# DEEP LEARNING

## UNIT - I: Feed Forward Neural Network

### Syllabus

**Introduction:** Various paradigms of learning problems, Perspectives and Issues in deep learning framework, review of fundamental learning techniques.

**Feed forward neural network:** Artificial Neural Network, activation function, multi-layer neural network.

# I) Introduction:

A **Feed forward Neural Network (FNN)** is one of the simplest types of artificial neural networks. It consists of multiple layers of neurons, where information flows in only one direction: forward, from the input layer through hidden layers and to the output layer. There are no cycles or loops in the network, hence the term "feedforward."

**Key Components:**

1. **Input Layer**: This layer receives the input data. Each neuron in this layer represents one feature of the input data.
2. **Hidden Layers**: These layers sit between the input and output layers. Each neuron in a hidden layer receives weighted inputs from the previous layer, applies an activation function, and passes the output to the next layer. Multiple hidden layers allow the network to learn complex representations.
3. **Output Layer**: This layer produces the final output, which could be a single value (e.g., for regression problems) or multiple values (e.g., for classification problems).
4. **Weights and Biases**: Each connection between neurons has a weight associated with it, which determines the strength of the connection. Additionally, each neuron has a bias that is added to the weighted sum of the inputs before passing it through the activation function.
5. **Activation Function**: The activation function introduces non-linearity into the model, enabling the network to learn complex patterns. Common activation functions include the sigmoid, hyperbolic tangent (tanh), and rectified linear unit (ReLU).

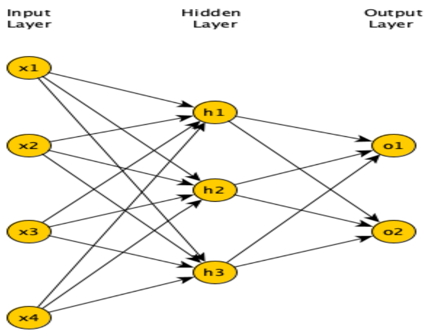
**Working:**

- In an FNN, data flows from the input layer, through the hidden layers, to the output layer.
- The network adjusts weights and biases during training using a process called **back propagation**, combined with an optimization algorithm like **gradient descent**.
- During training, the network minimizes the loss function, which measures the difference between the predicted output and the actual target.

**Applications:**

- Image recognition
- Natural language processing
- Time series prediction

- Regression and classification tasks




## 1.1 Various paradigms of learning problems:

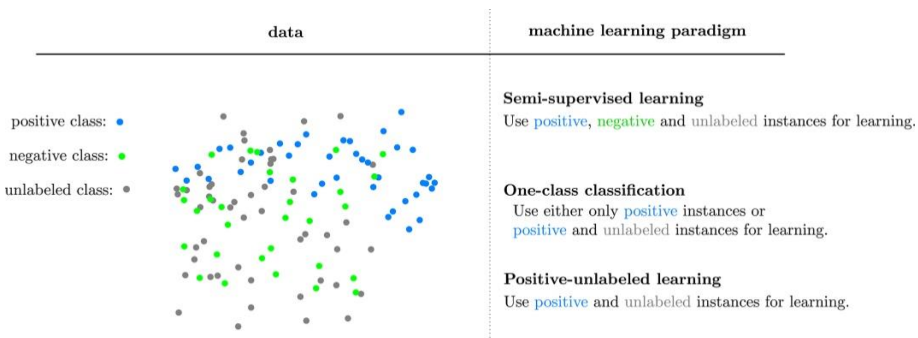
In Deep Learning (DL), different paradigms are used to categorize learning problems based on how data is structured and how models are trained.

- Hybrid Learning
- Composite Learning
- Reduced Learning

1. **Hybrid learning**:How can modern deep learning methods cross the boundaries between supervised and unsupervised learning to accommodate for a vast amount of unused unlabeled data?
   **Hybrid Learning Problems:**
   - Semi-Supervised Learning
   - Self-Supervised Learning
   - Multi-Instance Learning

   i. **Semi-Supervised Learning:**

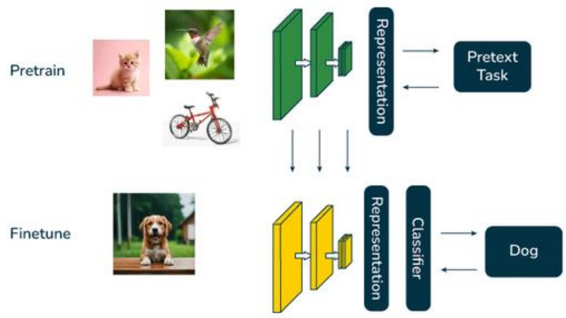
   - **Description**: This paradigm combines a small amount of labeled data with a large amount of unlabeled data.
   - **Goal**: Improve learning performance by leveraging both labeled and unlabeled data.
   - **Examples**: Label Propagation, Semi-Supervised GANs.




   ii. **Self-Supervised Learning:**

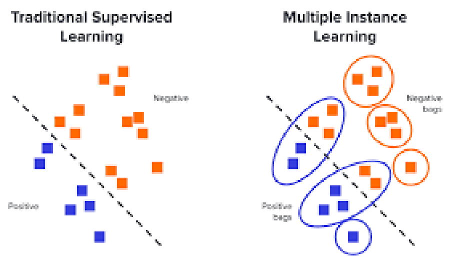
   - **Description:** Self-supervised learning uses unlabeled data to predict missing or altered parts of the input for model training.

