ARTIFICIAL INTELLIGENCE 06/05/19

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* ML and AI are not separate. ML is a subset/sub branch of AI.
* AI is mimicking of real nature.
* Any technique in this world which mimics human behavior, which has decision making capability/ability that is called an AI system.
* System that is trained is called intelligent system, making own decision making ability.
* Some of the use cases are:

a) INDUSTRIAL IOT (Industrial 4.0)

b) HEALTH CARE (Diagnosing diabetic retinotherapy)

c) SURVIALLENCE CAMERA

c.1) Finding culprit by image and sending to DB to police for immediate action).

c.2) Recognizing accident and intimating to nearest hospitals to send ambulance).

d) SMART COMPOSE (Included in Gmail to predict the next words)

Recurrent neural networks (RNN) for type chain analysis

(Smart suggestions are also provided to reply for a Mail.)

e) Image Recognition

* AI has number of branches :
* Machine learning
* deep learning
* predictive analytics
* Natural language processing (Uses CHATBOT): Processing and understanding human natural language is called as NLP.
* Translation
* Classification and clustering
* Information extraction
* Speech
* Speech to text
* Text to speech
* Expert systems ( To predict stock price)
* Planning , scheduling & optimization ( CRM- Customer Relationship Management
* Robotics( Sophia)
* Vision
* Image Recognition
* Machine vision

Research Goals of AI

**John McCarthy** who named the word AI.

* He wrote an article before his research in which it is given, can a machine be intelligence.
* In 1956, with huge size costs about 2500$ per month (rent).
* In 1980, size reduced cost is 3495$
* In 2019, 400gb of Hard Disk is 84$, size is very small.

AI requires two things:

* + Computational power
  + Relevant data

Deep learning is data hunger which requires huge data for training purpose.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Question: To process huge data, do we need huge computational power?

Answer: Yes.

**Big Data**: Due to increase of improvements of social networking (billion of active users). This data is stored in huge data centers. From there we got the word Big Data.

**Evolution:**

**AI: Any technique which enables computers to mimic human behavior.**

**ML: Subset of AI techniques which use statistical methods to enable machines to improve with experiences.**

**DL: It is a technique which makes computation of neural techniques or concept of neural networks/ multi layer neural networks feasible. It is the subset of ML.**

ML, Algorithms in ML & DL.

Basics of python programming language (Anaconda s/w)

Advance python (Python for AI)

ML (regression concepts, classifications of algorithms, deploying projects in IBM cloud)

ANN, CNN

Open Computer Vision (detecting human face, no .of topics…).

Q) What are different ways a human learn?

A) Experience, reading, listening, watching, practice, trial and error method, substitution method.

**LEARNING TYPES :**

* Supervised learning
* Unsupervised learning
* Reinforcement learning

**SUPERVISED LEARNING**: Whenever we have a labeled data (when we have i/p & o/p) to predict, this kind of learning type is called as supervised learning.

EG: House price prediction – Parameters are: Area, no. of stairs, age of house, infrastructure, locality.

Above parameters are inputs and the price is the output parameter.

NOTE: Technically *inputs* are also called as *independent variables*.

Technically *outputs* are called as *dependent variables*.

We have two kinds of data: a) Continuous (doesn’t have any fixed value)

b) Categorical (It has fixed value)

In supervised learning we have 2types: a) Regression

b) Classification

**Regression**: When the dependent variable data is of type continuous that type of problem type data is called regression.

Eg: Estimating of life expectancy of person, Weather forecasting, Population prediction.

**Classification**: When the dependent variable data is categorical then that type of problem type is classification.

Eg: Fraud detection, customer satisfaction prediction, prediction of cancer, Flowers species categorization.

**UNSUPERVISED LEARNING**: When we have input but no output to predict, i.e unlabeled data.

**Clustering**: Making similar category of data in to clusters.

Eg: supermarket arrangement, customer segmentation (Amazon, flipkart recommendations for a user depending on his search)

**REINFORCEMENT LEARNING:** Learning by rewards or mistakes.

(Reward based learning either be positive or negative)

Eg: Self driving cars.

AI in games: train the AI 1st and then slowly reward it with positive and negative rewards.

Package used for games: Open AI games.

QUEUE LEARNING used for self driving cars, etc.

**PYTHON**

Q) What is a programming language?

A) Language used to interact with computer.

Q) Why python if Java ruled the world for decades?

A) Because, python has easy syntax to learn and easy to code.

***GUIDO VAN ROSSUM*** Author of python programming.

**PYTHON:**

* Python is interpreted, object oriented, interactive, high level programming language.
* We will have interpreter rather than compilation. Compilation and execution will be in one step. We need not compile code, we can directly interpret the code.
* Object Oriented Programming: Alike java and C++, Python is also and high level and Object oriented language.
* It is also a beginner language, because it is easy to learn.

**SCOPE OF PYTHON:**

Python for web services

Django: Jango is a framework to develop *web application* using python

Flask: It is another framework.

KIVY is a framework to *develop mobile applications* using python programming language.

Scikit learn/SK learn is on a framework for ML.

Tensor flow is another framework.

NOTE: There are number of libraries to develop any number of applications.

We have python 2.x and 3.x version, we will be working on 3.7 version.

* INSTALLATION OF PYTHON.
* SPYDER IDE.

KEY POINTS:

* String to string concatenation is only possible.
* Whenever we use backslash we instruct the python to omit the functionality of that particular function.
* \n works as new line in print statement.
* The data type of a variable in python depends upon the value of the variable assigned.
* Type(variable) used to specify the variable type.
* Declaring variables means allocating memory and assigning values to it.
* Indexes in python starts from i.e index of H in HELLO is 0.
* C[0:2] specifies only 1st 2 letters and last character in 2nd position is omitted we will get only 0th position and 1st position character only.
* H e l l o

0 1 2 3 4

-5 -4 -3 -2 -1

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_DAY1\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

PYTHON 07/05/12

WORKING WITH:

Various data structure in python:

Functions (ways to define, python handlers, import statements….etc.);

**LIST:**

* List in python is denoted by square brackets.
* List is a group of elements.
* We need not declare data type to lists.
* L=[1,2,3,4]
* L [1:] 🡪 the element from 1 to everything in the length.
* L [2:3] 🡪 only 3rd element will be displayed.
* L [:2] 🡪 starting element to before element of the index.
* L [-1] 🡪 last element of the list.
* List has some in built methods.
* Lists can also have dissimilar elements, such as numbers and strings.

This is the main difference b/w array and list, list can have any type of elements but array should have fixed element type.

* We can replace element by assigning another value to particular index.

Eg: L[1]=3;

**Methods in lists:**

Append: Add object at end of the list.

Eg: L.append(Value to append)

Insert: Add object at any desired position of the list.

Eg: L.insert(index, data to be inserted)

Remove: Removes 1st occurrence of the value.

Eg: L.remove(Value to remove)

POP: Removes the value at that particular index.

Eg: L.pop(index value to be removed)

* If you do not mention the index value in pop, then last element in the list will be removed.

Eg: L.pop()

**Usage of lists**:

* Used in ML when we are working with data
* Prediction purpose, working with data sets.

TUPLE:

* Tuple is denoted by parenthesis or rounded brackets.
* Tuple is a group of similar or dissimilar elements.
* List are mutable i.e we can change the data by appending, removing, etc.. But tuple are immutable.

Eg: t[1]=5

Traceback (most recent call last):

File "<ipython-input-41-ba681835c344>", line 1, in <module>

t[1]=5

TypeError: 'tuple' object does not support item assignment.

* We can also pass duplicate elements in tuple.

Eg: t=(1,2,3,4,2)

* We can add 2tuples but we cannot change the data.

**Methods in tuples:**

Count: returns the number of occurrence of value.

Eg: t=(1,2,3)

t.count(1) 🡪 1

t.count(6) 🡪 0

**SET:**

* Set is denoted by curly brackets/ flower brackets.
* Set doesn’t support duplicate elements.
* Set is rarely used.

**Dictionaries:**

* Dictionary is also denoted by curly brackets but the data in the dictionary is in the form of key and value pair.
* The key and the value are separated by colon (:).
* Multiple keys and values are separated by comma (,).

Eg: d={"Name":"Sriram","Initial":"Velaga","Age":20}

* Keys are immutable in dictionary, but values can be changed.
* Tuple can also be a key in a dictionary. (Since, tuple is also immutable).
* Type(d) 🡪 dict
* D.keys() 🡪 returns all the keys in dictionary
* D.values() 🡪 returns all the values in dictionary.
* To modify data values: d['Key']="new value".

**CONDITIONS AND LOOPS IN PYTHON:**

***Indentation in python***: statements written with space (tab space) in if condition.

Eg:

a=10

if(a>5):

print("Sriram Velaga") ///Inside if condition

print(a) ///Inside if condition

print(a) //Out of if condition

print("Out of the loop") ///Out of if condition

**Using else condition:**

Eg:

a=10

if(a<5):

print("Sriram Velaga") ///In if conditon

print(a) ///In if condition

else:

print(a) ///Else condtion

print("Out of the loop") ///out of the statements

* *And* can be used when 2conditions need to be satisfied in if condition
* *Or* can be used when either of the condition can be satisfied in if condition.
* Alike else if () in C, we use *elif*(): in python.
* *Indentation error:* if we do not specify the tab space after any if or elif or else conditions for statements.

**While loop:**

* While(condition):

Stmts ///continuous iteration till while loop fails.

Stmts ///other statements executed after while loop terminated.

***U****se visualize python to see the line to line execution of the code.*

EG: while True:

Print(“True”)

Print(“Done”)

O/p: Infinite loop, to terminate use ctl+c , keyboard exception.

**For loop:**

Contains range

Range value contains 3 parameters initial, final, step size.

By default the step size is 1.

**Functions:**

Function in C programming: Block of organized and reusable code, which can be called any number of times.

**Def**: Def is a keyword in python to declare a function.

* We can pass arguments in function.
* We can pass multiple arguments too using comma (,).

EG: def add(a,b):

c=a+b;

return c

data=add(5,6)

O/P: No output

EG:

def add(a,b):

c=a+b;

print(c)

add(5,6)

O/P: 11

EG: def add(a,b):

c=a+b;

return c

add(5,6)

Out[92]: 11

***Another example related to Functions:***

***Position wise data entry:***

def form(name,age,city):

print("Name:", name)

print("Age:",age)

print("City",city)

O/P:

form("Sriram",21,"Hyderabad")

Name: Sriram

Age: 21

City Hyderabad

***Error in position arguments:***

def form(name,age,city):

print("Name:", name)

print("Age:",age)

print("City:",city)

O/P:

form("Sriram","Hyderabad",21)

Name: Sriram

Age: Hyderabad

City: 21

***Using reference arguments:***

def form(name,age,city):

print("Name:", name)

print("Age:",age)

print("City:",city)

O/P:

form(name="Sriram", city="Hyderabad", age=21)

Name: Sriram

Age: 21

City: Hyderabad

**EXPLORE:**

Variable length argument

Keyword variable length argument

**Import:**

Keyword for importing a package library in python is :

import package\_name

* as keyword is used for alias

Eg: import time as t

t.time()

* Without giving a reference in python is importing all the modules in that particular package.

