PAGE NO.: 1

2/7/24

Machine Learning

- Artificial Intelligence (AI): The ability of a machine to think, reason and act like a human. (mimic a human).
  - > The term was coined by 'John McCarthy' (american) in 1956 at the Dartmouth Conference.
  - > A British mathematician and logician, 'Alan Turing' introduced the 'Turing Test', originally called 'imitation game', was proposed as a way to assess a machine's ability to exhibit intelligent behaviour indistinguisable from that of a human (when put in a dark room).

- Machine Learning (ML): A subset of AI, that allows computers to learn from data and improve their performance over time without explicit programming. (glass / white box).
  - > T: Task
    P: Performance
    E: Experience.
    then only you can say that the machine is learning.

3/7/24

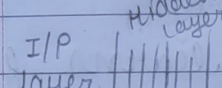
- Neural Networks (NN): They are computational models inspired by the human brain that learn from data through interconnected layers of neurons. Effective for image and speech recognition. (Artifical Neurons).
  - > Architecture of a neuron :-
    - > Synapsis - data collection
    - > Soma - able to process something
    - > Dentrites - data giving (pass the data to another neuron).
    - > Axon - a thread which carries the process data to dentrites

(Shallow learning)

* 60% of ML can be solved by : $y = mx + c$ eq
ML → { Probability / Statistics.

* Deep learning (DL) : A ML technique that uses deep neural networks with multiple layers to learn complex patterns from large datasets.

* Overfitting : Occurs when a ML model becomes excessively complex, memorizing the training data rather than learning underlying patterns. This leads to poor performance on new, unseen data. (Basically fitting only type of data, excessively).

* 

Activation function : Which fires the neurons.

* Neural Network / Deep learning → Black box testing
(Deep learning is a type of Neural Network)

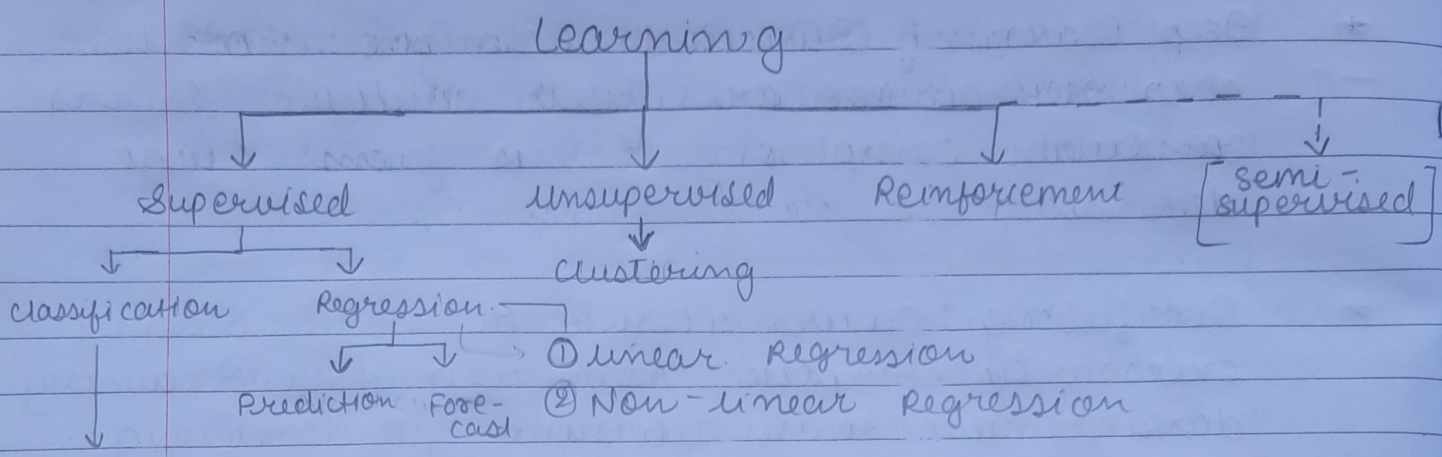
* Internal behaviour of the system is always different because no. of weights will be a learning parameter which will always change; that would be the input. And the output will be learning parameter.

Types of System learning in ML :-

* ChatGPT - Generative pre-trained transformer (2017)

* NLP - Natural language Processing.

9/7/24



- Supervised learning : A model learns from labeled data, where input data is paired with desired output. (like teacher's guidance)

- Unsupervised learning : A model learns from unlabeled data, discovering patterns and relationships within the data itself. (like learning independently)

- Reinforcement learning : A model learns by interacting with an environment, receiving rewards or penalties for its actions. (like learning through trial and error).

(optional)

- Semi-Supervised learning : A model learns from a combination of labeled and unlabeled data, leveraging both to improve its performance. (like learning with partial guidance).

16/7/24

Discriminant Functions : A discriminant functions takes an input vector 'x' and assigned it to one of 'k' classes ; where the class is denoted by C_k.

(i) Two class classification

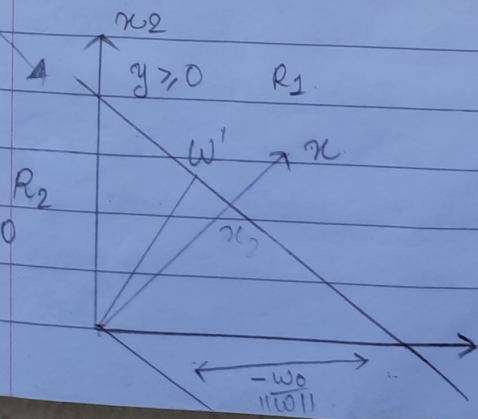
The simplest representation of a linear discriminant funn is obtained by taking a linear function of input vector 'x'.

