

Q. Diff basic b/w RNN, CNN, DNN, and KNN?

Answer:

1. **Recurrent Neural Networks (RNN):** RNNs are a class of neural networks designed to recognize patterns in sequences of data, such as time series or natural language.

Key Features:

- **Memory:** RNNs have a form of memory built into the network through loops in the architecture, allowing them to maintain information about previous inputs.
- **Sequence Handling:** They are adept at handling sequential data and maintaining temporal dependencies.

Uses:

- **Natural Language Processing (NLP):** Language modelling, text generation, and machine translation.
 - **Time Series Prediction:** Stock price prediction, weather forecasting.
 - **Speech Recognition:** Recognizing spoken words and phrases.
2. **Convolutional Neural Networks (CNN):** CNNs are a type of neural network particularly well-suited for processing data that has a grid-like topology, such as images.

Key Features:

- **Convolutional Layers:** These layers apply a convolution operation to the input, capturing spatial hierarchies in data.
- **Pooling Layers:** Used to reduce the dimensionality of feature maps while preserving important features.
- **Translation Invariance:** Capable of recognizing patterns regardless of their location in the input.

Uses:

- **Image Recognition and Classification:** Identifying objects within images.
- **Computer Vision Tasks:** Image **segmentation**, object detection, facial recognition.
- **Video Analysis:** Action recognition, video classification.

3. **Deep Neural Networks (DNN):** DNNs are neural networks with multiple layers (often more than three) between the input and output layers. The term is often used broadly to refer to any neural network with a significant number of layers.

Key Features:

- **Depth:** Having multiple hidden layers enables the network to model complex patterns.
- **Hierarchical Feature Learning:** Capable of learning abstract representations at different levels.

Uses:

- **General Machine Learning Tasks:** Can be applied to a wide range of problems including classification, regression, and anomaly detection.
- **Deep Learning Applications:** Used in various deep learning frameworks for tasks in computer vision, NLP, and more.

Note: -

RNN/CNN vs. DNN: Both RNNs and CNNs are specific types of DNNs tailored for different types of data (sequential for RNNs and grid-like for CNNs). DNN is a more general term for any deep network.

Stock Price Forecasting using concepts of Stacked LSTM – Deep Learning

Ways we follow:

Step1: Collect stock data

Step2: Preprocessing – Train and Test

Step3: Create an LSTM model (stacked)

Step4: Predict the future 30 days and plot the output

API keys are generated after signing in with email id: ankit485803@gmail.com

through these TIINGO

https://pandas-datareader.readthedocs.io/en/latest/remote_data.html

Table of Contents

Introduction	5
Machine Learning (Types):.....	5
Deep Learning & Neural Networks:.....	6
Chapter 2: Machine Learning	7
Chapter 3: DL and Neural Network.....	11
1. Artificial Neural Network	11
2. Recurrent Neural Network	27
3. Concurrent Neural Network	30
Natural Language Processing (NLP)	31

Chapter 01

Introduction

Command on 10th standard stats (mean, median, mode, std. deviation, Covariance, Corelation, normal or gaussian distribution) maths chapter

GDSC_IITP resources

<https://docs.google.com/document/d/1cM7WsWY7S-WlztBFZVk5WnJHqgTuthspGY2PDdvOfM/edit>

Machine Learning (Types):

- Supervised Learning
- Unsupervised Learning
- Semi-supervised Learning
- Reinforcement Learning
- Regression

Supervised Learning and Techniques

- I. Linear Classifier (numeric function)
- II. Parametric (probabilistic function)
- III. Non parametric (K nearest neighbour, Kernel regression, kernel density)
- IV. Non metric (symbolic function: Classification & Regression, Decision Tree)
- V. Aggression (Random Forest, bootstrap)

Note: **Reverse – Engineering** is also best sometimes!!!

Unsupervised Learning

- I. Clustering (K means & Spectral clustering)
- II. Density Estimation (Gaussian Mixed Model, Graphical model)
- III. Dimensionally reduction (PCA, Factor Analysis)

Resources by APNA COLLEGE

<https://docs.google.com/document/d/1AOXV76qKbJe9xHNqF9o00Kuj-1CKbnJ6LHCgIYlaKsQ/edit>

Deep Learning & Neural Networks:

- Artificial Neural Network (ANN)
- Recurrent Neural Network (RNN)
- Convolutional NN
- LSTM (Long short-term memory)
- Transform
- KNN (K nearest neighbour algo) – non parametric supervised learning classifier

Imp: **Emotional Detection** using Bidirectional LSTM concept

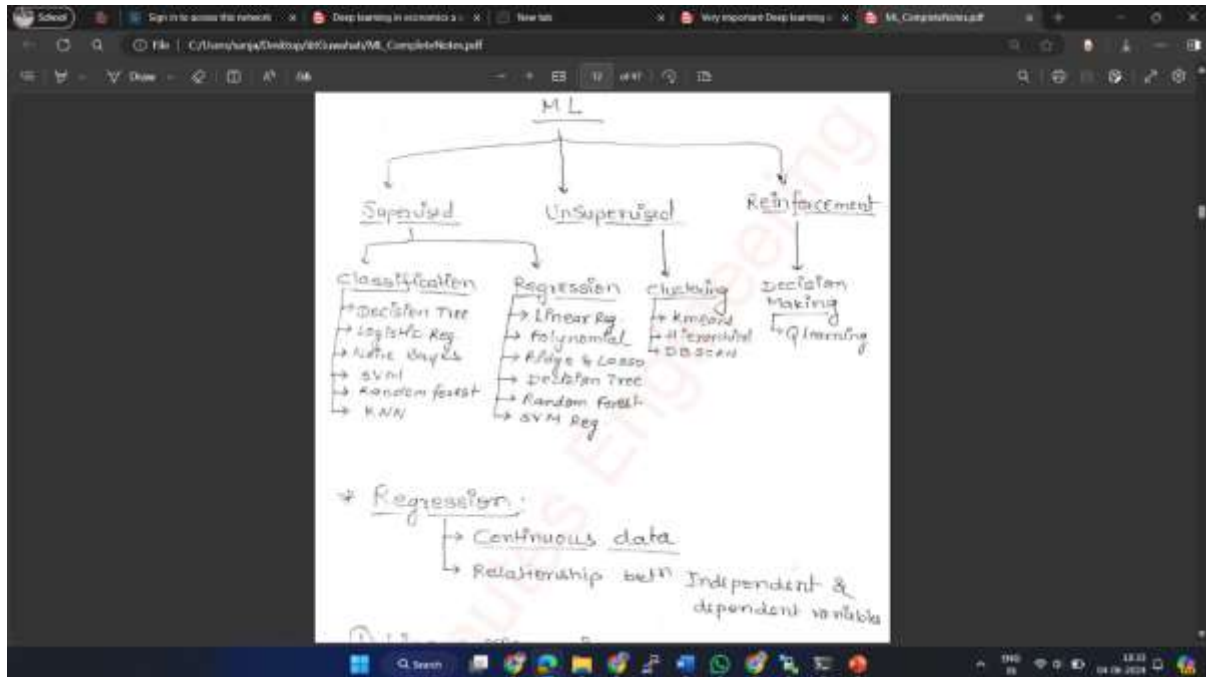
Note: PPT animation time – Start on click, **Duration: 00.20 sec** and Delay: **0.0** as Dr Hari sir and Debanjana Didi suggest on my first presentation on 5th June'24 Wednesday because audience or committee become bore.

Terms to be read soon:

MLE (Maximum Like hood Estimation),

Chapter 2: Machine Learning

In this chapter, we have covered the complete concept of fundamentals of ML, as well as in depth. Steps we followed: 1. Supervised, 2. Unsupervised, 3. Semi supervised, 4. Reinforcement



Note: This screenshot (GUIDE of ML) is taken from 5 minutes Engineering YouTube channel.

