient Descent) In Figure 1 Here need the correct value to balance **m** and **b** which cover minimum error on this line

Here in OLS ordinary least square we have also formula

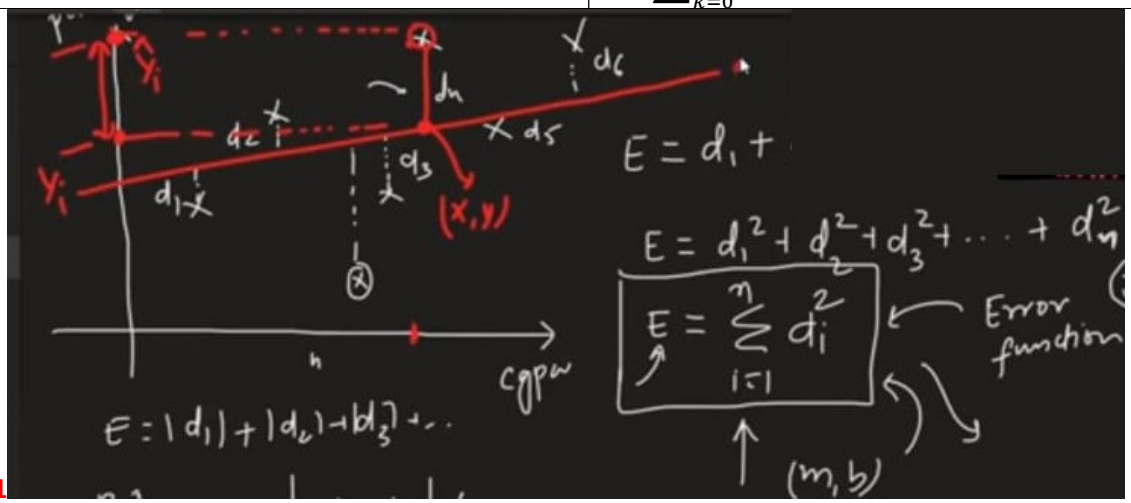
In Figure 2 shows an error because line is touched on every cgpa which shows distance from line



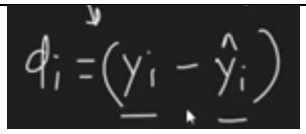
E= d1+d2+d3….+dn : in total error here + and – will cancel because some point in up side and down side

E= $d^2_1+d^2_2+d^2_3….+d^2_n$ and if it is done in square then both will be in positive

E= $\sum_{k=0}^{n} (d^2{}_i)$



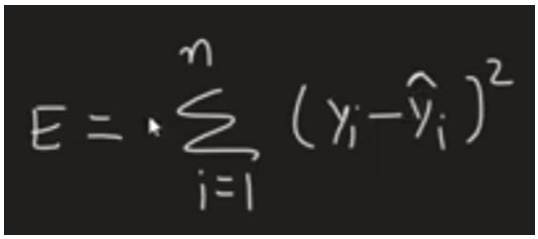
1

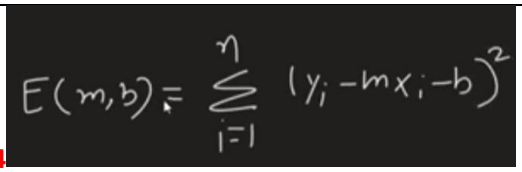


2

where d_i is distance **and y^ is the prediction output** and y_i is the actual output so the this d shows the difference with **error for a particular student** and total error is which add all errors. Total Error is for all student

3





4

Multiple Linear Regression

Here multiple input are required for this problem.

let x1,x2,x3 are input and independent data and y is output dependent

so multiple linear regression is just extension of simple linear regression and **in reverse order simple linear regression is just specialization of multiple linear regression**

let takes example for 100 students and we have cgpa, iq, gender, lpa

Cgpa x1	Iq X2	Gender X3	Lpa Y
1			
2			
. .			
100			

Here data is 4D 3 input and 1 output

while y= mx+b is 2D or we can say y = $\beta 0+\beta 1X$ here $\beta 0$ is b and $\beta 1$ is m

Hyperplanes are represented by equations and can be used to classify data points based on their position

now here we find out the hyperplane of 4D data in the equation

y=$\beta 0+\beta 1X1+\beta 2X2+\beta 3X3$ *Suppose here we want predicted value means* $\hat{y}$ and now

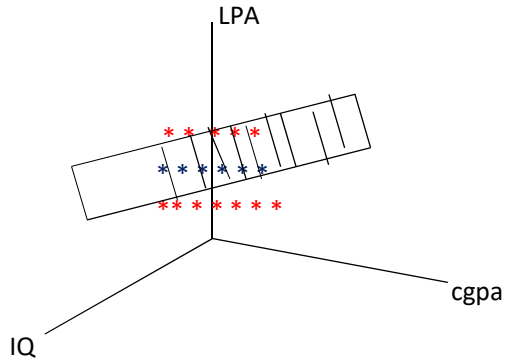
This involves more than one independent variable and one dependent variable. The equation for multiple linear regression is:
y=β0+β1X1+β2X2+……βnXn
\where:

- Y is the dependent variable
- X1, X2, …, Xn are the independent variables
- β0 is the intercept
- β1,β1 …, βn are the slopes

The goal of the algorithm is to find the best Fit Line equation that can predict the values based on the independent variables.

X1 : cgpa	X2 : iq	Y

here data is 3 D



in 2 D y= mx+b

here in 3D formula y= mx1+nx2+b

$y=\beta 0+\beta 1X1+\beta 2X2$ | $\beta 0, \beta 1, \beta 2$

here in 4D formula

$y=\beta 0+\beta 1X1+\beta 2X2+\beta 3X2$ | $\beta 0, \beta 1, \beta 2, \beta 3$

if Data Set n dimensional

$y=\beta 0+\beta 1X1+\beta 2X2+\beta 3X2\ldots\ldots\ldots+\beta nX2$

$$y = \beta 0 + \sum_{i=1}^{n} \beta i Xi$$

if n=1

$y=\beta 0 + \beta 1 x1$

so linear regression we try to find out coefficient value if we have 1 input column then we find out 2 coefficient value

and if we have 2 input column then we find out 3 coefficient value

if we have n input columns then we find out n+1 coefficient value

earlier we have 3D data set

cgpa –x1, iq -> x2, y ->lpa and suppose if you have value of $\beta 0$, $\beta 1$, $\beta 2$

$y=\beta 0+\beta 1X1+\beta 2X2$ here if you want to calculate lpa

$lpa = \beta 0 + \beta 1 * cgpa + \beta 2 * iq$

so if we can find out **lpa** for a new student then above formula we can evaluate it.

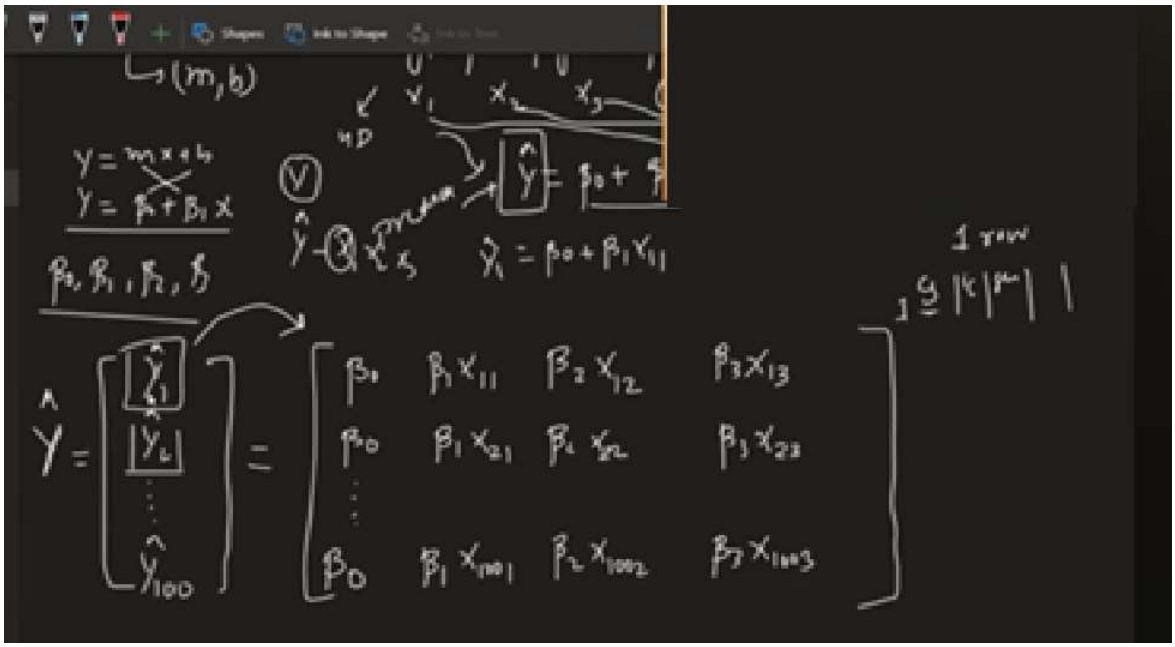
here coefficient represents like weight in the above formula $\boldsymbol{\beta 2}$ mentioned what is the weightage of **IQ** to calculate the LPA

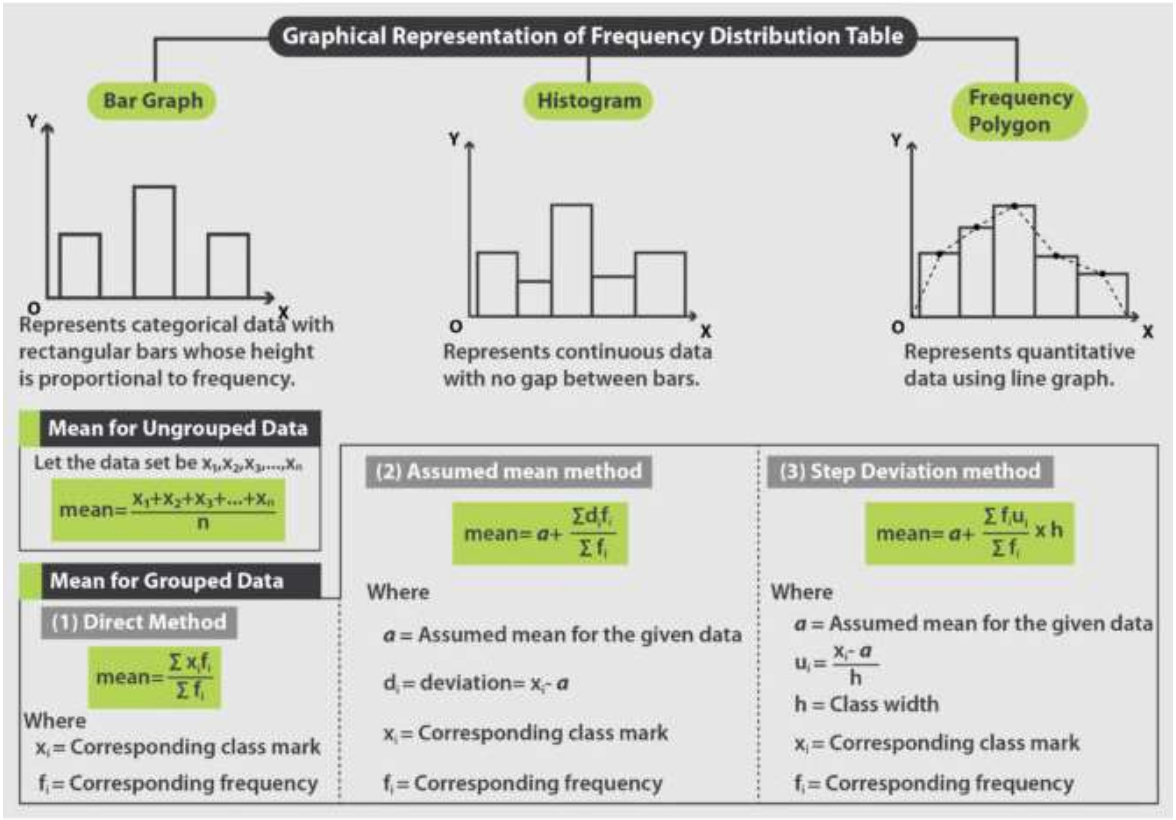
if $\beta 1 > \beta 2$ then we can imagine cgpa plays an important role as compared to iq if lpa is calculated

β0 is offset means intercept value and if cgpa =0 and iq = 0 then it is decided by intercept values to calculate the LPA

calculate 100 student predicted lpa $\hat{y}$

1 student	**Cgpa x1**	**IQ x2**	**Gender x3**


(m, b)
$y = mx + b$
$y = \beta_0 + \beta_1 x$
$\beta_0, \beta_1, \beta_2, \beta_3$
$\hat{y}_1 = \beta_0 + \beta_1 x_{11}$
1 row
$\hat{Y} = \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_{100} \end{bmatrix} = \begin{bmatrix} \beta_0 & \beta_1 x_{11} & \beta_2 x_{12} & \beta_3 x_{13} \\ \beta_0 & \beta_1 x_{21} & \beta_2 x_{22} & \beta_3 x_{23} \\ \vdots & & & \\ \beta_0 & \beta_1 x_{1001} & \beta_2 x_{1002} & \beta_3 x_{1003} \end{bmatrix}$



Fuzzy Logic : Lets take example of Doctor and Patient

its super set of Boolean logic means complete true or false Yes/No

But in fuzzy logic defines uncertanity

Patients shares his problem and the doctor considers the technique of cure how does it match the exact disease

But in some case any symptom that matches 60% to 70% for any disease on that basis doctor prescribes medicine while 30% to 40% in uncertainity

Hence in Fuzzy logic relates the uncertain data

Neural Network

A, a, **a,** ***a*** all is A by the knowledge base which learn any method

So soft computing is replacing Neural Network

Evolutionary / Generic Computing - > Decision base computing carried out Who will win any match from the last inning cricket match

What is Soft Computing?