$\longrightarrow$ learning parameters

Bias

$$\therefore Y(x) = \underbrace{W^T}_{\text{weight vector}} X + W_0 \qquad \text{——— (1)}$$

- The negative of the bias is called threshold.
- An input vector 'x' is assigned to some class C_1, if $Y(x) \geq 0$ and assigned to C_2 otherwise.
- The corresponding decision boundary is therefore defined by the relation of $Y(x) = 0$.
- $Y(x) = 0$, corresponds to 'D-1' dimensional hyperplane within D-dimensional input space

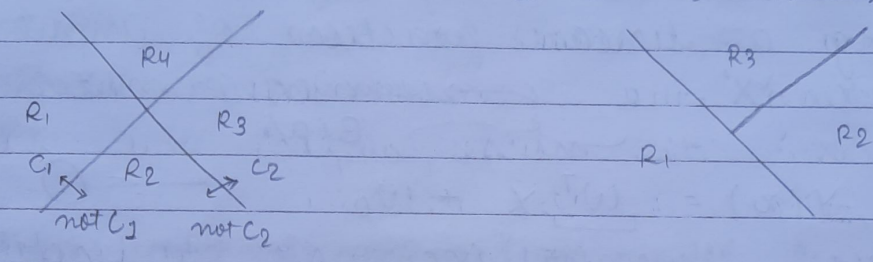


$\frac{y(x)}{\|w\|}$ $\rightarrow$ The orthogonal distance from point x to decision boundary

γ Displacement from the weight

ii) Multi-class Classification

It the extension of $K > 2$ classes. The use of $K-1$ classifier, where each of the classifier solves a two class problem of separating points in a particular class C_k from the points not available in that class is known as 'One - versus - the - rest' classifier.



Here, due to the classification arrangement we would get the fourth region and that is an unknown and un-identified region, so it is considered as to be the worst case arrangement. There are only two possible ways of this arrangement i.e. super accurate classification or problem with arrangement.

An Example involving three classes where this approach leads to a region of input space i.e. ambiguous in classified

$K(K-1)/2$

17/7/24

Dimensionality Reduction :

i) PCA (Principle Component Analysis)
ii) ICA (Independent Component Analysis)
iii) FA (Factor analysis)

The primary reasons to simplify the datasets are :-

(i) Making a dataset to use
(ii) Reducing the computation cost
(iii) Reducing the noise inside the dataset for better generalization
(iv) Making the result easier to understand.

Principle Component Analysis

A dataset is transformed from its original co-ordinate system to the new co-ordinate system.

The initial new Axis are chosen in the direction of higher variance of the data and later axis are the orthogonal projection of the initial axis.

The same procedure thats repeated for as many features as we have in the original data.

The PCA, reduces the complexity of the data and identifies the most important features which are known as Principle Component Feature.

~> Independent Component Analysis.

The ICA, assumes that the data is generated out of 'n' number of sources.

Here, the data is assumed to be the mixture of observation of the available sources. This sources are assumed to be statistically independent which represents that the data is uncorelated with each other.

If there are fewer no. of sources than the amount of observed data, then we can achieve the Dimensionality Reduction.

29/7/24

ML activities / General Process of ML :-

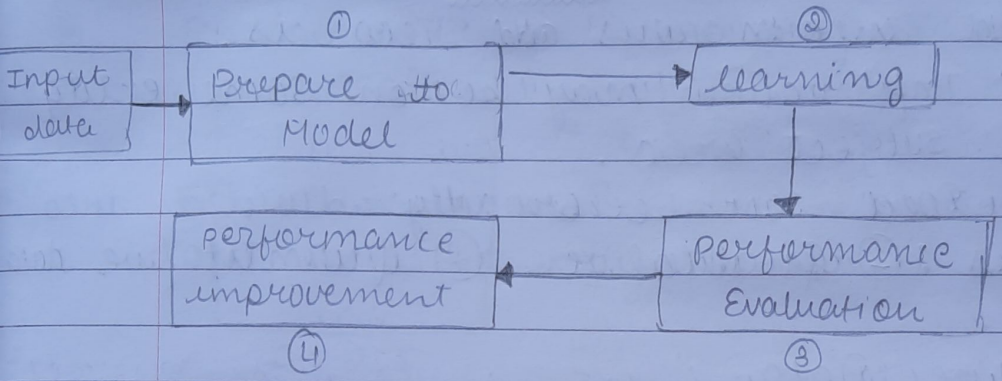
In order to develop ML system, multiple processing pipeline require. Along with some typical preparation activity, once the input data becomes available.

Few of the required activities consist of :

i) Understand the type of data in the given data set
ii) Explore the data to understand the nature, the quality and quantity.
iii) To understand inter-feature relationship
iv) Identify the potential issues in the

available dataset, if any.
→ To perform the division of the dataset

used for evaluation of the model performance
- Training Set 80%
- Validation Set (10%)
- Testing Set 20% (10%)



31/7/24

Step-1: Understand the type of data in the given dataset, which includes pre-processing part of dimensionality reduction and feature selection.

Step-2: learning. In the ML pipeline, the learning step includes the partition of the data, selection of model and cross validation.

Step-3: Performance Evaluation. Here, we conduct the evaluation of ML model with various performance matrices. Here, we should visualize the performance and the behaviour of the model.

Step-4: Performance Improvement. This include

training the model and ensembling, babbing and boosting.

Types of Data in Machine Learning :-

In ML, the dataset is a collection of related informations and records.

Information may be on some entity or some subject area.

Data can be broadly divided into two types : ① Qualitative ② Quantitative data

1. Qualitative Data.

It provides information about the quality of an object or information which cannot be measured. (eg: Good or Bad)

It is also called Categorical data, which can be further divided into two types : ① Nominal ② Ordinal.

— The Nominal Data does not contain any numerical value but a named value. This values cannot be further quantified. (eg: Nationality, blood groups, gender)

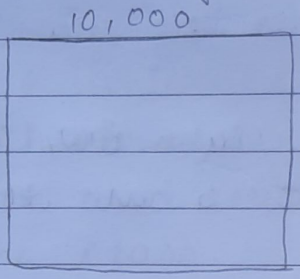
— The Ordinal Data, in addition to processing of nominal data can also be naturally ordered. (eg: Customer Satisfaction, Grades of Student, hardness of material)

* Dichotomons.