Eg: from time import \*

* \* implies all (to import all the modules in that package)
* It reduces the programming time.
* All libraries may not be installed

To install such libraries in python we use pip tool.

PIP: Python installation package.

* For any library to be installed the command is :

Pip install library

* The installed data will be stored in site packages folder in anaconda/python. (The existing and installed folders will be there.)
* I we want to install specific version of library:

pip install library == version

**FILE OPERATIONS:**

* Open() is an in-built function in python to handle file operations
  + In this we pass 2arguments a) file name b)mode of opening.
  + Mode: accessing the file either to read, write, append, etc.
  + If code and file in same working directory we can give direct file path.
  + If code and file are not in same working directory then we need to specify the path.
* Read():

Eg:

file = open("sriram.txt","r")

print(file.read())

file.close()

* Write(): removes the entire data and writes new data.

Eg:

file = open("sriram.txt","w")

file.write("Killerking is the name that is more likely by sriram")

file.close()

w mode creates a file not existing, or if exists just write the data.

* Append(): append the data to the existing file.

Eg:

fr = open("sriram.txt","a")

fr.write("Sriram Velaga")

fr.close()

* Rename():

Eg: import os

Os.rename(“oldfilename”,”newfilename”)

* Remove():

Eg: import os

os.remove(“filename”)

**EXCEPTION HANDLING IN PYTHON**

🡪To avoid the errors we use exception handling.

🡪Exception handling in python is done with try and except

🡪Try (when there is no exception executed)

🡪Exception statement executes when ever exception occurs.

Eg:

try:

f=open("data.txt","r")

except:

print("Exception handled")

print("Sriram thope")

* Here, exception is handled and the next occurring print statement after exception is displayed.
* Except is a global for all kind of exceptions, executed if any mentioned exception is not the case of that particular code.
* If multiple errors are mentioned, the first occurred except statement will be displayed other except statements are simply ignored even though there are similar exceptions occur in the code.
* In try and except, else can also be used. It is executed when there is no error occurred.

**TYPE CONVERSION IN PYTHON:**

* Converting of data from one type to another type.
* Eg:

a=10

type(a)

O/P: int

b=str(a)

type(b)

O/P: str

**GLOBAL VARIABLE & LOCAL VARIABLE:**

x=10 ///global variable

def display():

x=20

print(x)

display() ///prints 20

print(x) ///prints 10

* Scope of the local variable is within the method or function only.

Q) Change global variable value in my display function.

A) By using this keyword. (Used in OOP concept)

We have a keyword called global.

Eg:

x=10

def display():

global x /// value is set into global

x=20

print(x)

display() ///prints 20

print(x) ///prints 20

* In order to modify the global variable data in local blocks, we will be using global keyword.

**OBJECT ORIENTED PROGRAMMING LANGUAGE:**

* Object: To refer something like tuples, lists,.. we use objects.

It is a physical representation.

Every object has a blue print, properties, structure.

That design of an object is class.

* Class: All the blue print, design, structure, properties are defined in class.
* The properties of class is represented in the form of attributes and methods.

Objects---------class--------attributes/methods.

* In OOP, attributes are nothing but variables, methods are nothing but functions.
* Class is a keyword to create a class.
* Technically object is an instance/physical representation of a class.
* Constructor: It is a method. The method name is same as the class name, object of class is created by calling a constructor.

In java constructor is created as follows:

A a=new A();

* There are 2 kinds of constructors:
  + Default constructor
  + Parameterized constructor
* The current object will be passed as a parameter automatically when we call a method in the class. (when the class has the object with which we are calling.)
* Self: It is a keyword to represent current object/instance of a class.
* \_\_init\_\_ is a special method in python class which automatically calls when object of a class is created.
* Init method is used to initialize attributes/variables.

EG:

class OnePlusOne:

def \_\_init\_\_(self):

print("This is init method")

def camera(self):

print("8MP")

def ram(self):

print("3GB")

def memory(self):

print()

opo=OnePlusOne()

opo.camera()

O/P:

This is init method

8MP

EG:

class OnePlusOne:

def \_\_init\_\_(self,memory):

print("This is init method")

self.mem=memory #self.mem is used to create mem as current object

def camera(self):

print("8MP")

def ram(self):

print("3GB")

def memory(self):

print("memory is "+self.mem)

opo=OnePlusOne("64GB")

opo.memory()

opo.camera()

O/P:

This is init method

memory is 64GB

8MP

**IMPORTANCE OF SELF KEYWORD:**

Eg:

class OnePlusOne:

def \_\_init\_\_(self,memory):

print("This is init method")

self.mem=memory #self.mem is used to create mem as current object

def camera(self):

print("8MP")

def ram(self):

print("3GB")

def memory(self):

print("memory is "+self.mem)

opo=OnePlusOne("64GB")

opo.camera()

opo.ram()

opo.memory()

opo1=OnePlusOne("128GB")

opo1.ram()

opo1.memory()

opo1.camera()

O/P:

This is init method

8MP

3GB

memory is 64GB

This is init method

3GB

memory is 128GB

8MP

**INHERITANCE:**

**🡪**When object is created to child class in the concept of inheritance, automatically the object is also created for the parent class.

**🡪** So by using inheritance, child class can access all objects of parent class.

08/05/19

* Child class can access the properties of parent class.
* Parent class cannot access properties from child class as child class is created after the formation or design of parent class. So, only child class can access methods and properties but parent class cannot.

**POLYMORPHISM:**

* Method overloading is the main topic that is mostly used in ML.
* Overriding means updating the existing data.
* If we don’t do overriding it takes the values of parent class only, so if we do overriding we can add- updated and new features to existing or inherited system.

**TAKING INPUT FROM THE USER:**

* Input function will return in the form of string, so later on we need to convert it to our desired data type.
* In python 2.x we have raw\_input but coming to python 3.x it has been removed.
* If we use ‘int’ in front of input, then the type will be converted to int.
* If we do not use ‘int’ in front of input, then the type will be taken as string (str).

Eg:

a = int(input(“Enter value”))

b = int(input(“Enter value”))

c=a+b;

print(c) //24

type(c) ///int

Eg:

a = input(“Enter value”)

b= input(“Enter value”)

c=a+b;

print(c) ///24

type(c) ///str

**PYTHON FOR AI**

**JUPYTER**

* Jupyter is an interactive based web application, the code will be in the form of extension .ipynb.
* i stands for interactive

py python

nb notebook

**PYTHON LIBRARIES FOR DATA SCIENCE:**

Many popular python toolboxes/libraries:

* NumPy
* Pandas
* ScKit-Learn

Visualization libraries

* Matplotlib
* Seaborn

**NumPy:** NumPy stands for numerical python; it is mostly used in working with arrays in ML. Especially with multi dimensional array.

* Used in almost all numeric computations.
* NumPy is in built in anaconda platform, if we use any normal python- we need to install it ------- pip install package\_name.
* In jupyter notebook if we want to install any package use:

!pip install package\_name

* There are some **in built methods** in numpy to create an array.
  + array()
  + arange()
  + linspace()
  + logspace() ///takes a log
  + zeros() ///we will get all zeroes, we need to mention size of array
  + ones() ///we will get all ones, we need to mention size of array
* **numpy.ndarray** stands for numpy n dimensional array.

Eg: import numpy as np

arr=np.array([1,2,3,12])

arr

O/P: array([ 1, 2, 3, 12])

type(arr)

O/P: numpy.ndarray

* **arr.ndim** specifies the dimension of arrary. In ML we work on 2D and 3D arrays with images.

**arange:** It takes 3arguments a) lower value b) upper value c) step size.

If step size is not mentioned it takes default value as 1.

It is equally spaced between elements.

**Linspace:**

Linspace also takes 3arguments.

They are : a) Lower value b) Upper value c) number of elements we need in that array.

The space between elements is equally spaced.

**SLICING:** Part of something.

**FLATTEN():** It is a method to convert multi dimension array into single dimension array.

**RESHAPE():** It is a method to convert any dimensional array into other dimensional array.

**DATAVISUALIZATION:** Data visualization is representing a data in the form of graphical format.

Data visualization helps us to understand more about the data or get more information/insight from the data.

**MATPLOTLIB:** It is a library in python used to visualize the data in the form of graphical representation.

* It is the father library of data visualization
* Matplot lib has 2 packages: a) pyplot b) pylab
* The main difference b/w these are: If we import pyplot we will get diff. functions where as if we import pylab, it imports both pyplot and numpy.
* Pylab has features of NumPy. So, we can use all the attributes in NumPy library.
* Mat -- matrix
* Plot -- plotting
* Lib -- library
* Legend method is used to tell the position of label.
* Scatter
* Bar
* Pie
* Some attributes: linewidth, color, label, alpha, start angle(pie)

**PANDAS:** It is a library, it is a data analytical tool i.e working with data sets we use this library.

In pandas we have 2 kinds of data structures: a) series b) dataframe

***Series***: It is nothing but 1D array i.e having 1 single dimension we call it as series.

* We have method called as series in pandas to represent series kind of data sets.

***Data frame***: If we have multiple no. of rows and no. of columns we call it as panda’s data frame.

🡪Pandas can also act as pre-processing library for ML

(Of course NumPy is also one more.)

Importing panda libraries:

* Import pandas

**MACHINE LEARNING 09/05/19**

**WHAT IS ML:** In ML we use models by giving inputs and output required. Model learns from the data like how humans learn and forms a model.

Model is a trained system.

Whereas traditional programming, conditional statements increase the length of program.

So we use ML for efficient and accurate data.

Data is required to train our ML system.

Features of ML:  
🡪 We need make computer program to teach themselves on data presented to it.

**The steps to build ML model is :**

Collecting data: kaggle.com, uci repository, data.gov.in, imagenet, analytics vidhya

1. Data Wrangling:

i) Discover whether the data is structured data or unstructured data.

ii) If data is unstructured, convert the data into structured format.

iii) If the data is structured, clean the data.

iv) Enrich the data i.e make your data suitable for algorithm.

1. Analyze the data: By analyzing the data, a data scientist can understand what kind of algorithm should be applied to our data.
2. Train your algorithm: Train ML model with the algorithm.
3. Test the ML model with new data.
4. Deployment.

**Data Preprocessing or Data Wrangling:**

1. Import the libraries
2. Import the data sets [csv: Comma separator value]

There is a method called as read\_csv() to read the csv files as pandas data frame.

It takes the path of our csv file as arguments.

Give ***r,*** before the path of the file if we get Unicode escape error.

1. Finding and handling missing values.

A data of a missing value cannot be applied to a ML algorithm , so we have to handle the missing values.

The different ways to

1. Filling the missing values with mean, median, mode.
2. Deleting the row where missing value is there.

NOTE: For numerical data we can fill the missing values with mean, median and mode also but for text data we can only fill the missing value with mode (Most frequently used).