<https://www.youtube.com/@5MinutesEngineering>

SUPERVISED – LEARNING:

Major Topics: Classifier, Regression

Note: Started date = 30th May'24, Thursday and RESOURCES: 1st priority = GFG theory + sample code, 2nd priority = Tutorial by Krish Naik

<https://www.geeksforgeeks.org/machine-learning/>

Regression = show relation b/w Dependent variable (generally rep by Y) and independent variable (X) according to stats (maths chapter)

Types of Regression: Linear R, non-linear (Polynomial R, Decision Tree R, Random Forest R, SVR)

Linear R: used to predicate, focus on CONDITIONAL probability $P(A/B) =$

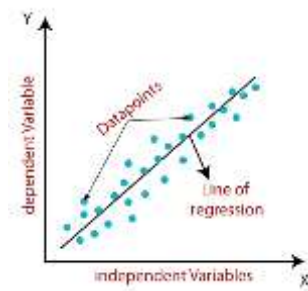
$$P(A \text{ intersection } B) / P(B)$$

Limitation: Overfitting

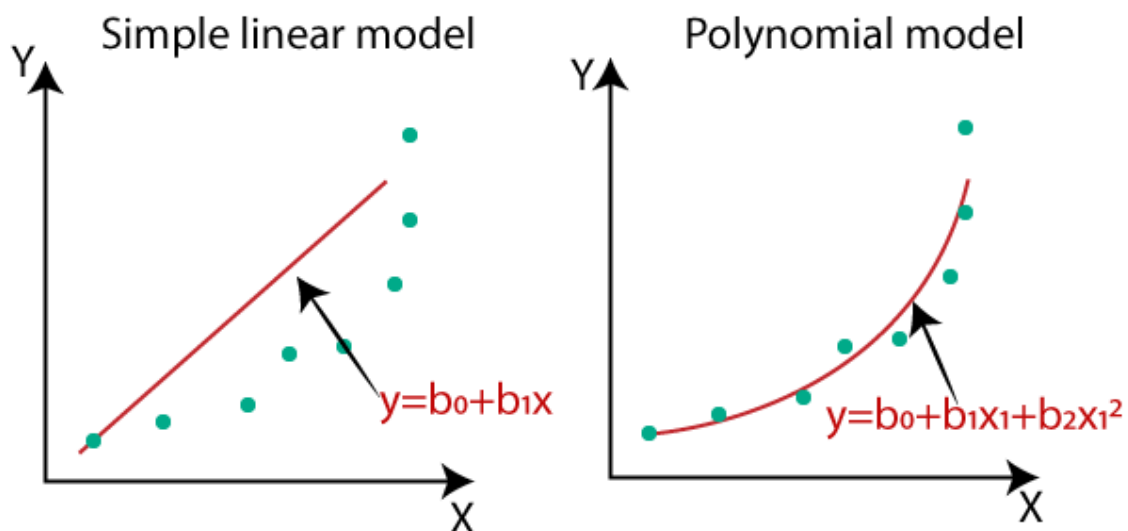
Equation: $y = mx + c$, where m is **model weight** or parameter, y is dependent PREDICATOR variable and x are independent c is bias

Req Module: sklearn

```
from sklearn.linear_model import LinearRegression
```



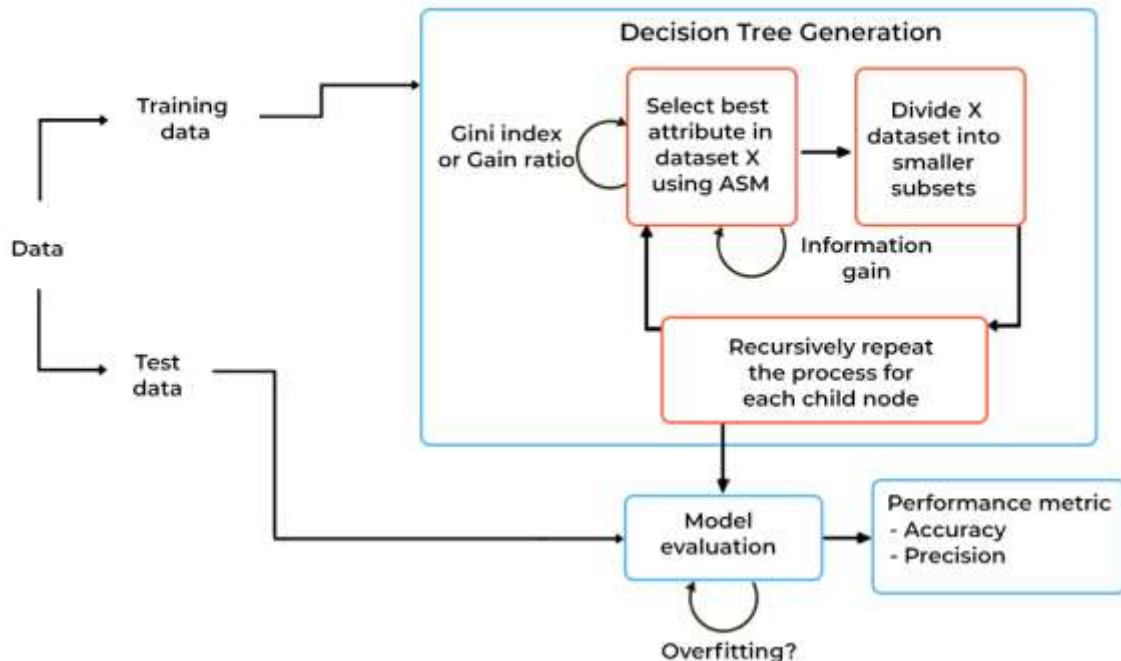
Polynomial R: used to show nonlinear relation b/w dependent and independent variable



Decision Tree R: most powerful tool for classifier and predication, flowchart like Tree structure (DSA chapter: Node, branch, internal node, leaf). There is **non parametric** method to model a decision tree to predicate a continuous outcome.

<https://www.geeksforgeeks.org/ensemble-methods-in-python/>

HOW DOES A DECISION TREE WORK?



Random Forest R: most multiple decision tree, using techniques **BAGGING** = Bootstrap + Aggregation. Bootstrap is row sampling and feature sampling from dataset.

SVR (Support Vector Regression): most uses linear and non-linear KERNEL for predication Linear Kernel means simple dot product of two input, whereas Non-linear uses more complex function to capture to more advance pattern in data.

<https://www.geeksforgeeks.org/multiclass-classification-using-scikit-learn/>

Logistic R: uses the binary classification (output in 0 & 1) with the help of sigmoid function
<https://www.geeksforgeeks.org/understanding-logistic-regression/>

.....
 Date: 31st May'24 (Friday)

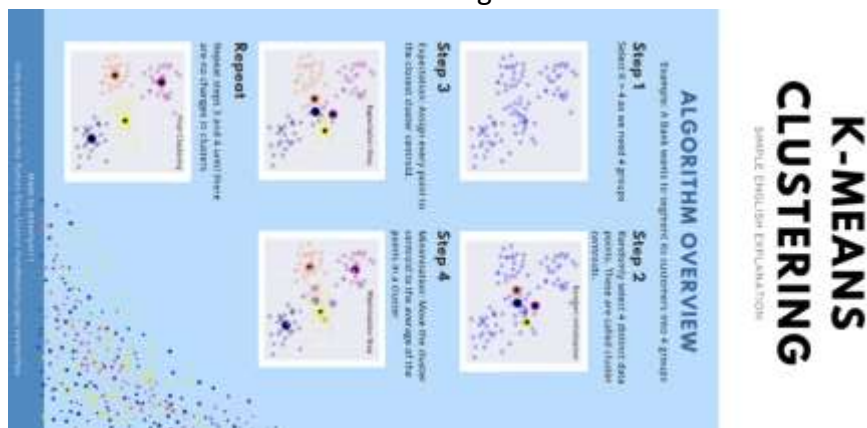
UNSUPERVISED – LEARNING:

Major Topics: Clustering, Associative Rule Learning, Dimensionally Reduction (PCA)

Clustering (K mean, Hierarchical clustering, Density based, Mean shift, Spectral clustering)

Aim of clustering = divide the datasets into sub set on the basis of similarity

1. **K mean clustering:** take K points (K1, K2) and initialize centroid C1, C2, from circular disks shaped and calculated mean distanced using Euclidean distance formula.