- **Goal:** To learn useful data representations from unlabeled data and reduce the need for large labeled datasets.
- **Example:** Image in painting predicts missing parts of an image; NLP models predict missing words in sentences.




### iii. Multi-Instance Learning:

- **Description:** Multi-Instance Learning (MIL) trains a model on labeled sets (bags) of instances where individual instance labels are unknown.
- **Goal:**To classify bags based on instance patterns, even without knowing the labels of individual instances.
- **Example:**In drug activity prediction, a bag represents a molecule's multiple conformations, and the goal is to predict whether any conformation is active.




## 2. Composite Learning:

- **Description:** Composite learning integrates multiple learning methods and sources to enhance overall understanding.
- **Goal:** To create a holistic learning experience by combining various educational approaches for deeper comprehension.
- **Example:** Using a mix of textbooks, online courses, and hands-on projects to master a programming language.

## 3. Reduced learning:

- **Description**:Reduced learning involves minimizing the amount of data or input required to achieve effective learning.

- **Goal:** To streamline learning by focusing on essential information and eliminating unnecessary complexity.
- **Example:** Utilizing condensed study guides to grasp key concepts of a subject efficiently.

# 1.2 Perspectives and Issues in deep learning framework

## Perspectives:

**1. Flexibility vs. Ease of Use**: High-level frameworks are easy to use but less flexible, while low-level frameworks offer more control. This trade-off depends on your project needs.
**2. Scalability:** Frameworks like TensorFlow scale efficiently for large datasets and complex models. They are designed to work across multiple devices.
**3. Community Support:** Strong community support aids learning and troubleshooting, with frameworks like PyTorch and TensorFlow having extensive resources. This boosts innovation and quick fixes.
**4. Deployment:** Some frameworks, like TensorFlow Lite, support easy deployment on cloud, mobile, and edge devices. They offer cross-platform compatibility.
**5. Research**: PyTorch is favored for its dynamic graph, making experimentation easier. Researchers prefer it for flexibility in designing new architectures.

## Issues:

**1. Deployment Challenges**: Transitioning models from development to production is often difficult and time-consuming. Optimizing for different environments adds complexity.
**2. Interpretability**: Deep learning models are powerful but hard to interpret, which raises concerns in fields requiring clear explanations. This "black box" nature limits trust.
**3. Debugging:** Debugging deep models is challenging due to the large number of parameters and layers. Even with tools, issues like vanishing gradients persist.
**4. Resource Needs:** Deep learning demands high computational power, like GPUs, making it resource-intensive. This can limit access for smaller organizations.
**5. Bias:** Models trained on biased data may perpetuate unfair outcomes, posing ethical issues. Handling bias requires careful attention to training data.
**6. Security:** Deep learning models are vulnerable to adversarial attacks, where small input changes lead to incorrect predictions. Securing models remains a priority.
**7. Privacy:** Using sensitive data introduces privacy risks during training, especially in healthcare and finance. Data privacy techniques are still evolving.

# 1.3 Review of fundamental learning techniques:

Deep learning is useful for data scientists who are responsible for gathering, analyzing, and understanding massive volumes of data.

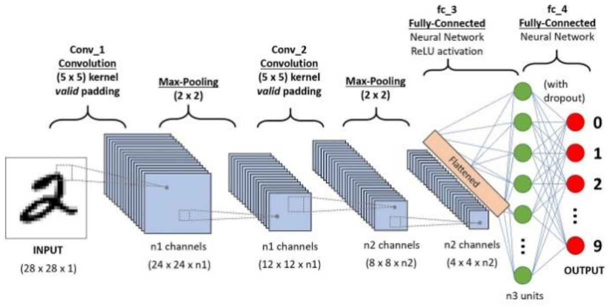
**Deep Learning Techniques:**

1. **Classic Neural Networks:**
   Also known as Fully Connected Neural Networks, it is often identified by its multilayer perceptrons, where the neurons are connected to the continuous layer.
   Functions included in this model are:

- **Linear function:** it represents a single line which multiplies its inputs with a constant multiplier.
- **Non-Linear function:** A nonlinear function is a function whose plotted graph does not form a straight line but a curved line.