🡪In Pandas frame work we have a method called as **fillna**, fills the Nan value with some approximate value (It may be either mean/mode/median).

🡪To search in which places we have missing value, we can find it using:

**df.isnull().any()**

🡪TO say particular column has null value:

df[“Column\_name”].isnull().any()

How to find the exact row number of missing value to remove the row from the data.

We have a special argument called INPLACE, which is used to update the value without assigning it to some variable.

**Inplace = True**, updates the data frame/variable.

**df.dropna** removes/drops/deletes the missing value of particular row.

We know the data what is missing, we can fill

1. The fourth step in our data wrangling is splitting dependent and independent variables as y and x respectively.

🡪The dependent and independent variables should be in the form of numpy array to apply to the machine learning algorithm.

So, if we add ***.values*** at the end of assigning we will get the data in numpy array.

1. Encoding the categorical text data.

🡪Sckit(Sk) learn developed especially for preprocessing and building the machine learning model.

Label encoder helps us to encode categorical text data.

Label encoder is in sklearn.preprocessing package.

There is a method called **Fit\_Transform** in LabelEncoder class to apply encoding to the data set.

🡪OneHot encoding

France – 0 – 00

Germany – 1 – 10

Spain – 2 – 01

FRANCE GERMANY SPAIN

1 0 0

0 0 1

0 1 0

0 1 0

1 0 0

🡪To apply oneHot encoding, we use oneHotEncoder class.

🡪We can only apply OneHotEncoding: when we have more than 2 categories, to avoid bias.

🡪We have fit\_transform() is a method used in OneHotEncoder to apply OneHotEncoding on data set.

🡪

🡪OneHotEncoder forms dummy variable.

1. Split Train and Test data

* We convert it into train and test because.
* The data is divided into train data and test data.
* Train data is used to train the model and test data is used to test the model.
* Train\_test\_split is in sklearn.cross\_validation or sklearn.model\_selection library.
* Train\_test\_split() constructor returns 4 values
* 80% data for training purpose and 20% is given for testing purpose

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_MACHINE LEANING\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Simple linear regression:**

It is the algorithm applying for data sets.

Y = a0 + a1 \* x

a0 is initial value.

a1 is slope of the line.

Our algorithm is used to find best a0 and a1 values.

🡪x is independent variable, y is dependent variable, a0 is constant and

a1 is coefficient.

🡪c = y-y^. c is also called as error cost function or loss.

C=(y-y^) 🡪 c=(y-y^)2 🡪 SUM(y-y^)2 [right shift or left shift the line] 🡪minimum error line. (is the best fit line).

NOTE:Initial line will be mean(x&y).

**STEPS:**

Step1: A line is drawn according to the mean(x&y) initially.

Step2: The error is found b/w the actual data points and the line data points, the error is also called as cost function or loss or error.

* C=y-y^

Step3: The sum of squares of the error is calculated for the line.

Step4: The line is either left shifted or right shifted [This line is left/right shifted based on gradient descent].

Step5: The line which has minimum error is considered as best fit line.

From the best fit line we can calculate the slope and constant values.

**10/05/19**

* Simple linear regression algorithm can be applied to a data set by using linear regression library in sklearn package.

from sklearn.linear\_model import LinearRegression

* By creating an object for linear regression class we are initializing the linear regression model.
* The object of LinearRegression itself is the Linear Regression model.

Before training our model we first initialize our model.

Using the object we need to train our model.

* fit() is a method used to train the ML model.
* In LinearRegression library there is a method called as predict() which is used to make the predictions.
* The predict() method returns the predicted values of the ML algorithm.
* Random\_state: It is a parameter. We use this to have same kind of splitting when using same data on different systems.
* Slope: Unit change in salary with respect to unit change in ………
* To find intercept: lr.intercept\_
* To find coefficient: lr.coef\_

**Multiple Linear Regression:**

Linear kind of algorithm with multiple independent variables.

Y = a0 + a1 \* x1 + a2 \* x2 + a3 \* x3 + ………. + ai \* xi

i = no. of independent variables

* After OneHotEncoding the encoded value moves to the initial position.

**Polynomial Linear Regression:**

* Formula to achieve curve in bent format are:

Y = a0 + a1 \* x1 + a2 \* x12 + ……..+ ai \* xi

* Previously we got best fit line, now we will get best fit curve.
* In polynomial regression the data is converted into polynomial features, regression is applied to polynomial features, then it becomes the multiple linear regression.
* A dataset can be converted into polynomial features by using polynomial features class in sklearn.preprocessing package.
* Fit\_transform() is the method in the polynomial features class to apply polynomial features to the data.
* Whenever the data set is converted into polynomial features we can achieve the curve.
* By applying the polynomial features with more degree, the curve is better

i.e more is the degree more better is the curve.

* When degree = 1, y = a0 + a1 \* x;
* When degree = 2, y = a0 + a1 \* x + a2 \* x2
* When degree = 3, y = a0 + a1 \* x + a2 \* x2 + a3 \* x3
* In polynomial regression, x(independent variables) is converted into polynomial features then linear regression is applied as the input is in the form of polynomial features the final end model will be the curve instead of line. 11/05/19

**CORRELATION:**

* Correlation is one of the statistical approaches to find a relation between any 2 variables.
* A relation means dependency.
* Correlation is dependency of one variable to other variable.
* Dependency: If change in one variable effects the other variable.
* Correlation is one of the parameter/factor/approach to say how much dependency is present between dependent variable and independent variable.
* The range of correlation is -1 to 1.
* We have 2 types of correlation:

a) Negative correlation

b) Positive correlation

* **Negative correlation**: When one variable is inversely proportional to another variable.
* **Positive correlation**: When one variable is directly proportional to another variable.
* Negative correlation produces negative values and range is -1 to 0.
* The range of positive correlation is 0 to 1.
* When correlation = 0 there is no correlation i.e no dependency between the variables.
* When correlation is either reaching 1 or -1 then it is positively highly correlated or negatively highly correlated respectively.

**Thresholds basically followed:**

* -0.2 to 0 (or) 0 to 0.2 It means that the correlation is very low.
* 0.2 to 0.6 moderate correlation.
* 0.6 to 1 very high correlation.
* If there is no correlation between one variable to another variable or less than 0.2 or 0.5, It is not necessary so such kind of independent variables can be removed without passing as parameters.
* If we remove the data that is not highly dependent: we can improve the model performance.
* In **pandas** data frame there is a method called as **corr** to find the correlation of different variables.
* HEATMAP is one of the visualization technique used to plot correlation between a dependent variable and another variable.

1. HeatMap is in seaborn library.
2. In seaborn library there is function called heatmap to plot the correlations.

**EVALUATING MODEL PERFORMANCE:**

* A performance measurement called as r2score
* Find Yavg = (Sum of all y variables)/(Total no. of y variables)
* Draw a line at Yavg.
* Calculate the sum of squares of errors between avg.line and actual data points.
* SSres= SUM(yi – yi^)2 && SStot = SUM(Yi - Yavg)2.
* R2 = 1- ( sum of squares of errors of best line, SSres )/(Sum of squares of errors of average line, SStot)
* The range of r2 is 0 to 1.
* Sum of squares of error of best fit line is inversely proportional to r2.
* Whenever R2 is 0 then it is worst model, if close to 1 then it is good model.
* If R2 > 0.6, the model is considered as good model, if it is >0.8 it is very good model.
* If it is < 0.5 it is worst/bad model.
* R2\_score is a library in sklearn.metrics package.

By eliminating/dropping the low correlated independent variable we can achieve good model performance or accuracy.

Dataset.drop([“Administration”], axis=1,inplace=True)

Axis = 0, means we are performing row operation.

Axis = 1, means we are performing column operation.

So, here we are removing the column administration and inplace = True is used to update the value in the dataset.

**LOGISTIC REGRESSION:**

* It is almost same as linear regression but it has some special function called as sigmoid function.
* P = 1/(1+e-y)
* The range of sigmoid function is 0 to 1
* Basically, sigmoid function is used to find probability.

Y =b0+b1 \* x 🡪 p= 1/(1+e-y) 🡪 ln(p/1-p) = b0+b1 \* x

(SIGMOID FUNCTION)

* Logistic regression is an algorithm mostly used for binary outcome or probability kind of outcome.
* When dependent variable is categorical we use logistic regression.

But categorical is type of classification, but we still call it as regression because the range is from 0 to 1, we can 0.5, 0.6,0.05 any value in between.

So after using threshold we are categorizing, till then it is under continuous, so it is named as regression though we use the algorithm for classification.

* After the generation of output, we apply threshold to categorize the binary outcomes.
* Logistic regression class is in sklearn.linear\_model.
* We use another metric called as accuracy\_score to calculate the percentage of correct predictions.
* It is in the sklearn.metrics package itself.

**Feature scaling:**

* Feature is nothing but our independent variables (Inputs).
* Scaling is nothing but obtaining the values in a range.
* Feature scaling is obtaining all the independent variables in a single range.
* By using standard scaling all the independent variable values ranges from -1 to1.

Why we need Scaling?

ANS:To avoid huge distance/difference between 2 data points we apply this.

* Feature scaling helps us to improve the model performance without any data loss by making distance between 2 data points minimum.

**Evaluating classification model performance:**

* Accuracy 🡪 score() 🡪percentage of correct prediction
* Confusion matrix
* ROC – Curve 🡪 (roc – receive operating characteristics).

Confusion matrix:

|  |  |
| --- | --- |
|  | Predicted class |
| Actual classes | |  |  |  | | --- | --- | --- | |  | Class=”No” | Class=”Yes” | | Class=”No” | True positive | False negative | | Class=”Yes” | False positive | True negative | |

**ROC Curve:**  It is a graph between true positive rate (TPR) & false positive rate (FPR).

* Area under the curve is AUC. Curve specifies how well the zeroes and ones are separated to each other.
* The range of AUC is between 0 and 1.
* Higher the AUC, good our model is.
* Degree of separability is our AUC.
* If AUC curve is near to 1, it is good model and if it is near to 0 then it’s a bad model.

