Hpc1

*The breadth-first search (BFS) algorithm is used to search a tree or graph data structure for a node that meets a set of criteria. It starts at the tree’s root or graph and searches/visits all nodes at the current depth level before moving on to the nodes at the next depth level. Breadth-first search can be used to solve many problems in graph theory.*

**Algorithm of Breadth-First Search:**

* **Step 1:** Consider the graph you want to navigate.
* **Step 2:** Select any vertex in your graph (say **v1**), from which you want to traverse the graph.
* **Step 3:** Utilize the following two data structures for traversing the graph.
* Visited array(size of the graph)
* Queue data structure
* **Step 4:** Add the starting vertex to the visited array, and afterward, you add v1’s adjacent vertices to the queue data structure.
* **Step 5:** Now using the FIFO concept, remove the first element from the queue, put it into the visited array, and then add the adjacent vertices of the removed element to the queue.
* **Step 6:** Repeat step 5 until the queue is not empty and no vertex is left to be visited.

*Depth-first search is an algorithm for traversing or searching tree or graph data structures. The algorithm starts at the root node (selecting some arbitrary node as the root node in the case of a graph) and explores as far as possible along each branch before backtracking.*

*So the basic idea is to start from the root or any arbitrary node and mark the node and move to the adjacent unmarked node and continue this loop until there is no unmarked adjacent node. Then backtrack and check for other unmarked nodes and traverse them. Finally, print the nodes in the path.*

Follow the below method to implement DFS traversal.

* **Step 1:** Create a set or array to keep track of visited nodes.
* **Step 2:** Choose a starting node.
* **Step 3:**Create an empty stack and push the starting node onto the stack.
* **Step 4:**Mark the starting node as visited.
* **Step 5:**While the stack is not empty, do the following:
  + Pop a node from the stack.
  + Process or perform any necessary operations on the popped node.
  + Get all the adjacent neighbors of the popped node.
  + For each adjacent neighbor, if it has not been visited, do the following:
    - Mark the neighbor as visited.
    - Push the neighbor onto the stack.
* **Step 6:** Repeat step 5 until the stack is empty.

Hpc2:

**Bubble Sort Algorithm**

*In this algorithm,*

* *traverse from left and compare adjacent elements and the higher one is placed at right side.*
* *In this way, the largest element is moved to the rightmost end at first.*
* *This process is then continued to find the second largest and place it and so on until the data is sorted.*

**Time Complexity:**O(N2)  
**Auxiliary Space:** O(1)

***Merge sort****is defined as a*[*sorting algorithm*](https://www.geeksforgeeks.org/sorting-algorithms/)*that works by dividing an array into smaller subarrays, sorting each subarray, and then merging the sorted subarrays back together to form the final sorted array.*

*Merge sort is a recursive algorithm that continuously splits the array in half until it cannot be further divided i.e., the array has only one element left (an array with one element is always sorted). Then the sorted subarrays are merged into one sorted array.*

O(N log(N))

Hpc 3

Parallel reduction refers to algorithms which combine an array of elements producing a single value as a result. Problems eligible for this algorithm include those which involve operators that are associative and commutative in nature. Some of them include

1. Sum of an array
2. Minimum/Maximum of an array

It works by using half the number of [threads](https://www.sciencedirect.com/topics/computer-science/thread) of the elements in the dataset. Every [thread](https://www.sciencedirect.com/topics/computer-science/thread) calculates the minimum of its own element and some other element. The resultant element is forwarded to the next round. The number of threads is then reduced by half and the process repeated until there is just a single element remaining, which is the result of the operation.

Hpc 5

Linear regression algorithm shows a linear relationship between a dependent (y) and one or more independent (y) variables, hence called as linear regression. Since linear regression shows the linear relationship, which means it finds how the value of the dependent variable is changing according to the value of the independent variable.

The linear regression model provides a sloped straight line representing the relationship between the variables



y= a0+a1x+ ε

Y= Dependent Variable (Target Variable)  
X= Independent Variable (predictor Variable)  
a0= intercept of the line (Gives an additional degree of freedom)  
a1 = Linear regression coefficient (scale factor to each input value).  
ε = random error

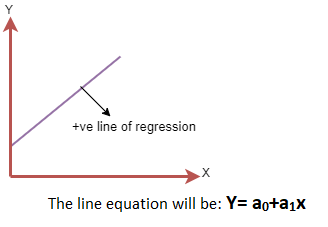
Linear regression can be further divided into two types of the algorithm:

* **Simple Linear Regression:**  
  If a single independent variable is used to predict the value of a numerical dependent variable, then such a Linear Regression algorithm is called Simple Linear Regression.
* **Multiple Linear regression:**  
  If more than one independent variable is used to predict the value of a numerical dependent variable, then such a Linear Regression algorithm is called Multiple Linear Regression.

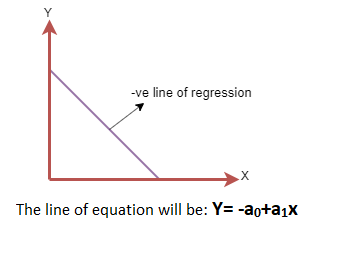
Linear Regression Line

A linear line showing the relationship between the dependent and independent variables is called a **regression line**. A regression line can show two types of relationship:

* **Positive Linear Relationship:**  
  If the dependent variable increases on the Y-axis and independent variable increases on X-axis, then such a relationship is termed as a Positive linear relationship.