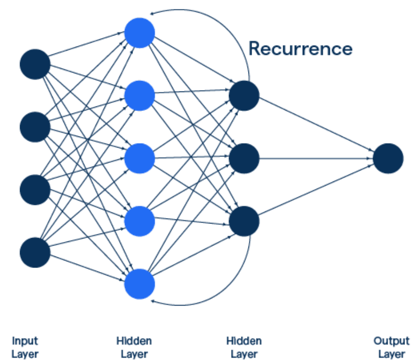
**2 Convolutional Neural Networks (CNN):**

A Convolutional Neural Network (CNN) is a specialized deep learning model designed to process grid-like data, such as images, by applying convolutional filters to detect features. It automatically learns hierarchical feature representations through multiple layers, reducing the need for manual feature extraction. CNNs are commonly used in tasks like image recognition, object detection, and image segmentation.



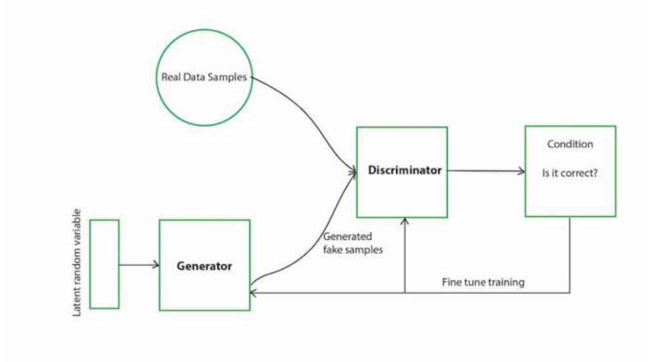

**3 Recurrent Neural Networks (RNN):**

A Recurrent Neural Network (RNN) is a type of neural network designed for processing sequential data by maintaining a state that captures information from previous inputs. It uses loops to allow information to persist across time steps, making it suitable for tasks like time series prediction and natural language processing. RNNs are effective for sequences where context and order are crucial.



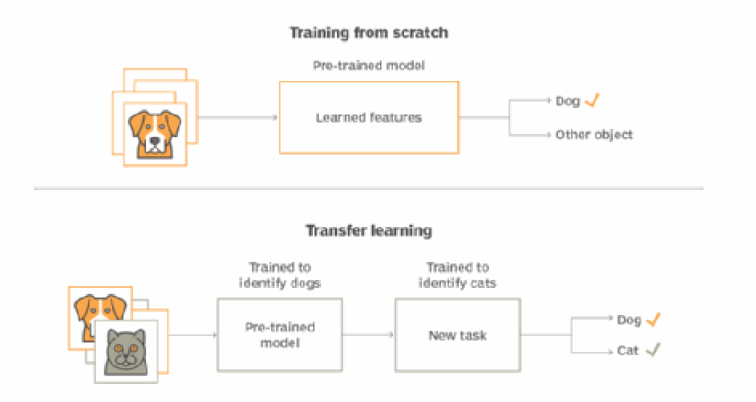

**4 Generative Adversarial Networks:**

Generative Adversarial Networks (GANs) consist of two neural networks—a generator and a discriminator—that compete against each other to produce and evaluate realistic data. The generator creates synthetic data, while the discriminator assesses its authenticity, leading to improved quality of generated data over time. GANs are widely used for tasks such as image synthesis, style transfer, and data augmentation.




## 5 Transfer Learning :

Transfer learning involves taking a pre-trained model on one task and adapting it to a new but related task, leveraging learned features to improve performance. This approach reduces the need for extensive training data and computational resources for the new task. It is commonly used in scenarios where labelled data is scarce but a similar, well-trained model is available.



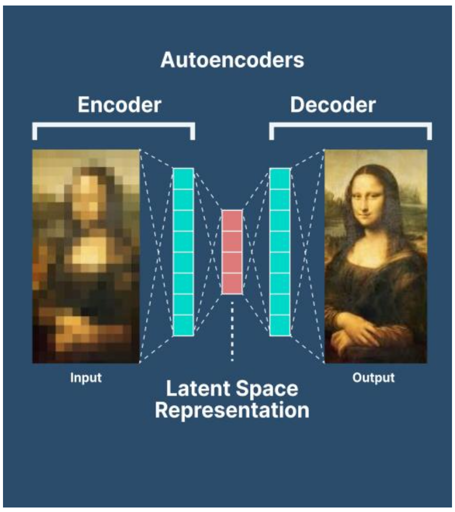

## 6 Auto encoders:

These are self-supervised machine learning models which are used to reduce the size of input data by recreating it.