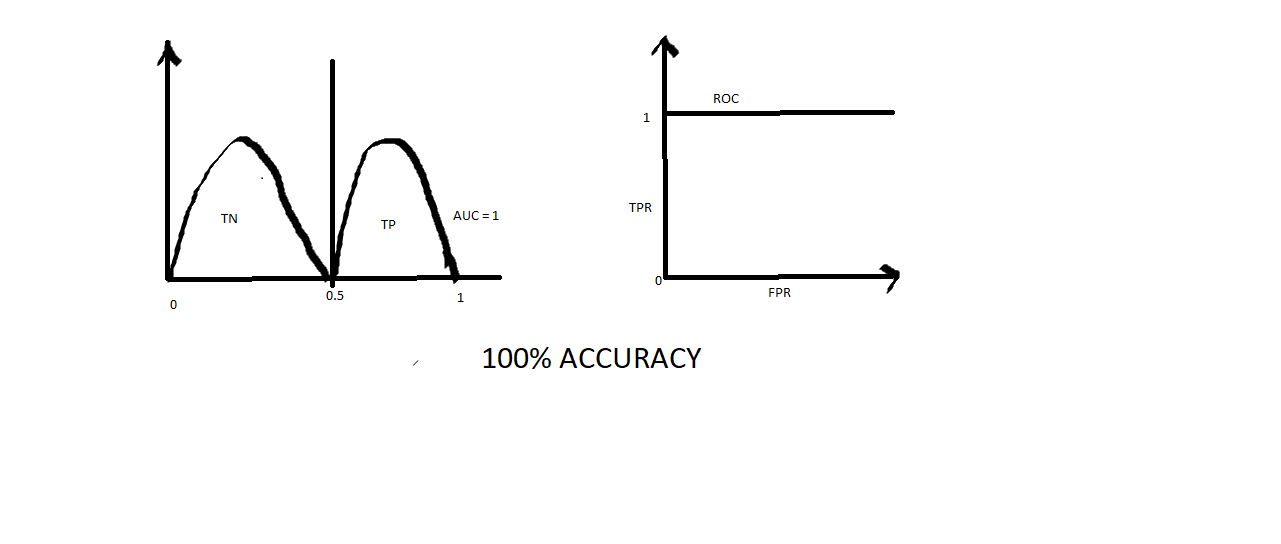
There are 3categories in this:

1. 100% accuracy
2. Wrong accuracy
3. Slightly overlapping
4. Overlapping (not recommended)

**100% accuracy**: When the two curves true positive and true negative exchanges their position there comes the 100% accuracy which leads to AUC=1.

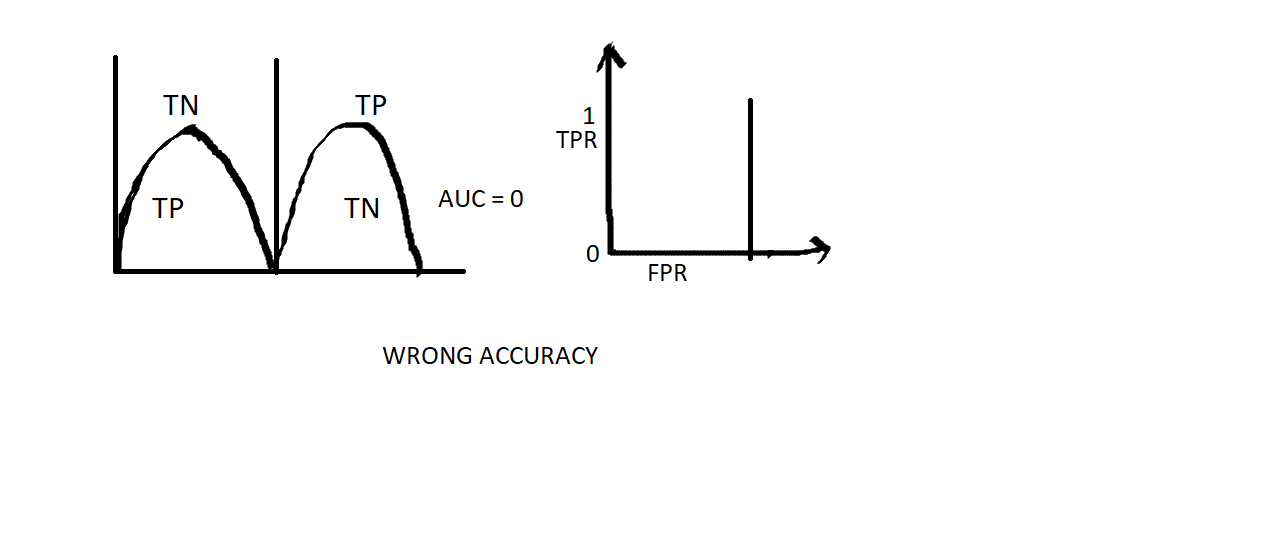
It can be obtained if both the curves true negative and true positive satisfies its position.

The curve placed between TPR and FPR will be horizontal straight line.

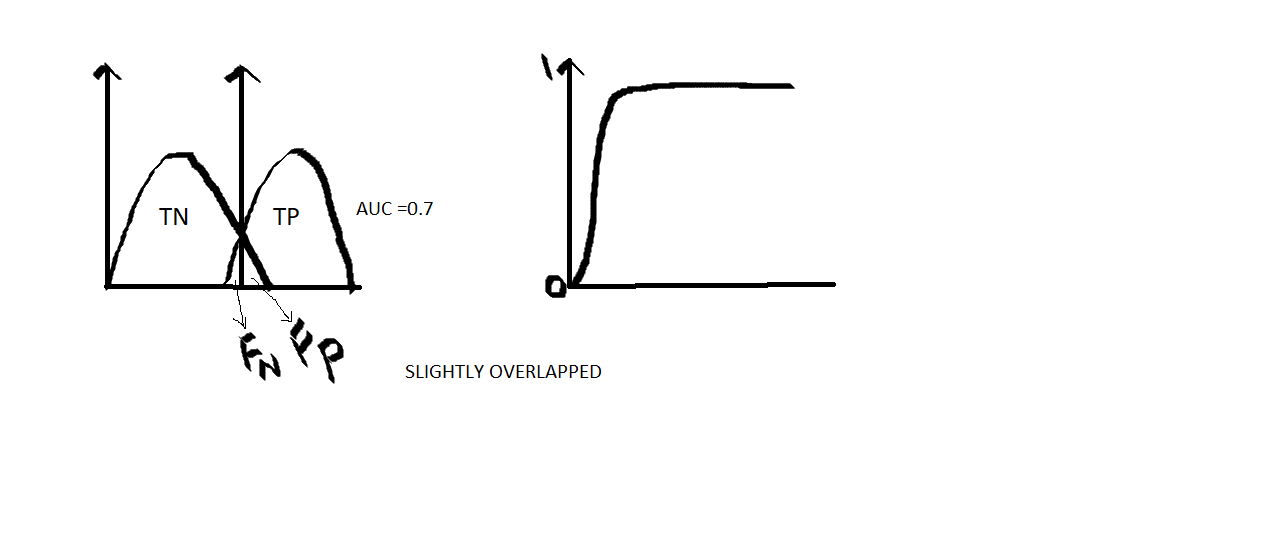


**Wrong Accuracy:**

* When the two curves true positive and true negative exchanges their position there comes the wrong accuracy which leads to AUC=0.
* Then the curve placed between TPR and FPR will be vertical straight line.

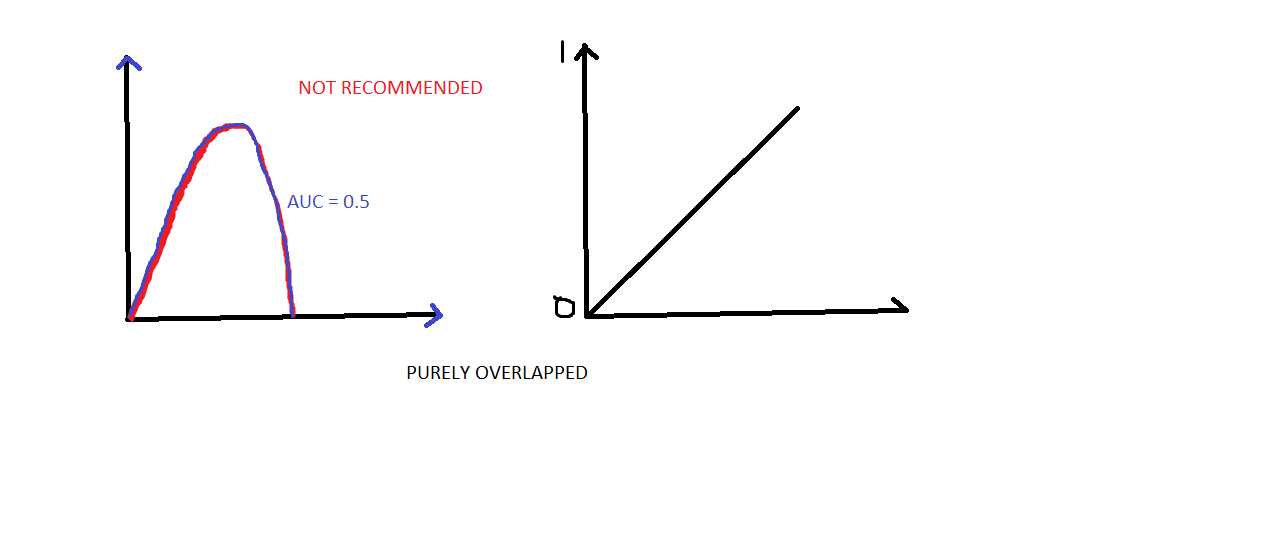


**Slightly Overlapping:**

Here TN and TP gets slightly merged at some point which leads the AUC value in between 0 & 1. 

**Overlapped:**

Here, the TN is purely overlapped with TP which is shown in the fig. Which leads to AUC = 0.5 which is not recommended.



TPR/Recall/Sensitivity = TP/TP+FN

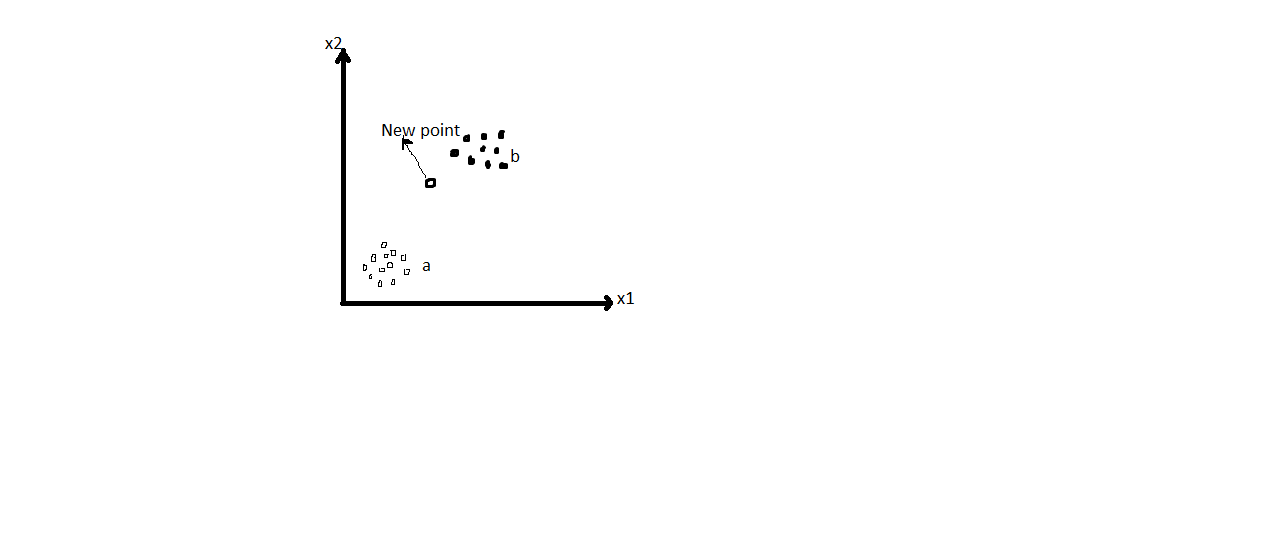
Specificity = TN/TN+FP and FPR = 1- Specificity

* The relation between TPR & FPR is directly proportional.
* If sensitivity increases specificity decreases.

