DEEP LEARNING

Module No: 01

Fundamentals of Neural Network

Traditional Programming

- Traditional Programming : We feed in DATA (Input) + PROGRAM (logic), run it on machine and get output. E.g. C Program to check the number is even or odd

```
#include <stdio.h>
int main() {
    int num;
    printf("Enter an integer: ");    INPUT
    scanf("%d", &num);

    // true if num is perfectly divisible by 2
    if(num % 2 == 0)    LOGIC
        printf("%d is even.", num);
    else                                        OUTPUT
        printf("%d is odd.", num);

    return 0;
}
```

Machine Learning

- Machine Learning : We feed in DATA(Input) + Output, run it on machine during training and the machine creates its own program(logic), which can be evaluated while testing.
- e.g. feed-in customer information/loan transactions (input) and how many defaulted on the loan (observed output), and it will create a model to predict who will default on the loan.

TP Vs ML

- Basic Difference in ML and Traditional Programming?

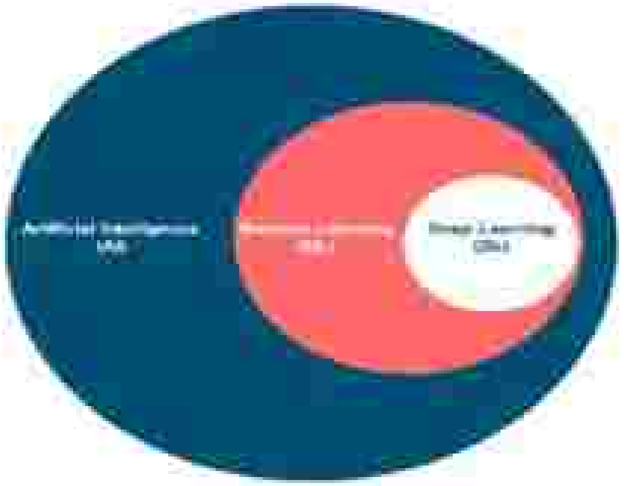
A biological neuron or Neuron

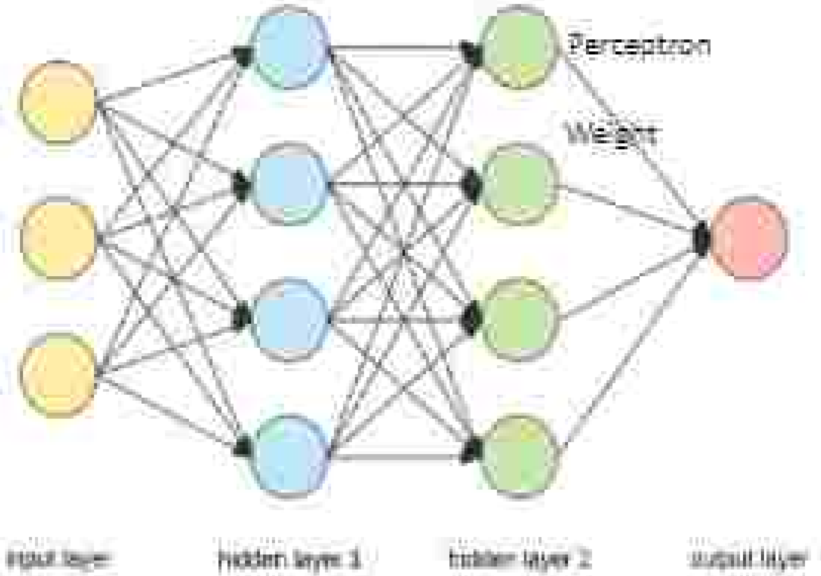
- A biological neuron/neuron, is the fundamental building block of the nervous system in living organisms, including humans.
- Neurons are specialized cells responsible for transmitting information through electrical and chemical signals.
- They are the basic units that form complex networks in the brain and other parts of the nervous system.

What is Deep Learning?

General Definition:

- Deep learning is a subfield of artificial intelligence and machine learning that is inspired by the structure of a human brain.
- Deep learning algorithms attempt to draw similar conclusions as humans would, by continuously analyzing data with a given logical structure called neural network.