Auto encoder is made up of two components:

**Encoder:** It works as a compression unit that compresses the input data.

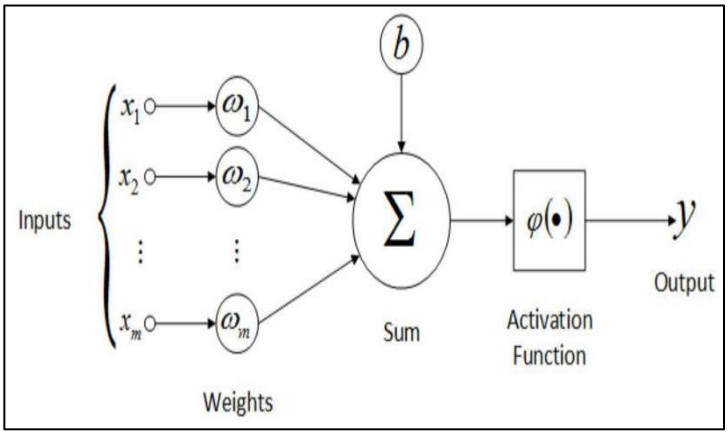
**Decoder:** It decompresses the compressed input by reconstructing it.



# II) Feed Forward Neural Network:

A Feedforward Neural Network is a type of neural network in which information flows in one direction only—from the input nodes, through any hidden layers, to the output nodes. It is used to approximate functions and perform tasks such as classification and regression. The network consists of layers of neurons, with each layer fully connected to the next layer.

- Feed-Forward Neural Network is a single layer perceptron.
- A sequence of inputs enter the layer and are multiplied by the weights in this model.
- The weighted input values are then summed together to form a total.
- If the sum of the values is more than a predetermined threshold, which is normally set at zero, the output value is usually 1, and if the sum is less than the threshold, the output value is usually -1.
- The single-layer perceptron is a popular feed-forward neural network model that is frequently used for classification.




**Single layer perceptron**

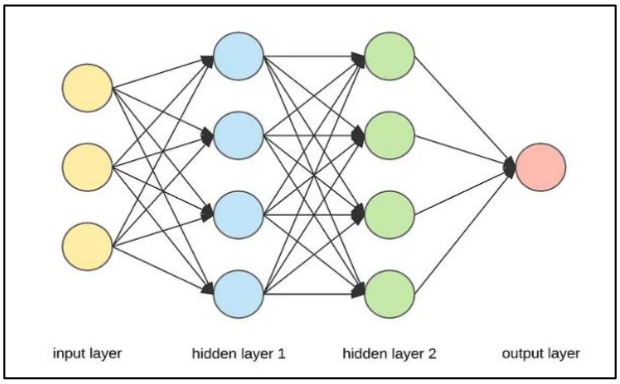
# 1.4 Artificial Neural Network:

- Artificial Neural Network (ANN) is a deep learning algorithm that emerged from the idea of Biological Neural Networks of human brains.
- ANN works very similar to the biological neural networks.
- ANN algorithm would accept only numeric and structured data as input.
- To accept unstructured and non-numeric data formats such as Image, Text, andSpeech, **Convolutional Neural Networks (CNN),** and **Recursive Neural Networks (RNN)** are used respectively.
- An example of a Deep Neural Network is

## Architecture of Artificial Neural Network(ANN):

### Input layer

- The input layer consists of inputs that are independent variables. These inputs can be loaded from an external source such as a web service or a CSV file.
- In simple terms, these variables are known as features




**Artificial Neural Network (ANN)**

## Weights

- Weights play an important role in Neural Network, every node/neuron has some weights.
- Neural Networks learn through the weights, by adjusting weights the neural networks decide whether certain features are important or not.

## Hidden Layer

- These lie between the input layer and the output layer.
- In this layer, the neurons take in a set of weighted inputs and produce an output with the help of the activation function.
- In this step, we apply activation function, these neurons apply different transformations to the input data.
- There are many activation functions used in deep learning some of them include ReLU, Threshold Function, Sigmoid.

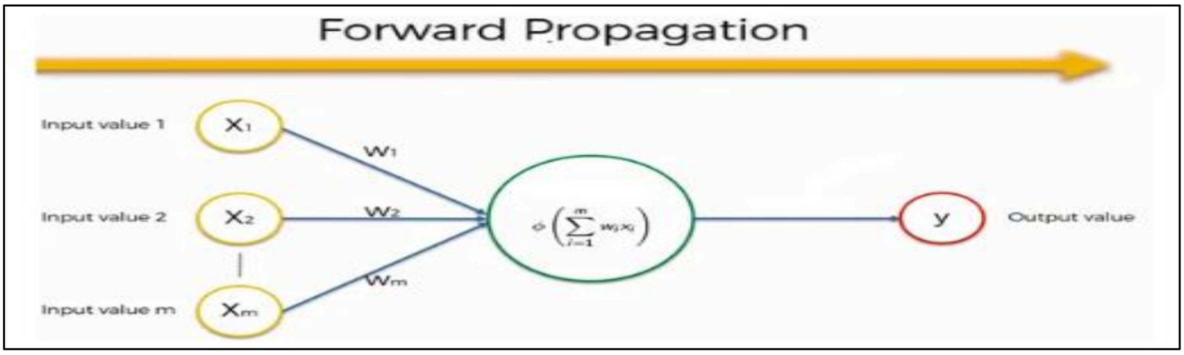
## Output Layer

- This is the last layer in the neural networks and receives input from the last node in the hidden layer. This layer can be
- Continous (stock price)
- Binary (0 or 1)
- Categorical (Cat or Dog or Duck.