13/05/19

**KNN ALGORITHM: KNN – K nearest neighbors.**

* In this thing, here x1, x2 are independent variables.
* Here if a new point occur, we should predict whether it belongs to category (a) or category (b).

**STEPS OF KNN:**🡪 In this, ‘K’ is value, which decides by the developer and it should be chosen as odd number. 

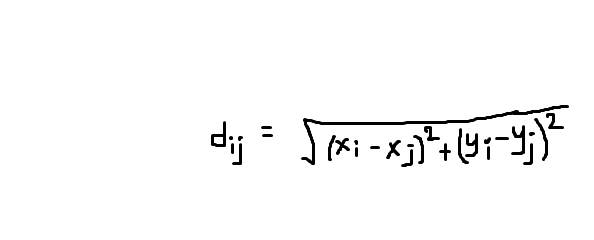
**STEPS FOR KNN:**

Step1: Choose the value of k (Mostly odd b=number).

Eg: k=5

Step2: Calculate the distance between the new data point to every other data points in our dataset.

Euclidean distance: This means distance between new point to other points.



Step3: As value of k=5 is assumed, we will calculate the distance between 5 nearest neighbors, i.e find the k nearest neighbors.

Step4: Based on the nearest neighbors, the new data point is classified based on the category which has highest number of neighbors.

NOTE:

* KNN is memory based algorithm.
* In KNN, testing time is greater than the training time.
* In KNN, during training time the data points are stored. In testing time, actual classification happens.

🡪From sklearn.neighbors import KNeighborsClassifier.

🡪knn=KNeighborsClassifier(n-neighbors=5,p=2)

Where p is euclidean distance.

**SVM algorithm: Support Vector Machine Algorithm**

* The main task of this algorithm is to find the decision boundary between categories.

NOTE:

The best decision boundary will be drawn based on two points (a, b).

If the distance between a to line and distance between b to line is nearer to equal then i.e best decision boundary.

The decision boundary is sum that is found out by using support vectors. The decision boundary is equidistant between support vector.

🡪from sklearn.sum import svc (SupportVectorClassification)

SV=SVC(kernel=”linear”)

NOTE:

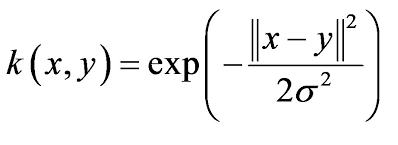
When data set is not linearly separable, then the data set is mapped to higher dimension by using mapping function.

Drawback:

Mapping to higher dimensional space can be highly compute-intensive.

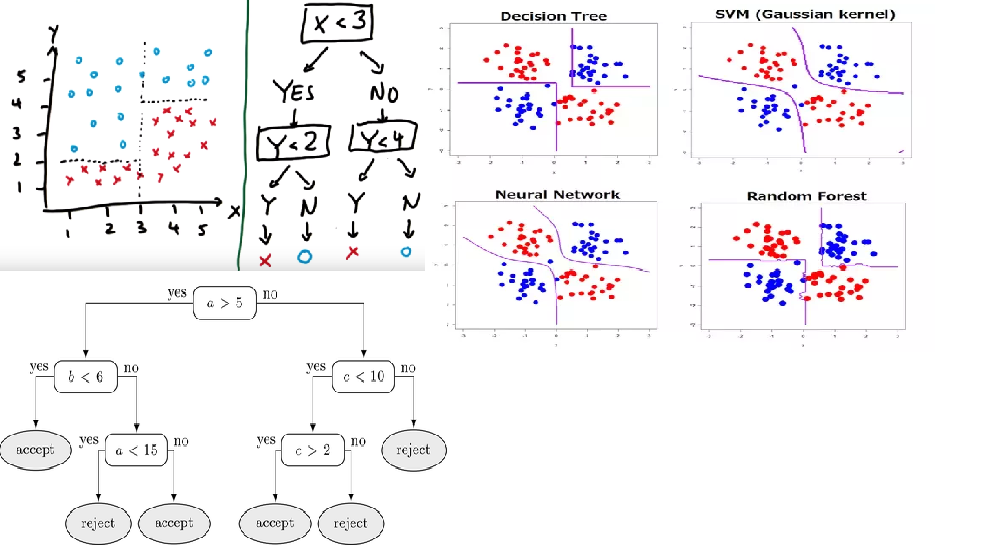
Gaussian RBF Kernel:

🡪It is a 3D kernel. When we apply this, instead of mapping dataset in 3D, our ML algorithm can imagine how they look like.



DECISION TREE ALGORITHM:

🡪CART means Classification And Regression Trees.



* Impurity = 0, when there are same categories.
* Impurity= 1, when there are different categories.

🡪Entropy is the metric used to measure the randomness of data (Impurity).

Entropy(s)= -p(yes/category-1)log2p(category-1)-p(no/category-2)lo2p(category-2)

Where s=total sample space.

EG: E= -1/2 log(1/2) – 1/2log(1/2) 🡺-log(1/2) 🡺 0.3

**ROOT NODE**: Represents the entire population or the sample and this further divided into.

**LEAF NODE**: Node cannot be divided further.

The first step of decision tree is to find out root node based on info gain.

**Formula of info gain:**

Info gain= Entropy(s)-[(weighted avg)\*Entropy(each feature)]

🡪The category which has highest info gain will be the root node of decision tree.

**Formula for WEIGHTED AVG:**

Weighted avg= total number of occurrences/total number

**RANDOM FORESTS ALGORITHM:**

Here we have number of decision trees and the final output is based on majority voting of decision tree.

Ensemble() is a method in random forests, which is group of decision trees.

n- Estimators tell number of decision trees.

**DEPLOYMENT DATA:**

The extension of ML model is .pkl

Framework: a) Django b) Flask

We can deploy our ML model in these frameweoks.

***Cloud Services:***

🡪Cloud computing is a concept of using the services hosted over the internet by third party cloud service providers.

1. Amazon web services (AWS sage Maker)
2. Microsoft Azure
3. IBM cloud (Watson Machine Learning)
4. Google cloud platform
5. Digital ocean
6. GoDaddy

🡪Watson studio is a data platform in IBM cloud which has various tools to train, test and deploy the AI model.

* + Visual recognition models: to analyze image
  + Natural Language classifier: to classify text
  + Analytical dashboards: to visualize analytical

**Ways to create ML:**

a) Model builder: Guides you step by step through algorithm.

b) Flow editor: Presents a graphical view.

c) Experiment builder

d) Notebooks

**Cloud Object Storage:** is a service in IBM cloud to store and retrieve files. The service is assigned to Watson studio, to store and retrieve data sets to train the object model.

Watson ML service is used to save and deploy ML model in the IBM cloud. We will be using Watson ML repository to store the ML model.

TO save and deploy our model I this service, use service credentials.

!pip install watson-machine-learning-client –upgrade is used to install Watson ml in our system.

🡪Watson-machine-learning-client library is used to save and deploy ML model.

From watson-machine-learning-client import WatsonMachineLearningAPIClient

WMI-credentials={“username”:””,”pswd”:””,……}

🡪client= WatsonMachineLearningAPIClient(WMI-credentials)

Model-props={client.repository.MOdelMetaNames.AUTHOR\_NAME:”SR”, client.repository.ModelMetaNames.AUTHOR\_EMAIL:[sr@gmail.com](mailto:sr@gmail.com), client.repository.ModelMetaNames.NAME:”MLR MODEL”}

**STEPS FOR DEPLOYMENT:**

1. Initialize watson machine learning client.
2. Initialize model properties
3. Save the model

Store\_model() is a method used to save ML model in ML repository.

Model= client.repository.store\_model(lr,meta-props=model-props)

Client.repository.list\_models()

After all these steps, deploy the save ML model to get the URL or scoring end point to access the ML model everywhere.

P\_uid= client.repositor.get\_model(model)

dep= client.deployments.create(p\_uid,name=”MLRModel”)

NODE RED: is flow based programming tool and the backend runtime of NODE-RED is NodeJs.

Using Node-Red, we can develop web application.

**STEPS TO GET PREDICTIONS USING SCORING ENDPOINT:**

🡪 Make an Http get request to wml url for getting the token as response.

🡪 Save the token.

🡪Make Http post request to scoring endpoint url using token and input to get predictions.

**Procedure:**

1. Place the wml url in http request node i.e., <url:https://........................com/v3/identity/token>

And then use basic authentication and then copy username and pswd for wml url and in return, place parsed JSON object.

1. To save that, use function node and to write java script code to save token.
2. Msg.header={“Authorization”:”Bearer “+token}
3. Extracting the wanted data from the JSON data is called as JSON parsing.
4. Dashboard nodes are used to create UI.

15/05/19

Syntax to create a global variables: global.set(var name,value)

Syntax to get the global variable value: global.get(variable name)

**DEEP LEARNING:** Deep learning is subset of machine learning.

* Deep learning is data hunger which requires huge data for training purpose.
* Advance concept of AI is DL.
* DL is subset of ML which makes the computation of multi-layer neural networks feasible.
* ML (Algorithms) may not handle huge data in some cases. Whereas DL algorithms can handle huge datasets.
* If we give small datasets we don’t get accurate output in DL.
* In ML feature extraction of images should be done by developer whereas in DL, it is done by model itself.

**NEURON:**

Dendrites are input.

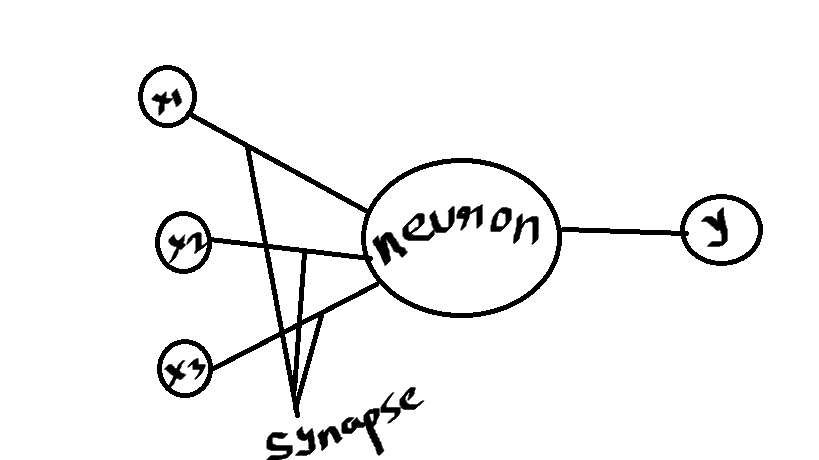
Nuclei are processing.