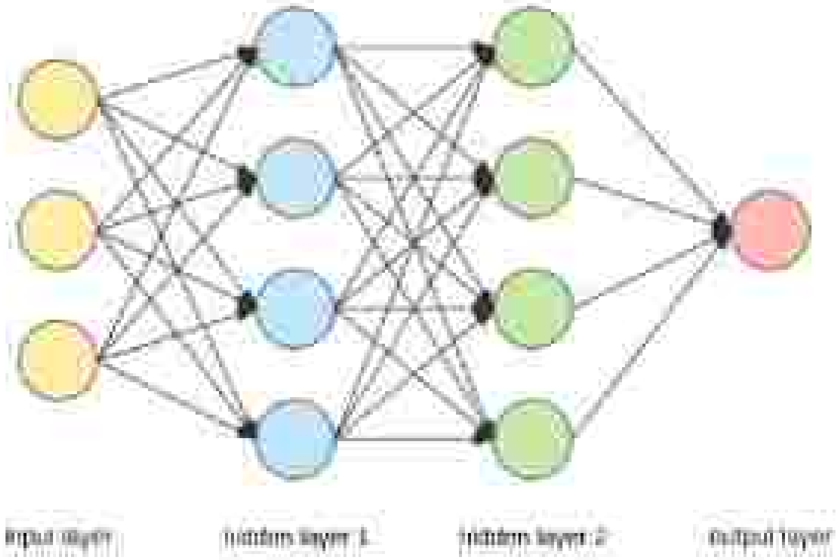




What is Deep Learning?

Technical Definition:

- Deep learning is a part of a broader family of machine learning methods based on artificial neural networks with representation learning. *(In machine learning, feature learning or* ***representation learning*** *is a set of techniques that allows a system to automatically discover the representations needed for feature detection or classification from raw data. This replaces manual feature engineering and allows a machine to both learn the features and use them to perform a specific task.)*
- Deep learning algorithms uses multiple layers to progressively extract higher-level features from the raw input.
- For example, in image processing, lower layers may identify edges, while higher layers may identify the concepts relevant to a human, such as digits or letters or faces.



Types of Neural Networks

Types included in our syllabus

- **ANN** (Artificial Neural Network): *simple form of NN*

  A computational model inspired by the human brain, comprising interconnected nodes (neurons) organized into layers. It is used for learning and modeling complex relationships within data.

- **CNN** (Convolutional Neural Network):*works best on Images*

  A specialized type of neural network designed for processing grid-like data, such as images. CNNs use convolutional layers to automatically learn hierarchical features.

- **RNN** (Recurrent Neural Network):*works best on speech/text data*

  A neural network architecture designed to handle sequential data by incorporating feedback loops. RNNs are suitable for tasks where the order of data elements is crucial.

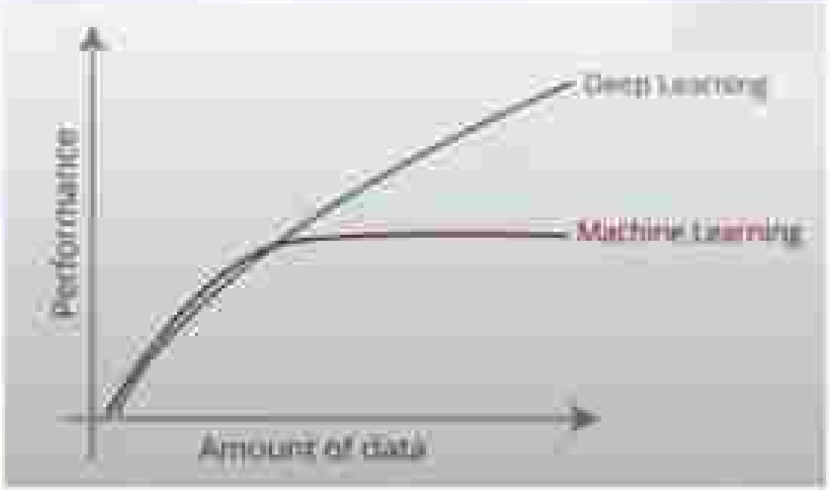
- **GAN** (Generative Adversarial Network):*works best to generate Images/Text*

  A type of neural network architecture consisting of two networks, a generator and a discriminator, which are trained adversarially. GANs are used for generating realistic synthetic data, such as images or text.