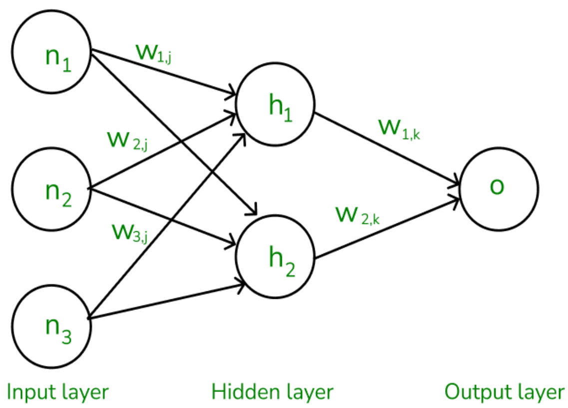
In the context of Artificial Neural Networks (ANNs), forward propagation, backward propagation, and gradient descent are interrelated processes used for training and optimizing the network.

- There are two phases in the Neural Network cycle, one is the training phase and the other is the prediction phase.
- The process of finding the weight and bias values occurs in the **training phase**.
- The process where the neural network processes our input to produce predictions comes under the **prediction phase.**
- Consider the learning process of a neural network as an iterative process of the pass and return.
- Pass is a process of Forward Propagation of Information and return is the Backward Propagation of Information.

- **Forward Propagation**:




- In Forward Propagation, Given some data, we compute the dot product of that input value with the assigned weight and then add all those and apply the activation function to the result in the hidden layer.
- This node acts as an input layer for the next layer. This is repeated until we get the final output vector ***y***.
- The obtained output value is known as a predicted value.



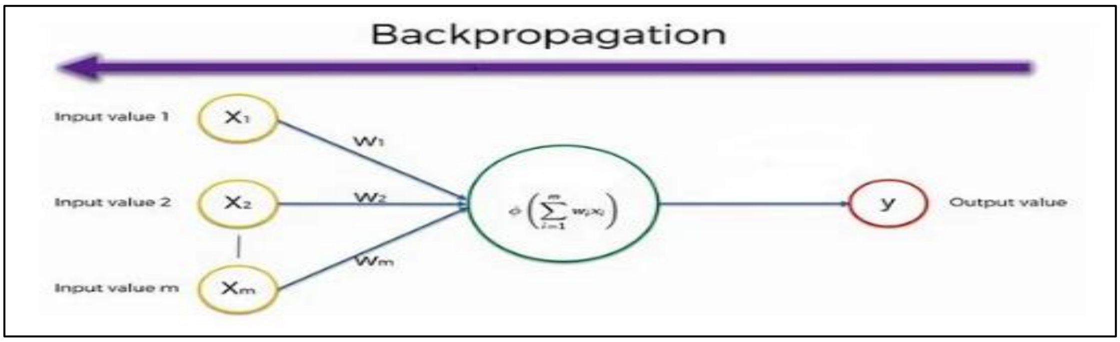

- we compare the predicted value with the actual value, the difference is known as an error which is known as **Cost Function.**
- **Loss Function** is inversely proportional to accuracy, less the cost function, more is the accuracy, our goal is to minimize the loss function.

The formula for loss function:

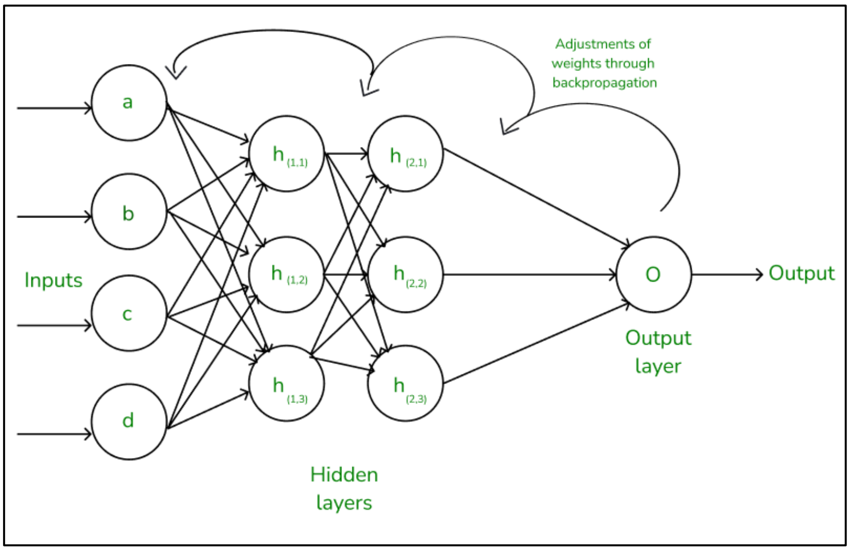
$$C = \frac{1}{2}(\hat{y} - y)^2$$

- After calculation the Loss Function, we feed this information back to Neural Network where it travels back through weights and weights are updated, this method is called **Back Propagation**.
- This process is repeated several times so that the machine understands the data and the weights for the features.