Soft computing is an approach where we compute solutions to the existing complex problems,

where output results are imprecise or fuzzy in nature, one of the most important features of soft computing is it should be adaptive so that any change in environment does not affect the present process. The following are the characteristics of soft computing.

- It does not require any mathematical modeling to solve any given problem
- It gives different solutions when we solve a problem of one input from time to time
- Uses some biologically inspired methodologies such as genetics, evolution, particles swarming, the human nervous system, etc.
- Adaptive in nature.

There are **three types of soft computing techniques** which include the following.

- Fuzzy Logic
- Artificial Neural Network
- Genetic algorithm

Example of Fuzzy Logic as comparing to Boolean Logic

The Fuzzy logic can be implemented in systems such as micro-controllers, workstation-based or large network-based systems for achieving the definite output. It can also be implemented in both hardware or software.

Fuzzy Logic

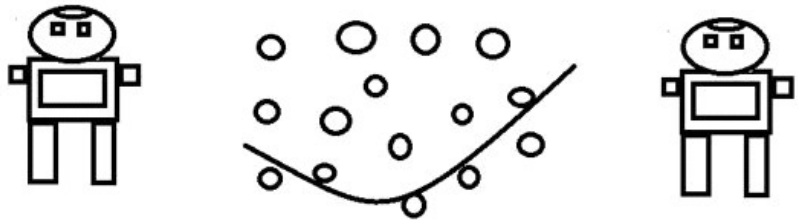
The fuzzy logic algorithm is used to solve the models which are based on logical reasoning like imprecise and vague. **It was introduced by Latzi** A. Zadeh in 1965. Who was a mathematician Fuzzy logic fuzzy logic provides stipulated truth value with the closed interval [0,1]. Where 0 = false value, 1= true value.

Fuzzy Logic is a super set of **Boolean Logic as we have a conventional method to represent it i.e, 0 and 1 means Yes or No, True or false**

But Fuzzy logic represent uncertainty means it works on the degree and Boolean logic will not work

Means we are representing degree the speed of

An example of a robot that wants to move from one place to another within a short time where there are many obstacles on the way. Now the question arises is that how the robot can calculate its movement to reach the destination point, without colliding to any obstacle. These types of problems have uncertainty problem which can be solved using fuzzy logic.



Destination Current location

Example 2

string1 = "xyz" and string2 = "view"

Problem 1

Are string1 and string2 same?

No, the solution is simply No. It does not require any algorithm to analyse this.

Let's modify the problem a bit.

Problem 2

How much string1 and string2 are same?

Solution

Through conventional programming, either the answer is Yes or No. But these strings might be 80% similar according to soft computing.

You have noticed that soft computing gave us the approximate solution.

	Fuzzy Set	Crisp Set
Basic	It is prescribed by vague or ambiguous properties.	It is defined by precise and specific characteristics.
Definition	It is a set of components with different membership degrees in the set.	It is a set of objects that have the same countability and finiteness qualities.
Applications	It is commonly utilized in fuzzy controllers.	It is commonly utilized in digital design.
Membership	It shows incomplete membership.	It shows the complete membership.
Logic	It follows the infinite-valued logic.	It follows the bi-valued logic.
Value	It specifies a number between 0 and 1, which includes both 0 and 1.	It specifies the value as either 0 or 1.
Degree	It defines the degree to which anything is true.	It is also referred to as a classical set.

Artificial Neural Network

Neural networks were developed in the 1950s, which helped soft computing to solve real-world problems, which a computer cannot do itself. We all know that a human brain can easily describe real-world conditions, but a computer cannot

This works on LLM (Large Level Model)

Some Basic Components of ANN

- Neurons (Nodes)
- Layers
- Connections

- Activation Function

Neurons

Neurons is like Human Brain what we think and gives the output Neuron Is Just Like Store the data

Data is stored and it will calculating the data now it is creating Layer when its processing the data

Layers Input data

Data Stores Hiiden layer data is breaking and associated

Layers are multiple Layers

Connection

The above picture is a Neuron Network

Activation Function

Output becomes sigmear data but we want that our value should come in None Linear Data between (0 & 1) then apply Activation function

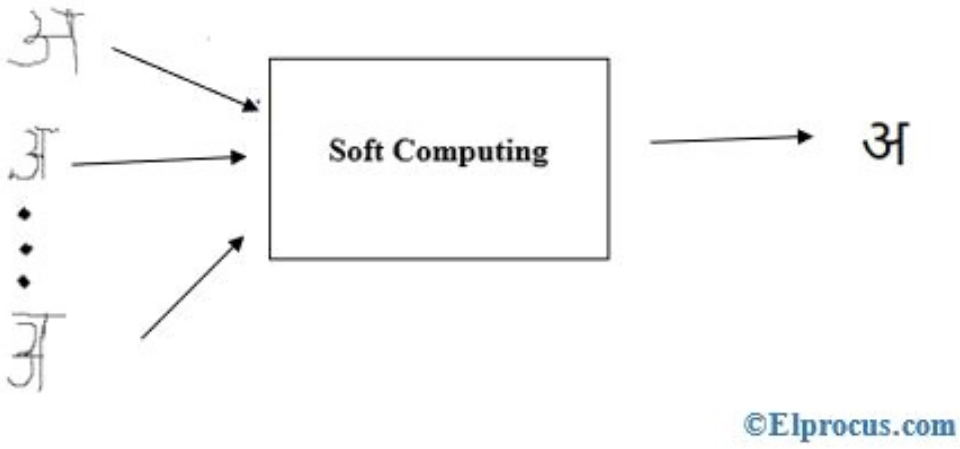
Over all the in above picture we are doing input as training data and it gives relevant data as we want because in every layer Data is relating as desired output

X1.w1 + X2.w2 + b (Biased Value)

- It is a connectionist modelling and parallel distributed network. There are of two types ANN (Artificial Neural Network) and BNN (Biological Neural Network). A neural network that processes a single element is known as a unit. The components of the unit are, input, weight, processing element, and output. It is similar to our human neural system. The main advantage is that they solve the problems in parallel, artificial neural networks use electrical signals to communicate. But the main disadvantage is that they are not fault-tolerant that is if anyone of artificial neurons gets damaged it will not function anymore.

Large Language Model (LLM) used Artificial Intelligence

An example of a handwritten character, where a character is written in Hindi by many people, they may write the same character but in a different form. As shown below, whichever way they write we can understand the character, because one already knows how the character looks like. This concept can be compared to our neural network system.



Evolutionary / Genetic Algorithm in Soft Computing

The genetic algorithm was introduced by Prof. John Holland in 1965. It is used to solve problems based on principles of natural selection, that come under evolutionary algorithm. **They are usually used for optimization problems like maximization and minimization of objective functions,** which are of two types of an ant colony and swarm particle. It follows biological processes like genetics and evolution.

This is Decision Based Computation Analysis

Who win the next IPL Match

CSK

KRK

RCB

Some characteristics of Soft computing

- Soft computing provides an approximate but precise solution for real-life problems.
- The algorithms of soft computing are adaptive, so the current process is not affected by any kind of change in the environment.
- The concept of soft computing is based on learning from experimental data. It means that soft computing does not require any mathematical model to solve the problem.
- Soft computing helps users to solve real-world problems by providing approximate results that conventional and analytical models cannot solve.
- It is based on Fuzzy logic, genetic algorithms, machine learning, ANN, and expert systems.

Example

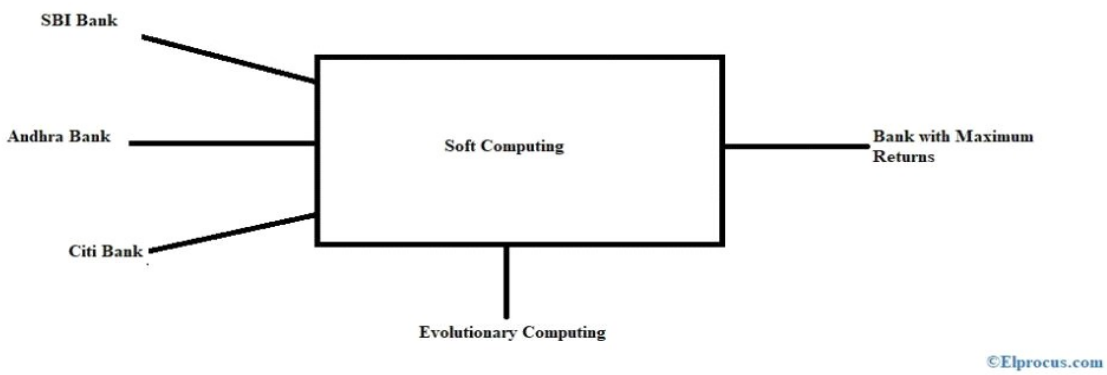
Soft computing deals with the approximation model. You will understand with the help of examples of how it deals with the approximation model.

Let's consider a problem that actually does not have any solution via traditional computing, but soft computing gives the approximate solution.

Functions of the Genetic Algorithm

The genetic algorithm can solve the problems which cannot be solved in real-time also known as the NP-Hard pLLMroblem. The complicated problems which cannot be solved mathematically can be easily solved by applying the genetic algorithm.

A simple way of understanding this algorithm is by considering the following example of a person who wants to invest some money in the bank, we know there are different banks available with different schemes and policies. Its individual interest how much amount to be invested in the bank, so that he can get maximum profit. There are certain criteria for the person that is, how he can invest and how can he get profited by investing in the bank. These criteria can be overcome by the “Evolutional Computing” algorithm like genetic computing.



Difference Between Hard Computing and Soft Computing

The difference between hard computing and soft computing are as follows

Hard Computing	Soft Computing
• The analytical model required by hard computing must be precisely represented	• It is based on uncertainty, partial truth tolerant of imprecision and approximation.
• Computation time is more	• Computation time is less
• It depends on binary logic, numerical systems, crisp software.	• Based on approximation and dispositional.
• Sequential computation	• Parallel computation
• Gives exact output	• Gives appropriate output
Examples: Traditional methods of computing using our personal computer.	• Example: Neural networks like Adaline, Madaline, ART networks, etc.

Advantages

The benefits of soft computing are

- The simple mathematical calculation is performed
- Good efficiency
- Applicable in real-time
- Based on human reasoning.

Disadvantages

The disadvantages of soft computing are

- It gives an approximate output value
- If a small error occurs the entire system stops working, to overcome its entire system must be corrected from the beginning, which is time taking process.