<https://www.geeksforgeeks.org/k-means-clustering-introduction/>

2. **Hierarchical clustering:** two things Agglomerative & Divisive, there are created Dendrograms like graph

Associative Rule Learning: very helpful relation b/w parameters of large dataset.

Dimensionally Reduction: process of reducing no. of feature in dataset while preserving as information as possible.

PCA (Principal Component Analysis): Learn first eigenvalue, eigenvectors, Covariance, correlation, covariance matrix computation which make edge to PCA.

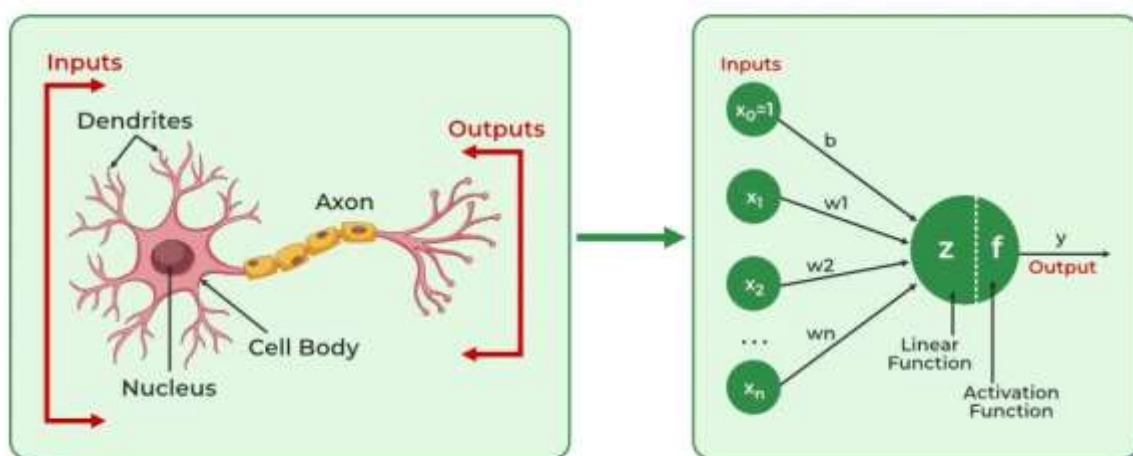
Reinforcement Learning:

RL is a science of decision making, data is accumulated from ML system that uses trial-error method or Reward based learning (feedback in the form of reward and penalties for wrong action is famous called Markov Decision Problem),

Chapter 3: DL and Neural Network

In this chapter, we have covered whole concept of neural network (used in deep learning) subparts as 1. ANN, 2.RNN, 3. CNN, 4. LSTM, 5.KNN

1. Artificial Neural Network



Research Paper **PATENTS**: means P

Patents are **legal** documents used to prevent other people from commercializing the process or devices they describe. Research papers typically report results of a scientific process, whether experimental or not, while patents describe processes and devices along with their respective intended applications.

Activation Function: Take all inputs from user and collect to pass through the activation function
3 types of activation function: Linear, Heviside Step function, Sigmoid function.

Feed Forward Neural Network:

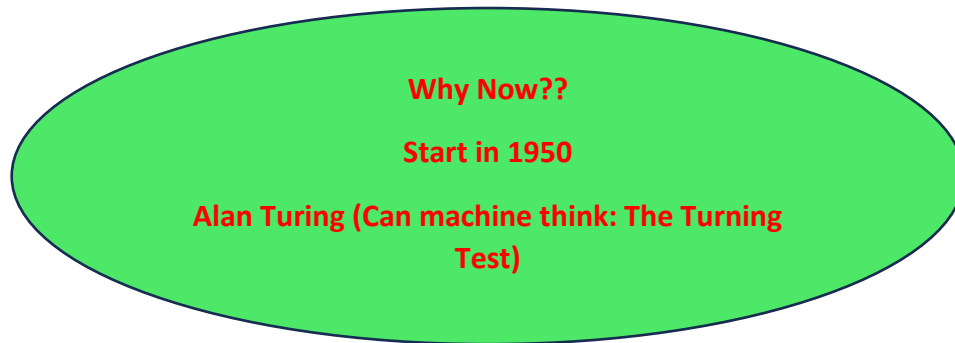
This **NN** don't form loop, once it goes forward then doesn't come back e.g. MLP and CNN

Feed Back NN: loop can form e.g. RNN & LSTM

<https://blog.paperspace.com/feed-forward-vs-feedback-neural-networks/>

Moore's Law: In every 2 years of interval, the no. of transistors is **double** and cost of production is Half.

NVIDIA, CUDA is a programming language to train the GPU



- **FPGA** = Field Programmable Gate Arrays, help to create integrated circuits (fast + Low power + reprogrammable) very costly
- ASIC = Application specific integrated circuits: e.g. TPU (by google), GPU, NPU (Neural Processing units)
- TensorFlow (+ KERASE) by google for Industry, PyTorch by Facebook for **Researcher**.
- **Transfer learning**: scientists & research already train a model on the major problems and make a free & publically available model (not doing so then waste of Time + Money + Efforts)

<https://www.geeksforgeeks.org/ml-introduction-to-transfer-learning/>

13th June'24 (Thursday)

imp Architecture (Transfer Learning):

Transfer learning algorithms, however, takes already-trained models or networks as a starting point. It then applies that model's knowledge gained in an initial source task or data

Popular arch.

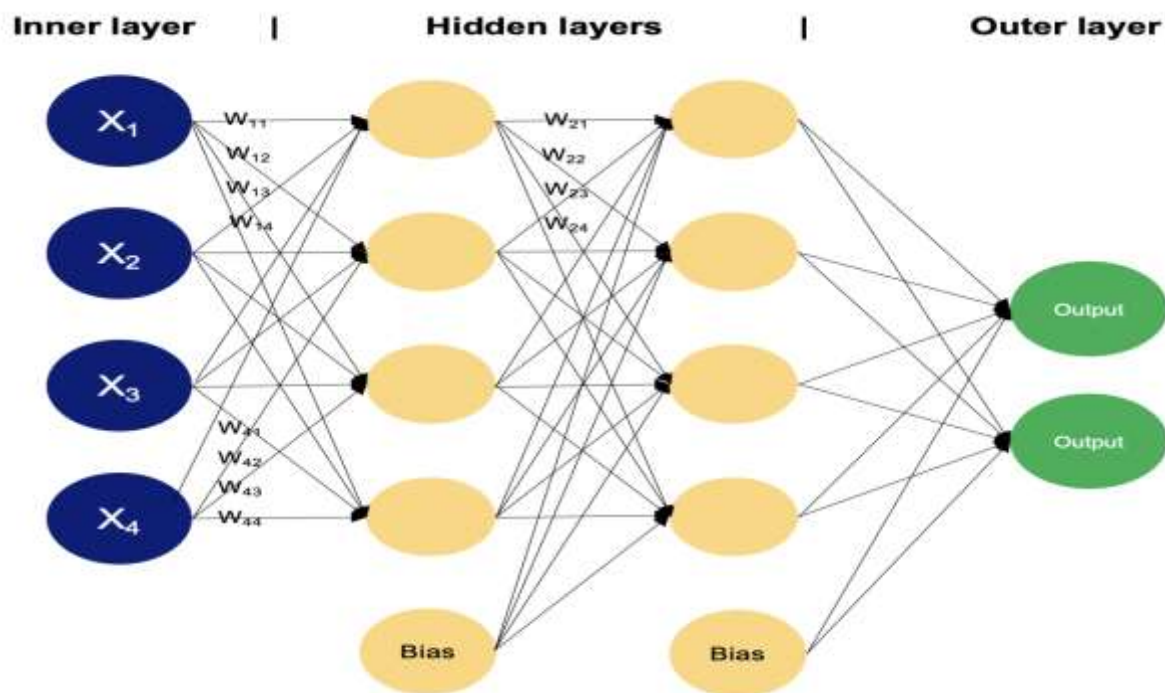
- (i) “**ResNet**” for image classification

https://www.researchgate.net/publication/346212393_Image_classification_based_on_RESNET

