UNIT-1

14-11-2023

Linear Algebra

Scalars, Vectors, Matrices and Tensors, Matrix operations, Types of matrices, Norms, Eigen decomposition, Singular Value Decomposition, Principal Component Analysis

---

Introduction to Deep learning :-

Deep learning is a branch of machine learning which is based on ANN-Artificial Neural Networks.

It is capable of learning complex patterns and relationships within data. In DL, we don't need to explicitly program everything.

ANN also known as DNN-Deep Neural Networks. These neural networks are inspired by the structure and function of the human brains biological neurons, and they are designed to learn from large amounts of data.

The key characteristic of DL is the use of DNN, which have multiple layers of interconnected nodes.

These networks can learn complex representations of data by discovering hierarchical patterns and features in the data.

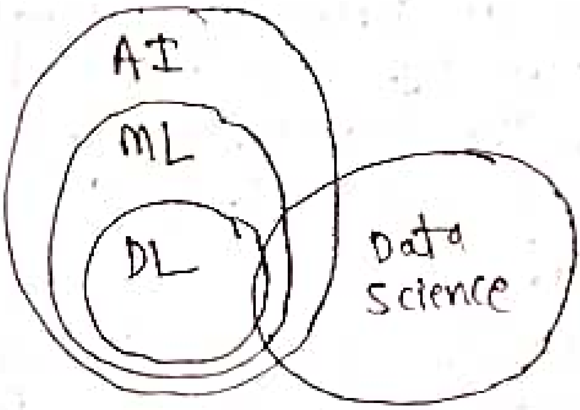
Deep learning algorithms can automatically learn and improve from data without the need for manual feature engineering.

Some of the popular DL architectures include CNN-Convolutional NN, RNN-Recurrent NN, and DBN-Deep Belief Networks.

What is DL:

DL is a branch of ML, which is based on ANN architecture. An ANN uses layers of interconnected nodes called neurons that work together to process and learn from the input data.

In fully connected DNN, there is an input layer and one or more hidden layers connected one after the other.
Each neuron receives input from the previous layer neurons or the i/p layer.
The o/p of one neuron becomes the i/p to other neurons in the next layer of the network, and this process continues until the final layer produces the o/p of the network.



Deep learning can be used for supervised, unsupervised as well as reinforcement ML algorithms.

Supervised Machine Learning:

SML is the ML technique in which the neural network learns to make predictions or classify data based on labeled datasets.
Here, we have input features along with target variables.

The neural network learns to make predictions based on the cost (or error) that comes from the difference b/w the predicted and the actual target, this process is known as back propagation.

DL algorithms like CNN, RNN are used for many supervised tasks like image classification and image recognition, sentiment analysis, and language translation, etc.

Unsupervised Machine Learning:

USML is the ML technique in which the neural network learns to discover the patterns to to cluster the dataset based on unlabeled datasets.

DL algorithms like autoencoders and generative models are used ~~to~~ for US tasks like clustering, dimensionality reduction, and anomaly detection.

Reinforcement ML:

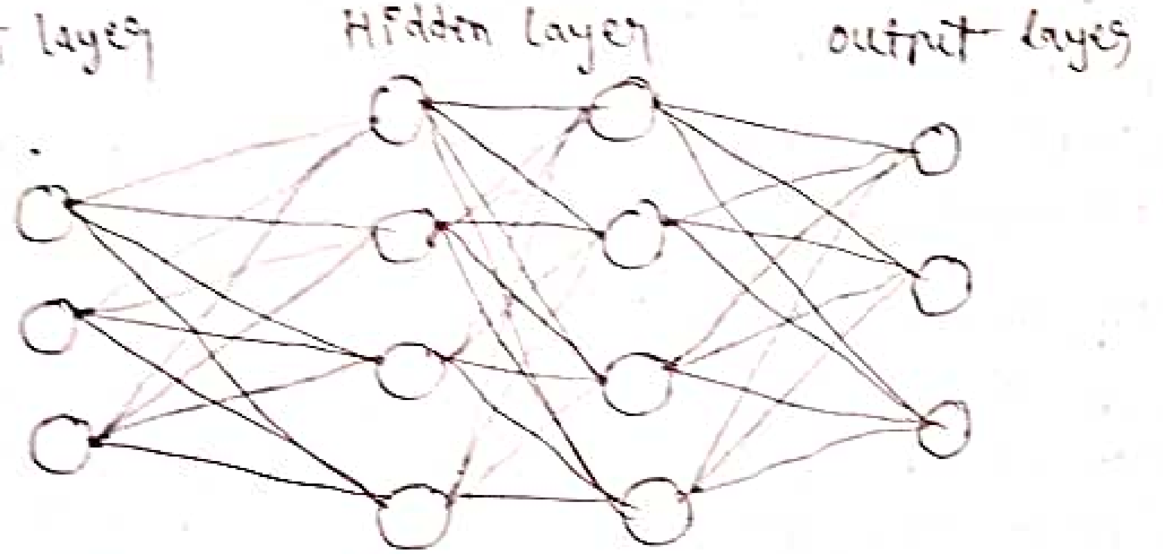
Reinforcement ML is the ML technique in which an agent learns to make decisions in an environment to maximize a reward signal.

DL can be used to learn policies, (or a set of actions, that maximizes the cumulative reward over time.

Deep reinforcement algorithms like Deep Q networks and Deep Deterministic Policy Gradient (DDPG) are used to reinforce tasks like robotics and game playing etc.

Artificial Neural Networks:-

ANN are built on the principles of the structure and operation of human neurons. It is also known as neural networks (or) neural nets.



Fully connected ANN

Artificial neurons, also known as units.

The complexities of neural networks will depend on the complexities of the underlying patterns in the dataset whether a layer has a dozen units (or) millions of units.

Each neuron receives input from the previous layer neurons (or) the input layer.

The o/p of one neuron becomes the i/p to other neurons in the next layer of the network, and this process continues until the final layer produces the o/p of the network.

Difference b/w ML and DL:-

ML and DL both are subsets of AI.

ML	DL
→ Can work on the smaller amount of dataset.	→ Requires the larger volume of dataset compared to ML.

→ Apply statistical algorithms to learn the hidden patterns and relationships in the dataset.	→ Uses ANNs architecture to learn the hidden patterns and relationships in the dataset.
→ Better for the low-label task	→ Better for complex task like image processing, NLP, etc.
→ Takes less time to train the model.	→ Takes more time to train the model.
→ A model is created by relevant features which are manually extracted from images	→ Relevant features are automatically extracted from images.
→ Less complex and easy to interpret the result.	→ More complex, it works like the black box interpretation.
→ It can work on the CPU (or) requires less computing power.	→ It requires high performance computer with GPU.

Types of Neural Networks:-

Deep learning models are able to automatically learn features from the data.

The most widely used architectures in DL are FNNs, CNNs, and RNNs.

FNN - Feed Forward Neural Networks:-

FNNs are the simplest type of ANN, with a linear flow of information through the network.

FNNs have been widely used for tasks such as image classification, speech recognition, and NLP.

CNNs – Convolutional Neural Networks:-

CNNs are specifically for image and video recognition tasks. CNNs are able to automatically learn features from the images.

CNNs used for tasks such as image classification, object detection, and image segmentation.

RNNs – Recurrent Neural Networks:

RNNs are a type of neural network that is able to process sequential data, such as time series and natural language.

RNNs used for tasks such as speech recognition, NLP, and language translation.

Applications of DL:

The main applications of DL can be divided into computer vision, NLP, and RL.

Computer Vision:

In computer vision, DL models can enable machines to identify and understand visual data.

→ **Object detection and recognition:**

Deep learning model can be used to identify and locate objects within images and videos.

Eg:- self-driving cars, surveillance, and robotics.

→ **Image classification:**

DL models can be used to classify images into categories such as animals, plants, and buildings.

Eg:- medical imaging, Quality control, and image retrieval.

→ **Image Segmentation:-**

DL models can be used for image segmentation into different regions, making it possible to identify specific features within images.

NLP:

In NLP, the DL model can enable machines to understand and generate human language.

→ Automatic Text Generation:

DL model can learn text like summaries, essays can be automatically generated using these trained models.

→ Language Translation:

DL models can translate text from one language to another language.

→ Sentiment Analysis:

DL models can analyze the sentiment of a piece of text, making it possible to determine whether the text is +ve, -ve, or neutral.

→ Speech Recognition:

DL models can recognize spoken words, making it possible to perform tasks such as speech-to-text conversion, voice search, and voice-controlled devices.

RL:

In RL, DL works as training agents to take action in an environment to maximize a reward.

→ Game playing:

Deep RL models have been able to beat human experts at games such as Go, Chess, and Atari.

→ Robotics:

Deep RL models can be used to train robots to perform complex tasks such as grasping objects, navigation, and manipulation.

Control Systems:

Deep RL models can be used to control complex systems such as power grids, traffic management, and supply chain optimization.