Applications

The following are the applications of soft computing

- Controls motors like induction motor, DC servo motor automatically
- Power plants can be controlled using an intelligent control system
- In image processing, the given input can be of any form, either image or video which be manipulated using soft computing to get an exact duplicate of the original image or video.
- In biomedical applications where it is closely related to biology and medicine, soft computing techniques can be used to solve biomedical problems like diagnosis, monitoring, treatment, and therapy.

Computing is a technique used to convert particular input using control action to the desired output. **There are two types of computing techniques hard computing and soft computing**. Here in our article, we are mainly focusing on soft computing, its techniques like fuzzy logic, artificial neural network, genetic algorithm, comparison between hard computing and soft computing, soft computing techniques, applications, and advantages. Here is the question “How are soft computing is applicable in the medical field?”

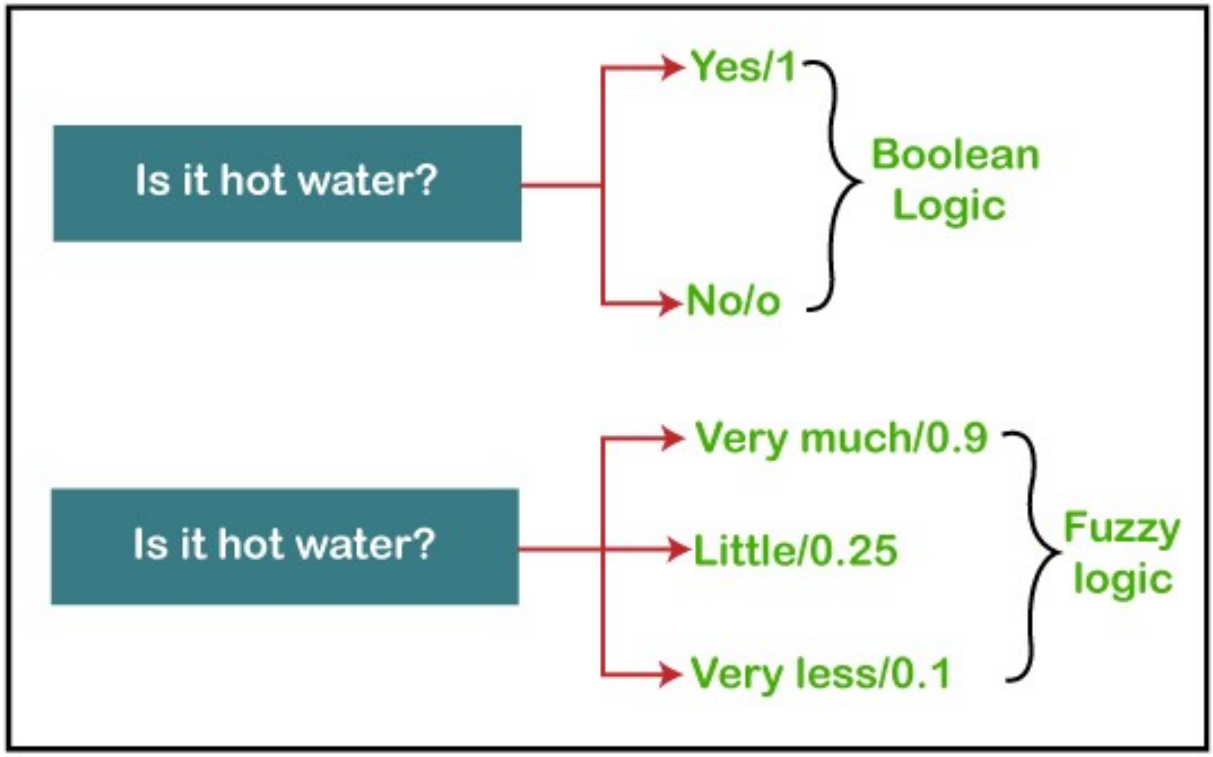
Evolution of Mathematic is Fuzzy Logic

There are **three types of soft computing techniques** which include the following.

- Fuzzy Logic
- Artificial Neural Network

- Genetic algorithm

Example of Fuzzy Logic as comparing to Boolean Logic



The Fuzzy logic can be implemented in systems such as micro-controllers, workstation-based or large network-based systems for achieving the definite output. It can also be implemented in both hardware or software

Artificial Neural Network

Neural networks were developed in the 1950s, which helped soft computing to solve real-world problems, which a computer cannot do itself. We all know that a human brain can easily describe real-world conditions, but a computer cannot

-

Here Neural Means Neuron and Network which is connected in them self

Like Internet which does sharing of Data

In Humans there are connected high number of neuron connected in each and other they think and Transfer the data in information and sharing of data

Neurons

Neurons is like Human Brain what we think and gives the output Neuron Is Just Like Store the data

Fuzzy Set

Universal Set and Our Set

Membership Belong

U (meun)

U = {1, 2, 3, 4, 5 }

S = {1, 2}

={(1,1) (2,1),(3,0),(4,0),(,5,0)}

x,u

Here 1 is with 1 which is membership values =1

Similiarly **Here 2 is with 1 which is membership values =1**

Here 3, 4 and 5 are with 0 which is membership values =0

Fuzzy set theory, introduced by Lotfi A. Zadeh, redefines the conventional notion of set theory by accommodating the granularity of membership within a set. Unlike classical set theory, which employs binary membership functions, fuzzy sets allow for a continuum of membership grades, thereby enabling the representation of partial truths.

Fuzzy Logic

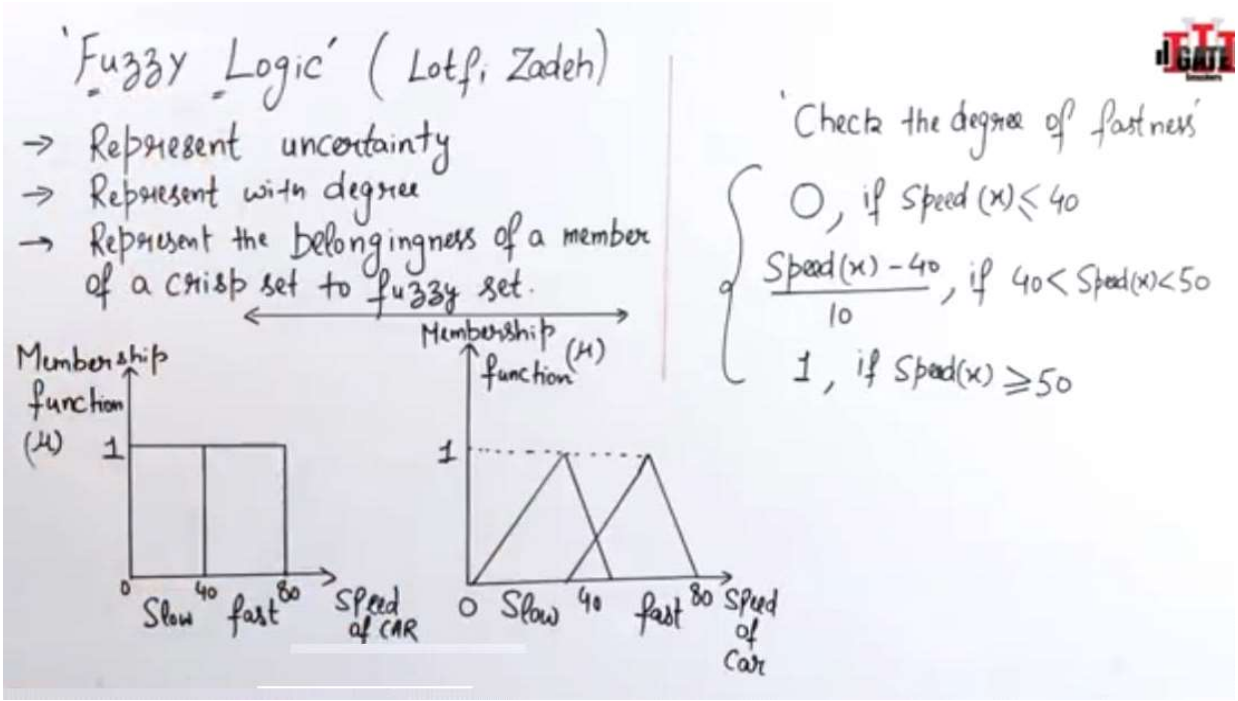
Fuzzy logic is a mathematical method for reasoning that's approximate and resembles human decision-making. It's based on the idea that real-world information is often vague and partially true, and that there are many possibilities between yes = 1 and no = 0

It is super set of Boolean logic means in Boolean Logic we represent [0 and 1]

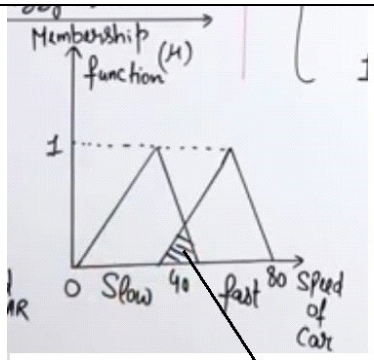
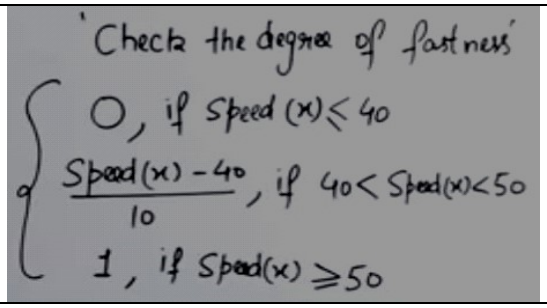
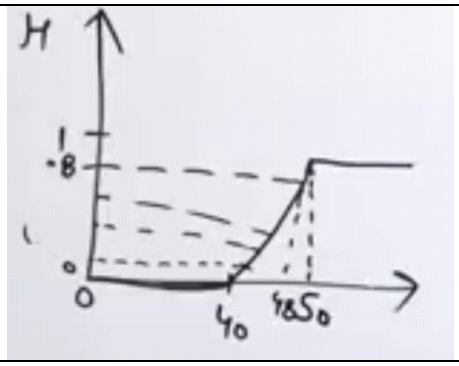
It represents uncertainty it represent the degree in membership function

Fuzzy set can have a progressive transition among many degrees of membership.

Means values will come between in 0 and 1 like 0.2, 0.3, 0.1



Speed Transition	Apply smart Agent	Result	Fuzzy logic
Slow : 30.4, 39.9 ->	**Acceleration**	**Oscillate**	Flexibility
Fast : 40, 40.1, 41	**Break**	**Oscillate**	Flexibility

In this pic we can represent flexible way via degree as given picture	 Represent flexible way in fuzzy logic
Boolean logic **to find out whether this sub set belong to U means S value belongs to Universal set U**	**Lets say we have universal set U = {1,2,3,4,5} and we have subset which value is S = {1,2}** **μ->membership** **{(1,1),(2,1),(3,0),(4,0),(5,0)}**
Check the degree of fastness of car through fuzzy logic Represents the belonging of a member function of a crisp set to fuzzy set	
0, if speed(x) <=40 **1, if speed(x) >=50** **(speed(x)-40)/10,** **if 40<speed(x)<50**	**x=30 (30,0)** **x=60 (60,1)** **42-40/10 =2/10=1/5 =0.2 up to degree** **48-40/10 = 8/10=4/5=0.8**
Membership function	

And we have to find out whether this sub set belong to U means S value belongs to Universal set U