- (ii) “BERT” for text classify.
(https://www.tensorflow.org/text/tutorials/classify_text_with_bert)
- (iii) **U-Next** for image segmentation
- (iv) Pix2Pix: image translation
- (v) Object detection: **Yolo** (<https://encord.com/blog/yolo-object-detection-guide/>)
- (vi) Speech generation: WaveNET

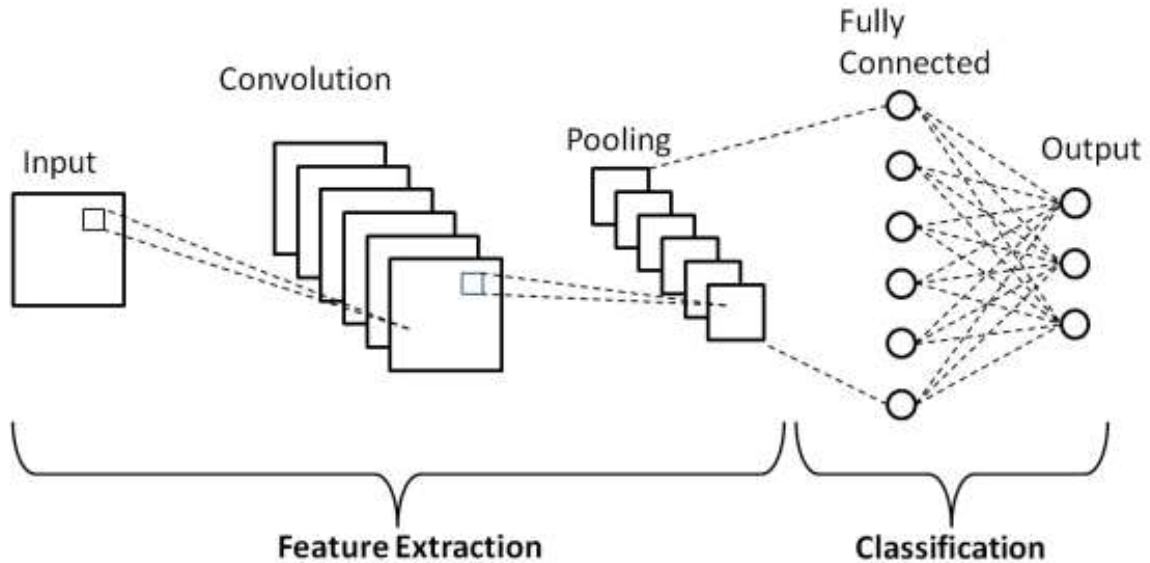
Q. Type of Neural Network

Ans: Multi-layer Perceptron (MLP): is a type of NN, commonly used for supervised learning tasks such as **classification and regression**. It consists of **multiple layers** of nodes, each connected to the nodes in the previous and next layers. MLPs are known for their ability to learn complex patterns in data and are widely used in a variety of applications in machine learning.



CNN stands for Convolutional Neural Network. It is a type of deep learning algorithm commonly used in image and video recognition tasks. CNNs are designed to automatically and adaptively learn spatial hierarchies of features from input data to perform tasks such as image classification, object detection, and image segmentation.

- CNNs are composed of multiple layers, such as **convolutional layers, pooling layers, and fully connected layers**. The convolutional layers extract features from the input using **filters or kernels**, while the pooling layers down sample the feature maps. The fully connected layers then perform the final classification or regression task.



- CNNs have revolutionized the field of **computer vision** and are widely used in various applications like facial recognition, self-driving cars, medical image analysis, and more.

Convolutional Neural Network (CNN) for image and video processing,

- Feed Forward NN: signal move in forward direction linearly from first to end (MLP + CNN)
- Feed Back NN: loop form in b/w hidden layer which help to improve the efficiency of output e.g. RNN and LSTM
- **Auto Encoder** (not very popular): used to compressed the video & audio without considering the quality <https://www.geeksforgeeks.org/auto-encoders/>
- Generative Adversarial Networks (**GAN**): the power of adversarial training which tends to produce much sharper and discrete outputs rather than blurry averages that MSE provides. Generator (create new photo which never exist) + Discriminator (to check real or fake). “Brother of CNN in computer version”

Note: Perceptron works on only on linear function but not in non-linear like XOR. Solution is Back propagation you can solve any function (weight, different algo. SVM, Random Forest). Deep

https://en.wikipedia.org/wiki/Deep_learning

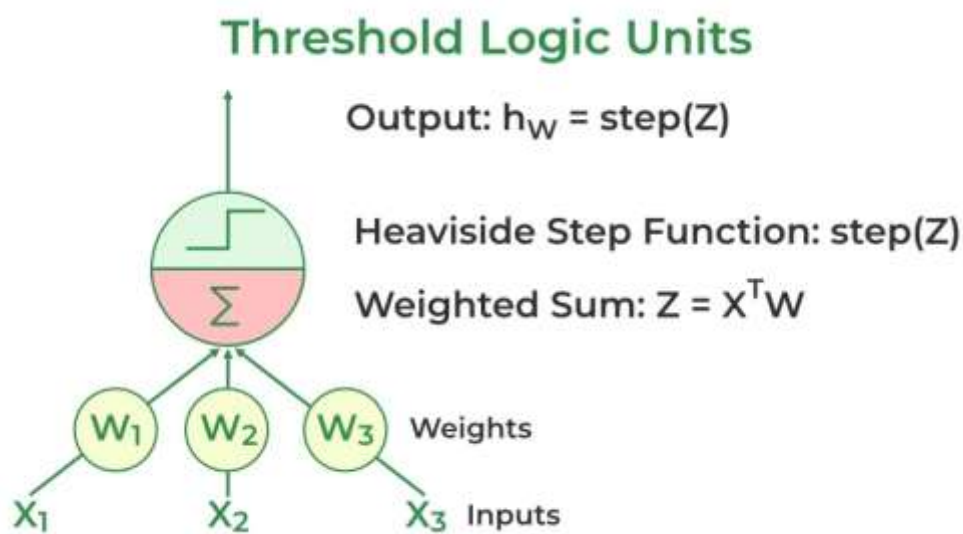
Perceptron v/s Neuron and Perceptron Geometric Intuition

Perceptron: is the simplest ANN and type of feed forward NN, consisting of a single layer of input nodes that are fully connected to a layer of output nodes. It can learn the linearly separable patterns. it uses slightly different types of artificial neurons known as threshold logic units (TLU).

<https://www.geeksforgeeks.org/what-is-perceptron-the-simplest-artificial-neural-network/>

Types of perceptron: Single-layer, multi-layer

We can say **MLP = ANN**



Difference b/w Neuron & Perceptron:

Neuron: Complex str. & Nucleus behind electronuclear reaction we don't know, Processing, Neuroplasticity

- Perceptron: Easy, only two operations (summation + step function), Connection,
- Perceptron is line which help us **to create a region** whether decide any student is placed or not (same concept read in IIT Patna math Sem2 Graph Theory, LPP: Linear Programming problems & Network by Dr Mahendra Kr Gupta sir)
- Perceptron is known for **Binary Classifier** because it divides the region or classes into two equal half using straight line.
- Perceptron acts like: In 2D = Line, 3D = plane (assume sheet), 4D = Hyperbola
- Limitation of perceptron: only **linear & sort of linear to classify** applicable in 2D, 3D, 4D in class cases. Be careful about this in nonlinear dataset!