Deep Learning Vs. Machine Learning

Basic difference:

- Data Dependency: Deep learning algorithms require a considerably larger volume of data compared to machine learning algorithms.
- Hardware Dependency: Deep learning algorithms typically demand the utilization of Graphics Processing Units (GPUs) for efficient execution, while machine learning algorithms can run on Central Processing Units (CPUs).



Deep Learning Vs. Machine Learning

- Training Time: Deep learning algorithms often entail an extended duration for training, ranging from days to weeks to train a model. In contrast, machine learning models typically have shorter training times, often measured in hours. Additionally, the inference or prediction time in deep learning is notably shorter than that in machine learning.
- Feature Selection: In deep learning, algorithms autonomously extract features from the data where as in ML you required to extract features manually.(e.g. resume selection)

Deep Learning Vs. Machine Learning

- Interpretability:
  - ✓ E.g. Consider a scenario where a deep learning model is trained for a social networking site to determine whether to ban a user based on their comments. If a user inquires about the reason for their ban, the deep learning model may struggle to provide a comprehensible explanation.
  - ✓ Deep learning models often lack interpretability, functioning as "black boxes" wherein the internal processes leading to a prediction are challenging to decipher.
  - ✓ In contrast, many machine learning models offer a higher degree of interpretability, allowing users to understand and explain the rationale behind their predictions.

Deep Learning Vs. Machine Learning

- Q. Differentiate between DL and ML using following points?
  1. Data Dependency
  2. Hardware Dependency
  3. Training Time
  4. Feature Selection
  5. Interpretability

Why is Deep Learning so popular and in demand these days?

- Datasets:
  - Smartphone revolution
  - Internet pricing revolution
  - Conversion of unlabeled data into labeled data by big companies
  - The availability of massive datasets in various domains, such as image, text, and speech, have played a crucial role in the success of deep learning.
  - Public datasets
    - ✓ Images: Microsoft COCO
    - ✓ Video: Youtube 8M (6.1 Million videos)
    - ✓ Text: SquAD by Wikipedia
    - ✓ Audio: Google Audioset (20L sound clips of 600 categories)
- Frameworks:
  - Open Source Frameworks: The development and accessibility of powerful deep learning frameworks, such as TensorFlow and PyTorch, have significantly contributed to the popularity of deep learning.
  - These frameworks simplify the implementation of complex models, offering pre-built modules and efficient tools for model development and training.

Why is Deep Learning so popular and in demand these days?