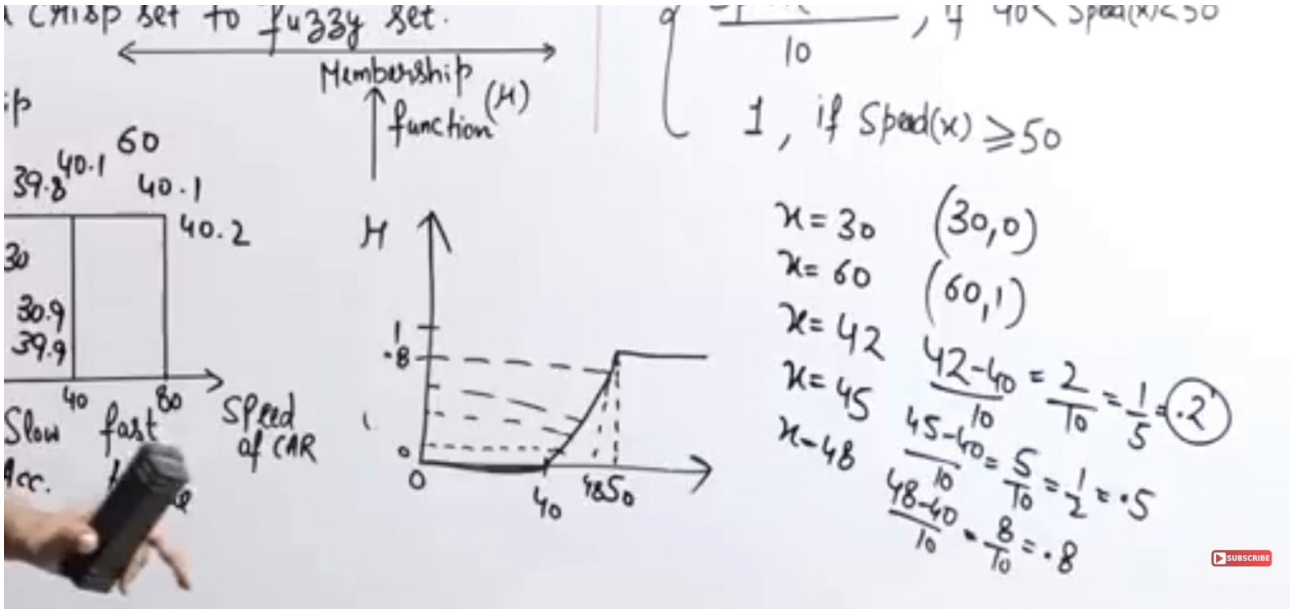
={1,1} 1 contains in U set Yes then 1 here U=1->x and 1 Yes 1 -> μ (meun) then its membership value is 1

Next ={(1,1), (2,1)}

Again check from Universal set

{(1,1),(2,1),(3,0),(4,0),(5,0)}

Hera membership values lies between 0 and 1 in Boolean but fuzzy shows the belongingness of a membership



It denotes Degree

Speed (x) -50/10 :

Fuzzy Set

A fuzzy set is a mathematical technique that uses a membership function to describe the degree of membership of an element in a set

Universal Set X ->x

The fuzzy set A defined on X is a collection of ordered pair

A = {(x, μ_A (x)}

Where μ_A (x): X-> [0,1] is called membership function

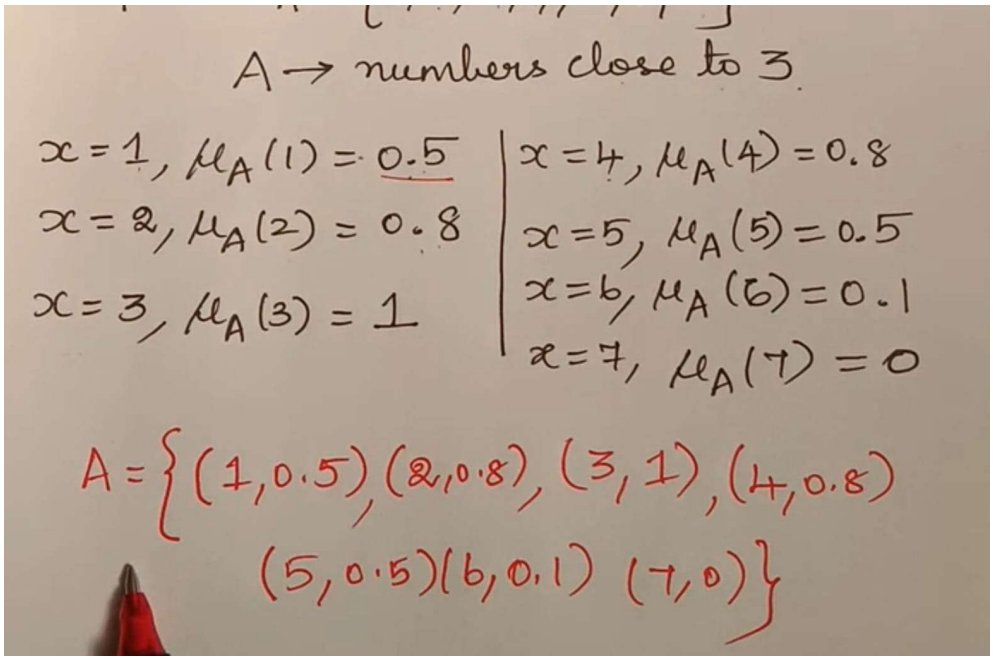
It represents of **degree of membership function** of element x with set A

which is value between 0 and 1

we consider Universal Set X and consider fuzzy set A i.e, closed to 3

X = {1,2,3,4,5,6,7}

A->number closed to 3



Given x to be a universe of discourse and A and B to be fuzzy sets $\mu_A(x)$ and $\mu_B(x)$ are their respective membership function

Union : Max $\{\mu_A(x),\mu_B(x)\}$, x Ɛ U

or operation in Boolean

x μ

A = {(10, 0.2), (20, 0.4), (25,0.7), (30,0.9), (40,1)}
B = {(10,0.4), (20, 0.1), (25,0.9),(30, 0.2), (40,0.6)}

A U B = {(10,0.4),(20,0.4), (25,0.9), (30,0.9), (40,1)}

Example in python :

```
key=[]
k=0
for i in range(5):
    k+=10
    key.append(k)
print("Set Value : ",key)


i=0
μc=[]
x=[0.8,0.3,0.4,0.6,0.2]
y=[0.5,0.7,0.9,0.1,0.3]
while i < len(x):
    if x[i] > y[i]:
        μc.append(x[i])
    elif y[i] > x[i]:
        μc.append(y[i])
    i+=1

print(μc)
bold=dict(zip(key,μc))
print(bold)
```

Output :
Set Value : [10, 20, 30, 40, 50]
[0.8, 0.7, 0.9, 0.6, 0.3]
{10: 0.8, 20: 0.7, 30: 0.9, 40: 0.6, 50: 0.3}

Intersection : Min $\{\mu_A(x),\mu_B(x)\}$, x Ɛ U

And operation in Boolean

A ∩ B = {(10,0.2), (20,0.1), (25,0.7), (30, 0.2), (40,0.6)}

some time we need to find out **μ** value ex: μ=x/(x+2)

Complement : $\mu_{\bar{A}}(x) = [1 - \mu(x)]$ x Ɛ U

Let A = {(10, 0.2), (20, 0.4), (25,0.7), (30,0.9), (40,1)}

Solution :

$\mu_{\bar{A}}(x)$ = (10, 0.8), (20,0.6), (25, 0.3), (30,0.1), (40, 0)

Example of complement in python

the uniform() method returns a random floating number between the two specified numbers.

```
import random
num=int(input("Enter the size of Element in set.."))
if num > 5:
    print("Enter size 5 ")
else:
    b=[]
    h=[]
    j=0
    for n in range(num):
        t=random.uniform(0,1)
        g=round(t,1)
        b.append(g)
print("membership : =",b)

if num > 5:
    print("Enter size 5")
else:
    c=[]
    k=0
    for m in range(num):
        k+=10
        c.append(k)
print("Set Value : ",c)
print("\n")
i=0
μA=list()
while i < len(c):
    t=1-b[i]
    s=round(t,2)
    μA.append(s)
    i+=1
print(μA)
comp=dict(zip(c,μA))
print("Fuzzy Complement ")
print(comp)
```

Output :

Enter the size of Element in set..4

membership : = [0.9, 0.8, 0.8, 0.2]
Set Value : [10, 20, 30, 40]

[0.1, 0.2, 0.2, 0.8]
Fuzzy Complement
{10: 0.1, 20: 0.2, 30: 0.2, 40: 0.8}

Bold Union : $\mu_{A \oplus B} = \text{Min}[1, \mu_A(x) + \mu_B(x)]$

Here first Add both A and B μ Value and arrange Minimum Value also check from 1 if the μ vaalue is greater than 1 then place 1 μ value [0 – 1]

ex : if μ=1 and 1.6 then min. value of μ=1

Min(1,0.6) -> 0.6, Min(1,0.5) -> 0.5, Min(1,1.6) -> 1

(10,0.6), (20,0.5), (25,1), (30,1), (40,1)

x μ
A = {(10, 0.2), (20, 0.4), (25,0.7), (30,0.9), (40,1)}
B = {(10,0.4), (20, 0.1), (25,0.9),(30, 0.2), (40,0.6)}

Bold Intersection : $\mu_{A \circ B}(x) = \text{Max}[0, \mu_A(x) + \mu_B(x) - 1]$

$\mu_{A \circ B}(x) = (0.2 + 0.4) - 1$

= (0.6) – 1 = – 0.4 compare to 0 which is greater than – 0.4 so it will be maximum value 0

= (10, 0)

= (20, 0.4) and (20, 0.1) again (0.4+0.1) – 1 = (0.5) – 1 = – 0.5

again – 0.5 it will compare to 0 then maximum is 0 i.e, (20,0)

again check (25,0.7) and (25,0.9) = (0.7+0.9) = (1.6) – 1 = 0.6

so 0.6 it will compare to 0 then maximum is 0.6 i.e, (25, 0.6)

= (30,0.9) and (30,0.2) again (0.9+0.2) = 1.1 – 1 = 0.1 and compare to 0 then maximum value is 0.1 (30,0.1)

= (40,1) and (40,0.6) again (1+0.6)=1.6 – 1 = 0.6 and compare to 0 then maximum value is 0.6 (40,0.6)

= {(10,0), (20,0), (25,0.6), (30,0.1), (40,0.6)}

Equality

A=B if $\mu_A(x) = \mu_B(x) \forall x \in S$

A = {(10, 0.2), (20, 0.4), (25,0.7), (30,0.9), (40,1)}
B = {(10,0.4), (20, 0.1), (25,0.9),(30, 0.2), (40,0.6)}

here A = (10, 0.2) $\mu_A(x) = 0.2$

here B = (10,0.4) $\mu_B(x) = 0.4$

Both are not equal

Product

$\mu_A(x) . \mu_B(x) = (10, 0.2) . (10, 0.4) = (10, 0.08)$

Ex : Dictionary :

```
key=["A", "B", "C", "D"]
value=[0.2,0.8,0.1,0.6]
print(dict(zip(key,value)))
```

Ex :

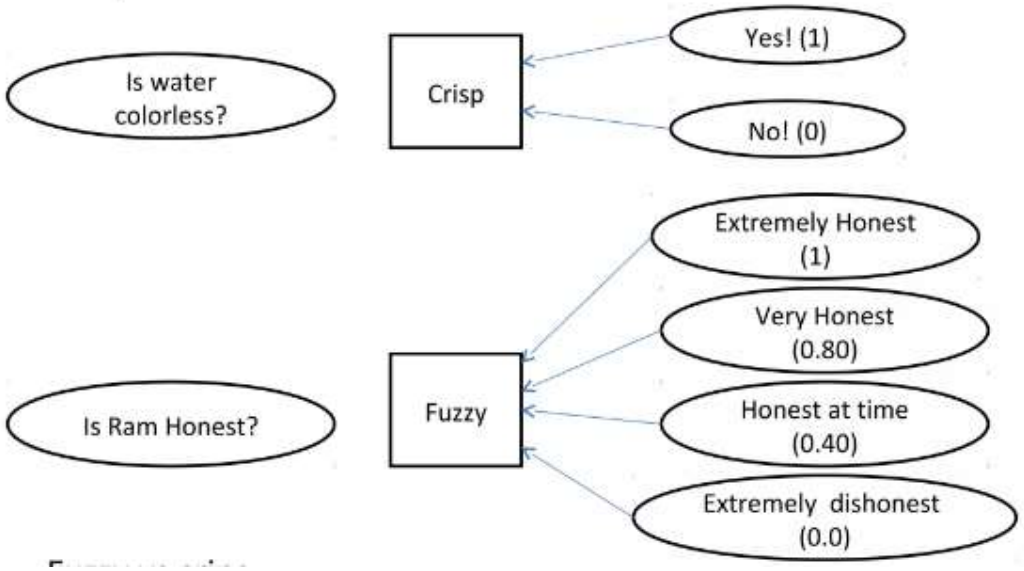
```
dicts = {}
keys = range(4)
values =[0.2,0.8,0.1,0.6]
for i in keys:
    dicts[i] = values[i]
print(dicts)
```

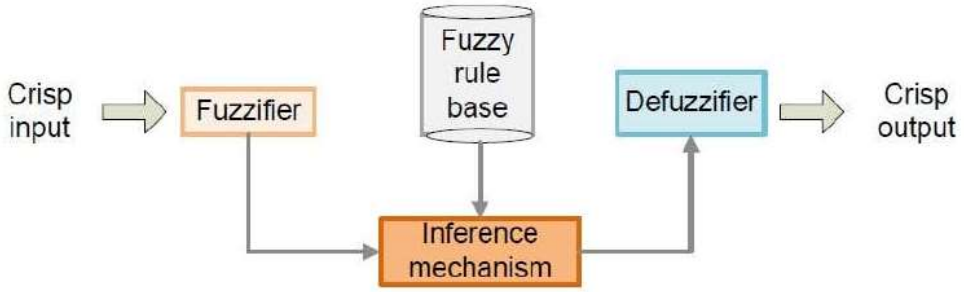
Output :

```
{0: 0.2, 1: 0.8, 2: 0.1, 3: 0.6}
```

Introduction (Continue)

Example





Crisp Set or Classical set

Classical set theory also termed as Crisp

Theory of crisp set had its root of Boolean logic

Crisp set is also called classical set

Crisp sets are classical sets defined in boolean logic

In crisp set known as function known as characteristic function

Crisp set defines the value is either 0 or 1.