Axon is the output.

One neuron’s axon is connected to dendrites of other neuron.

There will not be physical connection but transfer of information.

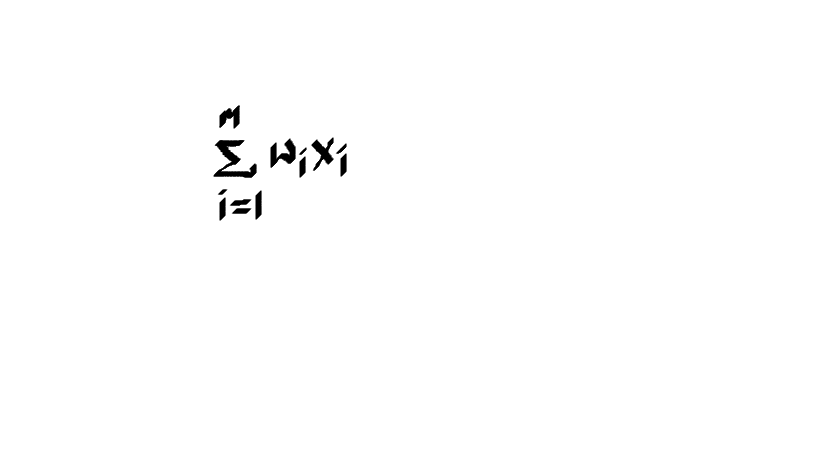
* Every neuron is connected to another neurons, this connection is called as synapse.
* Independent variables are the input to our Neural Networks.
* The output of neural network may be continuous, binary or categorical.



X1, x2, x3 are independent variables.

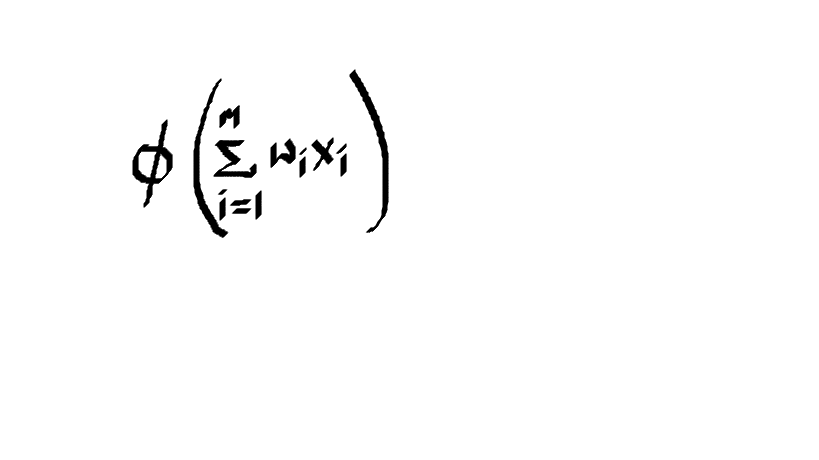
* Number of input neurons depends upon number of independent variables.
* If categorical, based on the categories the output divides.
* Single neuron is present it is named as perseptron.
* Every synapse in a neural network we will have a weight.

1st step in neural network in neuron:



m= no. of independent variables.

2nd step is activation function:

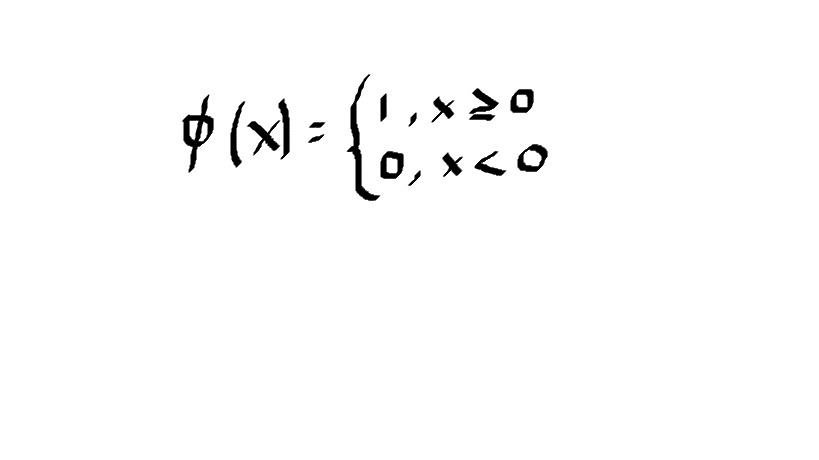
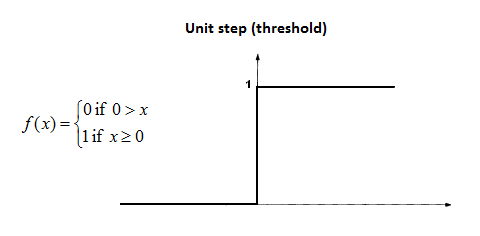


3rd step is y=f(x).

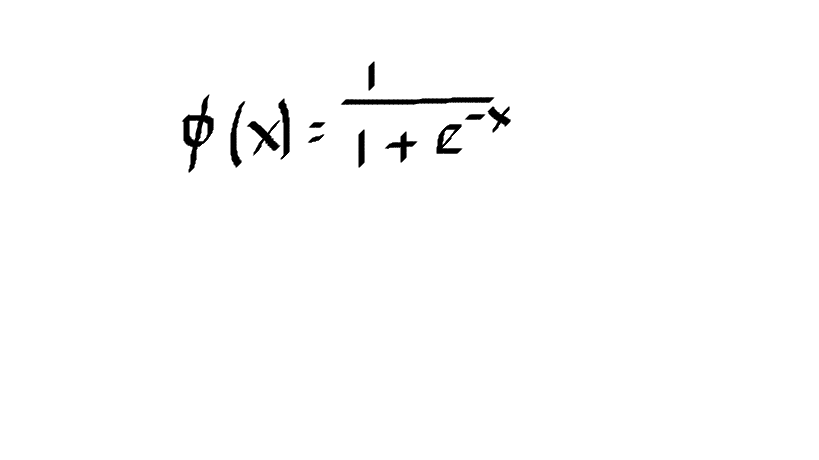
Activation function is used for transformations in the output.

Different type of activation functions available are:

1. Threshold activation function
2. Sigmoid function
3. Rectifier linear unit or (relu).
4. Hyperbolic tangent (tanh)

**Threshold activation function:** 

**Sigmoid function:** The output of sigmoid functions ranges from 0 to 1.



**Rectifier Linear unit:**

Values that are greater than zero – remains same.

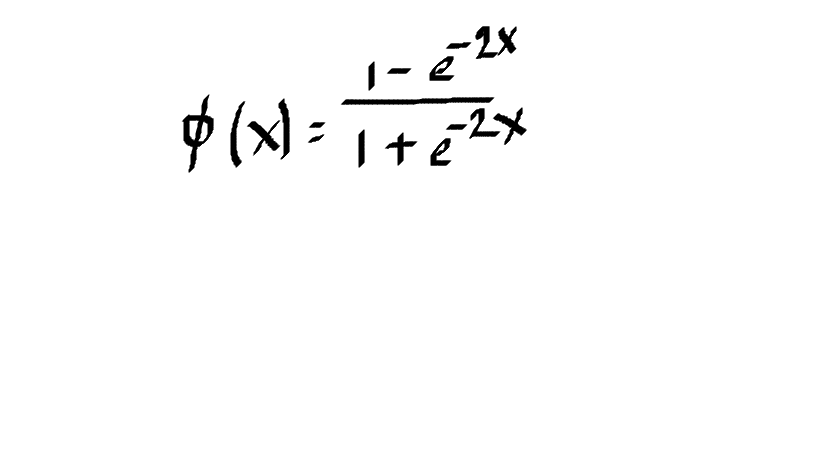
Values less than zero – is done as 0

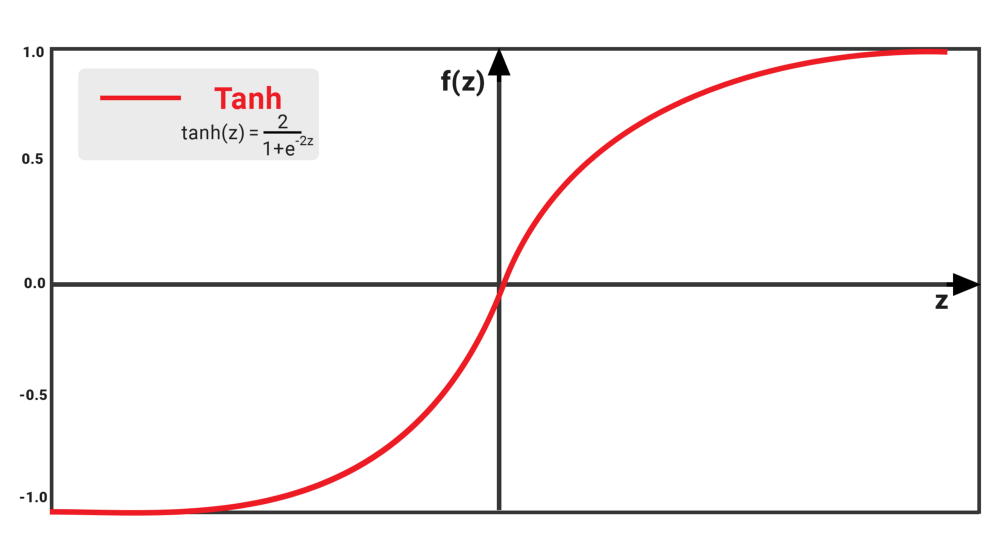
RELU activation function removes negative values.