Challenges in DL:-

→ Data availability.
→ Computational Resources.
→ Time consuming.
→ Interpretability.
→ Overfitting.

Advantages:

→ High accuracy
→ Automated feature engineering.
→ Scalability.
→ Flexibility
→ Continual Improvement.

Scalars:-

A scalar is just a single number.
Scalar can be written in italic.
We usally give scalars lower-case variable names.

$s \in \mathbb{R}$ be the slope of the line.
$n \in \mathbb{N}$ be the number of units.

Vectors:-

A vector is an array of numbers. The numbers are arranged in order. We can identify each individual number by its index in that ordering.

We give vectors lower-case names written in bold typeface, such as $\mathbf{x}$.

The elements of the vector are identified by writing its name in italic typeface, with a subscript.

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

We use the "−" sign to index the complement of a set. For example x_{-1} is the vector containing all elements of x except for x_1.

Matrices:-

A matrix is a 2-D array of numbers, so each element is identified by two indices.

We usually give matrices upper-case variable names with bold type face, such as A. (italic)

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \quad A^T = \begin{bmatrix} A_{11} & A_{21} \\ A_{12} & A_{22} \end{bmatrix}$$

Tensors:-

An array of numbers arranged on a regular grid with a variable number of axes is known as "tensor".

We denote a tensor with type face : $\mathbf{A}$.

We identify the element of A at coordinates (i,j,k).

The transpose of a matrix is the mirror image of the matrix across a diagonal line, called the main diagonal.

ex

$$A_{ij} = (A^T)_{ji}$$

Vectors can be thought of as matrices that contain only one column.

$$x = [x_1, x_2, x_3 \cdots, x_n]^T$$

A scalar can be thought of as a matrix with only a single entry.

$$a = a^T$$

1	$\begin{bmatrix} 1 \\ 2 \end{bmatrix}$	$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$	$\begin{bmatrix} [1 & 2] & [3 & 2] \\ [1 & 7] & [5 & 4] \end{bmatrix}$
Scalar	Vector	Matrix	Tensor

Scalar → 0^{th} order tensor
Vector → 1^{st} order tensor
Matrix → 2^{nd} order tensor
Tensor → 3^{rd} order tensor.

Linear Algebra :-

Linear algebra is a branch of mathematics that is widely used throughout science and engineering.

Linear algebra is a form of continuous rather than discrete mathematics, many computer scientists have little experience with it.

A good understanding of linear algebra is essential for understanding and working with ML algorithms, especially DL algorithms.

Matrix Operations:-

One of the most important operations involving matrices is multiplication of two matrices.

$$A_{m\times n}\ ,\ B_{n\times p}$$

$$\begin{array}{r} m\times n \\ n\times p \\ \hline m\times p \end{array}$$

$$A\cdot B = C_{m\times p}$$

$$C = AB$$

$$C_{ij} = \sum_{K} A_{i,K} B_{K,j}$$

Note that the standard product of two matrices is not just a matrix containing the product of the individual elements. Such an operation exists and is called the element-wise product (or Hadamard product), and is denoted as $A \odot B$

Matrix multiplication is distributive

$$A(B+C) = AB + AC$$

It is also associative

$$A(BC) = (AB)C$$

Matrix multiplication is not commutative. $AB \neq BA$.

$$x^T y = y^T x$$

$$(AB)^T = B^T A^T$$

$$x^T y = (x^T y)^T = y^T x$$

Identity and Inverse Matrices:

An identity matrix is a matrix that does not change any vector when we multiply that vector by that matrix. Identity matrix (I_n).

$$A_{ij} = 0, \text{ when } i \neq j$$

$$A_{ij} = 1, \text{ when } i = j$$

All of the entries along the main diagonal are 1, while all of the other entries are zero.

The matrix inverse of A is denoted as A^{-1}.

$$A^{-1} A = I_n$$

$$Ax = b \Rightarrow x = A^{-1} b.$$

$$A^{-1} = \frac{1}{|A|} \text{adj} A.$$

Matrix Addition & Subtraction:

The shapes of the matrices are same the addition and subtraction is possible.

Shape and Size:

Shape, the number of rows and columns in the given matrix.

Size, The number of elements in the matrix.

Ex: $\begin{bmatrix} 1 & 2 & 3 \end{bmatrix}_{1 \times 3}$ = shape

Size = 3

Dense and Sparse Matrices:

A sparse matrix is a matrix that consists of mostly zero values. And sparse matrices are different from matrices with mostly non-zero values, which are known as dense matrices.

Mean, Variance, and SD:-

$$mat = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

~~Mean~~

```
np.mean(mat)
np.var(mat)
np.std(mat)
np.sqrt(np.var(mat))
```

Trace of a Matrix:

The sum of all diagonal elements in the matrix.

$$\text{Trace}(mat) = 1+4 = 5$$

Min and Max elements of a matrix:

The elements with the highest and lowest value among all the elements.

$$\min(mat) = 1$$
$$\max(mat) = 4$$

Determinant:

$$|A| = \begin{vmatrix} a & b \\ c & d \end{vmatrix} = ad - bc$$

Reshape:

Changing the shape of the given matrix.

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}_{2\times 3} \quad \text{reshaping into} \quad \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}_{3\times 2}$$

Types of Matrices:-

A matrix having m rows and n columns is given by $A_{m \times n}$

Each member of the matrix is called an element of the matrix.

Order of a Matrix:

If a matrix has m rows and n columns, then it is said to be a matrix of orderm mn.

Singleton Matrix:

A matrix having only one element is known as a singleton matrix.

Ex:- $[1]$, $A = [a_{ij}]_{m \times n}$ if, $i = j = 1$

Row Matrix:

A matrix having only one row is called a row matrix.

Ex:- $\begin{bmatrix} 1 & 2 & 3 & 4 \end{bmatrix}_{1 \times 4}$

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \end{bmatrix}_{1 \times n}$$

Column Matrix:

A matrix having only one column is called a column matrix.

Ex:- $\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}_{3 \times 1}$ $\quad A = \begin{bmatrix} a_{11} \\ a_{21} \\ a_{31} \\ \vdots \\ a_{n1} \end{bmatrix}_{n \times 1}$

Null Matrix | Zero Matrix:

If all the entries of a matrix are zero, it is called a null matrix.

$$A = [a_{ij}]_{m \times n} \quad , \text{ if } a_{ij} = 0$$

Ex: $\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$

Square Matrix:

Any matrix is said to be a square matrix, if it has the same number of rows and a same number of columns.

$$A = [a_{ij}]_{m \times n} \quad , \text{ if } m = n$$

Ex: $A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$

Note: A vector of values from the diagonal of the matrix from top to the bottom right is known as a principal diagonal.

Diagonal Matrix:

In a square matrix, all the elements of a principal diagonal are non-zero, and all the other entries are zero, then it is called a diagonal matrix.

$$A = [a_{ij}]_{m \times n}, \quad a_{ij} = 0, \text{ if } i \neq j$$

Ex: $A = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}$

Note: Null matrix is also a diagonal matrix.

Scalar Matrix:

It is a particular case of diagonal matrix in which all the elements of the diagonal are identical.

Ex: $A = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$ $\quad a_{ij} = 0, \text{ if } i \neq j$

$a_{ij} = a, \text{ if } i = j$

Identity Matrix (Unit Matrix):

It is a particular case of the scalar matrix. If all the non-zero elements of the scalar matrix are equal to 1, then it is termed as identity matrix.

Ex: $A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = I_2$, $\quad a_{ij} = 0, \text{ if } i \neq j$

$a_{ij} = 1, \text{ if } i = j$

Triangular Matrix:

A square matrix that has all the values in the upper-right or lower left and the remaining values are filled with zero, then such matrices are called Triangular matrix.

These are 2 types.

→ Upper Triangular matrix
→ Lower Triangular matrix

Note: zero/null matrix is also a type of triangular matrix.

Upper Triangular Matrix:

A square matrix in which all the elements below the principal diagonal are zero is known as a upper triangular matrix.

$A = [a_{ij}]_{mn}$, if $a_{ij} = 0$, when $i > j$

Ex: $\begin{bmatrix} 1 & 2 & 3 \\ 0 & 4 & 5 \\ 0 & 0 & 6 \end{bmatrix}$

Lower Triangular Matrix:

A square matrix in which all the elements above the principal diagonal are zero is known as lower triangular matrix.

$A = [a_{ij}]_{mn}$, if $a_{ij} = 0$, when $i < j$.

Ex: $A = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 3 & 0 \\ 4 & 5 & 6 \end{bmatrix}$

Symmetric Matrix:

A square matrix that is equal to its transpose is called a symmetric matrix.