Roster notation or Tabular Form

A = {a, e, i, o, u}

B = (1,3,5,7,9}

- The elements in roster form can be in any order (they don't need to be in ascending/descending order).
- The elements should not be repeated in set roster notation.

One of the limitations of roster notation is that we cannot represent a large number of data in roster form.

if we want to represent the first 100 or 200 natural number in a set B the limitation we can be overcome by representing data with the help of a dotted line. then it is necessary to represent large data i.e, odd number between 1 to 100 {1,3,5,7,……..,199}

Builder Notation

Such that' is expressed by the symbols '|' (a vertical bar separator) or ':' (colon).

the set {a,e,i,o,u} is written as A = {x | x is a vowel alphabet element }

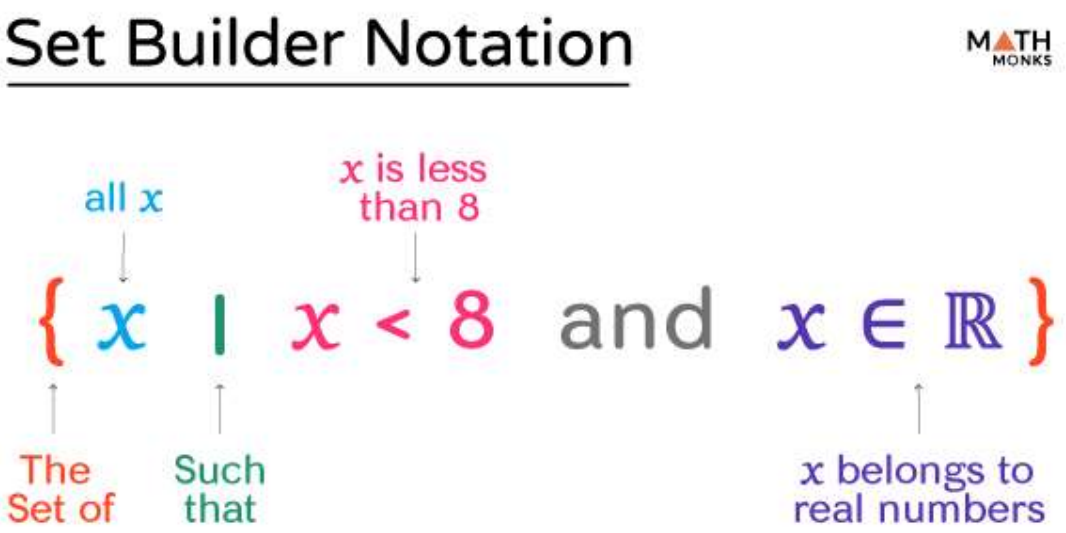
the set {1,3,5,7,9} is written as A = {x | x ∈ ℕ, 1 < x < 10, x is odd} and is read as "set A is the set of all 'x' such that 'x' is a odd number between 1 and 10."

The set {1,3,5,7,9} is writing as B = {x | x < 10 and (x%2 $\not=$ 0) }

A = {x | x ∈ ℕ, 5 < x < 10} and is read as "set A is the set of all 'x' such that 'x' is a natural number between 5 and 10."

Real Number :-> union of rational number and irrational number, these number can be expressed on a number line

A set of real numbers less than 8 is written in set builder notation as follows:



Types of Set

Singleton Set

A set which contains a single element is called a singleton set.

Example: There is only one apple in a basket of grapes.

Check whether the given sets are singleton sets or not?

S = {10}

Example : The given set is A = {1, 3, 5, 7, 11}

Therefore the five singleton sets which are subsets of the given set A is {1}, {3}, {5}, {7}, {11}.

Example :

Set Q = {|x|: $x^2 = 16$}

Disjoint Set

Disjoint

In order to find if two sets are disjoint sets, we need to perform the intersection of sets operation on these two sets. The condition for any given sets to be disjoint can be given as $A \cap B = \phi$

P = {1, 2}, Q = {2, 3} and R = {5, 3}.
P ∩ Q = {1, 2} ∩ {2, 3}

P ∩ Q = {1,2} ∩ {2,3}= {2}

And,

Q ∩ R = {2, 3} ∩ {5, 3} = {3}

And then the last pair,

P ∩ R = {1, 2} ∩ {5, 3}

P ∩ R = **ϕ**
Example in Python

```
p = {1, 2}
q = {2, 3}
r = {5, 3}

res = p.intersection(q, r)
print(res)
```

Output :
set()

Example :

```
p = {1, 2,1,4,6}
q = {2,3,4,6}
r = {5,4,3}
s = {4,9,12,16}
m = p.intersection(q)
print(m)
n = q.intersection(r)
print(n)
t = p.intersection(q,r,s)  // p ∩ q ∩ r ∩ s
print(t)
```

Output :
{4}

So, among the given sets, P and R can be considered as disjoint sets.
Disjoint sets:
Two sets A and B are said to be disjoint, if they do not have any common element in them, i.e. A ∩ B = { }. For sets

A = Set of even numbers = {2, 4, 6, 8} and

B = Set of odd numbers = {1, 3, 5, 7}

A ∩ B = { }, so here A and B are disjoint sets

Finite set

Finite sets can be easily represented in roster notation form. For example, the set of vowels in English alphabets, Set A = {a, e, i, o, u}

The finite set is countable and contains a finite number of elements

A set which contains a definite number of element called a finite set

Example – S = {x | 1< x < 10 and (x%2 ≠ 0)}

Here x is range of 1 to 10 and all odd no. 1,3,5,7,9

Infinite set

A set which contains infinite number of element is called an infinite set

The elements of infinite sets are endless, that is, infinite.

Example – S = {x | x ∈ N and x > 10}

Here x such that x is a integer and x is greater than 10 i.e, 11,12,13,……. infinite no.

Example : The set of whole numbers, W = {0, 1, 2, 3, ……..} is an infinite

Empty Set or Null Set

An empty set contains no element it is denoted by ϕ

Example – S = {x | x ∈ N and 7 < x < 8} = ϕ

Here x is a integer and x is greater than 7 and less than 8, so there is no number which is greater than 7, and less than 8 means it contains 0 no. element

so this is called empty set here

Subset

A set X is a subset of Y (written as X ⊆ Y) if every element of X is an element of set Y

Here X is subset of Y and Y is Superset of X

Example Y = {1,2,3,4,5,6} and X = {1,2} so we can write X ⊆ Y

Example A = {0,1,2} and B = {0,1,2} so here all element of A is in B

$A \subseteq B$ and we can also define $B \subseteq B$

suppose if we have C = {0,1,2,3} and B = {0,1,4,6}

$C \not\subseteq B$ Here **C is not subset of B**

Empty set is subset of every set and Every set is subset of itself

Ex: if A = { 1,2,3 } find all possible subset of set A

ϕ, {1}, {2}, {3}, {1,2},{1,3}, {2,3}, {1,2,3} Now Here total no. of Subset is 8

Here we can say if any set have nth element then the total no. of subset will be 2^n and

Subset are two types

Subsets are

- classified as

- Proper Subset
- Improper Subsets

Proper Subset Symbol
Let A and B are two sets, Set A is said Proper subset of set B if every element of set A is also element of set B but But $A \neq B$

A proper subset is denoted by $\subset$ and is read as 'is a proper subset of'. Using this symbol, we can express a proper subset for set A and set B as;

$$A \subset B$$

Set A is considered to be a proper subset of Set B if Set B contains at least one element that is not present in Set A.

Example: If set A has elements as {12, 24} and set B has elements as {12, 24, 36}, then set A is the proper subset of B because 36 is not present in the set A.

$$A \subset B$$

But $A \neq B$

Ex: : if A = { 1,2,3 }

So the subset is ϕ, {1}, {2}, {3}, {1,2},{1,3}, {2,3}, {1,2,3}

proper subset = ϕ, {1}, {2}, {3}, {1,2},{1,3}, {2,3} and Improper set is {1,2,3}

Improper Subset : Every set is a **subset of itself**, called an Improper subset.

: if A	{ 1,2,3 }
Total subset	ϕ, {1}, {2}, {3}, {1,2},{1,3}, {2,3}, {1,2,3}
Proper subset	ϕ, {1}, {2}, {3}, {1,2},{1,3}, {2,3}, remaining subset
Improper subset	{1,2,3} because again we're getting

Ex : A = ϕ here we don't find any subset so ϕ is Improper Subset

If a set has n element then total proper subset = 2^n - 1

All Subsets of a Set

The subsets of any set consists of all possible sets including its elements and the null set. Let us understand with the help of an example.

Example: Find all the subsets of set A = {1,2,3,4}

Solution: Given, A = {1,2,3,4}

Subsets =

{}

{1}, {2}, {3}, {4},

{1,2}, {1,3}, {1,4}, {2,3},{2,4}, {3,4},

{1,2,3}, {2,3,4}, {1,3,4}, {1,2,4}

{1,2,3,4},

Example: Find all the subsets of set X = {1,5,7,9}

Solution: Given, X = ϕ, {1} {5}, {7}, {9},

{1,5}, {1,7}, {1,9}, {5,7}, {5,9}, {7,9}

{1,5,7} {1,5,9} {1,7,9}, {5,7,9}

{1,5,7,9}

if A = {2,4,6}

Number of subsets: {2}, {4}, {6}, {2,4}, {4,6}, {2,6}, {2,4,6} and Φ or {}.

Proper Subsets: {}, {2}, {4}, {6}, {2,4}, {4,6}, {2,6}

Improper Subset: {2,4,6}

itertools.combinations():

The function itertools.combinations() is intended to provide every possible combination of the items in a given iterable

The itertools.combinations() function in Python takes an iterable as its first argument and an integer r as its second argument. The iterable can be any object like a list, tuple, string, or set.

Here is a simple program to showcase the different iterators that can be used.

the combinations() function basically returns an iterator. We can loop through the iterator to get our results. But, we can also convert it into a list and get all the combinations generated through the iterator in a list. Let us look at the example below to convert it into a list.

Example

```
import itertools


s = {1, 2, 3}
k = 2

# Find all subsets of size k

subsets = list(map(set,itertools.combinations(s, k)))
# Print the subsets
print(subsets)
```

Output:
[{1, 2}, {1, 3}, {2, 3}]

Example :

```
x = ['A', 'B', 'C', 'D']
result = list(map(set, itertools.combinations(x, 2)))
print(list(result))
```

Output :
[{'A', 'B'}, {'C', 'A'}, {'D', 'A'}, {'C', 'B'}, {'D', 'B'}, {'D', 'C'}]

Example :

```
import itertools

def findsubsets(s, n):
    return list(itertools.combinations(s, n))

s = {1, 2, 3}
n = 2

print(findsubsets(s, n))
```

Output:
[(1, 2), (1, 3), (2, 3)]

Example

```
from itertools import combinations
def getsubset(s):
    subset=[]
    for i in range(len(s)+1):
        for m in combinations(s,i):
            subset.append(m)
    return subset

set1=[1,2,3]
subs=getsubset(set1)
print(list(map(set,subs)))
```

Output:
[set(), {1}, {2}, {3}, {1, 2}, {1, 3}, {2, 3}, {1, 2, 3}]

Here nested loop are required :
print("\n")

```
for i in range(4):
    print("\n")
    for j in range(i+1):
        print(j,end=" ")
```

Output:

```
0
0 1
0 1 2
0 1 2 3
```

Improper Subset

A = B

Improper	Proper
⊆	⊂

A = B

Fuzzy Set

Union

Fuzzy Set

Fuzzy Logic is derived from fuzzy set theory

Many degree of membership (between 0 to 1) are allowed. Thus a membership function is associated with a fuzzy sets Ã such that the function maps every element of universe of discourse X to the interval [0,1].