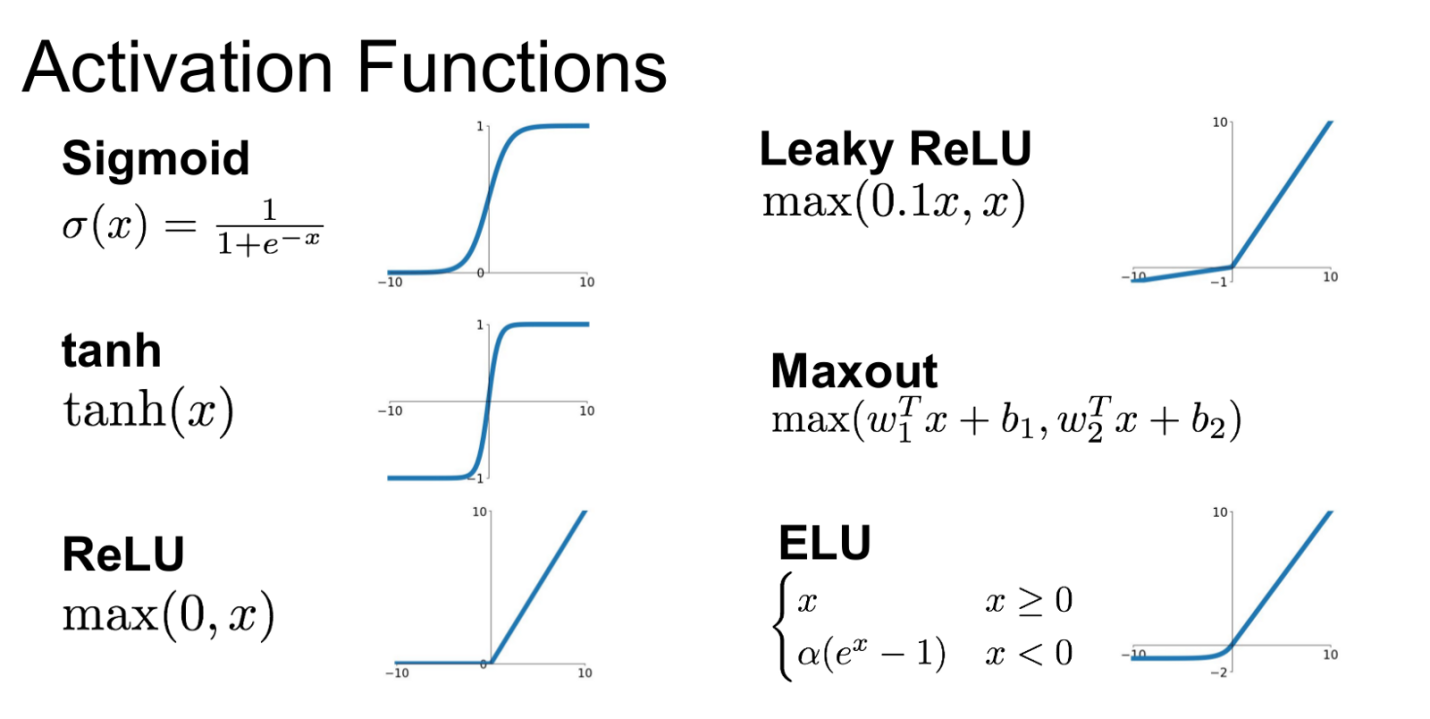
**Tanh**:

Tanh range is -1 to 1.





**ALL ACTIVATION FUNCTIONS:**



**When dependent variable is continuous activation function is not applied.**

How do neural network learn?

**STEPS FOR NEURAL NETWORK LEARNING PROCESS**:  
**step1**: Input the 1st observation and initialize the weights randomly close to 0 but not 0.

**Step2:** forward propagation. (Send the input values along with the weights, multiply them add them and apply activation function.) It is from left to right

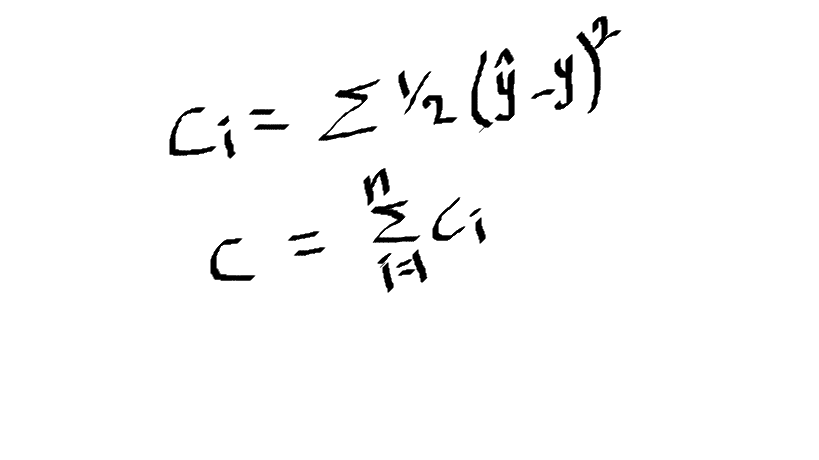
**Step3:** calculate the error between the actual value and the predicted value.

The error is measured based on the respective weight responsibility.

**Step4:**  Backward propagation from right to left.

In this the weights are changed i.e optimized.

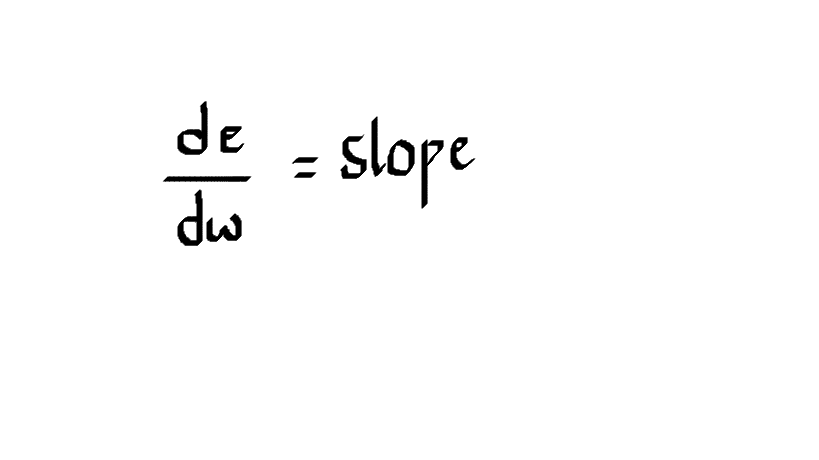
We send all the rows as input, then we will get y and y^. Now, we will calculate errors for all rows and update the weight. This is called as gradient descent.



🡪 This backward propagation continues until C becomes 0. To C become 0, c1, c2, c3,… must become 0.

🡪 To calculate the best cost function for world’s super fast computer it takes

3.14 \* 1050 years.

🡪Change in error with respective change in weight = SLOPE. 

🡪 If error increases with increase in weight, then the weight should be decreased. This is also called as positive slope.

🡪 If error decreases with increase in weight, then the weight should be increased. This is also called as negative slope.

**Local Minima problem**: When errors are not parabolic type, then optimal solution finds the local minimum.

To avoid this problem, we go through stochastic gradient descent.

**Gradient Descent:** Adjusting the weights after calculating all the errors of the row in the dataset.

**Stochastic Gradient Descent:** Adjusting the weights for each and every row.

It prevents the problem of local minima.

**Batch Gradient Descent:** It is used for adjusting the weights for a particular batch of size.

Mostly followed Gradient descent in real time is Batch Gradient Descent.

🡪 Epoch: When the whole training set passed through an ANN.

🡪Multiple epochs are performed for more accuracy.

LINK: Playground.tensorflow.org

* The layer which has inputs is called as input layer.
* The layer which processes the data is called hidden layer.
* The layer which has outputs is called as output layer.

**Tensorflow:** It is a library to develop neural n/w, by using this we need to write it from scratch.

**Keras:**

***Steps to build neural network model:***

***Part1: Data Processing***

***Part2: Building neural network model***

***Step1: Importing the libraries***

🡪Sequential class is used to initialize the neural network model and Dense is in the keras.Layers packagr i.e Dense is the neural network layers.

***Step2: Initialize the neural network model. (Init NN model)***

Neural network model is **initialized** by **calling the constructor of sequential class**.

***Step3: Add input layer.***

Layers are added into Neural Networks by using a **function** called **add.**

**Add function** is inn sequential class.

**Input\_dim** represents number of neurons in the input layer.

**init** represents **weights initialization strategy.**

**activation** tells which function to be used.

**output\_dim** represents how many neurons to have in next layer.

***Step4: Add Hidden layer.***

***Step5: Add Output layer***

16/05/19

***Step6: Compile the model.***

In compilation we mention the strategies.

1st strategy is weight optimization/ updating strategy.

2nd parameter is error calculating strategy.

3rd parameter is model performance evaluating strategy.

During compilation of the model we will mention what kind of strategies to be followed during the training process.

Optimizer tells us which method can be used.

Optimizer = “adam” means batch gradient descent.

Loss (calculating error) is calculated by binary\_crossentropy

Formula: **-(ylog(p)+(1-y)log(1-p))**

Where y is actual value and p is predicted value.

Metric is model evaluation strategy.

***Step7: Train the neural network model.***

In fit method we also mention number of epochs along with x\_train and y\_train.

To improve accuracy, we can increase epochs, batch size, number of neurons.

***Part3: Making predictions.***

As sigmoid function is applied the prediction range is from 0 to 1.

In batch size, for total number of rows specified, rows are considered as per batch number mentioned and weights are updated for each particular batch after execution.

*Eg model:*

Input layer: output\_dim=15

Hidden layer: output\_dim= (15+2)/2=8.

To improve the model performance we can improve the number of neurons and increase the number of epochs.

In tensorflow we will have pre defined datasets. In this we will take hand written datasets.

Independent data is images.

**Mnist**  us predefined and written digit images dataset, located in tensorflow repository.

Basically in mnist class we have a method called as load\_data() which downloads the images and returns 4 parameters x\_train,x\_test,y\_train and y\_test.

* The images are in the form of numpy array nothing but 3D array.
* Image can be shown by using a function called as imshow in matplotlib library.

Normalize is in keras.util package, which normalizes the image.

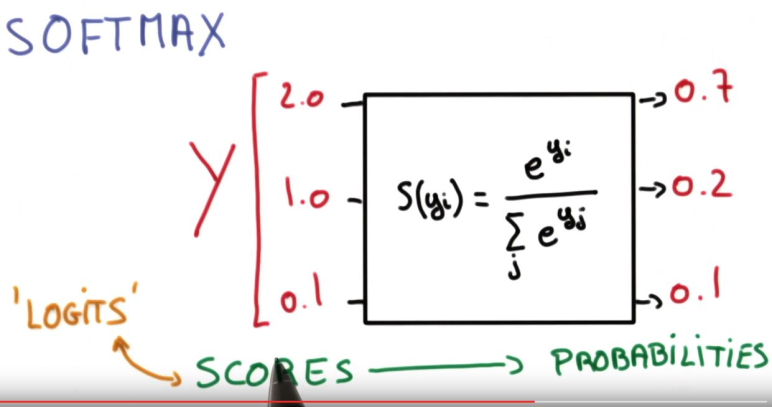
Axis=1: represents column wise normalization/operation.

ANN algorithm (any of the ML algorithms) accepts only 2D array for training.

To convert 3D to 2D we use: x\_train=x\_train.reshape(no.of images, size\*size)

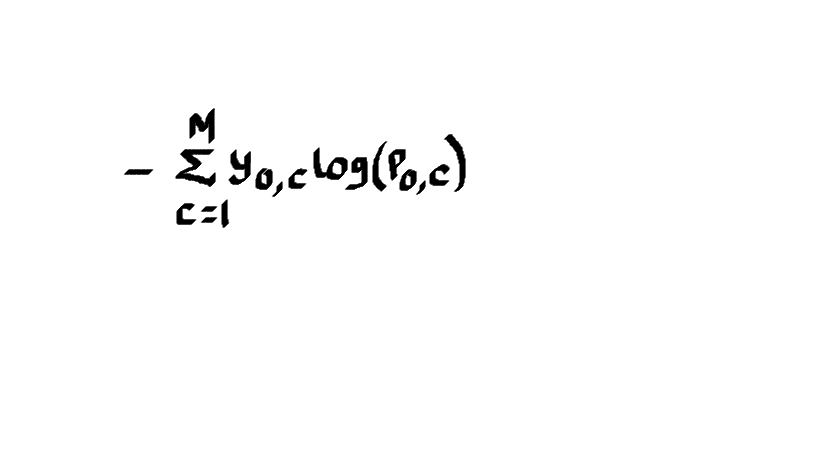
* As we have to apply oneHotEncoding to label, labels should be in 2D.