$A = A^T$ then $a_{ij} = a_{ji}$

Ex: $\begin{bmatrix} 1 & 2 \\ 2 & 0 \end{bmatrix}$, $\begin{bmatrix} 1 & 2 & 3 \\ 2 & 9 & 4 \\ 3 & 4 & 7 \end{bmatrix}$

Skew-Symmetric Matrix:

A square matrix is equal to its - of transpose is called a skew-symmetric matrix.

$A = -A^T$, $a_{ij} = 0$, for $i = j$

Ex: $A = \begin{bmatrix} 0 & 2 & -1 \\ -2 & 0 & -4 \\ 1 & 4 & 0 \end{bmatrix}$

Idempotent Matrix:

A square matrix A is said to be Idempotent, if

$A^2 = A$

Ex: $A = \begin{bmatrix} 2 & -2 & -4 \\ -1 & 3 & 4 \\ 1 & -2 & -3 \end{bmatrix}$

Singular matrix $\Rightarrow |A| = 0$

Non-singular $\Rightarrow |A| \neq 0$

Hermitian Matrix $\Rightarrow A = A^\theta$

Skew-Hermitian $\Rightarrow A^\theta = -A$

Nilpotent $\Rightarrow A^n = 0$

Involutory Matrix:

A square matrix A is said to be involutory, if

$A^2 = I$

Ex: $A = \begin{bmatrix} -5 & -8 & 0 \\ 3 & 5 & 0 \\ 1 & 2 & -1 \end{bmatrix}$

Orthogonal Matrix:

A square matrix A is said to be an orthogonal matrix if it satisfies, $A \cdot A^T = I = A^T A$

$A^{-1} = A^T$ //

Ex: $A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

~~Horizon~~ Horizontal Matrix:

A matrix of order $m \times n$ is a horizontal matrix, if $n > m$.

Ex: $\begin{bmatrix} 1 & 2 & 3 & 4 \\ 2 & 5 & 1 & 1 \end{bmatrix}$

Vertical Matrix:

A matrix of order $m \times n$ is a vertical matrix, if $m > n$

Ex: $\begin{bmatrix} 2 & 5 \\ 1 & 1 \\ 3 & 6 \\ 2 & 4 \end{bmatrix}$

Periodic Matrix:-

A square matrix which satisfies the relation $A^{K+1} = A$, for some +ve integer K, then A is periodic with period K.

Ex $\begin{bmatrix} 2 & -3 & -5 \\ -1 & 4 & 5 \\ 1 & -3 & -4 \end{bmatrix}$ has period 1.

Period of an idempotent matrix is 1.

Norms:-

In ML, we usually measure the size of vectors using a function called a norm.

The L^p norm is given by

$$\|x\|_p = \left(\sum_i |x_i|^p\right)^{\frac{1}{p}}$$

The L^2 norm, with $p=2$, is known as the Euclidean norm. It is simply the Euclidean distance from the origin to the point.

The L^1 norm is $\|x\|_1 = \sum_i |x_i|$

The L^1 norm is commonly used in ML when the difference b/w zero and non-zero elements is very important.

One other norm that commonly arises in ML is the L^∞ norm, also known as the max norm.

$$\|x\|_\infty = \max |x_i|$$

Frobenius norm, $||A||_F = \sqrt{\sum_{i,j} A_{i,j}^2}$ for matrix

The dot product of two vectors can be rewritten in terms of norms.

$$x^T y = ||x||_2 \, ||y||_2 \cos\theta .$$

where, θ is the angle b/w x and y.

A unit vector is a vector with unit norm: $||x||_2 = 1$

Vector Norm:

The length of a vector is a non-negative number that describes the extent of the vector in space, and is sometimes referred to as the vector's magnitude (or) the norm.

Vector L1 norm:

The notation for the L1 norm of a vector is $||v||_1$ and this type of norm is also referred to as Manhattan Norm (distance).

$$||v||_1 = |a_1| + |a_2| + |a_3|$$

Vector L2 Norm:

The notation for the L2 norm of a vector is $||v||_2$. and this type of norm is also known as Euclidean Norm.

$$||v_2|| = \sqrt{(b_1)^2 + (b_2)^2 + (b_3)^2}$$

Vector Max Norm:

The notation is $||v||_\infty$

The max norm is calculated as returning the max value of the vector.

$$\|v\|_\infty = \max(|a_1|, |a_2|, |a_3|)$$

<u>Norm of a Matrix:</u>

The Frobenius norm of a matrix is defined as the square root of the sum of the squares of the elements of the matrix.

→ 1-norm, $\|A\|_1$

→ Infinity norm, $\|A\|_\infty$

→ 2-norm, $\|A\|_2$

→ Frobenius norm, $\|A\|_F$ (Euclidean Norm).

→ The max norm, $\|A\|_{max}$

1-norm of the matrix by summing each column of A and selecting the maximum.

Ily, Infinity norm of the matrix by summing each row of A and selecting the maximum.

2-norm of the matrix by taking the largest eigen value of A^TA and calculating its square root.

$$\|A\|_2 = \sqrt{\lambda_{max}(A^TA)}$$

Frobenius norm by summing the elements on A^TA's diagonal (its trace) and taking its square root.

$$\|A\|_F = \sqrt{trace(A^TA)}$$

Max norm of A can be obtained by taking largest value of A.

Q7. Calculate $\|A\|_\infty$, $\|A\|_1$, $\|A\|_F$ of $A = \begin{bmatrix} 2 & -3 & 1 \\ 4 & 3 & 0 \\ 5 & 2 & -1 \end{bmatrix}$ and $\|A\|_2$

Sol:
$$\|A\|_1 = \max(2+4+5,\ 3+3+2,\ 1+0+1)$$
$$= \max(11, 8, 2) = 11 \quad \downarrow \text{column wise}$$

$$\|A\|_\infty = \max(2+3+1,\ 4+3+0,\ 5+2+1)$$
$$= \max(6, 7, 8) = 8 \quad \downarrow \text{row wise}$$

$$\|A\|_F = \sqrt{\text{Trace}(A^TA)}. \qquad A^T = \begin{bmatrix} 2 & 4 & 5 \\ -3 & 3 & 2 \\ 1 & 0 & -1 \end{bmatrix}$$

$$A^TA = \begin{bmatrix} 45 & 16 & -3 \\ 16 & 22 & -5 \\ -3 & -5 & 2 \end{bmatrix}$$

$$\|A\|_F = \sqrt{45+22+2} = \sqrt{69} = 8.3066$$

(or)

$$\|A\|_F = \sqrt{\text{all the sum of squares of individual values}}$$
$$= \sqrt{2^2+4^2+5^2+(-3)^2+3^2+2^2+1^2+0^2+(-1)^2}$$
$$= \sqrt{69} = 8.3066$$

$$\therefore \|A\|_2 = \sqrt{\lambda_{max}(A^TA)}$$

Eigen Decomposition:-

Eigen decomposition is not commonly used directly in deep learning, but a related concept in SVD. SVD is often used for dimensionality reduction and feature extraction.

Decomposition of matrices gives information about their functional properties.
One of the most widely used kinds of matrix decomposition is called eigen decomposition, in which we decompose a matrix into a set of eigen vectors and eigen values.
A eigen vector of a square matrix A is a non-zero vector 'v', and the scalar λ is known as the eigen value corresponding to the eigen vector.
Eigen decomposition only applicable for square matrices.

$$A = EDE^{-1} \cdot (\text{or}) \quad A = Q\Lambda Q^{-1}$$

Here, E is an orthogonal matrix composed of eigen vectors of A.

D is a diagonal matrix with eigen values of A as its diagonal elements.

E^{-1} = inverse of E.

A matrix whose eigen values are all positive is called positive definite.
A matrix whose eigen values are all positive or zero-valued is called positive semi-definite.

~~A matrix or~~
Similarly, negative definite, negative semi-definite.

Ex: $A = \begin{bmatrix} 3 & 2 \\ 6 & 8 \end{bmatrix}$, find eigen decomposition?