2. Quantitative Data.

It relates to information about the quantity of an object and hence it can be measured. (we can take interval & ratio as well)

* Exploring the numerical data :-



* If bunch of random quantities are given and we need to segregate the data, then we can see by taking mean, median and variance & look up on the histogram.

• Understanding the Central Tendency :-

To understand the nature of numeric variables, we can apply the measures of central tendency like mean and median.

• Understanding the data Spread :-

After central tendency of numerical attributes, we have a clear idea of which attributes have a large deviation between mean and median. Here, we can apply two approach : ① Dispersion of data ② Position of the data values.

14/8/24

SVM (Support Vector Machine) is highly accurate given the appropriate size of data and SVM can be used for numerical prediction and classification.

In SVM, the separative hyperplane can be written as,

$$W \cdot X + b = 0 \quad ;$$

W → weight vector, X → input tuples, b → bias

where W is weight vector, $W = \{w_1, w_2, \cdots w_n\}$ n is a no. of attributes and b is the scalar.

- X is input vectors, $X = \{x_1, x_2, \cdots, x_n\}$ where x is the values of the given attributes.
- Any points that lies between the separative hyperbane is denoted by,

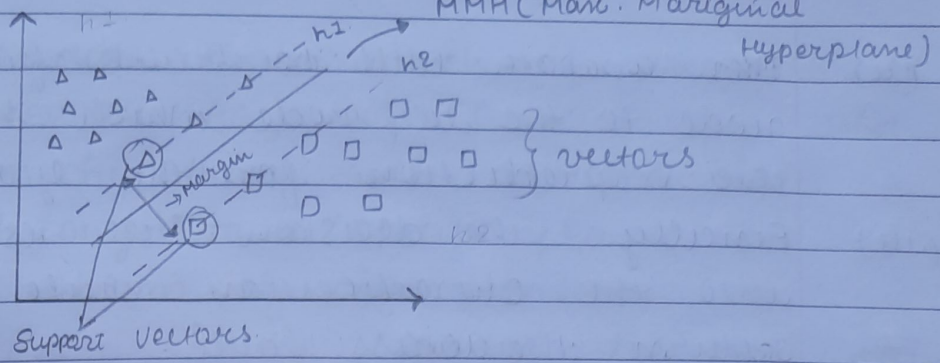
$$W \cdot X + b > 0$$

- Similarly, that lies below the hyperbane is denoted by,

$$W \cdot X + b < 0.$$

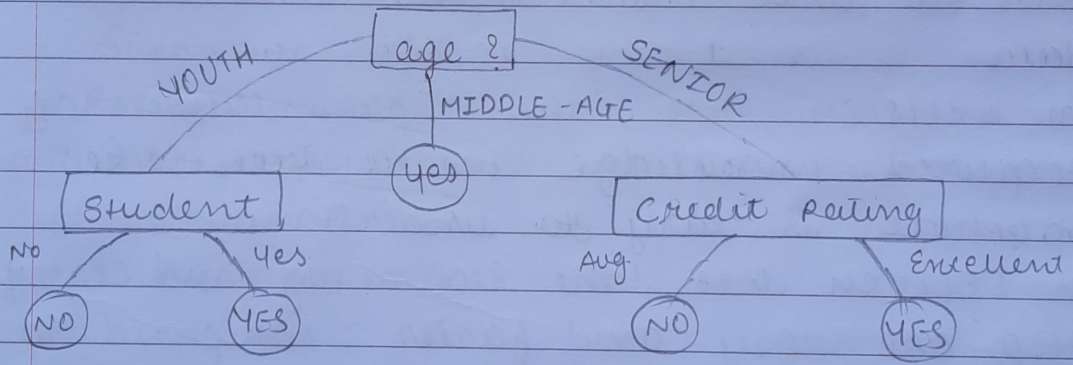
- Any training tuples that fall on the hyperplanes h_1 & h_2 are called support vectors.

The Support vectors are most difficult tuples to classify and gives most crucial information regarding classification.



* Decision Tree Classifier :- (Approach)

A Decision tree is a structure where each internal node denotes a test or attribute. Each branch represents the output of the test and each leaf node holds the label of the class.



In decision tree, the classification algorithm will generate a binary tree structure, which can lead the testing data to the appropriate class label.

—* How the decision tree classifier are used for classification &

(i) Given a tuple 'X' for which the associated class label is unknown, the attribute vector of the tuple are evaluated against the decision tree

(ii) Then a path will be traced from the root node to the leaf node which holds the class prediction for the given tuple.

(iii) Finally, the decision tree is converted into the classification rule which can return the class label.

→ Why the decision tree classifier are so popular? (Significance)

(i) The construction of the decision tree classifier does not require any domain related knowledge.

(ii) Decision tree can handle multi-dimensional data.

(iii) In Decision tree, the representation of acquired knowledge in a tree like structure is easy to understand.

(iv) In Decision tree, the learning and classifications step are easy and faster, compared to other similar techniques

(v) Decision Tree classifier, offers good accuracy

21/8/24

Bayes' Classifier :- $P(H|X)$ (Posterior ← , → Prior)
(Bayesian Theorem)

(1) They are statistical classifier

(2) They can predict class membership probabilities such as the probability that a given tuple belongs to a particular class.

(3) Let 'x' be a data tuple, which is considered as an evidence in Bayesian terminology and 'H' be some hypothesis such as the data tuple 'x' belongs to a specific class ; say 'C_1'.

(4) For the problem of classification, we want to determine; P(H|X), where the probability that hypothesis 'H' holds given the data tuple 'X'.

$$\therefore P(H|X) = \frac{P(X|H) \times P(H)}{P(X)} \quad ; \text{ Bayesian Equation.}$$

* Likelihood Calculation

SR.	Weather	Play
1.	Sunny	NO
2.	Rainy	YES
3.	Sunny	YES
4.	Overcast	YES
5.	Rainy	NO
⋮	⋮	⋮
14.		