* **Negative Linear Relationship:**  
  If the dependent variable decreases on the Y-axis and independent variable increases on the X-axis, then such a relationship is called a negative linear relationship.

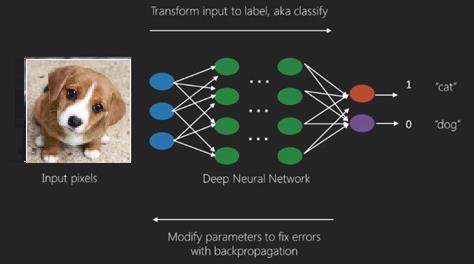


DL

A deep neural network (DNN) is an [artificial neural network](https://en.wikipedia.org/wiki/Artificial_neural_network) (ANN) with multiple layers between the input and output layers.[[10]](https://en.wikipedia.org/wiki/Deep_learning#cite_note-BENGIODEEP-10)[[13]](https://en.wikipedia.org/wiki/Deep_learning#cite_note-SCHIDHUB-13) There are different types of neural networks but they always consist of the same components: neurons, synapses, weights, biases, and functions.[[137]](https://en.wikipedia.org/wiki/Deep_learning#cite_note-Nokkada-137) These components as a whole function similarly to a human brain, and can be trained like any other ML algorithm. There are hidden layers in between which do all calculations

A deep neural network (DNN) is an ANN with multiple hidden layers between the input and output layers. Similar to shallow ANNs, DNNs can model complex non-linear relationships.

The main purpose of a neural network is to receive a set of inputs, perform progressively complex calculations on them, and give output to solve real world problems like classification.



Sentiment analysis, also known as opinion mining, is a computational technique used to determine and analyze the sentiment or emotion expressed in a piece of text, such as social media posts, customer reviews, or news articles. It involves the application of natural language processing (NLP) and machine learning algorithms to classify the sentiment of the text as positive, negative, or neutral.

The main objective of sentiment analysis is to extract subjective information from text data and understand the overall sentiment of a particular subject or topic. It has gained significant importance in various domains, including marketing, customer feedback analysis, brand management, and public opinion monitoring.

The process of sentiment analysis typically involves several steps. First, the text data is preprocessed by removing noise, such as special characters or stopwords, and tokenizing the text into individual words or phrases. Then, the sentiment analysis algorithm assigns a sentiment polarity to each word or phrase, classifying it as positive, negative, or neutral based on a predefined sentiment lexicon or a trained model. Finally, the sentiments of individual words or phrases are aggregated to determine the overall sentiment of the text.

Sentiment analysis can be approached using different techniques, including rule-based methods, machine learning models, or deep learning algorithms. Rule-based methods rely on predefined rules or dictionaries, while machine learning models and deep learning algorithms learn from labeled training data to predict sentiment. Common machine learning algorithms used for sentiment analysis include Naive Bayes, Support Vector Machines (SVM), and Recurrent Neural Networks (RNN).

Overall, sentiment analysis provides valuable insights into public opinion, customer feedback, and brand perception. By automating the analysis of large volumes of textual data, businesses can make data-driven decisions, monitor their reputation, improve customer satisfaction, and gain a competitive edge in the market.

CNNs are a class of Deep Neural Networks that can recognize and classify particular features from images and are widely used for analyzing visual images. Their applications range from image and video recognition, image classification, medical image analysis, computer vision and natural language processing.

The term ‘Convolution” in CNN denotes the mathematical function of convolution which is a special kind of linear operation wherein two functions are multiplied to produce a third function which expresses how the shape of one function is modified by the other. In simple terms, two images which can be represented as matrices are multiplied to give an output that is used to extract features from the image.



* **Input Layers:** It’s the layer in which we give input to our model. In CNN, Generally, the input will be an image or a sequence of images. This layer holds the raw input of the image with width 32, height 32, and depth 3.
* **Convolutional Layers:**This is the layer, which is used to extract the feature from the input dataset. It applies a set of learnable filters known as the kernels to the input images. The filters/kernels are smaller matrices usually 2×2, 3×3, or 5×5 shape. it slides over the input image data and computes the dot product between kernel weight and the corresponding input image patch. The output of this layer is referred ad feature maps. Suppose we use a total of 12 filters for this layer we’ll get an output volume of dimension 32 x 32 x 12.
* [**Activation Layer:**](https://www.geeksforgeeks.org/activation-functions-neural-networks/)By adding an activation function to the output of the preceding layer, activation layers add nonlinearity to the network. it will apply an element-wise activation function to the output of the convolution layer. Some common activation functions are **RELU**: max(0, x),  **Tanh**, **Leaky RELU**, etc. The volume remains unchanged hence output volume will have dimensions 32 x 32 x 12.
* [**Pooling layer:**](https://www.geeksforgeeks.org/cnn-introduction-to-pooling-layer/) This layer is periodically inserted in the covnets and its main function is to reduce the size of volume which makes the computation fast reduces memory and also prevents overfitting. Two common types of pooling layers are **max pooling** and **average pooling**.
* **Flattening:**The resulting feature maps are flattened into a one-dimensional vector after the convolution and pooling layers so they can be passed into a completely linked layer for categorization or regression.
* **Fully Connected Layers:**It takes the input from the previous layer and computes the final classification or regression task.
* **Output Layer:** The output from the fully connected layers is then fed into a logistic function for classification tasks like sigmoid or softmax which converts the output of each class into the probability score of each class.