Sol: $\begin{vmatrix} 3-\lambda & 2 \\ 6 & 8-\lambda \end{vmatrix} = 0$, $\quad (3-\lambda)(8-\lambda) - 12 = 0$

$24 - 3\lambda - 8\lambda + \lambda^2 - 12 = 0$

$$\lambda^2 - 11\lambda + 12 = 0$$

$$\lambda^2 + \lambda - 12\lambda + 12 = 0$$

$$\lambda_1 = 9.77, \quad \lambda_2 = 1.22.$$

at $\lambda_1 = 9.77$,

$$\begin{bmatrix} 3-9.77 & 2 \\ 6 & 8-9.77 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} -6.77 & 2 \\ 6 & -1.77 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$x + y = 1 \rightarrow (1)$$

$$-6.77x + 2y = 0 \rightarrow (2)$$

$$6x - 1.77y = 0 \rightarrow (3)$$

solving (1) & (2) (or (1) & (3)

we get $x = 0.227, \ y = 0.772$

$$X_1 = \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 0.227 \\ 0.772 \end{bmatrix}$$

at $\lambda_2 = 1.22$,

$$\begin{bmatrix} 1.78 & 2 \\ 6 & 6.78 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$1.78x + 2y = 0 \rightarrow (4)$$

$$6x + 6.78y = 0 \rightarrow (5)$$

$$x + y = 1 \rightarrow (6)$$

solving (4) & (6) (or (5) & (6)

we get $x = 8.69, \ y = -7.69$

$$X_2 = \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 8.69 \\ -7.69 \end{bmatrix}, \quad D = \begin{bmatrix} 9.77 & 0 \\ 0 & 1.22 \end{bmatrix}$$

$$E = \{X_1 \ X_2\} = \begin{bmatrix} 0.22 & 8.69 \\ 0.77 & -7.69 \end{bmatrix}, \quad E^{-1} = \begin{bmatrix} 0.917 & 1.036 \\ 0.091 & -0.026 \end{bmatrix}$$

Now, $RHS = EDE^{-1}$

$$RHS = \begin{bmatrix} 0.22 & 8.69 \\ 0.77 & -7.69 \end{bmatrix} \begin{bmatrix} 9.77 & 0 \\ 0 & 1.22 \end{bmatrix} \begin{bmatrix} 0.917 & 1.036 \\ 0.091 & -0.026 \end{bmatrix}$$

$$\therefore RHS = \begin{bmatrix} 2.93 & 1.95 \\ 6.04 & 8.03 \end{bmatrix} \approx \begin{bmatrix} 3 & 2 \\ 6 & 8 \end{bmatrix} = LHS = A \text{ //}$$

Singular Value Decomposition :-

SVD is a factorization method commonly used in linear algebra and ML. It decomposes a matrix into three other matrices, revealing important information about the original matrix's structure.

Steps :

(i) Given $m \times n$ matrix - A

(ii) Compute Transpose : A^T

(iii) Matrix multiplication : $A \cdot A^T$ and $A^T \cdot A$

(iv) Eigen decomposition : Perform eigen decomposition on $A \cdot A^T$ and $A^T A$ to get its eigen values and corresponding eigen vectors.

(v) Singular values : The singular values are the square roots of the eigen values of matrix A.

(vi) Singular value matrix (Σ) : Diagonal matrix Σ with the singular values.

(vii) Left Singular ~~Matrix~~ Vectors (U): The columns of U are the eigen vectors of $A \cdot A^T$.

(viii) Right singular vectors (V): The columns of V are the eigen vectors of $A^T \cdot A$.

This decomposition is useful in various applications including dimensionality reduction and solving linear systems.

Every real matrix has a singular value decomposition. If a matrix is not square, the eigen decomposition is not defined, and we must use a SVD.

SVD is useful for rectangular matrices.

$$SVD = U \Sigma V^T$$

$U_{m \times n}$, and $V_{n \times n}$ both are orthogonal matrices.

The determinant is equal to the product of all the eigen values of the matrix.

Det(A) is a function mapping matrices to real scalars.

If the determinant is 0, then space is contracted completely along at least one dimension, causing it to lose all of its volume.

If the determinant is 1, then the transformation is volume preserving.

~~Ex~~ Find the SVD of a matrix, $A = \begin{bmatrix} -4 & -7 \\ 1 & 4 \end{bmatrix}$

Soln

$$A = \begin{bmatrix} -4 & -7 \\ 1 & 4 \end{bmatrix}, \quad A^T = \begin{bmatrix} -4 & 1 \\ -7 & 4 \end{bmatrix}$$

$$AA^T = \begin{bmatrix} 65 & -32 \\ -32 & 17 \end{bmatrix}, \quad A^T A c$$

$$\begin{vmatrix} 65-\lambda & -32 \\ -32 & 17-\lambda \end{vmatrix} = 0$$

$$\lambda^2 - 82\lambda + 81 = 0$$

$$\therefore \lambda = 1, 81.$$

at $\lambda = 81$, eigen vector is $u_1 = \begin{bmatrix} -2 \\ 1 \end{bmatrix}$

at $\lambda = 1$, eigen vector is $u_2 = \begin{bmatrix} 0.5 \\ 1 \end{bmatrix}$

For $\lambda = 81$, $L = \sqrt{(-2)^2 + 1^2} = 2.236$

Normalized eigen vector $= \begin{bmatrix} \frac{-2}{2.236} \\ \frac{1}{2.236} \end{bmatrix} = \begin{bmatrix} -0.894 \\ 0.447 \end{bmatrix}$

For $\lambda = 1$, $L = \sqrt{0.5^2 + 1^2} = 1.118$

Normalized eigen vector $= \begin{bmatrix} \frac{0.5}{1.118} \\ \frac{1}{1.118} \end{bmatrix} = \begin{bmatrix} 0.447 \\ 0.894 \end{bmatrix}$

$$U = [u_1 \ u_2] = \begin{bmatrix} -0.894 & 0.447 \\ 0.447 & 0.894 \end{bmatrix}$$

$A^T A = \begin{bmatrix} 17 & 32 \\ 32 & 65 \end{bmatrix}$, eigen values are, $\lambda = 1, 81.$

at $\lambda = 81$, $v_1 = \begin{bmatrix} 0.5 \\ 1 \end{bmatrix}$

at $\lambda = \cancel{8}1$, $v_2 = \begin{bmatrix} -2 \\ 1 \end{bmatrix}$

For $\lambda = 81$, $L = \sqrt{0.5^2 + 1^2} = 1.118$

Normalized eigen vector $= \begin{bmatrix} \frac{0.5}{1.118} \\ \frac{1}{1.118} \end{bmatrix} = \begin{bmatrix} 0.447 \\ 0.894 \end{bmatrix}$

For $\lambda = 1$, $L = \sqrt{(-2)^2 + 1^2} = 2.236$

Normalized eigen vector $= \begin{bmatrix} \frac{-2}{2.236} \\ \frac{1}{2.236} \end{bmatrix} = \begin{bmatrix} -0.894 \\ 0.447 \end{bmatrix}$

$$\Sigma = \begin{bmatrix} \sqrt{81} & 0 \\ 0 & \sqrt{1} \end{bmatrix} = \begin{bmatrix} 9 & 0 \\ 0 & 1 \end{bmatrix}$$

$$V = [v_1 \ \ v_2] = \begin{bmatrix} 0.447 & -0.894 \\ 0.894 & 0.447 \end{bmatrix}$$

$$V^T = \begin{bmatrix} 0.447 & 0.894 \\ -0.894 & 0.447 \end{bmatrix}$$

Now, $SVD = U \Sigma V^T$

RHS $= U \Sigma V^T$

$$RHS = \begin{bmatrix} -0.894 & 0.447 \\ 0.447 & 0.894 \end{bmatrix} \begin{bmatrix} 9 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0.447 & 0.894 \\ -0.894 & 0.447 \end{bmatrix}$$

$$RHS = \begin{bmatrix} -3.996 & -6.993 \\ 0.999 & 3.996 \end{bmatrix} \approx \begin{bmatrix} -4 & -7 \\ 1 & 4 \end{bmatrix} = LHS = A$$

Principal Component Analysis:-

One simple ML algorithm, PCA can be derived using only knowledge ~~o~~ of basic linear algebra.

PCA is one of the dimensionality reduction technique.

It transforms the variables into a new set of variables is called as principal components.

These principal components are linear combinations of original ~~variables~~ variables and that are orthogonal.

$(AA^T = I)$ Orthogonal.

It is a way of identifying patterns in data and expressing the data in such a way to highlight their similarities and differences.

PCA Algorithm:

(i) Get data.

(ii) To find mean of each data set.

(iii) Calculate the covariance matrix.

(iv) Calculate the eigen values and its corresponding eigen vectors of covariance matrix.

(v) Choosing ~~one~~ top 'k' components and forming a feature vector.

(vi) Project the original data on to the new subspace formed by the selected principal components.