- **Backward Propagation :**




- Backward propagation (or backpropagation) is used to update the weights of the network based on the error between the predicted output (from forward propagation) and the actual target values
- During backward propagation, the error is propagated backward through the network from the output layer to the input layer. This involves calculating the gradient of the loss function with respect to each weight in the network. The gradients indicate how much each weight contributed to the error, which is essential for adjusting the weights to minimize the error.



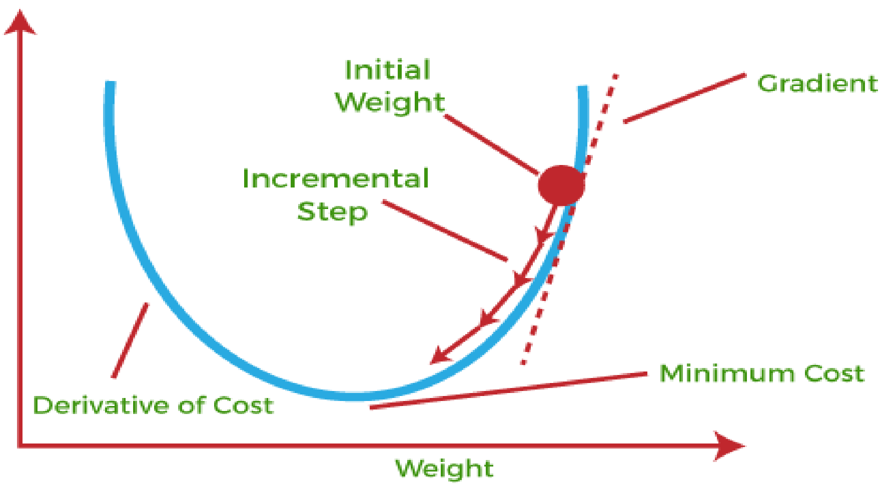

## Gradient Descent:

- It is an optimization technique that is used to improve the neural network-based models by minimizing the cost function.
- This process occurs in the **backpropagation step.**
- It allows us to adjust the weights of the features in order to reach the global minima.
- A global minimum is a point where the function value is smaller than at all other feasible points.

- Using the gradients obtained from backpropagation, gradient descent adjusts the weights in the direction that reduces the loss. The update rule for each weight is generally of the form:

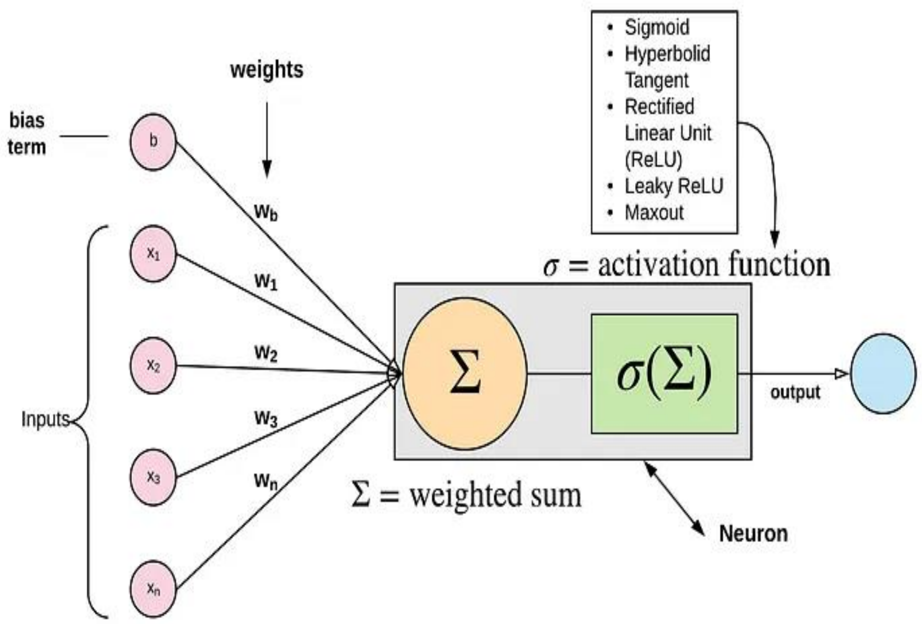
$$w_{new} = w_{old} - \eta \cdot \frac{\partial L}{\partial w}$$

where η is the learning rate, and ∂L/∂w is the gradient of the loss function with respect to the weight w.