The mapping is written as: (x): X → [0,1].

Fuzzy Logic is capable of handing inherently imprecise (vague or inexact or rough or inaccurate) concepts

Example

Fuzzy Sets (Continue)

• Let X = {g, g, g, g, g} be the reference set of students.

• Let A be the fuzzy set of "smart" students, where "smart" is fuzzy term.

A = {(g_1,0.4) (g_2,0.5) (g_3,1)(g_4,0.9)(g_5,0.8)}

Here Ã indicates that the smartness of g, is 0.4 and so on

Property of Fuzzy Set

Fuzzy sets are defined as sets that contain elements having varying degrees of membership values. Given A and B are two fuzzy sets, here are the main properties of those fuzzy sets

A = {1, 2, 3},

B = {2, 3, 4},

C = {5, 6}

Commutativity :- (A ∪ B) = (B ∪ A) (A ∩ B) = (B ∩ A)	 A ∪ B = {1, 2, 3, 4} → LHS B ∪ A = {1, 2, 3, 4} → RHS A ∩ B = {2, 3} → LHS B ∩ A = {2, 3} → RHS
Associativity :- • (A ∪ B) ∪ C = A ∪ (B ∪ C) • (A ∩ B) ∩ C = A ∩ (B ∩ C)	A ∪ B = {1, 2, 3, 4} (A ∪ B) ∪ C={1, 2, 3, 4, 5, 6} B ∪ C = {2, 3, 4, 5, 6} A ∪ (B ∪ C) = {1, 2, 3, 4, 5, 6} → RHS A ∩ B = {2, 3} (A ∩ B) ∩ C = ϕ → LHS B ∩ C = ϕ A ∩ (B ∩ C) = ϕ → RHS
Distributivity :- • A ∪ (B ∩ C) = (A ∪ B) ∩ (A ∪ C) • A ∩ (B ∪ C) = (A ∩ B) ∪ (A ∩ C) A = {1, 2, 3}, B = {2, 3, 4}, C = {5, 6}	A ∪ (B ∩ C) = (A ∪ B) ∩ (A ∪ C) B ∩ C = ϕ A ∪ (B ∩ C) = {1, 2, 3}→LHS A ∪ B = {1, 2, 3, 4} A ∪ C = {1, 2, 3, 5, 6} (A ∪ B) ∩ (A ∪ C) = {1, 2, 3} → RHS A ∩ (B ∪ C) = (A ∩ B) ∪ (A ∩ C) B ∪ C = {2, 3, 4, 5, 6} A ∩ (B ∪ C) = {2, 3} → LHS

	$A \cap B = \{2, 3\}$ $A \cap C = \phi$ $(A \cap B) \cup (A \cap C) = \{2, 3\} \rightarrow$ RHS
Idempotent :- $A \cup A = A$ $A \cap A = A$	Idempotency is defined as, $A \cup A = A$ $A \cap A = A$ For the given data, $A \cup A = \{1, 2, 3\} = A$
Identity :- $A \cup \Phi = A \Rightarrow A \cup X = X$ $A \cap \Phi = \Phi \Rightarrow A \cap X = A$	Mathematically, we can define this property as, $A \cup X = X$ $A \cap X = A$ $A \cup \phi = A$ $A \cap \phi = \phi$ For the given data, $A \cup X = \{1, 2, 3, 4, 5, 6\} = X$ $A \cap X = \{1, 2, 3\} = A$ $A \cup \phi = \{1, 2, 3\} = A$ $A \cap \phi = \{\ \} = \phi$
Transitivity :- If $A \subseteq B$, $B \subseteq C$, then $A \subseteq C$	
Involution :- $\mathbf{(A^c)^c = A}$	

μ = {1,2,3,4,5}

A={2,3}

$A^c =$ **μ - A** = {1,4,5}

L.H.S. $(A^c)^c$ => {1,2,3,4,5} – {1,4,5} = {2,3}

Example

μ = {a,b,c,d}

A={a,b}

A^c = μ - A = {a,b,c,d} - {a,b} = {c,d}

L.H.S. = **$(A^c)^c$ =** {a,b,c,d} – {c,d} = a,b

$(A^c)^c$ = A

Sets are used to store multiple items in a single variable.

Set is one of 4 built-in data types in Python used to store collections of data, the other 3 are List, Tuple, and Dictionary, all with different qualities and usage.

A set is a collection which is unordered, unchangeable*, and unindexed.

* Note: Set items are un changeable, but you can remove items and add new items.

Union

```
set1 = {"a", "b" , "c",5.2}
set2 = {1, 2, 3}
t=set1.union(set2)
print(set1.union(set2))
print(t)
```

Output :

```
{1, 2, 'a', 3, 5.2, 'b', 'c'}
```

Example

```
set1 = {"a", "b" , "c",5.2}
0set2 = {1, 2, 3}
set1.update(set2)
print(set1)
```

Output:

```
{1, 2, 3, 'b', 5.2, 'c', 'a'}
```

typecast into set

```
x=["ab",4,9,12,3.5,12,9.2]
m=set(x)
print(m)
```

output :

{3.5, 4, 9, 9.2, 12, 'ab'}

Subset

```
a={1,2,3,}
b={1,2,3,4}
print(a.issubset(b))
```

Output:

True

Intersection

```
a={3,6,0,7.6,"a",1,False,15}
b={1,3,6,9,12,15,0}
z=a.intersection(b)
print(z)
```

output :

{0, 1, 3, 6, 15}

Intersection using loop

```
l1 = ['one', 'two', 'three']
l2 = ['one', 'six', 'three']
l3 = ['one', 'four', 'five']
l4 = ['one', 'three','five']

check_list = list(set(l) for l in (l2, l3, l4))

print(check_list)
print("\n")
result = set(l1)
for s in check_list:
    result = result.intersection(s)
print(result)
```

output :

[{'three', 'one', 'six'}, {'four', 'one', 'five'}, {'three', 'five', 'one'}]

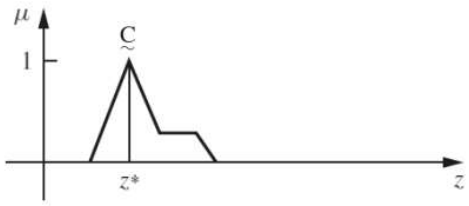
{'one'}

Fuzzy to Crisp Conversion

We have various method of defuzzification

1. Max – membership principal
2. Centroid method
3. Weighted average method
4. Mean – max membership
5. Center of sums
6. Center of Largest area
7. First of maxima, last of maxima

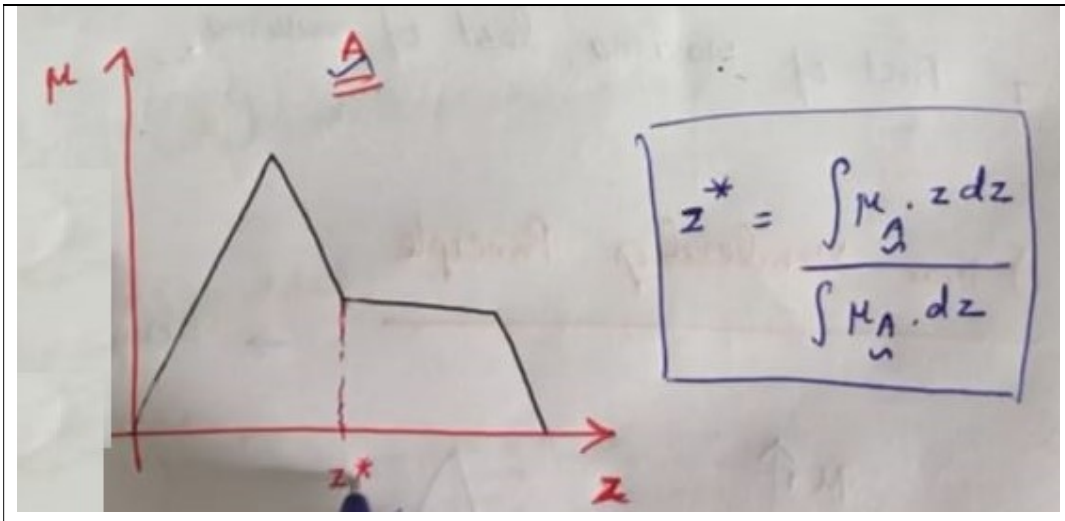
Max – membership principal
Also known as the height method, this scheme is limited to peaked output functions. This method is given by the algebraic expression where $z*$ is the defuzzified value. ∫

$\mu_{\tilde{C}}(z^*) \geq \mu_{\tilde{C}}(z)$, for all $z \in Z$, 	Here membership value μ given in highest peak on C point which is the highest position on 1 and highest value on z axis is z* so z* is the crisp value or defuzzified value by using maximum membership principal

Centroid method

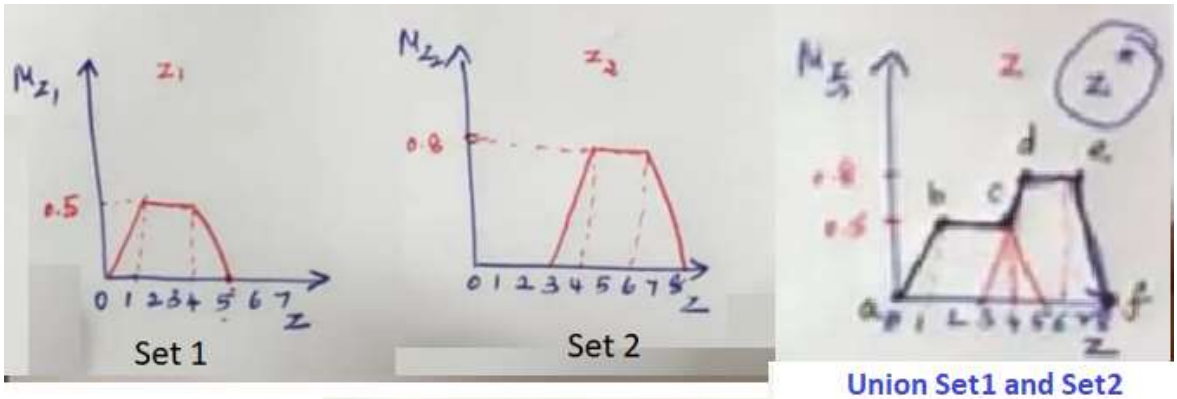
This method is also known as center of mass or center of area or center of gravity which needs to find out defuzzyfied value.

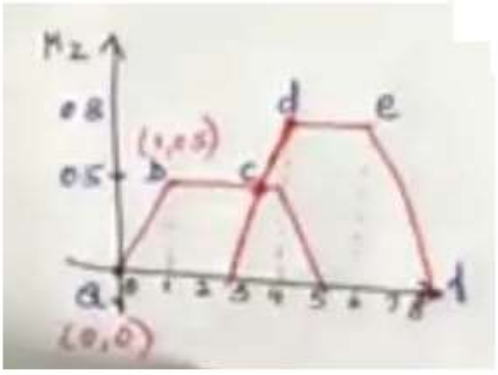
Let take an example by taking the union of two fuzzy set so in this diagram two axis y is μ membership value and x axis and defuzzified value received by centeroid two fuzzy set the centroid value in between two fuzzy set



Here centroid value is Z* which defuzzified value and here A~ is fuzzy set value	Gives the z point which is located at the center of gravity

Let take another example





Here Set1 and Set 2 is the union of two fuzzy set set1 and set 2 and use integration method and considering all point a,b,c,d,e so check coordinate value x,y value

for a,b the coordinate value is (1,0.5) from o to 1 check the x and y value

Weighted average method

The weighted average method is a defuzzification technique that converts a fuzzy quantity to a precise quantity. It's used to convert fuzzy sets to crisp values and is computationally efficient. The method is only valid for fuzzy sets with symmetrical output membership functions.