P(P(YES)	P(P(NO)	Weather	YES	NO	
4/9	0/5	Overcast	4	0	= 4/14 = 0.29
3/9	2/5	Sunny	3	2	= 5/14 = 0.36
2/9	3/5	Rainy	2	3	= 5/14 = 0.36
		Total =	9	5	

9/14 = 0.64 5/14 = 0.36

(i) $$P(\text{YES}|\text{Overcast}) = \frac{P(\text{Overcast}|\text{YES}) \times P(\text{YES})}{P(\text{Overcast})}$$

$$= \frac{4/9 \times 9/14}{4/14} = \frac{0.44 \times 0.64}{0.29} = 0.97$$

(ii) $P(NO/Overcast) = \frac{P(Overcast/NO) \times P(NO)}{P(Overcast)} = 0.03$

* Steps of Bayes' Classification :- (Single Feature)

Step-1: Calculate the prior probability for given class label.

Step-2: Find likelihood probability with each attribute with each class.

Step-3: Calculate posterior probability, putting the values of likelihood probability.

Step-4: Check and select the class/hypothesis having highest probability.

* Steps of Bayes' Classification :- (Multiple Feature)

Step-1: Calculate prior probability for given class label.

Step-2: Calculate conditional probability with each attribute with each class.

Step-3: Multiply same class conditional probability.

Step-4: Multiply prior probability with the conditional probability of (Step-3).

Step-5: Check and select the class label or hypothesis for highest probability.

→ P(YES / W = Overcast, Temp = Mild) =
P(W = Overcast, Temp = Mild) · P(YES)
↳ ①

→ P(W = Overcast, Temp = Mild / YES) = P(Overcast / YES) · P(Mild / YES)
↳ ②

→ The Goal in classification is to take an input vector 'X' and to assign it to one of the hypothesis based on the probability calculated with an evidence.

→ In Bayes' Theorem, we are looking at the probability that a given tuple 'X' having certain no. of feature belongs to one of the 'C' classes having the prior knowledge of attribute description of given tuple 'X'.

28/8/24

Clustering :-

1. Partition Based Methods

This category of methods uses mean values of the available data sample to represent the center of the cluster.

2. Hierarchical Methods.

This category of methods propose hierarchical or Tree like structure through

decomposition or merging of data samples

3. Density Based Methods

This category of methods are applicable for identifying arbitarily shaped cluster

* K-means Clustering :-

In supervised learning, we try to match input to some existing pattern while in unsupervised learning we will try to discover the patterns in raw and unlabeled data.

Clustering is the process of grouping similar type of data by isolating these similar data.

We want the data points in a cluster with a common property that separates the data points of other clusters.

Clustering is used to identify the potential group in a dataset while classification is used to match an input to an existing group.

We can define the task of clustering as :

Input : Training Pairs

$D_{train} = [x_1, x_2, \cdots, x_n]$

Output : Assignment of each point to cluster

$Z = [z_1, z_2, \cdots, z_n]$; where $z_i \in \{1, \cdots, K\}$

The task of clustering is take a set of data points as input and return a partioning of the points into K-clusters.

Step-1: Select K-points in the data space and make them as initial centroids

Step-2: Assign each point in the data space to the nearest centroid to form K-cluster.

Step-3: Measure the distance of each point in the cluster from the centroid.

Step-4: Calculate the error (with SSE) to measure the quality of the cluster.

Step-5: Identify new centroids of each cluster on the basis of distance between the points.

Step-6: Repeat the steps ② to ⑤ until centroids needs no changes.

Probability of Machine Learning :-

In ML, we train the system by using a limited set of data called the training data. And based on the confidence level of the training data, we expect the ML algorithm to depict the behaviour of the

larger set of actual data.

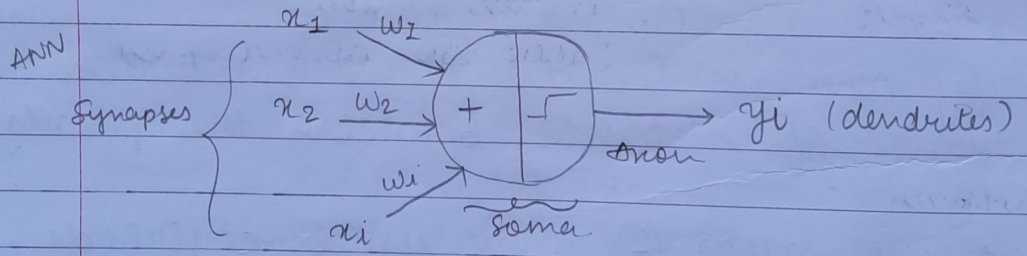
If we have a observation on a subset of events called a sample, then there will be some uncertainity in attributing the sample results to the whole set or population.

Question arises, how a limited knowledge of a sample set can be used to depict the behaviour of the real set with the same confidence?

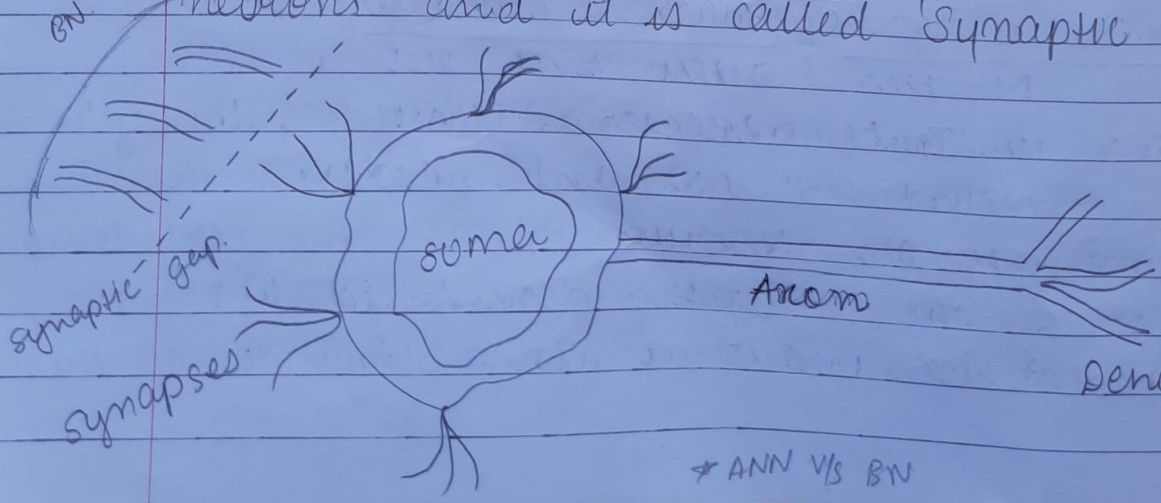
Knowledge of Sample	+	Known Uncertainity	=	useful Knowledge