<https://github.com/campusx-official/100-days-of-deep-learning/blob/main/day3/placement.csv>

SKLEARN python module supports the perceptron model that's why we don't need to write code by own

https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Perceptron.html

Start: 14th June (Friday)

How to train a perceptron? Perceptron - Trick

We use Line of equation in Linear regression: $y = mx + c$

But in Logistic regression is $y = AX_1 + BX_2 + C = 0$ (general form of line)

<https://www.geeksforgeeks.org/deep-learning-tutorial/>

positive and negative region visualize tool (I've used in Maths IITP this tool many times) transform of line <https://www.desmos.com/calculator>

Learning rate = 0.01

```
# make classification name se aap dataset bana sakte ho

from sklearn.datasets import make_classification
import numpy as np
X, y = make_classification(n_samples=100, n_features=2,
                          n_informative=1, n_redundant=0,
                          n_classes=2, n_clusters_per_class=1,
                          random_state=41, hypercube=False, class_sep=10)
```

The Perceptron Trick is a technique used to adjust the weights of a perceptron algorithm in a supervised learning scenario. It aims to minimize errors in classification tasks by updating the weights iteratively based on misclassified instances.

<https://medium.com/@Anm.Y/understanding-what-is-the-perceptron-trick-and-how-it-works-39c5025f6ebe>

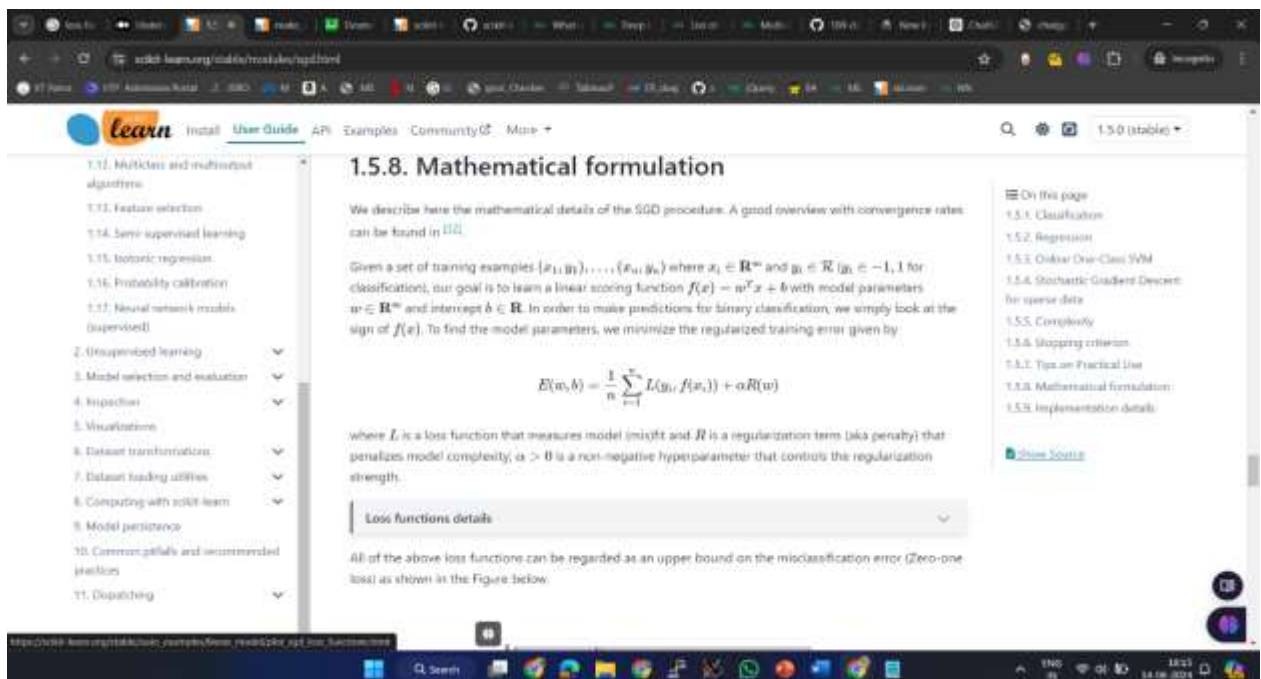
Perceptron Loss Function and Hinge Loss & Binary Cross Entropy

Loss function: way to determine how much our ML model is correct or wrong

- Loss function, also known as a cost function or objective function, is a measure of how well a machine learning model is performing on a given dataset. The loss function calculates the difference between the predicted values generated by the model and the true values in the training data.
- to **minimize this loss** function, which indicates how far off the predicted values are from the actual values.
- Some imp loss function in different algo 1. Linear regression: MSE (mean square error) 2. Logistic regression: log x, 3. SVM (Support vector machine): Hinge loss and 0 & 1
- You can also make own loss function e.g. number of **misclassified points**, this can be your loss function.
- **Working:** for every line it gives a number, this number tells you model error, direct count the point of miss-classified points == error, magnitude of error

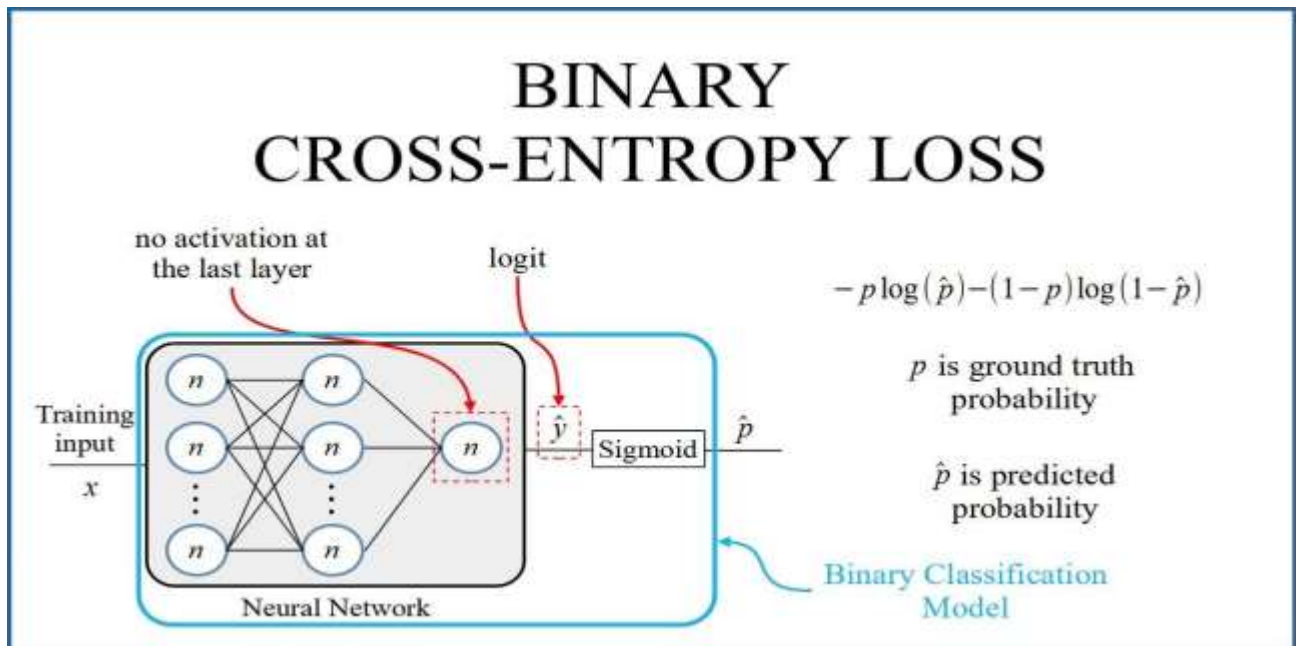
Loss function: way to determine how much our ML model is correct or wrong

Actual Loss function used in SKLEARN library <https://scikit-learn.org/stable/modules/sgd.html>



Now calculate the value of w_1 , w_2 , b that make whole equation of loss function $L = 0$, coming the concept of GRADIENT DECENT

Binary Cross Entropy: is often used in binary classification tasks to measure the uncertainty or impurity of a dataset. It is a common metric used in decision tree algorithms, such as ID3 or C4.5, for determining the best split at each node.



Perceptron = is a real mathematical model used in multiple ways & flexible by design

Note: 3 things- Change activation function, Loss function and apply **SGD** (Stochastic Gradient Descent)

If activation F = Step then used in -1 to +1 while sigmoid then 0 to 1 which is used for probability-based outcome asking in question.