Eg:- Find (or) Calculate the PCA for the following data

x	4	8	13	7
y	11	4	5	14

Sol:- Step-1:

given data

x	4	8	13	7
y	11	4	5	14

Step-2:
$$\bar{x} = \frac{4+8+13+7}{4} = \frac{32}{4} = 8$$
$$\bar{y} = \frac{4+4+5+14}{4} = \frac{17}{2} = 8.5$$

Step-3: Covariance matrix

x	y	$(x-\bar{x})$	$(y-\bar{y})$	$(x-\bar{x})^2$	$(y-\bar{y})^2$	$(x-\bar{x})(y-\bar{y})$
4	11	-4	2.5	16	6.25	-10
8	4	0	-4.5	0	20.25	0
13	5	5	-3.5	25	12.25	-17.5
7	14	-1	5.5	1	30.25	-5.5

$$Var(x) = \frac{\Sigma(x-\bar{x})^2}{n-1} \qquad \Sigma(x-\bar{x})^2 = 42 \qquad \Sigma(x-\bar{x})(y-\bar{y}) = -$$
$$\Sigma(y-\bar{y})^2 = 69$$
$$Covar(x,y) = \frac{\Sigma(x-\bar{x})(y-\bar{y})}{n-1}$$

$$\text{Covariance matrix} = \begin{bmatrix} Var(x) & Cov(x,y) \\ Cov(x,y) & Var(y) \end{bmatrix}$$

$$Var(x) = \frac{42}{3} = 14, \quad Var(y) = \frac{69}{3} = 23$$

$$Cov(x,y) = \frac{-33}{3} = -11$$

$$\text{Matrix} = \begin{bmatrix} 14 & -11 \\ -11 & 23 \end{bmatrix}$$

Step-4: Eigen decomposition
$$\begin{vmatrix} 14-\lambda & -11 \\ -11 & 23-\lambda \end{vmatrix} = 0$$

$$\lambda^2 - 37\lambda + 201 = 0$$
$\lambda_1 = 30.3849 \Rightarrow$ first principal component

$\lambda_2 = 6.6151 \rightarrow$ second principal component.

at $\lambda_1 = 30.3849$, eigen vector is $\begin{bmatrix} 11 \\ -16.38 \end{bmatrix}$

The normalized eigen vector is $L = \sqrt{11^2 + (-16.38)^2}$

$$e_1 = \begin{bmatrix} \frac{11}{L} \\ \frac{-16.38}{L} \end{bmatrix} = \begin{bmatrix} 0.5574 \\ -0.8303 \end{bmatrix}$$

lly, at $\lambda_2 = 6.6151$, normalized eigen vector is $e_2 = \begin{bmatrix} 0.8303 \\ 0.5574 \end{bmatrix}$

Step 5: Deriving new dataset

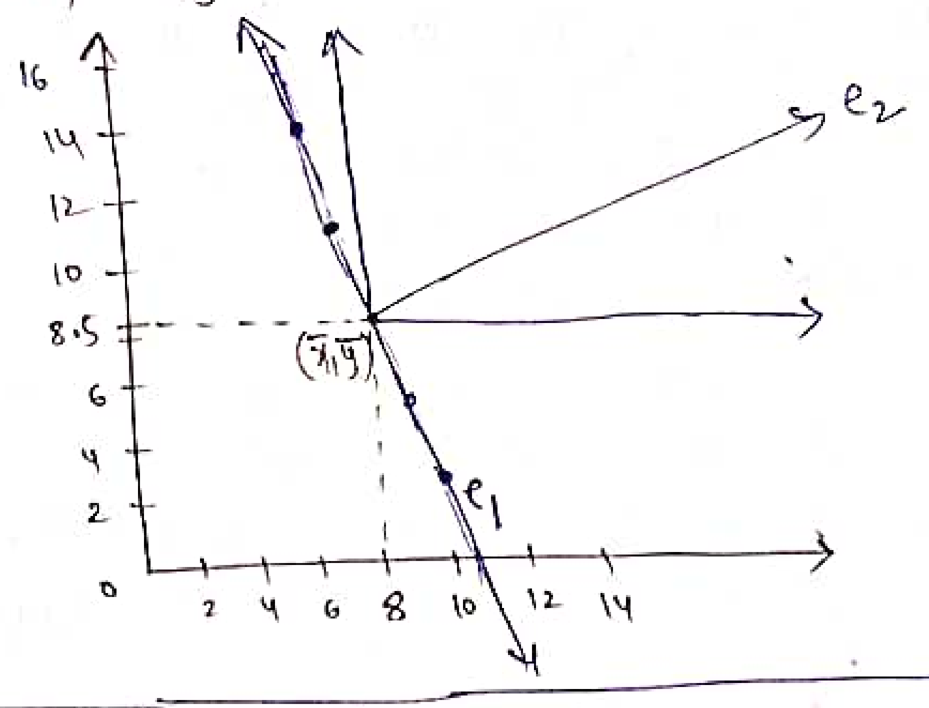
First PC1	P_{11}	P_{12}	P_{13}	P_{14}

$$P_{11} = e_1^T \begin{bmatrix} 4-8 \\ 11-8.5 \end{bmatrix} \qquad e_1^T \begin{bmatrix} x_i - \bar{x} \\ y_i - \bar{y} \end{bmatrix}$$

$$P_{11} = \begin{bmatrix} 0.5574 & -0.8303 \end{bmatrix} \begin{bmatrix} 4 \\ 2.5 \end{bmatrix} = -4.3052$$

lly, $P_{12} = 3.7361$, $P_{13} = 5.6928$, $P_{14} = -5.1238$

Step 6: Projection



Probability and Information Theory

Random Variables, Probability Distributions, Marginal Probability, Conditional Probability, Expectation, Variance and Covariance, Bayes Rule, Information theory.

Probability theory is a ~~fo~~ mathematical framework for representing uncertain statements. Many branches of computer science deal mostly with entities that are entirely deterministic and certain.

ML must always deal with uncertain quantities, and sometimes may also need to deal with stochastic (non-deterministic) quantities.

There are three possible sources of uncertainty.

(i) Inherent stochasticity in the system being modeled.

(ii) Incomplete observability.

(iii) Incomplete modeling.

~~If a doctor analyzes a patient and says that the patient has a 40% chance of having the flu~~

In the case of the doctor diagnosing the patient, we use probability to represent a degree of belief, with 1 indicating absolute certainty that the patient has the flu and '0' indicating absolute certainty that the patient does not have the flu.

The former kind of probability, related directly to the rates at which events occur, is known as frequentist probability, while the latter, related to qualitative levels of certainty, is known as Bayesian probability.

Random Variables :-

A random variable is a variable whose possible values are outcomes of a random experiment.

A random variable is a variable that can take different values randomly.

We denote the random variable with a lower-case letter and can have lower-case sub-script letters.

Random variables can be classified into 2 main types.

→ Discrete Random Variables

→ Continuous Random Variables.

A discrete RV is a RV that can take on a countable number of distinct values.

Ex:- The no. of heads obtained when flipping a coin.

The no. of cars passing through a toll booth in an hour.

The outcome of rolling a 6-sided die.

A continuous RV is a RV that can take any value within a specified range (or) interval.

Ex:- The height of a person.

The temparature on a given day.

The time taken for computer to process a task.

Discrete Random Variable:

$$\Sigma p(x) = 1$$

Expectation $= E(x) = \Sigma x p(x)$

Mean $= \mu = \Sigma x p(x) = E(x)$

Variance $= \sigma^2 = \Sigma x^2 p(x) - \mu^2$

$$\sigma^2 = E(x^2) - [E(x)]^2$$

$SD = \sqrt{\text{variance}} = \sigma$

Continuous Random Variables:

$$\int_{-\infty}^{\infty} f(x)\,dx = 1$$

$$\text{Mean} = \mu = E(x) = \int_{-\infty}^{\infty} x f(x)\,dx$$

$$\text{Variance} = \sigma^2 = \int_{-\infty}^{\infty} x^2 f(x)\,dx - \mu^2$$

Probability Distributions :-

Probability distribution describe how the likelihood of different outcomes is spread. They are essential for modeling uncertainty and randomness.

Probability distribution is also called as theoritical distribution.

There are 2 types of probability distributions.

(i) Discrete Probability distribution (PMF)

→ Bernoulli Distribution

→ Binomial Distribution

→ Poisson Distribution

~~→ Rectangular Distribution~~

→ Negative Binomial distribution

→ Geo-metric distribution

(ii) Continuous Probability Distributions (PDF)

→ Uniform Distribution (Rectangular distribution)

→ Normal / Gaussian distribution

→ Exponential distribution

→ Gamma distribution

→ Logistic distribution.

PMF – Probability Mass Function, PDF – Probability Density Function.

Bernoully Distribution:

Models a single binary experiment with two outcomes. (success (or) failure).

$$PMF = P(X=x) = p^x (1-p)^{1-x} \quad \downarrow \quad q = 1-p, \quad p+q=1$$

Binomial Distribution:

Models the number of successes in a fixed number of independent Bernoulli trails.

$$PMF = P(X=x) = {}^nC_x \, p^x \, q^{n-x} \quad \downarrow \quad q = 1-p$$

$$\text{Mean} = \mu = \sum x\,p(x) = np.$$

$$\text{Variance} = \sigma^2 = npq$$

Poisson Distribution:

Models the number of events occurring in fixed intervals of time (or) space.