## 1.5 Activation Function :

- It's just a function that is used to get the output of node, It is also known as Transfer Function.
- It is used to determine the output of neural network like yes or no. It maps the resulting values in between 0 to 1 or -1 to 1 etc. (depending upon the function).
- The activation function decides whether a neuron should be activated or not by calculating the weighted sum and further adding bias to it. The purpose of the activation function is to introduce non-linearity into the output of a neuron.

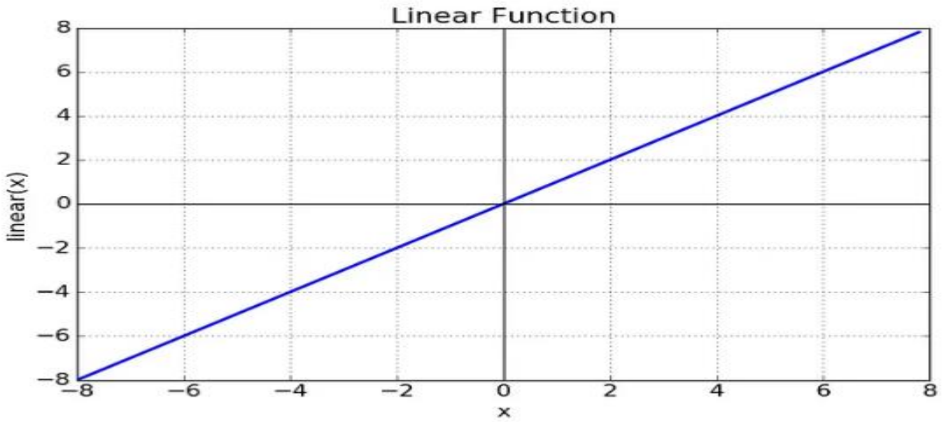



- The Activation Functions can be basically divided into 2 types-

Linear Activation Function

Non-linear Activation Functions

- Linear Activation Function:
  The linear activation function, also known as **"no activation,"** or "**identity function",** is one of the **most straightforward activation functions**, where the **output is identical to the input**.



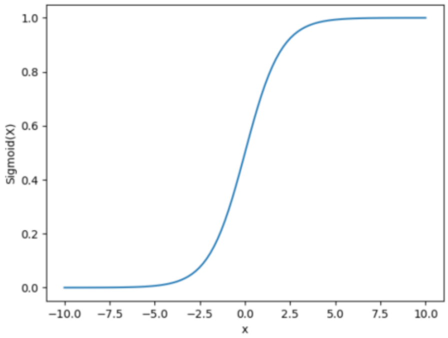

- Non-linear Activation Functions:
  A **nonlinear function** used to **send the output signal either on or off** to **a neuron** is known as an **activation function.**

**There are different types of Non Linear Activation Functions:**

- (a) Sigmoid Activation Function
- (b) Tanh Activation Function
- ( c) ReLU Activation Function

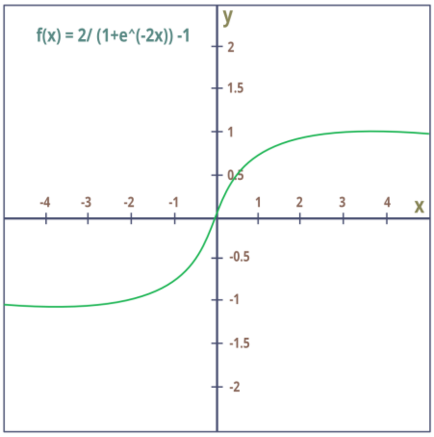
  Leaky ReLu Activation Function
- (d) Softmax Activation Funcion

1. **Sigmoid Function**:




- It is a function which is plotted as **'S'** shaped graph.
- **Equation :** A = 1/(1 + e^-x)
- **Nature :** Non-linear. Notice that X values lies between -2 to 2, Y values are very steep. This means, small changes in x would also bring about large changes in the value of Y.
- **Value Range :** 0 to 1
- **Uses :** Usually used in output layer of a binary classification, where result is either 0 or 1, as value for sigmoid function lies between 0 and 1 only so, result can be predicted easily to be ***1*** if value is greater than **0.5** and ***0*** otherwise.

## 2. Tanh Function:

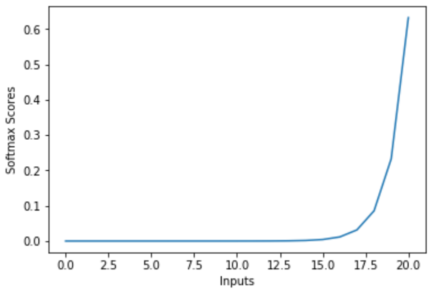



- The activation that works almost always better than sigmoid function is Tanh function also known as **Tangent Hyperbolic function**. It's actually mathematically shifted version of the sigmoid function. Both are similar and can be derived from each other.
- **Equation :-**
  f(x) = tanh(x) = 2/(1 + e-2x) – 1
  OR
  tanh(x) = 2 * sigmoid(2x) – 1
- **Value Range :-** -1 to +1
- **Nature :-** non-linear
- **Uses :-** Usually used in hidden layers of a neural network as it's values lies between **-1 to 1** hence the mean for the hidden layer comes out be 0 or very close to it, hence helps in *centering the data* by bringing mean close to 0. This makes learning for the next layer much easier.