When activation function = sigmoid, Loss function = Binary Cross Entropy then perceptron acts like a Logistic regression.

- Multiclass model making: Activation function is SOFTMAX, and loss function is **Categorical Cross Entropy**
- CCE is used when true labels are one-hot encoded, for example, we have the following true values for 3-class classification problem [1,0,0], [0,1,0] and [0,0,1].
- Categorical Cross Entropy is also known as SoftMax Loss. It's a SoftMax activation plus a Cross-Entropy loss used for multiclass classification. Using this loss, we can train a Convolutional Neural Network to output a probability over the N classes for each image.
- If loss function = MSE (Mean Square Error), then Linear regression == $(Y_i - \hat{Y}_i)^2$

Learn Perceptron Loss F table (4 cases):

Loss F	Activation F	Output
1.Hinge Loss	Step	Perceptron – Binary classifier (-1 or 1)
2. log 10 ss (say: Binary Cross Entropy)	Sigmoid	Logistic Regression (0 - 1) binary classifier Ques. help in probability type
3. Categorical Cross Entropy	SoftMax	SoftMax regression, multiclass classification prob. Based (0 to 1)
4.MSE (Mean Square Error)	Linear or not present Or not used	Linear regression, Output = any number

<https://www.simplilearn.com/tutorials/deep-learning-tutorial/perceptron>

Problem with Single Layer Perceptron (why not so popular)

Answer: this is proved by logic gate concept (Ankit read in Sem1 IITP) through G. Colab coding experiment, Non-linear in data.

And_Data = when all are 1 then it gives 1 otherwise 0

OR_Data = when any 1 is one from two then give 1 otherwise 0

```
# 3 dataframe hm banaye
or_data = pd.DataFrame()
and_data = pd.DataFrame()
xor_data = pd.DataFrame()
```

nice ideas of script

```
!pip install mlxtend
from mlxtend.plotting import plot_decision_regions
import pandas as pd
import numpy as np

# Assuming 'clf3' and 'xor_data' are already defined
plot_decision_regions(xor_data.iloc[:,0:2].values,xor_data.iloc[:,-1].values, clf=clf3, legend=2)
```

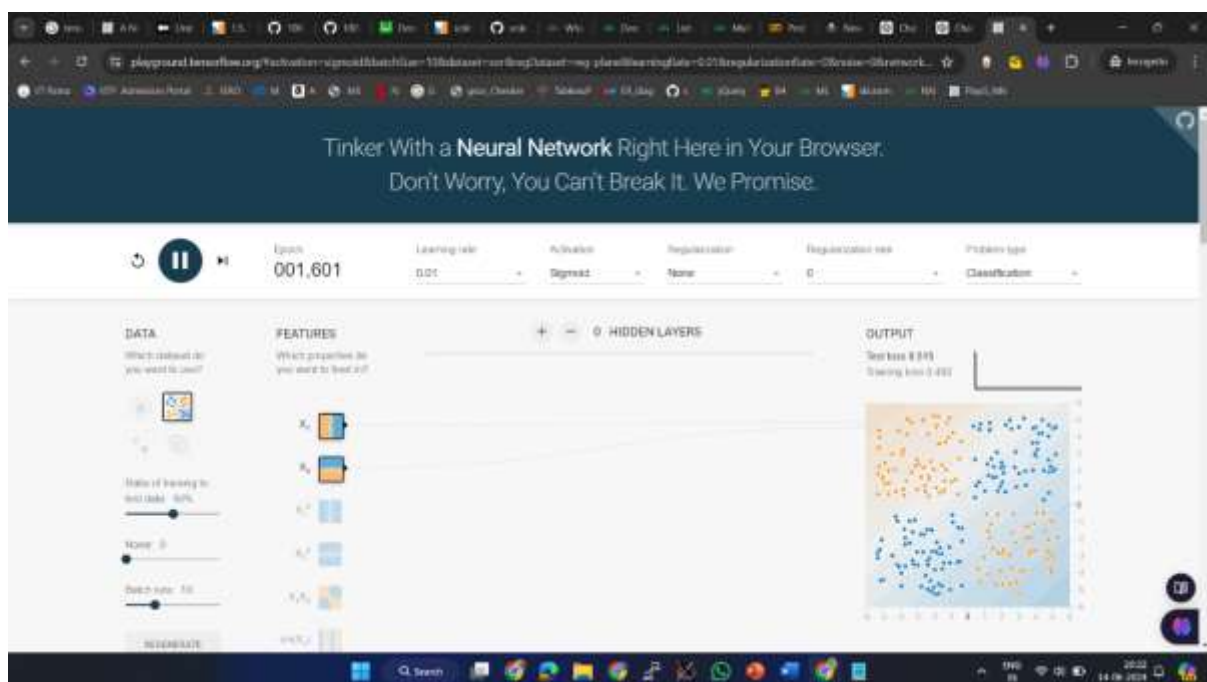
TensorFlow playground: is open source where you can run NN directly (any model like perceptron, ML model) GUI tools, create NN and run

<https://playground.tensorflow.org>

e.g. of Run Perceptron: first dataset (linearly separable), and remove hidden layer because perceptron doesn't have hidden layer like other NN, select input, Learning rate = 0.01, Activation F = Sigmoid, Problem type = Classification

<https://playground.tensorflow.org/#activation=sigmoid&batchSize=10&dataset=gauss®Dataset=reg-plane&learningRate=0.01®ularizationRate=0&noise=0&networkShape=&seed=0.05167&showTestData=false&discretize=false&percTrainData=50&x=true&y=true&xTimesY=false&xSquared=false&ySquared=false&cosX=false&sinX=false&cosY=false&sinY=false&collectStats=false&problem=classification&initZero=false&hideText=false>

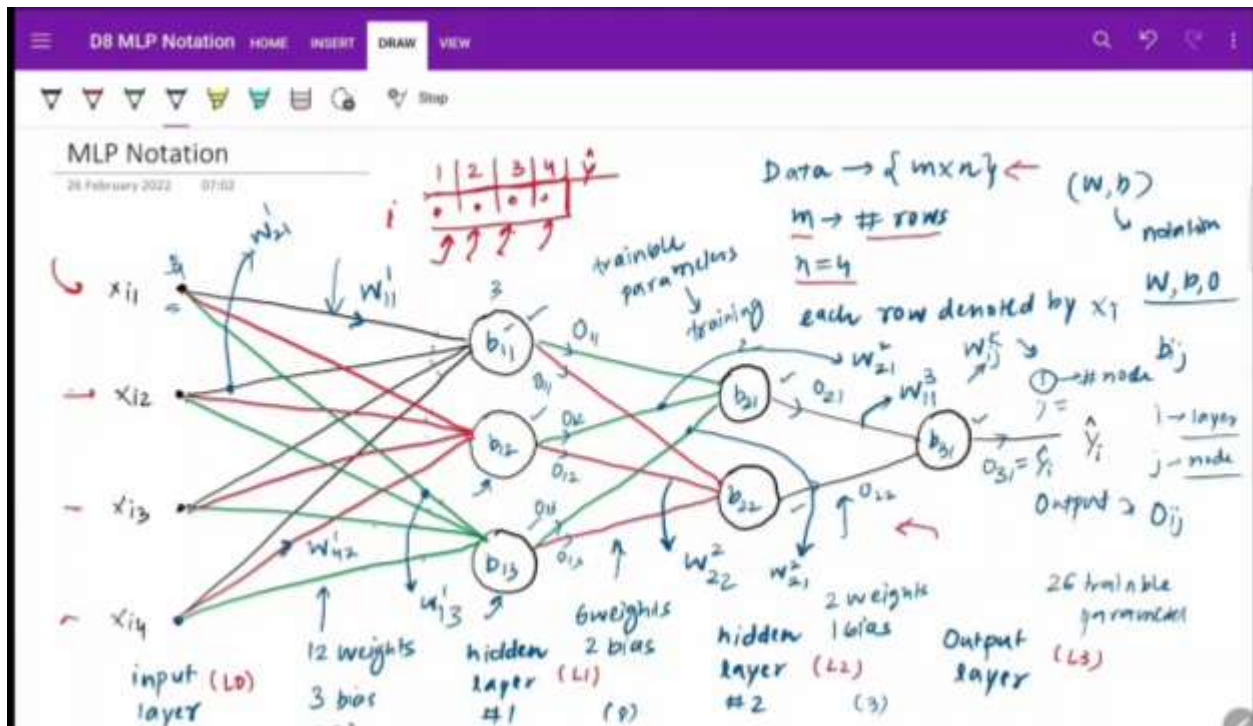
That's why We need MLP (Multi-Layer Perceptron)