It is limiting case of Binomial distribution. Here 'n' is very large.

$$PMF = P(X=x) = \frac{e^{-\lambda}\lambda^x}{x!}$$

$$\text{Mean} = \mu = \lambda$$

$$\text{Variance} = \sigma^2 = \lambda$$

Geometric Distribution:

Models the number of trails needed before a success occurs in a sequence of independent Bernoulli trials.

$$PMF = P(X=x) = (1-p)^{x-1} \cdot p. \text{ for } x = 1,2,3,\dots$$

$$\text{Mean} = \mu = \frac{1}{p}, \quad \text{Variance} = \frac{1-p}{p^2} = \sigma^2$$

Uniform / Rectangular Distribution:

Uniform distribution occurs when all outcomes in a range are equally likely. It is called rectangular because the PDF is a constant within the range.

$$PDF = f(x) = \frac{1}{b-a} \quad \text{for } a \le x \le b$$

$$\text{Mean} = \mu = \frac{a+b}{2}, \quad \text{Variance} = \sigma^2 = \frac{(b-a)^2}{12}$$

Normal Distribution:

Normal distribution, that is symmetric and bell-shaped. It is completely characterized by its mean and SD.

$$PDF = f(x) = \frac{1}{\sqrt{2\pi}\,\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Exponential Distribution:

Models the time b/w events in a Poisson process, where events occur continuously and independently at a constant avg rate.

$$PDF = f(x) = \lambda e^{-\lambda x}$$

$$\text{Mean} = \frac{1}{\lambda} = \mu, \quad \text{Variance} = \sigma^2 = \frac{1}{\lambda^2}$$

Gamma Distribution:

The Gamma distribution is a family of continuous probability distributions with 2 parameters, shape (k), and rate (λ).

$$PDF = f(x; k, \lambda) = \frac{\lambda^k x^{k-1} e^{-\lambda x}}{\Gamma(k)}$$

Here, $\lambda = \frac{1}{\theta}$

if $k=1$, reduces to exponential distribution.
if k is an integer, it is called the Erlang distribution.

Mean $= \mu = \frac{k}{\lambda}$, Variance $= \sigma^2 = \frac{k}{\lambda^2}$

$\mu = k\theta$. $\sigma^2 = k\theta^2$

Logistic Distribution:

The logistic distribution is a continous probability distribution that resembles the normal distribution but has heavier tails.

$$PDF = f(x;\mu,s) = \frac{e^{-\frac{(x-\mu)}{s}}}{s\left(1+e^{-\frac{(x-\mu)}{s}}\right)^2}$$

Here, μ = location
s = Scale

Mean $= \mu$

Variance $= \sigma^2 = \frac{s^2\pi^2}{3}$

Marginal Probability:-

The probability distribution over the subset is known as the marginal probability distribution.

It refers to the probability of a specific event occuring without considering the occurence or non-occurence of other events.
It is derived from the joint probability distribution of multiple random variables by summing or integrating over the values of the other variables.

For two discrete random variables X and Y,

The marginal probability of x is $P(X=x) = \sum_{y} P(X=x, Y=y)$

The marginal probability of y is P(

$$P(Y=y) = \sum_x P(X=x, Y=y)$$

For two continuous random variables X and Y, the marginal probability of x is

$$f_x(x) = \int f_{xy}(x,y)\,dy$$

The marginal probability of y is

$$f_y(y) = \int f_{xy}(x,y)\,dx$$

Conditional Probability:-

It is the probability of an event occurring given that another event has already occurred. It is denoted by $P(A|B)$.

$$P\left(\frac{A}{B}\right) = \frac{P(A\cap B)}{P(B)}$$

Here, $P(A\cap B)$ = joint probability.

Any joint probability distribution over many random variables may be decomposed into conditional distributions over only one variable.

$$P(x^{(1)}, \ldots, x^{(n)}) = P(x^{(1)}) \prod_{i=2}^{n} P(x^{(i)} | x^{(1)}, \ldots, x^{(i-1)}).$$

This observation is known as the chain rule (or product rule) of probability.

$$P(a,b,c) = P\left(\frac{a}{b,c}\right) \cdot P(b,c)$$

we know, $P(b,c) = P\left(\frac{b}{c}\right) \cdot P(c)$

$$P(a,b,c) = P\left(\frac{a}{b,c}\right) \cdot P\left(\frac{b}{c}\right) \cdot P(c)$$

Formulas,

$P\left(\frac{A}{A}\right) = 1$, $P\left(\frac{\phi}{A}\right) = 0$

if A, B are mutually exclusive ↓ $A \cap B = \phi$

then $P\left(\frac{A}{B}\right) = P\left(\frac{B}{A}\right) = 0.$

$$P\left(\frac{A}{\bar{B}}\right) = \frac{P(A) - P(A \cap B)}{P(\bar{B})}$$

$$P\left(\frac{\bar{A}}{B}\right) = \frac{P(B) - P(A \cap B)}{P(B)}$$

$$P\left(\frac{\bar{A}}{\bar{B}}\right) = \frac{1 - P(A \cup B)}{1 - P(B)}$$

Ex:- Event - A : Drawing a red card

Event - B : Drawing a heart

Find $P\left(\frac{A}{B}\right)$, the probability of drawing a red card given that the card is heart.

Sol:- $P\left(\frac{A}{B}\right) = \frac{P(A \cap B)}{P(B)}$

Total cards = 52 { 26 Red cards, 26 Black cards.

$P(A \cap B) = P(A) \cdot P(B) = \frac{\cancel{26}^{1}}{\cancel{52}_{2}} \cdot \frac{1}{4} = \frac{1}{8}$ | $P(B) = \frac{1}{4}$.

$$P\left(\frac{A}{B}\right) = \frac{\frac{1}{\cancel{8}_2}}{\frac{1}{\cancel{4}}} = \frac{1}{2} \,//.$$

Expectation :-

The expectation or expected value of some function $f(x)$ with respect to a probability distribution $P(x)$.

The expectation of a random variable is a measure of the central tendency or avg value that we can expect from the variable.

For discrete random variable,

$$\text{Expectation} = E(x) = \Sigma x\, p(x)$$

For continuous random variable

$$E(x) = \int_{-\infty}^{\infty} x\, f(x)\, dx$$

Properties :

(i) Linearity of expectation

$$E(aX+b) = a\,E(X)+b$$

(ii) Expectation of constant

$$E(c) = c$$

(iii) Expectation of sum

$$E(X+Y) = E(X) + E(Y).$$

Variance :-

Variance is a measure of the spread or dispersion of a set of values.

It quantifies how far each data point in the set is from the mean.

For discrete RV,

$$\text{Variance} = \sigma^2 = Var(x) = \Sigma x^2 p(x) - \mu^2 \quad \text{or} \quad \Sigma (x-\mu)^2 p(x)$$

For continuous RV,

$$Var(x) = \sigma^2 = \int_{-\infty}^{\infty} x^2 f(x) - \mu^2 \quad (or) \quad \int (x-\mu)^2 f(x)\, dx$$

Properties:

(i) Variance of constant

$$Var(c) = 0$$

(ii) Variance of sum

$$var(x+y) = var(x) + var(y) + 2\,cov(x,y)$$

(iii) Variance of a constant times of a RV.

$$var(aX) = a^2\, var(X).$$

A small variance indicates that the data points tend to be close to the mean.
A large variance indicates that the data points are spread out over a wide range.

Covariance:-

Covariance gives how much two values are linearly related to each other.

$$cov(x,y) = \frac{\Sigma (x-\bar{x})(y-\bar{y})}{N-1} = E\left[(X-\mu_x)(Y-\mu_y)\right]$$

if $cov(x,y) = 0$, no linear relationship b/w x and y.

if $cov(x,y) > 0$, indicates as X increases Y also increases.

if $cov(x,y) < 0$, indicates as X increases, Y decreases.

Properties:

(i) Covariance of a variable with it self

$$cov(x,x) = Var(x)$$

(ii) Bilinearity.

$$\text{cov}(aX+b, Y) = a\cdot\text{cov}(X, Y)$$

$$\text{cov}(X, aY+b) = a\cdot\text{cov}(X, Y).$$

(iii) Symmetry

$$\text{cov}(X, Y) = \text{cov}(Y, X)$$

$\text{cov}(X,Y) > 0$, x and y move same direction

$\text{cov}(X,Y) < 0$, x and y move opposite direction.

Baye's Rule :-

Baye's rule, also known as Baye's theorem or Bayes law that describes how to update probabilities based on new evidence. It is named after the Reverend Thomas Bayes, who introduced the theorem.

Bayes rule derived from the definition of conditional probability.

For events A and B, Baye's rule expressed as,

$$P\left(\frac{A}{B}\right) = \frac{P(A)\cdot P\left(\frac{B}{A}\right)}{P(B)}.$$

where,

$P\left(\frac{A}{B}\right)$ is the posterior probability of event A given evidence B.

$P\left(\frac{B}{A}\right)$ is the likelihood of evidence B given that A has occurred.

$P(A)$ is the prior probability of event A.