**SOFTMAX:**



As the output is categorical we will go for categorical\_crossentropy

**Categorical\_crossentropy formula:**



🡪Instead of using rmax we can use predict\_classes so that we can get directly yield the field without other predictions.

🡪To save the trained model in keras model extension is: model.h5

For further usage: model.save(“ann\_mist.h5”)

Whenever we use save method it saves the whole data in the defined path of system.

**OPENCV is the Open source computer vision library:**

The function used to read the image in library: **imread()**

**imread()** returns the image in the form of numpy array.

3 represent RGB

1 represent Grey scale

**ROI: REGION OF INTEREST:**

From the whole image, the part of image user is interested in, here we can apply slicing.

In opencv we have a function called as **imshow()**. It is used to show the image in the GUI.

We need pass 2 parameters a) GUI window name b) Image or frame we want to show.

In opencv there is a function called as blur to blur the image, the parameters are a) Image to be blur b) Intensity of a image to blur.

In opencv there is a function called as video capture to read the video from the camera/webcam/live video.

**Video.read()** in opencv package reads each and every frame from the video.

Read function returns 2paramters. a) Boolean (t/f) whether there is next frame or not. b) the frame.

In opencv package we have function called as **cvtColor()** to convert the color of our image. The parameters passed are: frame we want to convert and 2nd parameter is flag, to which color we want to convert.

**Haar cascade** classifier is a pre trained classifier which detects the haar like features.

(edge features, line features, four-rectangular features).

Line features: white pixels and black pixels.

Haar cascade fontal face detects the human face. (It uses position of particular human)

Haar cascade eyes detects position of eyes in the face.

We can train our own classifier but those 2(haar cascade fontal face and eyes) are trained methods.

CascadeClassifier() method loads the pre define classifiers in to the code.

**Haar cascade** classifiers are trained to work for gray scale images.

So, using cvt color we are converting RGB into Gray.

🡪There is a method called as detectMultiScale which helps us to find the position of the features in the image.

🡪In detectMultiScale the parameters is gray scale image, scale factor, number of neighbors.

🡪 cv2 has a function called as putText which helps us to display a text in a window.

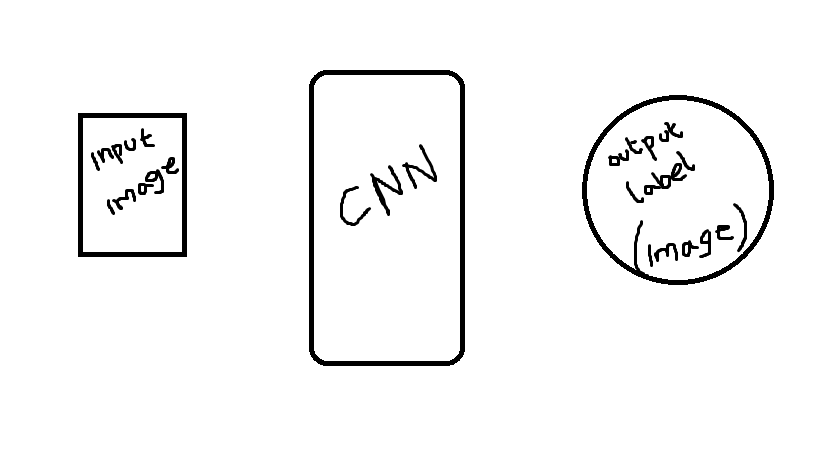
17/05/19

**CNN: CONVOLUTION NEURAL NETWORK:**

🡪 It is used especially for feature extraction.

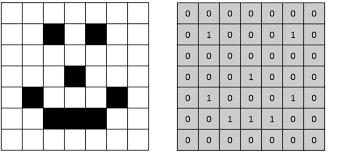
🡪To classify an image we need perfect features.

(For training purpose we need to give good featured images, but not confusing image)



🡪We represent the image in the form of 0’s and 1’s wherever we have particular features we give 1 and rest of them we will give 0.

🡪CNN is used to extract the important features from the image.



**STEPS OF CONVOLUTION :**  
a) Convolution

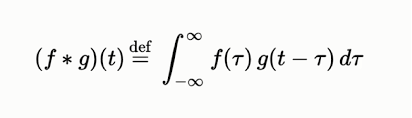
b) Max pooling

c) Flattering

d) Full connection

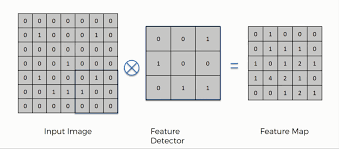
If we follow these steps out convolution algorithm will find the best features in the image.

Formula for convolution:



**CONVOLUTION:**

🡪In convolution operation input image is convolved with the feature detector or filter to form a feature map.



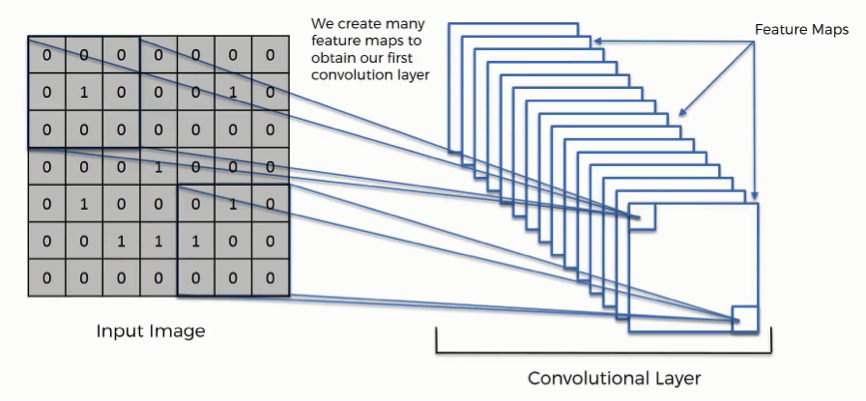
🡪By applying convolution the size of the image is reduced but still the important features are extracted.

🡪Feature detector helps to extract main features in the image.

🡪Based upon the images feature detector is optimized during the training process.

🡪Multiple filters or feature detetctors are applied to avoid the loss of important features in the image.

🡪Group of feature maps is called convolution layer.



**MAX POOLING:**

Spatial invariance can be reduced/removed by applying pooling.

Fitting: training something.

Over Fitting: Largely training. Only one particular angle/image/structure is trained.

This can be reduced by pooling.

Under Fitting: Always rejecting.

🡪 We will be applying pooling operations on feature map.

Pool size: The size of pooling.

🡪By applying pooling 75% of unwanted features are removed, by extracting the maximum features from the feature map.

🡪Pooling prevents over fitting, spatial invariance and distortions of images.

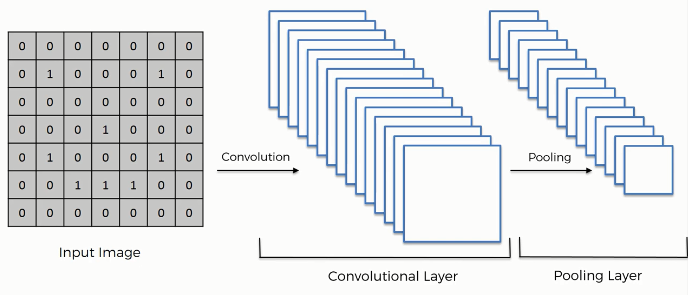
Spatial invariance: Features in different position.

There are different kinds of pooling like:

1. Sum pooling (Extracting sum of the features)
2. Mean pooling (Extracting mean of the features)

Mostly used is Max Pooling.

🡪Multiple pooled feature maps forms pooling layer.

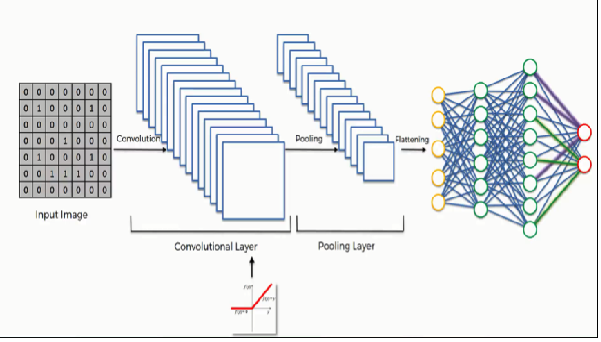


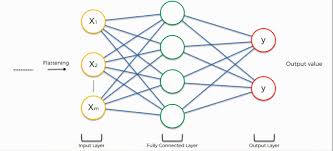
**FLATTEN:**

🡪In flattening multi dimensional pooled feature map is converted into single dimension.

🡪Each feature is formed as single neuron.

🡪Flattening output is the input layer of ANN.





🡪In CNN during backward propagation feature detector is also optimized, along with the weights.

**STEPS TO BUILD CNN MODEL:**

1. Import library

Conv2D is used to add convolution.

1. Initialize the Neural Network model
2. Add Convolution layer

The 1st parameter passed in Conv2D is number of feature detector to be applied to the input image. 2nd parameter is size of the feature detector that is applied tot our image.

3rd parameter is input\_shape we have to define the size of the image. Parameters in input\_shape are Size of the image (64-width, 64-height), RGB (3-number of channels).

4th parameter is activation. Relu activation function is applied in the convolution layer to avoid/remove negative pixels. This helps us to avoid non-linearity in the images.

1. Add pooling layer
2. Add Flatten layer (we need not add input layer because flatten layer is itself acts as input layer to ANN).
3. Add hidden layer

Output\_dim is number of neurons in next layer.

1. Add output layer.

When we have 2 categories we can apply sigmoid. If more than 2 we apply softmax.

***Follow these steps to add code:***

Keras.io 🡪 Image Processing 🡪  .flow\_from\_directory(directory)

* ImageDataGenerator applies geometrical operations to the images like re-scaling, zooming, flipping (vertical & horizontal), ranging to avoid over fitting.
* As we do not use test dataset for training purpose we do not use all the geometrical operations.
* **flow\_from\_directory()** is a method in ImageDataGenerator class to import the images from the local directory.

The 1st parameter is the path of training dataset.

Target size represents the size of the image to be converted.

**Class\_mode:** If the dependent variable has only 2 categories then it is binary, otherwise it is categorical.

* Fit only trains the model, but fit\_generator simultaneously trains and tests the model.
* Steps\_per\_epoch = number of images/batch size.
* To find which category got which numbering we use: x\_train.class\_indices.

🡪To use a model, from keras.models import load\_model

Model= load\_model(‘Path’)

🡪To deploy keras model into IBM Watson studio we have to convert .h5 file to .tgz file.