18/9/24

Artificial Neural Networks :~

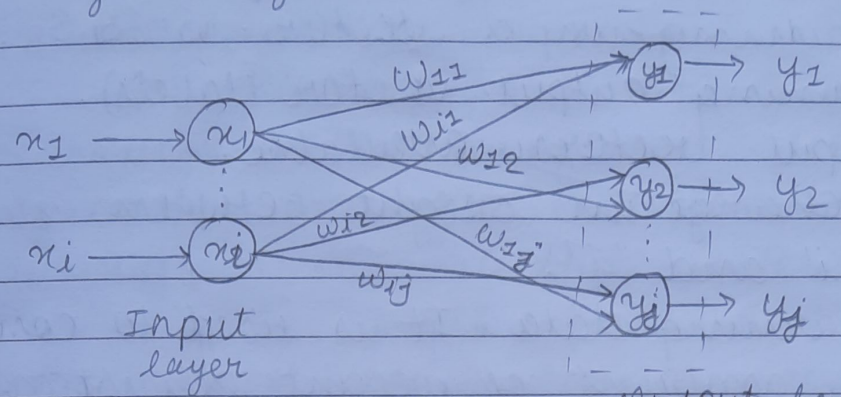


* There is always a gap between two interconnected neurons and it is called 'Synaptic Gap'.

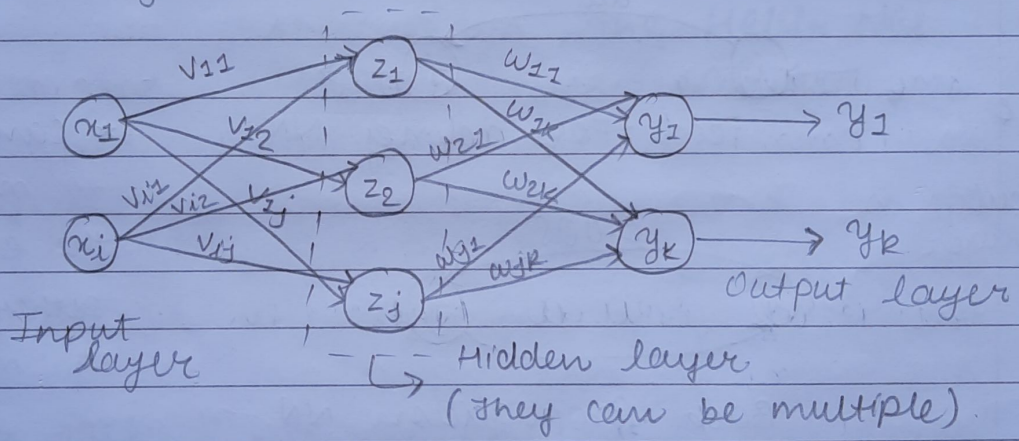


* ANN V/S BN

i) Single layer ANN



ii) Multi-layer ANN

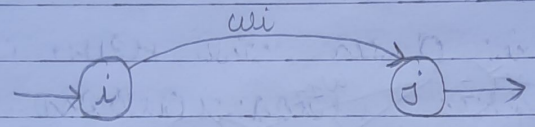


* Notations :-

- x_i → inputs signal
- y_j → output signal (produced by $f(y\text{-}in_j)$)
- w_{ij} → weight between 'i' and 'j'. Usually between hidden and output
- v_{ij} → weights between 'i' and 'j', Usually between input and hidden
- b_j → Bias on Neuron 'j'
- w → weight magnets formulated, $w = \{w_{ij}\}$
- $y\text{-}in_j$ → Net input to unit 'y_j'.

- θ_j → threshold on the unit 'j'
- S → training input vector, $S = \{S_1, S_2, \cdots, S_i\}$
- t → training output vector (labels)
- x → input vectors available.
- Δw_{ij} → change in weight between units 'i' and 'j'.
- η (eta) → learning data. It is used to control the amount of weight adjustment during each step of training.

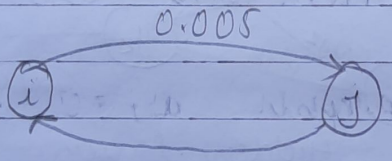
23/9/24



Time & Space complexity is much higer

Feed - Forwading NN

* Can be used to minimize the error in either single or multi-layer



Back Propogation NN

Hebb Net :- (> Hebbian learning Rule, > Extended Hebbian Rule)

The earliest and simplest learning rule for a NN is generally known as Hebb Rule.

Heb proposed that the learning occurs by modification of synaptic strength or weights.

In a manner, such that if two interconnected neurons are in activated state at a same time then the synaptic

strength between those neurons should be increased.

However, the strongest form of learning occurs if we determine the synaptic strength between neurons during their inactive state.

This result, extended hebbian rule which is a single layer feed-forward NN architecture, popularly known as Hebb Net.

In early days, the Hebb Net was used for pattern classification.

* Algorithm for Hebbian Rule :-

Step-1: Initialize all weights, $w_i = 0$ {for $i = 1$ to n}.