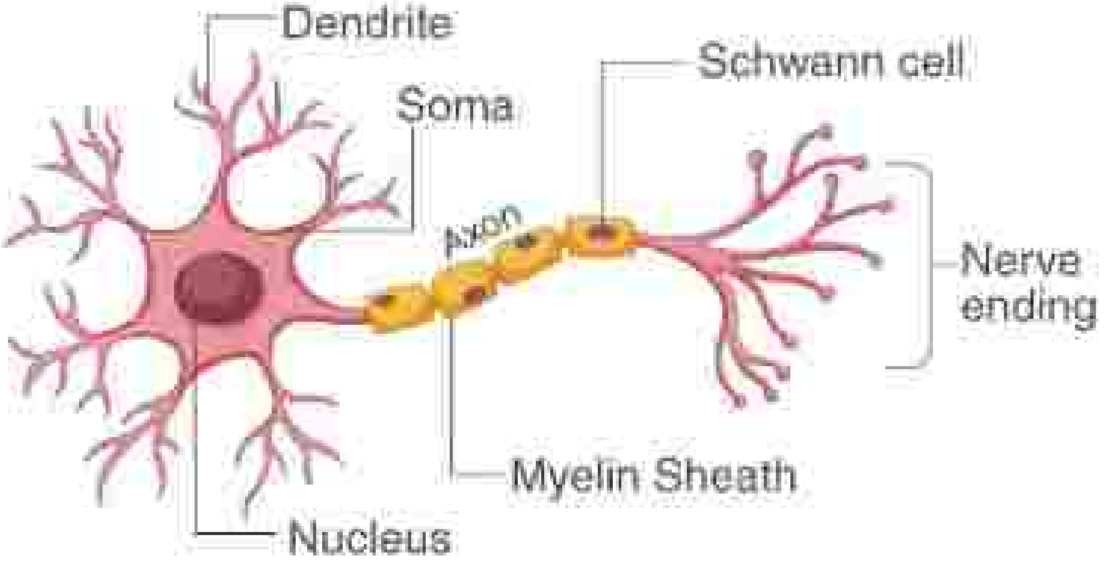
- Model Architecture:
  - Complex Model Capabilities: Deep learning's success can be attributed to its ability to automatically learn hierarchical representations and features from data.
  - The architecture of deep neural networks, with multiple layers, enables them to capture intricate relationships in the data, making them suitable for a wide range of tasks, from image recognition to natural language processing.
- Hardware:
  - Advancements in GPU Technology: The parallel processing capabilities of Graphics Processing Units (GPUs) have greatly accelerated the training of deep learning models.
  - GPU technology, especially when compared to traditional Central Processing Units (CPUs), enables deep learning practitioners to train complex models in significantly shorter time frames.
- Community:
  - Collaborative Knowledge Sharing: The strong and active deep learning community has fostered collaborative knowledge sharing.
  - Researchers, practitioners, and enthusiasts contribute to the field through open-source projects, research papers, and forums.
  - This collaborative ecosystem has accelerated the pace of innovation and made deep learning more accessible.

Why is Deep Learning so popular and in demand these days?

- Q. Why is Deep Learning so popular and in demand these days? Explain using following points.
  1. Datasets
  2. Frameworks
  3. Model Architecture
  4. Hardware
  5. Community

A biological neuron or Neuron

- A biological neuron/neuron, is the fundamental building block of the nervous system in living organisms, including humans.
- Neurons are specialized cells responsible for transmitting information through electrical and chemical signals.
- They are the basic units that form complex networks in the brain and other parts of the nervous system.



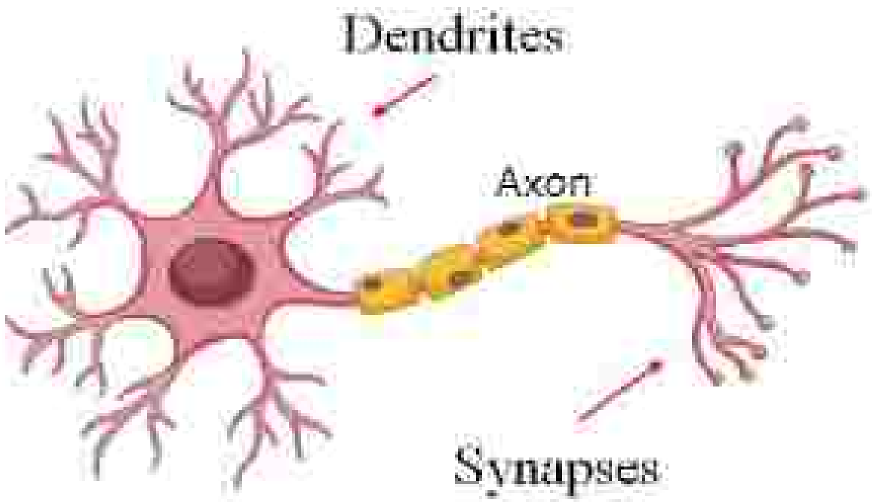
A biological neuron or Neuron

- Neuron Structure
  - A neuron varies in shape and size depending on its function and location. All neurons have three different parts – dendrites, cell body and axon.
- Parts of Neuron
  - Dendrites
    - These are branch-like structures that receive messages from other neurons and allow the transmission of messages to the cell body.
  - Cell Body
    - Each neuron has a cell body with a nucleus, Golgi body, endoplasmic reticulum and other components.
  - Axon
    - Axon is a tube-like structure that carries electrical impulse from the cell body to the axon terminals that pass the impulse to another neuron.
  - Synapse
    - It is the chemical junction between the terminal of one neuron and the dendrites of another neuron.