## 3. RELU Function:

- It Stands for *Rectified linear unit*. It is the most widely used activation function. Chiefly implemented in *hidden layers* of Neural network.
- **Equation :-** ***A(x) = max(0,x)***. It gives an output x if x is positive and 0 otherwise.
- **Value Range :-** [0, inf)
- **Nature :-** non-linear, which means we can easily backpropagate the errors and have multiple layers of neurons being activated by the ReLU function.
- **Uses :-** ReLu is less computationally expensive than tanh and sigmoid because it involves simpler mathematical operations. At a time only a few neurons are activated making the network sparse making it efficient and easy for computation.
  In simple words, RELU learns *much faster* than sigmoid and Tanh function.

## 4. Softmax Function:

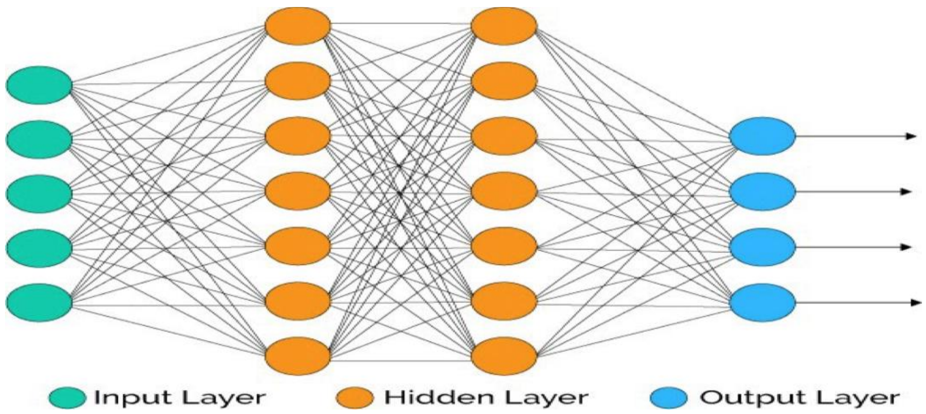


The softmax function is also a type of sigmoid function but is handy when we are trying to handle multi- class classification problems.

- **Nature :-** non-linear
- **Uses :-** Usually used when trying to handle multiple classes. the softmax function was commonly found in the output layer of image classification problems.The softmax function would squeeze the outputs for each class between 0 and 1 and would also divide by the sum of the outputs.
- **Output:-** The softmax function is ideally used in the output layer of the classifier where we are actually trying to attain the probabilities to define the class of each input.
- The basic rule of thumb is if you really don't know what activation function to use, then simply use *RELU* as it is a general activation function in hidden layers and is used in most cases these days.
- If your output is for binary classification then, *sigmoid function* is very natural choice for output layer.
- If your output is for multi-class classification then, Softmax is very useful to predict the probabilities of each classes.

## 1.6 Multi-Layer Neural Network:

- Multi-Layer Neural Network is a type of neural network with an architecture consisting of input, hidden, and output layers of interconnected neurons. It is capable of learning complex patterns and performing tasks such as classification and regression by adjusting its parameters through training.
- A fully connected Multi-Layer Neural Network is called a Multilayer Perceptron (MLP)




- A multi-layered perceptron model has considered as artificial neural networks having various layers in which activation function does not remain linear
- In this Networks, we used the Non Linear Activation Functions used.ie, Sigmoid, Relu, Tanh etc. Since the probability of any event lies between 0 and 1

- **Applications**:

  **Image Recognition** : Multi-layer neural networks, particularly Convolutional Neural Networks (CNNs), are widely used for tasks like image classification and object detection.

**Natural Language Processing**: Recurrent Neural Networks (RNNs) and Transformers, which are types of multi-layer neural networks, are used for tasks such as machine translation, sentiment analysis, and text generation.

**Speech Recognition**: Used to transcribe spoken language into text and for voice-activated assistants

- **Advantages :**

- **Complex Pattern Recognition**: Can model and recognize complex patterns in data by learning multi-level abstractions.
- **Flexibility**: Applicable to a wide range of tasks including image, text, and speech processing.

  - **Description:** Composite learning integrates multiple learning methods and sources to enhance overall understanding.
  - **Goal:** To create a holistic learning experience by combining various educational approaches for deeper comprehension.
  - **Example:** Using a mix of textbooks, online courses, and hands-on projects to master a programming language.