Step-2: For each training vector and target output pair 's:t' perform step (3) to (5)

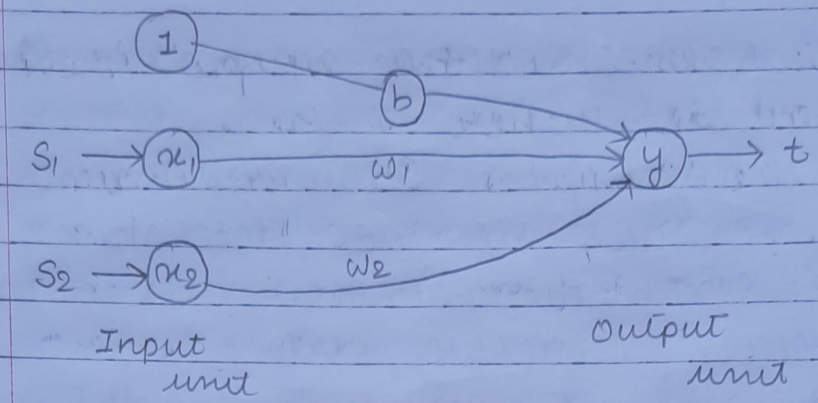
Step-3: Set activations for input unit,
$x_i = s_i$ (for $i = 1$ to n).

Step-4: Set activations for output unit,
$y = t$

Step-5: Adjust the weights,
$w_i(new) = w_i(old) + x_i y$
(for $i = 1$ to n)

Always activated; '1' ← $b_i(new) = b_i(old) + y$.

Step-6: For the stopping condition.



Perceptron :~

After the Hebb Net, the perceptron are having far reaching impact on NN community in the early days.

The perceptron learning rule is a more powerful rule compared to the Hebbian Rule. Under the specific assumption its iterative learning procedure can be proved with convergence.

The no. of different types of perceptron are proposed, since 1962, having the linear capabilities. The perceptron can be decomposed into 3 major blocks :

i) The Sensory unit
ii) The Associatory unit
iii) The Response unit

The activation fn for each associater unit is the binary step fn with the arbitary but fixed threshold 'θ'.

So, the signal sent from the

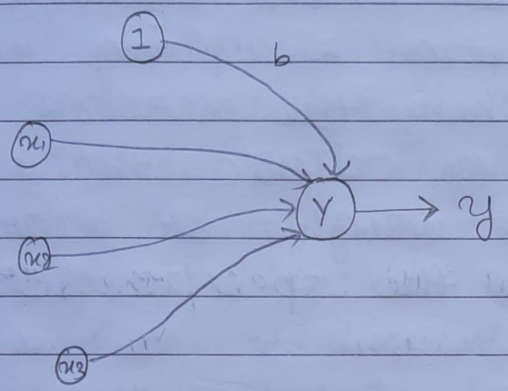
associator units to the output / response unit, will be either '1' or '0'.

The Output of the Perceptron is

$y = f(y\_in)$;

$$f(y\_in) = \begin{cases} 1 & ; \ y\_in > \theta \\ 0 & ; \ -\theta < y\_in < \theta \\ -1 & ; \ y\_in < -\theta \end{cases}$$

25/9/24



The weights from associator unit to the response unit are getting adjusted using 'perceptron learning rule'. For each training input, the net would calculate the response of the output unit. Then the net would determine whether an error occured for the given pattern or not, comparing the output of the net with the targeted value of the given input pattern.

The Perceptron Learning Rule includes an additional known as 'learning Rate' and the rule can be state as,

$\therefore w_i(New) = w_i(old) + \eta t x_i$.

If the error occured during learning a specific pattern then the

weight of the network are getting updated using the given Perceptron learning rule.

The training of the network would continue until no error remains or the possible error minimization.

The Perceptron learning rule converge only when the network meets a specific learning criteria that results the best possible weights or learning parameters, given the specific set of input sequences.

* Perceptron Algorithm :-

Step-1: Initialize the weight and bias. For simplicity set weights and bias to '0'. Set the learning rate between, ($0 < \eta \leq 1$) [we can set the learning rate to 1].

Step-2: While stopping condition is false, repeat step ③ to ⑥

Step-3: Set activations of input unit, $xi = si$

Step-4: Compute the response of output unit, $y\text{-}in = b + \Sigma xi \cdot wi$

Step-5: update the weights and bias, if the error occured for the given pattern.

If, $y \neq t$; $wi(New) = wi(old) + \eta \cdot txi$

$b(new) = b(old) + \eta \cdot t$

else ; $w_j(New) = w_j(old)$
$b(new) = b(old)$

Step-6: For the Stopping Condition.

30/9/24

Back-Propogation ANN :-

↳ Nomenclatures

① $x = (x_1, \ldots, x_i, \ldots, x_n)$

② $t = (t_1, \ldots, t_i, \ldots, t_n)$

③ $\delta_k = w_{jk} \cdot y_k$

↳ proportation of error correction weight adjustment for w_{jk} i.e. due to an error at the output unit, y_k. Also the information about the error at the unit y_k i.e propogated back to the hidden units that fit unto the y_k.

④ $\delta_j = v_{ij} \cdot z_j$

↳ Proportation for v_{ij} i.e. due to the back propogation of error information from the output layer to the hidden unit z_j.

⑤ η = learning rate

⑥ $x_i = s_i$

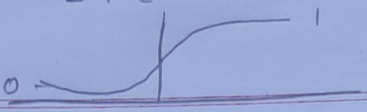
Input unit 'i' for an input unit, the input signal and the output signal are the same.

⑦ v_{oj} = Bias on hidden 'j'

Bias on the hidden unit 'j'.

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

Sigmoidal



⑧ $Z_j = Z\_in_j$

The hidden unit 'j'. The net input to 'Z_j' is denoted by 'Z_in_j' and will be calculated by,

$\therefore Z\_in_j = v_{oj} + \sum x_i . v_{ij}$

and the output signal or activation of 'Z_j' is denoted by 'z_j'. (small 'z')

$\therefore z_j = f(Z\_in_j)$

⑨ W_{ok} = Bias on output unit 'k'.