Here's how the weighted average method works:

Weight membership functions

Weight each membership function in the output by its maximum membership

Weighted average method: With this technique, the membership function has been weighted with its maximum membership value. The output x * can be calculated with the formula (2.72). Mean-max membership: This method can be recognized as the middle of the maxima.

Mean – max membership

The Mean of Maxima (MoM) defuzzification method is a technique that calculates the most plausible result by selecting the typical value of the most valid output linguistic term. It's often used in pattern recognition applications.

Here are some things to know about the MoM defuzzification method:

- **How it works**: The MoM method calculates the most plausible result by selecting the typical value of the most valid output linguistic term.
- **Output characteristics**: The MoM method results in stepped output characteristics.
- **Challenges:** The main challenge with the MoM method is generating a timetable that is practical, fairly, and good as possible.
  Defuzzification is the process of converting a fuzzy output to a single or crisp output value. It's also known as "rounding it off". Other defuzzification methods include: Centroid method, Weighted average method, and Max Criterion Method (MCM).

Centre of Sums Method:

This process involves the algebraic sum of individual output fuzzy sets instead of their union. Two drawbacks to this method are that the intersecting areas are added twice, and the method also involves finding the centroids of the individual membership functions

The defuzzified value x^* is defined as :

$$x^* = \frac{\sum_{i=1}^{N} x_i \cdot \sum_{k=1}^{n} \mu_{A_k}(x_i)}{\sum_{i=1}^{N} \sum_{k=1}^{n} \mu_{A_k}(x_i)} ,$$

Here, n is the number of fuzzy sets, N is the number of fuzzy variables, $\mu A_k (x_i)$ is the membership function for the k-th fuzzy set.

The defuzzified value x^* is defined as :

$$x^* = \frac{\sum_{i=1}^{k} A_i \times \bar{x_i}}{\sum_{i=1}^{k} A_i} ,$$

Here, A_i represents the firing area of i^{th} rules and k is the total number of rules fired and $\bar{x_i}$ represents the center of area.

The aggregated fuzzy set of two fuzzy sets C_1 and C_2 is shown in Figure 1. Let the area of this two fuzzy sets are A_1 and A_2.

A_1 = ½ * [(8-1) + (7-3)] * 0.5 = ½ * 11 * 0.5 = 55/20=2.75

A_2 = ½ * [(9-3) + (8-4)] * 0.3 = ½ * 10 * 0.3 = 3/2 =1.5

Now the center of area of the fuzzy set C_1 is let say $\bar{x_1}$ = (7+3)/2= 5 and

the center of area of the fuzzy set C_2 is $\bar{x_2}$= (8+4)/2=6.

Now the defuzzified value $x^* = \frac{(A_1.\bar{x_1} + A_2.\bar{x_2})}{A_1 + A_2} = \frac{(2.75 * 5 + 1.5 * 6)}{(2.75 + 1.5)}$ = 22.75/4.25 = 5.35

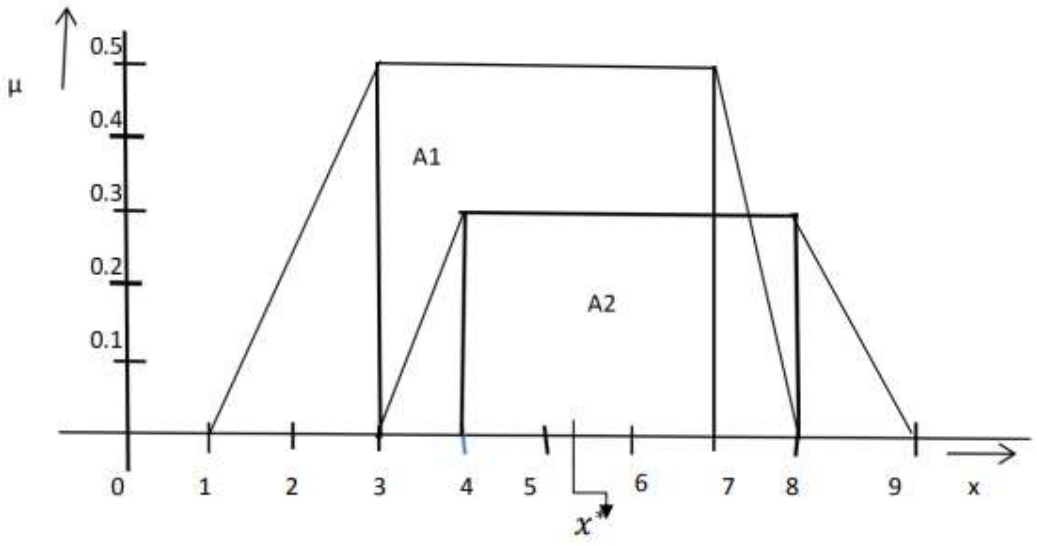


Figure 1 : Fuzzy sets C_1 and C_2

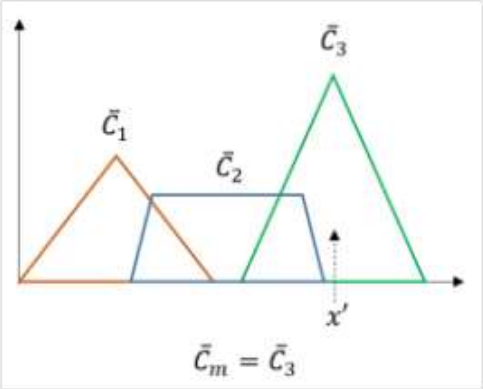
Center of Largest Area (CoA)

Center of Largest Area (CoA) method is simple, computationally effective and widely used defuzzification.
If the fuzzy set has two sub-regions, then the *center of gravity of the sub-region with the largest area* can be used to calculate the defuzzified value.

If the fuzzy set has two sub-regions, then the ***center of gravity of the sub-region with the largest area*** can be used to calculate the defuzzified value.

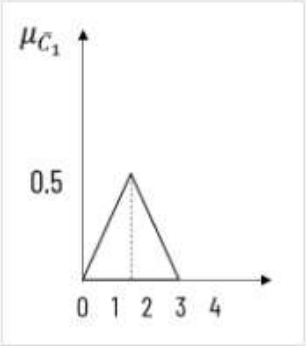
$$x^* = \left(\int \mu_{\underline{C}m}(x) \cdot x' \, dx\right) / \left(\int \mu_{\underline{C}m}(x) \, dx\right)$$

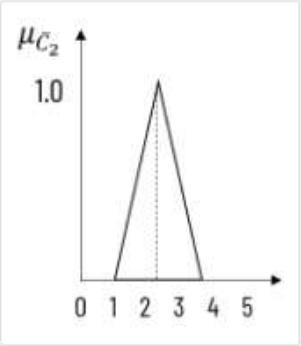
Here, $\underline{C}_m$ is the region with the largest area, x′ is the center of gravity of $\underline{C}_m$

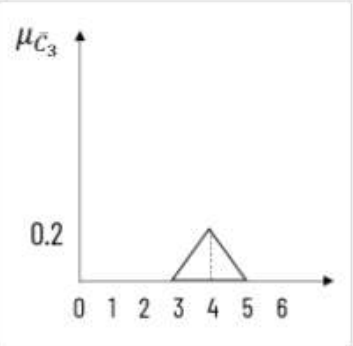


Example: Center of Largest Area (CoA) method

Find the crisp value corresponding to given fuzzy output functions



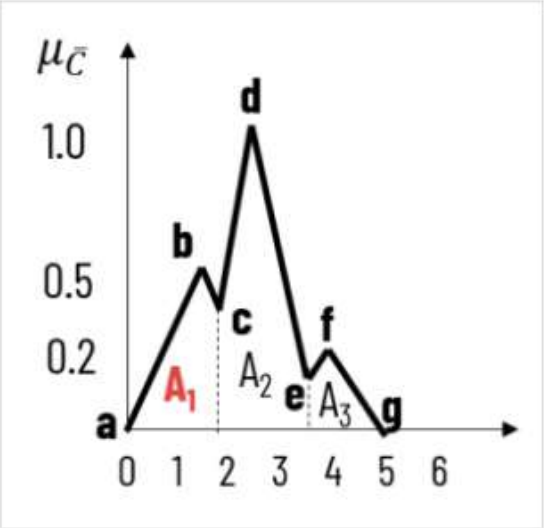




Solution:

To compute the crisp value for given fuzzy output sets, we shall place them on one axis and should find the area of nonoverlapping regions. The region with the highest value is used to compute the crisp value.

The aggregated fuzzy output functions are shown below. First, we will compute the area of region A1.



To find the area of region A_1, we need to find the line equations for the ab and bc lines

line ab:

$(y - y_1) / (x - x_1) = (y_2 - y_1) / (x_2 - x_1)$

for line ab, $(x_1, y_1) = (0, 0)$, and $(x_2, y_2) = (1.5, 0.5)$

$\Rightarrow (y - 0) / (x - 0) = (0.5 - 0) / (1.5 - 0)$

$\Rightarrow y = 0.33x$

The line ranges from 0 to 1.5 on the X axis.

line bc:

for line bc, $(x_1, y_1) = (1.5, 0.5)$, and $(x_2, y_2) = (x, 0)$.

We don't know what is the x coordinate of point c. We shall compute the equation of line cd also. The x coordinate of both the line would be the same at the intersection point

for line bc:

$(y - 0.5) / (x - 1.5) = (0 - 0.5) / (3 - 1.5)$

$\Rightarrow y = 0.33x + 1$

for line cd:

line cd:

$y = 0.67x - 0.67$

By comparing equations of line bc and line cd,

$0.33x + 1 = 0.67x - 0.67$

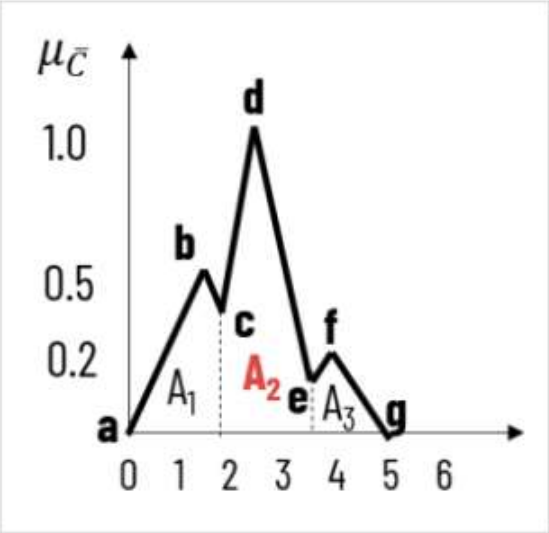
$x = 1.67 \approx 1.7$

so line bc ranges from 1.5 to 1.7 on the X axis.

The area enclosed by line ab and bc:

$$A_1 = \int_0^{1.5} 0.33x\,dx + \int_{1.5}^{1.7} (0.33x + 1)x\,dx$$

$A_1 = 0.466$



To find the area of region A_2, we need to find the line equations for the cd and de lines

line cd:

$(y - y_1) / (x - x_1) = (y_2 - y_1) / (x_2 - x_1)$

for line cd, $(x_1, y_1) = (1, 0)$, and $(x_2, y_2) = (2.5, 1)$

$\Rightarrow (y - 0) / (x - 1) = (1 - 0) / (2.5 - 1)$

$\Rightarrow y = 0.67x - 0.67 \quad [1.7, 2.5]$

The line ranges from 1.7 to 2.5 on the X axis.