15th June'24 (Saturday)

MLP (Multi-Layer Perceptron)

Learn Notation of bias, weight, and output of network of perceptron



Task: we can draw any network and practice & write BWIO in IJ form

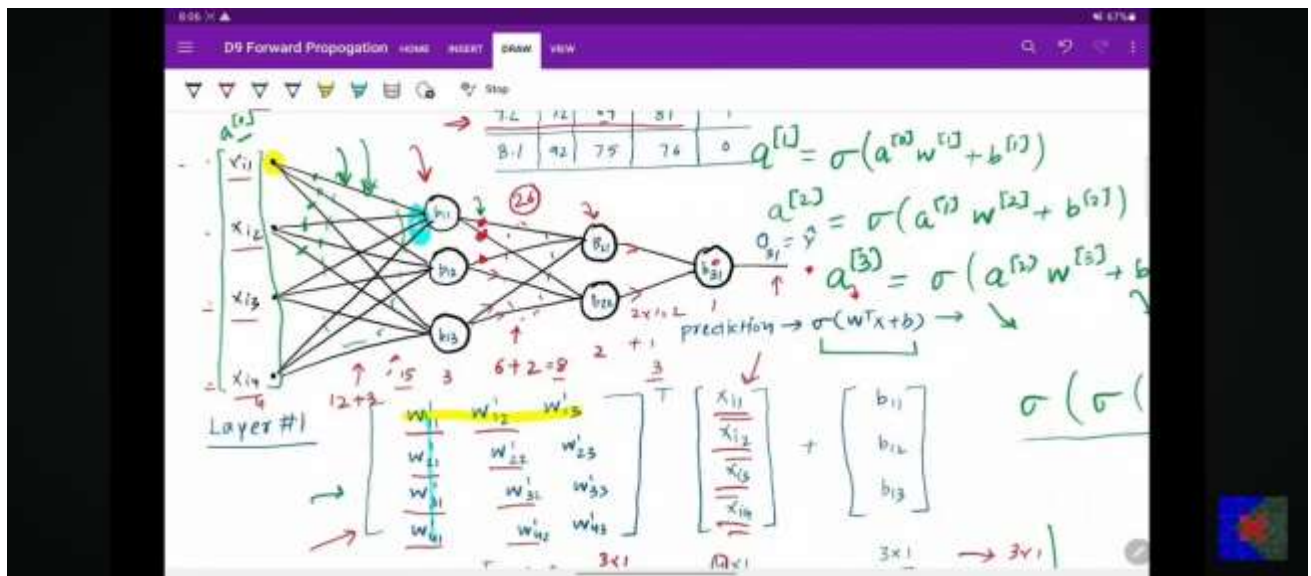
Take MLP idea graphically by **super-impose** of two graphs + decision boundary and **Smoothing**: this is a solution of limitation of SLP. We are trying to find out **linear combination of two perceptron**

Note: you can solve any problem Non linear data by giving (enough or increase Hidden Layer + Training time) known for UFA (Universal Function Approximator) of neural network [NN = UFA]

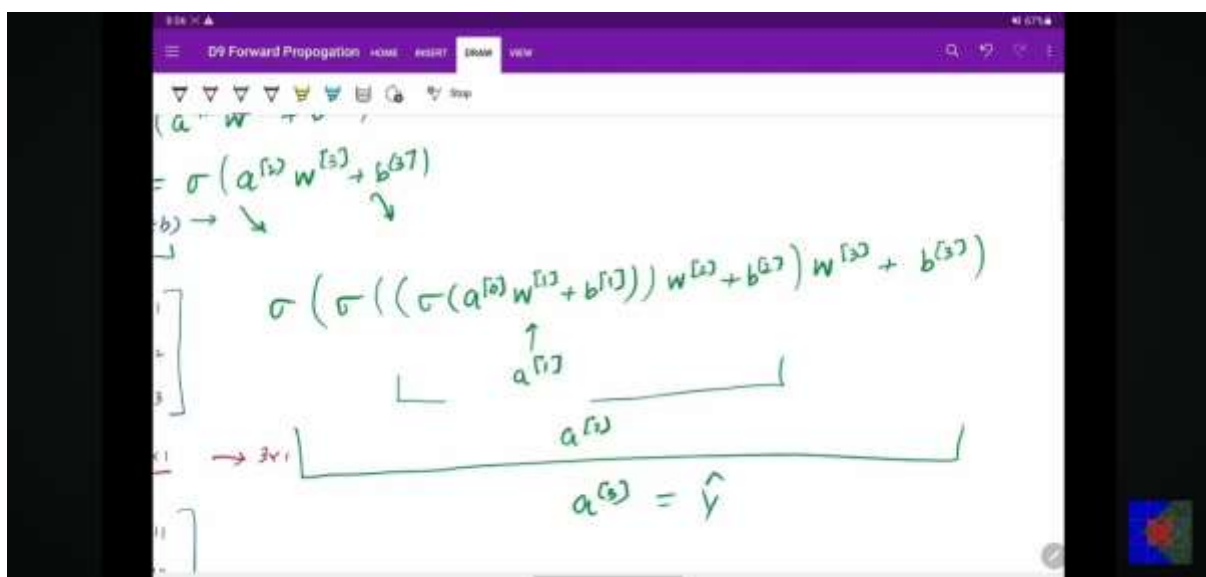
Back propagation journey start:

Observe the difference: In forward: data input moves toward forward direction, but in case of backward: the **update weights** in backward direction that's how **training occurs**.

Behind the scene PREDICATION: how NN work – Linear Algebra (dot product of matrix multiplication)
 [Ankit have learnt in Sem2 IITP maths very easy also by MatLab]



Prediction: maths behind



Very easy complex by structure – wise but become simple on applying linear algebra.

From back- propagation NN learn

The `get_dummies` function in pandas is used to convert categorical variables into dummy/indicator variables. This process is also known as one-hot encoding.

```
# change form : categorical data into dummy or indicator called one hot encoded

df = pd.get_dummies(df, columns=['Geography', 'Gender'],
drop_first=True)
```

https://pandas.pydata.org/pandas-docs/version/0.19/generated/pandas.get_dummies.html

```
# Q. cheque how many percent I train my data, and how many percent I test my data from this command

from sklearn.model_selection import train_test_split

# Split data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=0)

# Calculate the percentage of data used for training and testing
train_percent = len(X_train) / len(X) * 100
test_percent = len(X_test) / len(X) * 100

print(f"Percentage of data used for training: {train_percent:.2f}%")
print(f"Percentage of data used for testing: {test_percent:.2f}%")
```

Scaling: refers to the process of standardizing the range of features or variables in your dataset. It ensures that each feature contributes **equally to the analysis and avoids biases** that could arise from features with larger ranges dominating those with smaller ranges.

22nd June'24 (Saturday + Sunday)

```
# convert pixel value to img

import matplotlib.pyplot as plt

# command imshow se aap plot kar sakte ho pixel to img
# 1st img of x train
plt.imshow(X_train[0])
```


Multi-class classification problem i.e. MNIST dataset is given

Range value is similar so that it gives accurate value

Model creating

```
# yah liye tensorflow ka keras se Sequential

model = Sequential()

model.add(Flatten(input_shape=(28,28))) # input layer Data ka shape
bataao

# 1st DENSE layer
model.add(Dense(128,activation='relu'))

# 2nd layer = hidden layer
model.add(Dense(32,activation='relu'))

# output layer
model.add(Dense(10,activation='softmax')) # softmax use karo ge jab
bhi ek se jayada output layer ho for multiclass classification
```

Problem3: Regression

In this case, output is a number

Scaling data because of large difference, Min-Max = use when upper bound is known & Standard

Scaling i.e. Z-Score

Q. Uses of random state?

`random_state` in `train_test_split` is a tool to control the randomness in data splitting, providing reproducibility and consistency in model evaluation.