⑩ $Y_k = (Y\_in_k$

Output unit 'k'; k is no. of possible outcomes and the net input to 'Y_k' is denoted by, 'Y_in_k'.

$\therefore Y\_in_k = W_{ok} + \sum_j z_j . W_{jk}$

and

$\therefore y_k = f(Y\_in_k)$

* Back Propogation algorithm :-

Step-1: Initialize the weights and bias.

Step-2: While stopping condition is false, perform step ③ to ⑩

Step-3: Feed Forwarding — For each available training pair, perform step ④ to ⑨.

Step-4: Each input unit 'X_i' receives input signal 'x_i' and broadcast this signal to all the units in layer above ie hidden units

Step-5: Each hidden unit 'Zj' sums the weighted input signal given by,

$$\therefore z\text{-}in_j = v_{0j} + \sum_{i=1}^{n} x_i \cdot v_{ij}$$

& It applies activation fn to computes its output signal, given by

$$\therefore z_j = f(z\text{-}in_j)$$

& Sends this output signal to the unit in the layer above i.e. Output units.

Step-6: Each output unit 'Yk' sums its weighted input signal, given by,

$$\therefore y\text{-}in_k = w_{0k} + \sum_{j=1}^{m} z_j \cdot w_{jk}$$

& applies its activation fn to compute it output signal, given by,

$$\therefore y_k = f(y\text{-}in_k)$$

Step-7: Back propagation: Each output unit 'Yk' receives a target pattern for corresponding to the input training pattern. We can compute the Error Information Term by,

$$\therefore \delta_k = (t_k - y_k) \cdot f'(y\text{-}in_k)$$

To calculate the weight correction term, we have,

$$\therefore \Delta w_{jk} = \eta \cdot \delta_k \cdot z_j$$

To calculate its bias correction term, we have,

$$\therefore \Delta w_{0k} = \eta \cdot \delta_k$$

Step-8: Each hidden unit 'z_j' sums its δ inputs from given above hidden layers,

$$\therefore \delta\_in_j = \sum_{i=1}^{m} \delta_k \cdot w_{jk}$$

Multiply the quantity by the derivative of its activation fn to calculate its error information term.

$$\therefore \delta_j = \delta\_in_j \cdot f'(z\_in_j)$$

& To calculate its weight correct term, we are having,

$$\therefore \Delta v_{ij} = \eta \cdot \delta_j \cdot x_i$$

& To calculate its bias correction term, we have,

$$\therefore \Delta v_{0j} = \eta \cdot \delta_j$$

Step-9: Each output unit 'Y_k' updates its weight and bias using,

O/p unit: $\therefore w_{jk}(new) = w_{jk}(old) + \Delta w_{jk}$

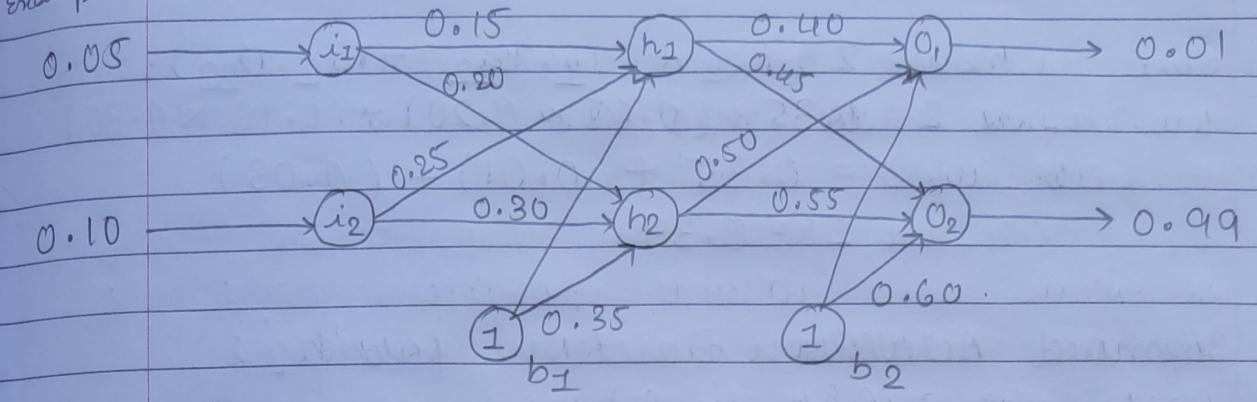
Each hidden unit 'z_j' updates its weights & bias

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

Step-10: It is for the stopping condition; it can be more than one.

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

Example:-



Given,

Inputs : $i_1 = 0.05$ $i_2 = 0.10$.

weights (input - hidden) : $v_{11} = 0.15$ $v_{21} = 0.25$

$v_{12} = 0.20$ $v_{22} = 0.30$

Biases (hidden) : $b_{h1} = 0.35$ $b_{h2} = 0.35$

weights (hidden - output) : $w_{11} = 0.40$ $w_{21} = 0.50$

$w_{12} = 0.45$ $w_{22} = 0.55$

Biases (output) : $b_{01} = 0.60$ $b_{02} = 0.60$.

Targets : $O_1 = 0.01$ $O_2 = 0.99$.

Forward Pass :-

(For input - hidden layer) $\left[z\text{-}in_j = v_{0j} + \sum_{i=1}^{n} x_i \cdot v_{ij}\right]$

$h\text{-}in\ 1 = b_{h1} + (i_1 \cdot v_{11}) + (i_2 \cdot v_{21})$

$\therefore h\text{-}in_1 = 0.35 + (0.05 \times 0.15) + (0.10 \times 0.25)$

$\therefore h\text{-}in\ 1 = 0.35 + (0.0075) + (0.025)$

$\therefore h\text{-}in\ 1 = 0.3825$.