$P(B)$ is the probability of evidence B occurring.

we know that conditional probability

$$P\left(\frac{A}{B}\right) = \frac{P(A\cap B)}{P(B)} \quad \text{and} \quad P\left(\frac{B}{A}\right) = \frac{P(A\cap B)}{P(A)}$$

$$P(A\cap B) = P\left(\frac{A}{B}\right)\cdot P(B) \rightarrow ①$$

$$P(A\cap B) = P\left(\frac{B}{A}\right)\cdot P(A) \rightarrow ②$$

from equations ① & ②

$$P\left(\frac{A}{B}\right)\cdot P(B) = P\left(\frac{B}{A}\right)\cdot P(A).$$

$$\therefore \quad P\left(\frac{A}{B}\right) = \frac{P(A)\cdot P\left(\frac{B}{A}\right)}{P(B)}. \quad //$$

Laplace Distribution:-

The Laplace distribution also known as the double exponential distribution is a continuous probability distribution.

It is characterized by its peakedness around the mean and heavy tails.

$$PDF = f(x;\mu,b) = \frac{1}{2b} e^{-\frac{|x-\mu|}{b}}$$

μ = location (median)

b = scale

Dirac Distribution :-

Also known as Delta function.

It is a theoritical probability distribution that represents a perfect "spike" at a specific point.

Mathematically it is represented by the Dirac delta function, denoted as $\delta(x-a)$, which is zero everywhere expect $x=a$, where it is infinite.
The area under the delta function is equal to 1.

Empirical Distribution:-

It is not a specific probability distribution, but a distribution based on observed data.
It is an approximation of the true distribution of a random variable based on a sample from that variable.

$$F_n(x) = \frac{1}{n} \sum_{i=1}^{n} I(X_i \leq x)$$

where, I = indicator function.
$F_n(x)$ = Empirical distribution function.

Information Theory:-

Information theory is a branch of applied mathematics that revolves around quantifying how much information is present in a signal.
It was originally invented to study sending messages from discrete alphabets over a noisy channel, such as communication via radio transmission.
This field is fundamental to many areas of electrical engineering and computer science. This field was established by Claude Shannon in the 1940s.
Entropy: Entropy is a measure of uncertainty (or randomness in a set of possible outcomes.

$\text{Entropy} = H(X) = -\sum_i P(X=x_i) \log_2 P(X=x_i)$.

Higher entropy indicates greater uncertainty (or disorder).

Joint Entropy:

For two random variables X and Y, the joint entropy $H(X,Y)$ is the measure of uncertainty associated with both variables.

$$H(X,Y) = -\sum_{i,j} P(X=x_i, Y=y_j) \log_2 P(X=x_i, Y=y_j)$$

Conditional Entropy:

For two random variables X and Y, the conditional entropy $H\left(\frac{X}{Y}\right)$ measures the remaining uncertainty about X given the value of Y.

$$H\left(\frac{X}{Y}\right) = -\sum_{i,j} P(X=x_i, Y=y_i) \log_2 P(X=x_i \mid Y=y_i)$$

Mutual Information:

Mutual Information measures the reduction in uncertainty about X when Y is known.

$$I(X;Y) = H(X) - H\left(\frac{X}{Y}\right).$$

It quantifies the amount of information one random variable contains about another.

KL Divergence:

KL - Kullback - Leibler diavergence how one probability distribution diverges from a second, expected probability distribution.

$$D_{KL}(P \| Q) = \sum_i P(x_i) \log \left(\frac{P(x_i)}{Q(x_i)}\right)$$

It is used to measure the difference b/w two probability distributions.

Channel Capacity:

It represents the maximum rate at which information can be reliably transmitted over a communication channel, considering noise and other impairments.

For continuous random various, the shannon entropy $h(X) = -\int f(x) \log_2 f(x) dx$. It is also called differential entropy.

Information theory has applications in various fields, including data compression, cryptography, communication theory, ML, and more.

It provides a theoretical foundation for understanding and optimizing information processing systems.

Structured Probabilistic Models:-

Structured probabilistic models refer to a class of statistical models that capture dependencies and relationships among multiple variables in a structured way.

These models are designed to represent the inherent structure in complex systems, making them particularly useful for tasks such as pattern recognition, decision making, and probabilistic inference.

Markov Chain:

A sequence of random variables where the probability of each variable depends only on the state of the preceding one.

Hidden Markov Models (HMM):

HMM extends the idea of the markov chain to include -

unobservable states. Widely used in speech recognition and bio informatics.

Bayesian Networks (BN):

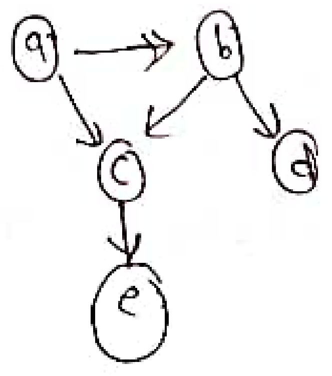
A DAG - Directed Acyclic Graph that represents probabilistic relationships among a set of variables. Nodes represent variables, and edges represent dependencies.

Markov Random Fields:

MRF, a undirected graph that represents dependencies among variables. Used in computer vision, image processing, and spatial modeling.

Structured probabilistic models are widely used in ML and AI for various tasks, including classification, regression, clustering, and generative modeling.

They provide a principled way to model complex relationships in data and make predictions in scenarios where the dependencies among variables are crucial for accurate modeling.



$$p(a,b,c,d,e) = p(a)\, p\left(\frac{b}{a}\right) p\left(\frac{c}{a,b}\right) p\left(\frac{d}{b}\right) \cdot p\left(\frac{e}{c}\right).$$

$$p(a,b,c,d,e) = \frac{1}{Z}\, \phi^{(1)}(a,b,c)\, \phi^{(2)}(b,d)\, \phi^{(3)}(c,e).$$

Numerical Computation

Overflow and Underflow, Gradient-Based Optimization, Constrained Optimization, Linear Least Squares.

Machine learning algorithms usually require a high amount of numerical computation. Common operations include optimization and solving systems of linear equations.

Numerical computation refers to the implementation and application of numerical algorithms to solve mathematical problems using computers.

It plays a crucial role in various fields such as science, engineering, finance, and computer science.

Numerical computation involves approximating mathematical solutions, especially when exact analytical solutions are difficult or impossible to obtain.

Overflow and Underflow :-

Overflow and underflow are phenomena that can occur in numerical computations when the result of an operation exceeds the representable range of the data type being used.

Overflow :-

Overflow happens when the result of an arithmetic operation is too large to be represented within the available numeric range.

In computer systems, numbers are typically represented using a fixed number of bits, and there is a limit to the largest and smallest values that can be expressed.

Overflow occurs when numbers with large magnitude are approximated as ∞ or $-\infty$.

Underflow:-

Underflow occurs when the result of an arithmetic operation is too close to zero to be represented within the available numeric precision.

One form of rounding error is underflow. It occurs when numbers near zero are rounded to zero.

One example of a function that must be stabilized against underflow and overflow is the softmax function.

The softmax function is often used to predict the probabilities associated with a multinoulli distribution.

$$\text{softmax}(x)_i = \frac{\exp(x_i)}{\sum_{j=1}^{n} \exp(x_j)}$$

Poor Conditioning:-

Conditioning refers to how rapidly a function changes with respect to small changes in its inputs.

Functions that change rapidly when their inputs are slightly changed can be problematic for scientific computation because rounding errors in the inputs can result in large changes in the o/p.

Optimization :-

Optimization is the process of finding the best solution to a problem from a set of possible solutions.

The best solution is typically the one that maximizes (or) minimizes a certain objective function while satisfying a set of constraints.

There are 2 types of optimization.

(i) Unconstrained Optimization:

The optimization problem involves minimizing (or maximizing a function without any constraints.

Ex :- finding min of a quadratic equation.

(ii) Constrained Optimization:

The optimization problem includes constraints that the solution must satisfy

Ex :- Maximizing profit subject to production constraints.

Gradient Based Optimization :-

Gradient based optimization is a class of optimization algorithms that leverages information about the gradient of a function to iteratively improve the solution.

These algorithms are widely used in ML, DL, and various scientific and engineering applications.

~~ch.~~

Optimizers are algorithms (or) methods used to update the parameters of the network such as weights, biases, etc to minimize the losses.

(GD)

Different instances of Gradient descent based optimizers are as follows:

→ GD (or) Batch GD (or) Vanilla GD

→ stochastic GD (SGD)

→ mini-batch GD (MBGD)

Batch GD :-

GD is an optimization algorithm, it iteratively updates the parameters in the direction opposite to the gradient, aiming to reach a min of the objective function.

$$\theta_{new} = \theta_{old} - \alpha \nabla J(\theta_{old})$$

Here, α is the learning rate.

J is the cost function.

θ is the parameter to be updated.

The GD optimization algorithm has many applications including -

→ Linear regression

→ Classification Algorithms

→ Backpropagation in neural networks etc.

Learning rate represents the size of the steps our optimization algorithm takes to reach the global minima.