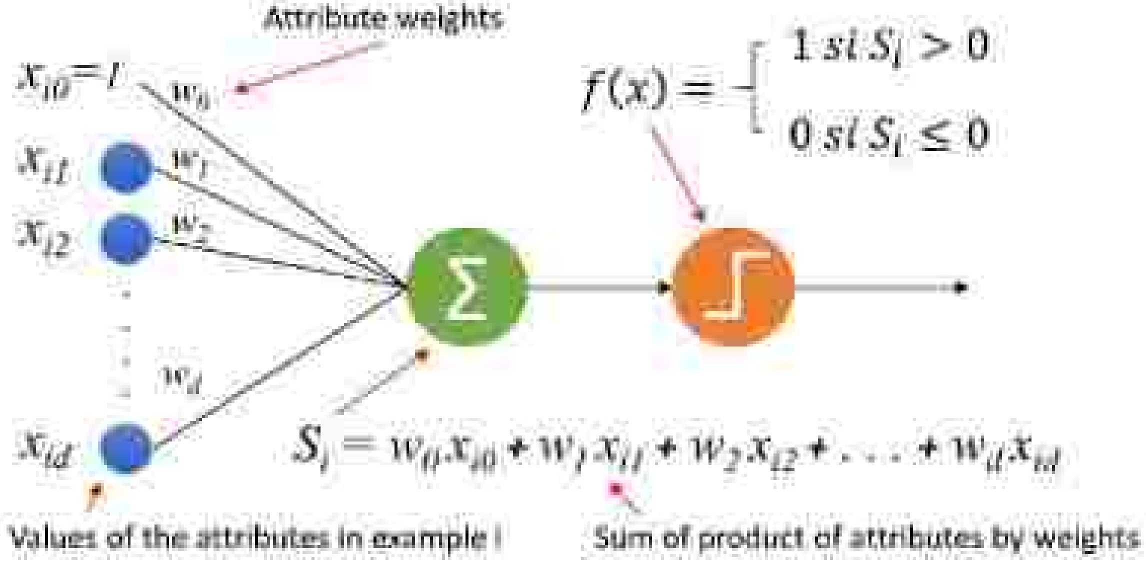
The McCulloch-Pitts neuron

- The McCulloch-Pitts neuron
  - The McCulloch-Pitts neuron is a mathematical model of a simplified artificial neuron, proposed by Warren McCulloch and Walter Pitts in 1943.
  - It is a formalization of how they believed biological neurons function.
  - The McCulloch-Pitts neuron is a binary threshold unit that takes multiple binary inputs, applies weights to them, and produces a binary output based on a threshold.