Sigmoid Activation Function for 'h_1' : $[z_j = f(z\text{-}in_j)$

$h_1 = f(h_1\text{-}in\ 1)$

$\therefore h_1 = \frac{1}{1+e^{-(0.3825)}} \approx \underline{0.5944759307}$

> $h\_in\ 2 = b_{h2} + (i_1 \cdot v_{12}) + (i_2 \cdot v_{22})$

$\therefore h\_in\ 2 = 0.35 + (0.05 \times 0.20) + (0.10 \times 0.30)$

$\therefore h\_in\ 2 = 0.35 + (0.01) + (0.03)$

$\therefore h\_in\ 2 = 0.39$

→ Sigmoid Activation Function for 'h_2':

$h_2 = f(h\_in\ 2)$

$\therefore h_2 = \frac{1}{1 + e^{-(0.39)}} \approx \underline{0.5962826992}$

(For hidden - output layer)

> $O\_in\ 1 = b_{O1} + (h_1 \cdot w_{11}) + (h_2 \cdot w_{21})$

$\therefore O\_in\ 1 = 0.60 + (0.5944759307 \times 0.40) + (0.5962826992 \times 0.50)$

$\therefore O\_in\ 1 = 1.185931722188$

> $O\_in\ 2 = b_{O2} + (h_1 \cdot w_{12}) + (h_2 \cdot w_{22})$

$\therefore O\_in\ 2 = 0.60 + (0.5944759307 \times 0.45) + (0.5962826992 \times 0.55)$

$\therefore O\_in\ 2 = 1.195469653375$

→ Sigmoid Activation Function for 'O_1':

$O_1 = f(O\_in\ 1)$

$\therefore O_1 = \frac{1}{1 + e^{-(1.185931722188)}} \approx \underline{0.7569319154}$

→ Sigmoid Activation Function for 'O_2':

$O_2 = f(O\_in\ 2)$

$\therefore O_2 = \frac{1}{1 + e^{-(1.195469653375)}} \approx \underline{0.7677178797}$

16/10/24

Convolutional Neural Network :~ (Feed Forward Archit ecture)

CNN is a type of feed forward NN architecture. Usually applicable for image classification.

It is also known as ConvNet.

The convolutional operation forms the basis of CNN.

In convolutional operation, the arrays are multiplied by element wise and the product is sum together to create a new feature array.

a = [5, 3, 2, 5, 9, 7] b = [1, 2, 3]

a * b → Elementwise Multiplication → Sum of the product

[17, 22, 39, 44]

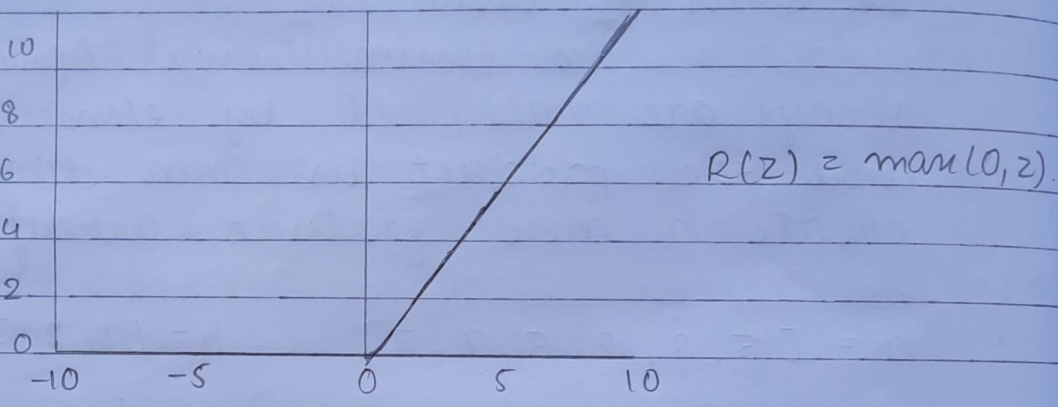
There are 7 possible layers in CNN generally, which are:-

1. Convolutional layer
2. ReLU
3. Pooling Layer
4. Fully connected layer
5. Activation layer
6. Flattening layer
7. Output layer

The ReLU stands for rectif

Once we have the feature maps extracted from the input then this vector is forwarded to ReLU layer.

The ReLU is responsible to perform elementwise operation and sets all the negative pixel to 0. Hence, the ReLU introduces non-linearity to the



the original input image is scanned with multiple convolutionals and ReLU layers for locating larger set of feature.

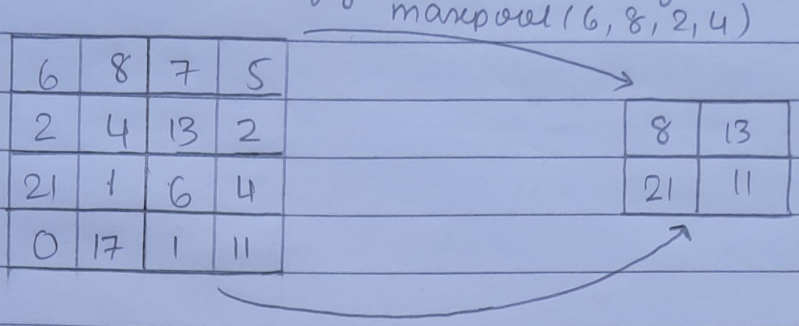
→ the pooling layer

The Pooling is down sampling operation that reduces the dimensionality of the feature map The Pooling layer uses variety of layers to filter the variety of features.

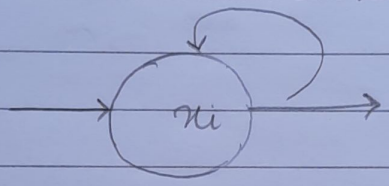
After the pooling operation, we apply the flatening to convert the resultant 2-D array into single linear vector.

The Flatened vector is further fed as an input to fully connected

network to classify the image.



Recurrent Neural Network :-



LSTM
↓
Recurrent cell
↓
LSTM cell
↓
RNN
↓
seq-to-seq learning
↓
seq-to-seq with Attention → Global, local.
↓
Transformer