Advantages include

→ Easy computation

→ Easy implementation

→ Easy to understand.

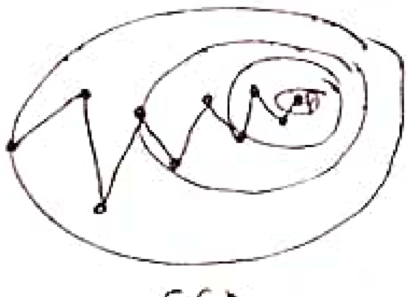
Disadvantages include

→ May trap at local minima

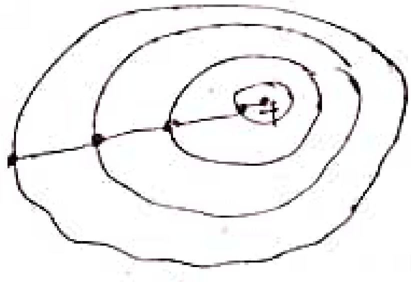
→ Requires large memory.

Stochastic GD :-

It is the extension of the batch GD optimization algorithm. In which, we compute the derivative by taking one data point at a time.
i.e, try to update the models parameters more frequently.



SGD



GD

It is observed that in SGD the updates take more iterations compared to GD to reach minima.

Advantages include
- → Takes less time to update
- → Requires less memory
- → May get new minimas

Mini-batch GD :-

It is the extension of the SGD algorithm.
It is considered the best among all the variations of gradient descent algorithms.
MB-GD algorithm takes a batch of points (or subset of points from the data set to compute derivative.

MB-GD

Challenges:

- → optimum learning rate.
- → constant learning rate.
- → local minimum.

Gradient-based optimization is used in training ML models, particularly in the content of DL. Efficient optimization algorithms are crucial for finding optimal model parameters and achieving good generalization performance on unseen data.

Constrained Optimization:-

Constrained optimization involves finding the min (m) max of a function subject to a set of constraints.

minimize (m) maximize $f(x)$ subject to $g(x) \leq 0$ and $h(x) = 0$, for $i = 1, 2, \cdots, m$ and $j = 1, 2, \cdots, n$.

Here, $f(x)$ = objective function to be optimized.

$g(x) \leq 0$ are inequality constraints

$h(x) = 0$ are equality constraints.

The goal is to find a vector x that minimizes or maximizes the objective function while satisfying the given constraints.

KKT Conditions:

KKT - Karush Kuhn Tucker.

A set of conditions that characterize the solutions to a constrained optimization problem.

The KKT approach provides a very general solution to constrained optimization. With the KKT approach we introduce a new function called the generalized lagrangian or generalized Lagrange function.

The generalized Lagrangian is

$$L(x, \lambda, \alpha) = f(x) + \sum_i \lambda_i g^{(i)}(x) + \sum_j \alpha_j h^{(j)}(x)$$

Here, λ_i and α_i are KKT multipliers.

Lagrange Multipliers:

Introduces Lagrange multipliers to transform the constrained optimization problem into an unconstrained one.

Challenges in Constrained Optimization:

(i) Feasibility and optimality.

(ii) Local minima / maxima

Constrained optimization problems may have multiple local minima or maxima.

(iii) Computational Complexity.

Solving constrained optimization problems can be computationally intensive, especially for large scale problems.

Constrained optimization is a fundamental problem in various fields, including engineering, economics, finance, and operations research.

Linear Least Squares :-

Linear least squares is a method to find the best fitting linear relationship b/w a dependent variable "y" and one or more independent variables "x_i"

The goal is to minimize the sum of squared differences b/w the observed values of y and the values predicted by the linear model.

$$y = \alpha_0 + \alpha_1 x_1$$

$$y = \alpha_0 + \alpha_1 x_1 + \alpha_2 x_2 + \cdots + \alpha_n x_n + \varepsilon_i$$

where,

y = observed value of the dependent variable.

x_i = independent variable.

α_i = coefficients to be estimated.

ε_i = Error term

Matrix Form :

The linear least squares problem can be expressed in matrix form as :

$$\text{minimize } \|y - X\alpha\|_2^2$$

$$\alpha = (X^T X)^{-1} X^T y$$

Assumptions :

Linear least squares relies on several assumptions, including :

→ Linearity : The relationship b/w variables is linear.

→ Independence : Observations are independent.

→ Homoscedasticity : Residuals have constant variance.
→ Normality : Residuals are normally distributed.

Linear least squares is widely used in various fields, including regression analysis, curve fitting, and ML

Q:- Fit the following data using linear least squares method (or) linear regression.

x	1	2	3	4
y	3	4	5	7

Sol:- It is in the form of $y = a_0 + a_1 x$

where, $a_0 = \dfrac{\Sigma y \Sigma x^2 - \Sigma x \Sigma xy}{n \Sigma x^2 - (\Sigma x)^2}$

x	y	xy	x^2
1	3	3	1
2	4	8	4
3	5	15	9
4	7	28	16

$a_1 = \dfrac{n \Sigma xy - \Sigma x \Sigma y}{n \Sigma x^2 - (\Sigma x)^2}$

$\Sigma x = 10$, $\Sigma y = 19$

$\Sigma x^2 = 30$, $\Sigma xy = 54$

$a_0 = \dfrac{19(30) - 10(54)}{4(30) - (10)^2} = \dfrac{3}{2} = 1.5$

$a_1 = \dfrac{4(54) - 10(19)}{4(30) - (10)^2} = \dfrac{26}{20} = 1.3$

$\therefore y = a_0 + a_1 x$

$\therefore y = 1.5 + 1.3(x)$ //

Optimizers in DL:-

Optimizers play a crucial role in training neural networks by adjusting the models parameters to minimize the loss function.
The optimization process involves finding the optimal set of weights and biases that make the model perform well on the given task.
Some common optimizers are
- → Stochastic GD
- → Adam – Adaptive Moment Estimation
- → RMS prop – Root Mean Squared Propagation.
- → Adagrad – Adaptive Gradient Algorithm.
- → Adadelota
- → Adamax

Stochastic GD:-

The basic optimization algorithm where the model parameters are updated in the direction of the negative gradient of the loss function with r to parameter.

$$\theta_{new} = \theta_{old} - \alpha \nabla J(\theta_{old})$$

Adam:

An adaptive learning rate optimization algorithm that combines ideas from RMSprop and momentum.
It adapts the learning rate of each parameter individually.
- Uses both first order (momentum) and second-order (RMSprop) moments to update parameters.

RMSprop:

An adaptive learning rate method that adjusts the learning rates of each parameter based on the magnitude of recent gradients.

$$\theta_{new} = \theta_{old} - \frac{\alpha}{\sqrt{v+\epsilon}} \nabla J(\theta_{old})$$

Adagrad:

An adaptive learning rate algorithm that adapts the learning rates for each parameter based on the historical gradient information.

$$\theta_{new} = \theta_{old} - \frac{\alpha}{\sqrt{G+\epsilon}} \nabla J(\theta_{old})$$

Adadelta:

An extension of adagrad that aims to address its aggressive, monotonically decreasing learning rates by using a running avg of squared parameter updates.

Adamax:

A variant of Adam that uses the ∞-norm (max) of the gradients in place of the L2 norm.

Activation Functions in DL:-

Activation functions are crucial components in DL models that introduce non-linearity to the network, enabling it to learn complex patterns and relationships in data.

Some common activation functions used in DL are

→ Sigmoid Function.

→ Hyperbolic Tangent Function.
→ ReLU – Rectified Linear Unit.
→ ELU – Exponential Linear Unit.
→ Softmax Function.

Sigmoid Function:

Historically used in the o/p layer for binary classification problems. However, its vanishing gradient problem makes it less common in hidden layers.

Formula : $\sigma(x) = \frac{1}{1+e^{-x}}$

Range : $(0,1)$

Hyperbolic Tangent Function:

Similar to the sigmoid, but with a range b/w -1 to 1. It mitigates the vanishing gradient problem better than the sigmoid.

Formula : $\tanh(x) = \frac{e^{2x}-1}{e^{2x}+1}$

Range : $(-1,1)$

ReLU :

One of the most widely used activation function. It introduces non-linearity and helps with the vanishing gradient problem.

Formula: $ReLU(x) = \max(0,x)$

Range : $[0,+\infty)$

ELU :

Similar to ReLU but with a smooth curve for negative values, which can help with the dying ReLU problem.

Formula : $ELU(x) = \begin{cases} x & \text{if } x > 0 \\ \alpha(e^{x} - 1), & \text{if } x \leq 0 \end{cases}$

Range : $(-\infty, +\infty)$

Softmax :

Primarily used in the o/p layer for multi-class classification.

Formula : $Softmax(x)_i = \dfrac{e^{x_i}}{\sum_{j=1}^{N} e^{x_j}}$

Range : $(0,1)$ and
the sum of all elements is 1.

15-11-2023