A biological neuron and McCulloch-Pitts neuron



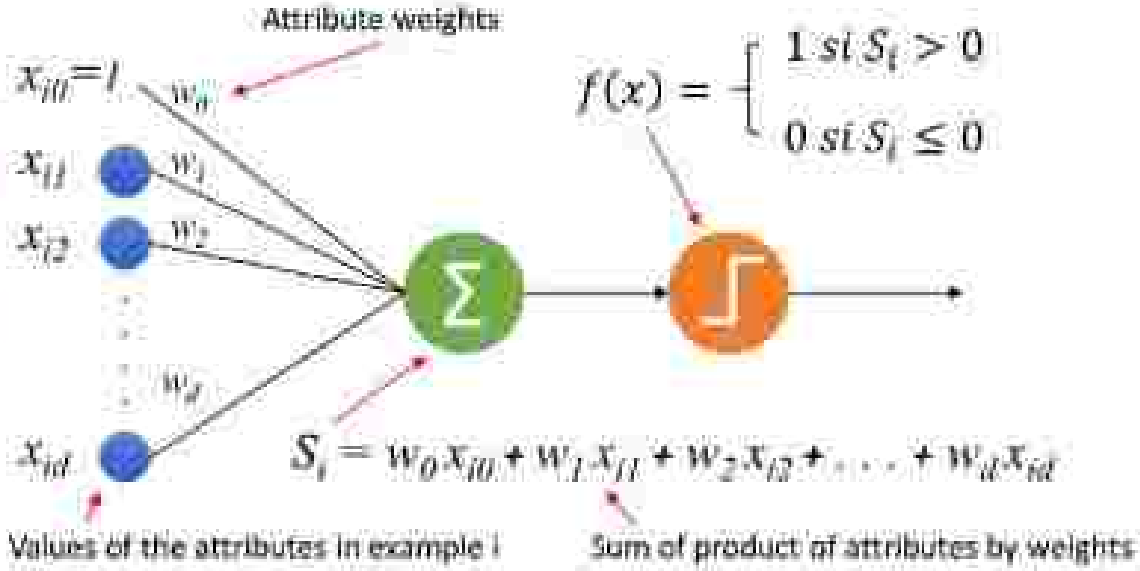
NEURON



PERCEPTRON

Perceptron

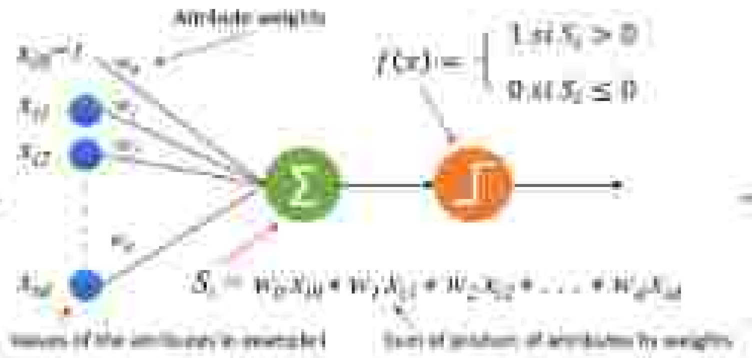
- The 'perceptron' is a mathematical model/function or an algorithm, like linear regression, logistic regression, and support vector machines (SVM).
- Similar to these algorithms, the perceptron is used for binary classification tasks under supervised learning.



PERCEPTRON

- A perceptron is a type of artificial neuron or the simplest form of a neural network.
- It was introduced by Frank Rosenblatt in 1957.
- The perceptron is the fundamental building block of neural networks and serves as a binary classifier.
- It takes multiple binary inputs, processes them with weights, sums them up, and produces a binary output.

Perceptron



PERCEPTRON

- Here are the key components of a perceptron:
  - Inputs (x): Each input represents a feature, and it can take a binary value (0 or 1) or a real value.
  - Weights (w): Each input is associated with a weight. Weights represent the importance of the respective input. A higher weight means the input is more influential in determining the output.
  - Summation Function (∑): The inputs are multiplied by their corresponding weights, and the products are summed up.

$$\text{Sum} = \sum_{i=1}^{n} (x_i \cdot w_i)$$

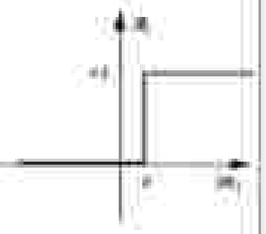
  - Activation Function (Step Function): The sum is then passed through an activation function. The traditional activation function for a perceptron is a step function, which outputs 1 if the sum is above a certain threshold and 0 otherwise.

$$\text{Output} = \begin{cases} 1, & \text{if Sum} \geq \text{Threshold} \\ 0, & \text{otherwise} \end{cases}$$

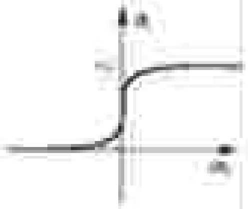
Limitations of Perceptron

1. Linear or Sort of linear Data Only:
    - The perceptron is limited to classifying data that is linearly(or sort of linearly) separable.
2. Inability to Handle Nonlinear Data:
    - Perceptrons fail to effectively classify non-linear data.
3. Restricted to Binary Classification:
    - The perceptron is specifically designed for binary classification tasks, meaning it can distinguish between only two classes.

Types of activation function



(a) Step function



(c) Sigmoid function

In a perceptron or a neural network, activation functions play a crucial role by introducing non-linearity to the model.