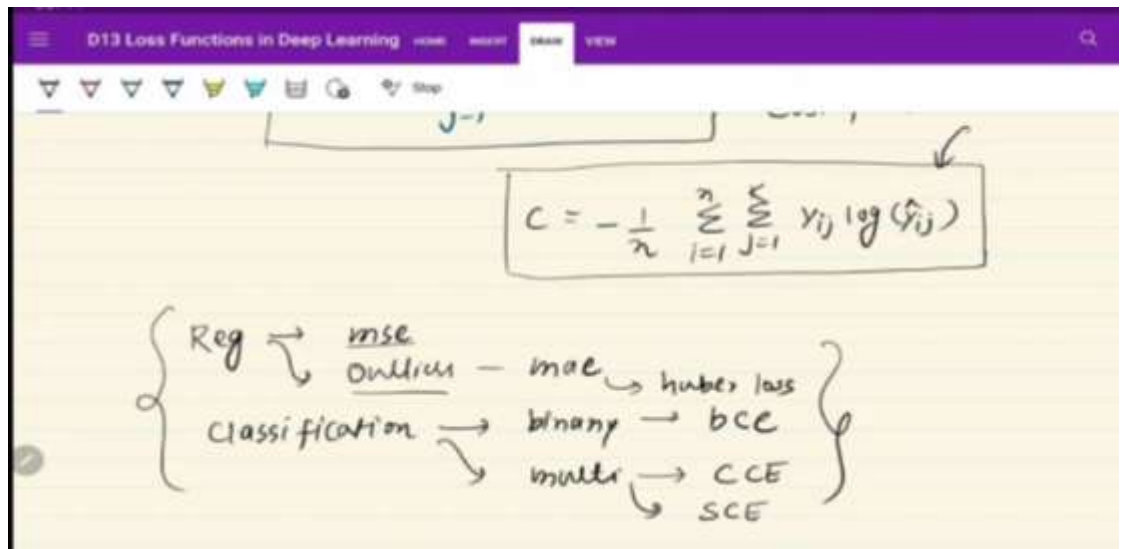
Always use Activation function = Linear in regression problems, while in multi-class classification problems, use SoftMax.

Loss Function: is a method of evaluating how well your algorithm is modelling your details.

Q. Difference b/w Loss function & Cost function?

Ans: Loss function = when calculating for single, Cost = for whole batch is also said Error function

1. Mean Squared Error (MSE) or Squared Loss
2. Mean Absolute Error (MAE)
3. Huber Loss: use for data have so many outliers
4. Binary cross entropy or log loss
5. Categorical cross entropy: used in SoftMax regression



24th June'24 (Monday) Backpropagation – concepts

Loss function: is a function of all trainable parameters.

Concept of **Gradient**: is a fancy word of derivatives or gradient descent

<https://developers.google.com/machine-learning/crash-course/fitter/graph>

change steps size and enjoys the minima graph how to reach.

Resource to see <https://developers-dot-devsite-v2-prod.appspot.com/machine-learning/crash-course/backprop-scroll>

25th June'24 (Tuesday): MLP Memorization concept in NN

Best ways: Take multiple hidden layers with fewer neurons – give more accuracy because deep Learning use the concept of Representation Learning

2. Recurrent Neural Network

Sentiment Analysis, Gated Recurrent Unit (GRU), LSTM, Bi – Directional (Emotional Detection)

Sentiment Analysis is a natural language processing task that involves determining the sentiment or opinion expressed in a piece of text and well-suited for sequence data like text, as they have the ability to capture **dependencies and patterns** in sequences.

Gated Recurrent Unit (GRU): is a method to solve the limitations of simple RNN and apart from GRN, best way is LSTM as well Bi-Directional LSTM.

7th June'24 Friday (My first ML model)

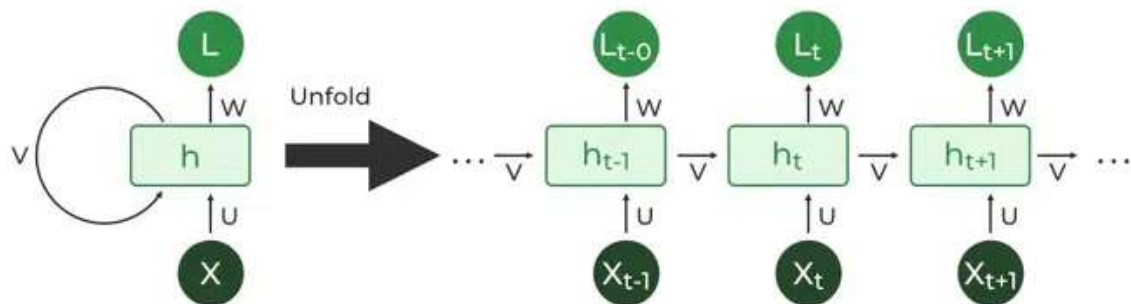
We can directly Load dataset without download and upload to google drive through raw data link GitHub or any other platform where data is publicly accessible.

First, we check the nature of dataset Ankit ways (like no. of rows, columns, size, which become X and Y == Independent & Dependent variable, just draw a rough sketch and ER diagram on <https://app.eraser.io/dashboard>)

Split arrays or matrices into random train and test subsets.

<https://towardsdatascience.com/how-to-build-a-machine-learning-model-439ab8fb3fb1>

MSE = Mean Square Error



10th June'24 (Monday)

This Neural Network is Recurrent in nature because it **generates same** type of function. It's different from the **Feed Forward NN** as there is no cycle or LOOP and storage (previous + current). This limitation is tackled by Recurrent NN.

Why to say Recurrent? Because it tries to perform same function on every input data. Here we use most of the time activation function is tanx

```
# Mnist data load from keras
(X_train, y_train), (X_test, y_test) = keras.datasets.mnist.load_data()
```

Load data from Keras modules

```
import tensorflow as tf
from tensorflow import keras
import numpy as np
import matplotlib.pyplot as plt
```

mnist.npz file format = numpy multiple array

.npy file = binary data in numpy

```
# model ka compilation
model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])
```

```
# considering hidden layer also to make new model2
# hmko hidden layer mai 300 neural need h

model2 = keras.Sequential([
keras.layers.Dense(300, input_shape=(784,), activation='relu'),
keras.layers.Dense(10, activation='sigmoid')
])
```

3. Concurrent Neural Network

Chapter 04

Natural Language Processing (NLP)

```
!sudo pip install nltk
```

This chapter is started on 9th June'24 (Sunday)

Here SUDO is stand for “**superuser do**” and command used in Unix-like operating systems to run a specific command with elevated privileges. By using SUDO before a command, the user is able to execute that command as the superuser or administrator, allowing them to perform actions that regular users do not have permission to do.

NLTK = Natural Language ToolKit

```
import nltk
nltk.download('punkt')

from nltk.tokenize import sent_tokenize, word_tokenize
```

PUNKT refers to a pre-trained model used for sentence boundary detection. In simpler terms, it helps a computer figure out where sentences begin and end within a text.

Why NLP?

Ans: Computer can understand only the structured data type, like a spreadsheet and tables But Human being can understand on an unstructured data, such as text, video, audio, etc.

To fulfil these gaps between the structure and unstructured data type we need NLP.

Application: Sentence segmentation, Word tokenization, Predicating parts of speech for each token, Lemmatization, NER (Name Entity Precogitation).

Token means **breaking - down** of word or sentence into smaller units.

References

1. Screenshot (GUIDE of ML) is taken from 5 minutes Engineering YouTube channel.
<https://www.youtube.com/@5MinutesEngineering>