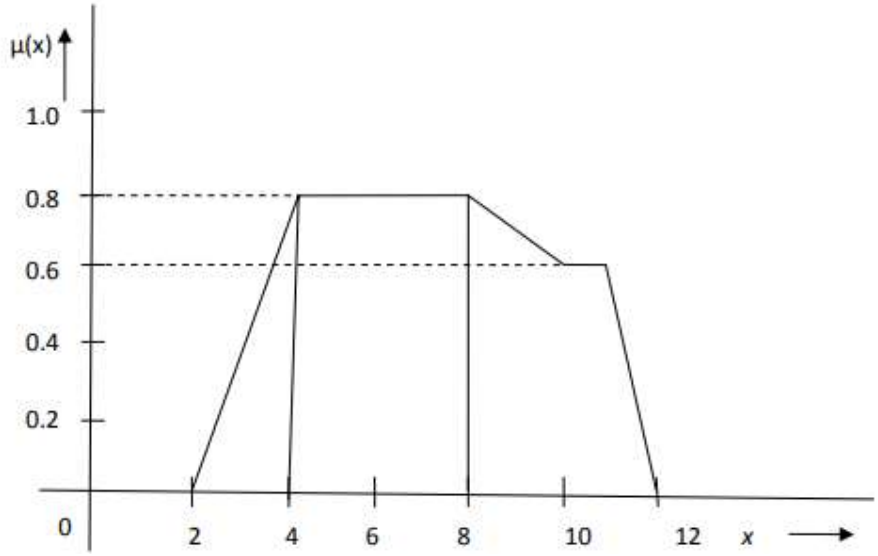
First of Maxima, Last of Maxima

First of Maxima Method (FOM)

This method determines the smallest value of the domain with maximum membership value.

This method determines the smallest value of the domain with maximum membership value.
Example:

The defuzzified value x^* of the given fuzzy set will be x^*=4.



Last of Maxima

- ### Last of Maxima Method (LOM)

  Determine the largest value of the domain with maximum membership va

  In the example given for FOM, the defuzzified value for LOM method will

- ### Mean of Maxima Method (MOM)

  In this method, the defuzzified value is taken as the element with the hig
  When there are more than one element having maximum membership v
  the maxima is taken.
  Let A be a fuzzy set with membership function $\mu_A(x)$ defined over $x \in X$,
  discourse. The defuzzified value is let say x^* of a fuzzy set and is defin

$$x^* = \frac{\sum_{x_i \in M} x_i}{|M|},$$

Here, $M = \{x_i | \ \mu_A(x_i)$ is equal to the height of the fuzzy set A$\}$ and $|M|$ is the cardinality of the set M.

Example

In the example as shown in Fig. , x = 4, 6, 8 have maximum membership values and hence |M| = 3

According to MOM method, $x^* = \frac{\sum_{x_i \in M} x_i}{|M|}$

Now the defuzzified value x^* will be $x^* = \frac{4+6+8}{3} = \frac{18}{3} = 6.$

Genetic Algorithm in Soft Computing

A genetic algorithm (GA), which is a subset of the larger class of evolutionary algorithms (EA), is a metaheuristic used in computer science and operations research that draws inspiration from the process of natural selection. Genetic algorithms frequently employ biologically inspired operators, including mutation, crossover, and selection, to produce high-quality solutions to optimization and search problems. Optimization of decision trees for improved performance, resolving sudoku puzzles, hyperparameter optimization, causal inference, etc., are a few examples of GA applications.

How the genetic algorithm works

Let’s consider an example that involves optimizing a common task: finding the best route to commute from home to work.

Imagine you want to optimize your daily commute and find the shortest route to go from your home to your workplace. You have multiple possible routes to choose from, each with different distances, traffic conditions, and travel times. You can use a GA to help you find the optimal route.

1. Encoding the solutions

In this case, potential solutions can be encoded as permutations of the cities or locations along the commute route. For example, you can represent each possible route as a string of city identifiers, such as “A-B-C-D-E-F,” where each letter represents a location (e.g., a street, intersection, or landmark).

2. Initialization

Start by creating an initial population of potential routes. You can randomly generate a set of routes or use existing routes as a starting point.

3. Evaluation

Evaluate each route in the population by considering factors such as distance, traffic conditions, travel time, and other relevant criteria. The evaluation function should quantify the quality of each route, where lower values indicate better solutions (e.g., shorter distance, less time spent in traffic).

4. Selection

Perform a selection process to choose which routes will be part of the next generation. Selection methods aim to favor fitter individuals, in this case, routes with lower evaluation values. Common selection techniques include tournament selection, roulette wheel selection, or rank-based selection.

5. Crossover

Apply crossover to create new routes by combining genetic material from two parent routes. For instance, you can select two parent routes and exchange segments of the routes to create two new offspring routes.

6. Mutation

Introduce random changes in the routes through mutation. This helps explore new possibilities and avoid getting stuck in local optima. A mutation operation could involve randomly swapping two cities in a route, inserting a new city, or randomly changing the order of a few cities.

7. New generation

The offspring generated through crossover and mutation and a few fittest individuals from the previous generation form the new population for the next iteration. This ensures that good solutions are preserved and carried forward.

8. Termination

The GA continues the selection, crossover, and mutation process for a fixed number of generations or until a termination criterion is met. Termination criteria can be a maximum number of iterations or reaching a satisfactory solution (e.g., a route with a predefined low evaluation value).

9. Final solution

Once the GA terminates, the best solution, typically the route with the lowest evaluation value, represents the optimal or near-optimal route for your daily commute.

By iteratively applying selection, crossover, and mutation, GAs help explore and evolve the population of routes, gradually converging toward the shortest and most efficient route for your daily commute.

It's important to note that GAs require appropriate parameter settings, such as population size, selection strategy, crossover and mutation rates, and termination criteria, to balance exploration and exploitation.

Methodology

Issues with optimization

In a genetic algorithm, a population of potential solutions to an optimization issue (people, creatures, organisms, or phenotypes) evolves toward superior solutions. Traditionally, solutions are represented in binary as strings of 0s and 1s, although other encodings are also feasible. Each candidate solution has a set of properties (its chromosomes or genotype) that can be changed and modified.

A generation is a term used to describe the population in each iteration of the evolution, which typically begins with a population of randomly generated individuals. Every member of the population has their fitness assessed once every generation; the fitness is typically the value of the objective function in the optimization issue being addressed. A new generation is created by stochastically selecting the fittest people from the current population, recombining their genomes, and introducing random mutations. The following algorithm iteration uses the fresh generation of candidate solutions. The algorithm typically ends when the population has reached a desirable fitness level or the maximum number of generations has been produced.

Genetic Operator

A Genetic Operator in computer science refers to a mechanism that selects and modifies individuals in a population based on their fitness, allowing them to reproduce and generate subsequent populations through processes like crossover and mutation.

A conventional genetic algorithm must represent the solution domain genetically and be evaluated using a fitness function.

Operators

Genetic variation is a necessity for the process of evolution. Genetic operators used in genetic algorithms are analogous to those in the natural world: survival of the fittest, or selection; reproduction (crossover, also called recombination); and mutation.

Introduction to Mutation

In simple terms, mutation may be defined as a small random tweak in the chromosome, to get a new solution. It is used to maintain and introduce diversity in the genetic population and is usually applied with a low probability – p_m. If the probability is very high, the GA gets reduced to a random search.

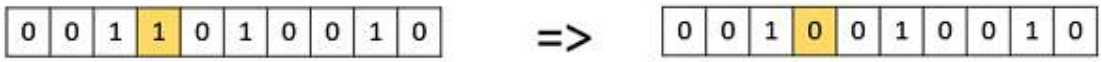
Mutation is the part of the GA which is related to the "exploration" of the search space. It has been observed that mutation is essential to the convergence of the GA while crossover is not.

Mutation Operators

In this section, we describe some of the most commonly used mutation operators. Like the crossover operators, this is not an exhaustive list and the GA designer might find a combination of these approaches or a problem-specific mutation operator more useful.

Bit Flip Mutation

In this bit flip mutation, we select one or more random bits and flip them. This is used for binary encoded GAs.

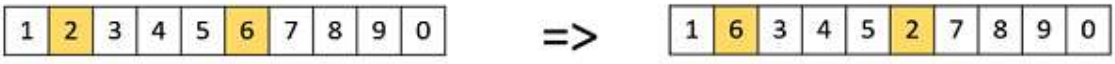


Random Resetting

Random Resetting is an extension of the bit flip for the integer representation. In this, a random value from the set of permissible values is assigned to a randomly chosen gene.

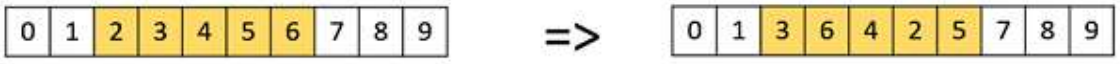
Swap Mutation

In swap mutation, we select two positions on the chromosome at random, and interchange the values. This is common in permutation based encodings.



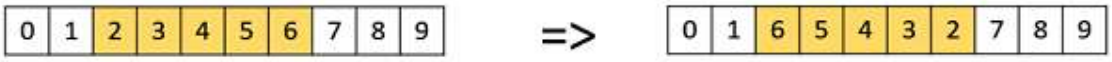
Scramble Mutation

Scramble mutation is also popular with permutation representations. In this, from the entire chromosome, a subset of genes is chosen and their values are scrambled or shuffled randomly.



Inversion Mutation

In inversion mutation, we select a subset of genes like in scramble mutation, but instead of shuffling the subset, we merely invert the entire string in the subset.



The Survivor Selection Policy determines which individuals are to be kicked out and which are to be kept in the next generation. It is crucial as it should ensure that the fitter individuals are not kicked out of the population, while at the same time diversity should be maintained in the population.

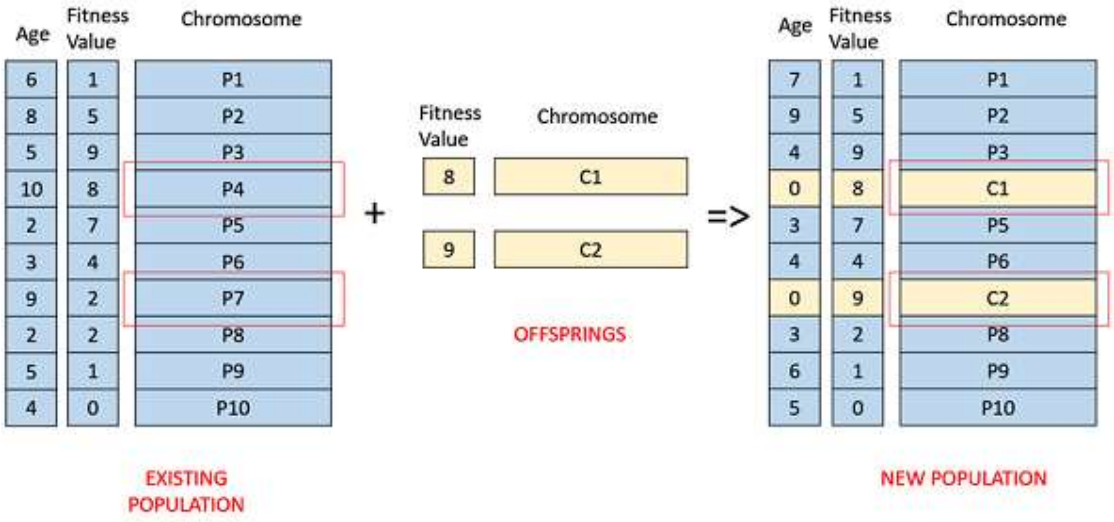
Some GAs employ **Elitism**. In simple terms, it means the current fittest member of the population is always propagated to the next generation. Therefore, under no circumstance can the fittest member of the current population be replaced.

The easiest policy is to kick random members out of the population, but such an approach frequently has convergence issues, therefore the following strategies are widely used.

Age Based Selection

In Age-Based Selection, we don't have a notion of a fitness. It is based on the premise that each individual is allowed in the population for a finite generation where it is allowed to reproduce, after that, it is kicked out of the population no matter how good its fitness is.

For instance, in the following example, the age is the number of generations for which the individual has been in the population. The oldest members of the population i.e. P4 and P7 are kicked out of the population and the ages of the rest of the members are incremented by one.


Age
Fitness Value
Chromosome
6 1 P1
8 5 P2
5 9 P3
10 8 P4
2 7 P5
3 4 P6
9 2 P7
2 2 P8
5 1 P9
4 0 P10
EXISTING POPULATION
+
Fitness Value
Chromosome
8 C1
9 C2
OFFSPRINGS
=>
Age
Fitness Value
Chromosome
7 1 P1
9 5 P2
4 9 P3
0 8 C1
3 7 P5
4 4 P6
0 9 C2
3 2 P8
6 1 P9
5 0 P10
NEW POPULATION