Here are some common types of activation functions used in perceptrons:

1. **Step Function:**
   - Description: The step function is a binary activation function. If the input is above a certain threshold, it outputs one; otherwise, it outputs zero.
   - Mathematical Form:

$$f(x) = \begin{cases} 0 & \text{if } x < \text{threshold} \\ 1 & \text{if } x \geq \text{threshold} \end{cases}$$

2. **Sigmoid Function:**
   - Description: The sigmoid (logistic) function squashes input values to the range (0, 1). It is commonly used in the output layer of binary classification models.
   - Mathematical Form: $f(x) = \frac{1}{1+e^{-x}}$

Types of activation function

3. Hyperbolic Tangent (tanh) Function:

- Description: Similar to the sigmoid function, the tanh function maps input values to the range (-1, 1). It is often used in hidden layers of neural networks.
- Mathematical Form:

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

4. Rectified Linear Unit (ReLU):

- Description: ReLU is a popular activation function that outputs the input for positive values and zero for negative values. It introduces non-linearity and is computationally efficient.
- Mathematical Form:

$$f(x) = \max(0, x)$$

Delta Learning Rule

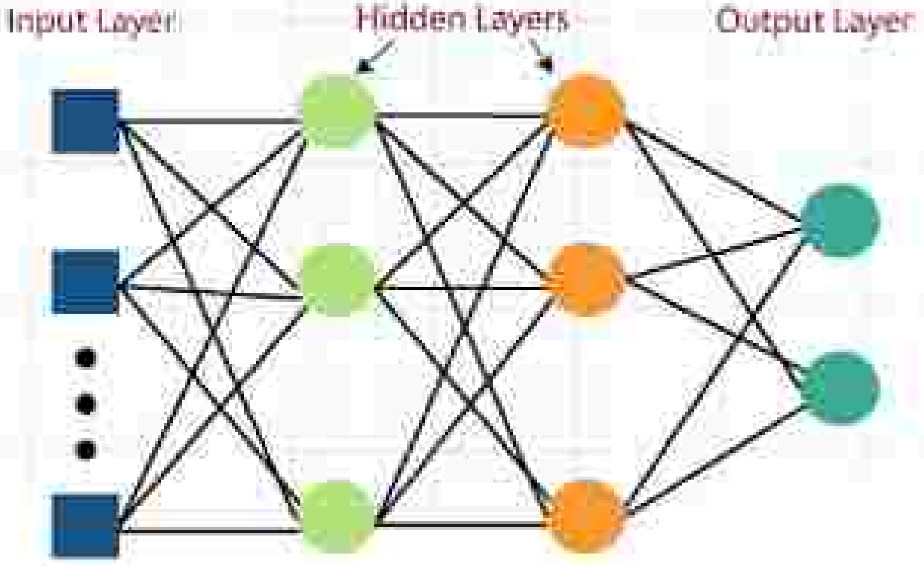
- The delta rule in an artificial neural network is a specific kind of backpropagation that assists in refining the machine learning/artificial intelligence network, making associations among input and outputs with different layers of artificial neurons. The Delta rule is also called the Delta learning rule.
- Delta rule is introduced by Widrow and Hoff, which is the most significant learning rule that depends on supervised learning.
- This rule states that the change in the weight of a node is equivalent to the product of error and the input.

Delta Learning Rule

- The given equation gives the mathematical equation for delta learning rule:
- $\Delta w = \mu . x . z$
- $\Delta w = \mu (t-y) x$
- Here, Δw = weight change. μ = the constant and positive learning rate. X = input value from pre-synaptic neuron.
- z= (t-y) is the difference between the desired input t and the actual output y. The above mentioned mathematical rule can be used only for a single output unit.
- The different weights can be determined with respect to these two cases.
- Case 1: When $t \neq k$, then $w(new) = w(old) + \Delta w$
- Case 2: When $t = k$, then No change in weight

Multilayer Perceptron

- A Multilayer Perceptron (MLP) is a type of artificial neural network characterized by its multiple layers of interconnected nodes or neurons.
- It consists of an input layer, one or more hidden layers, and an output layer.
- Each layer contains nodes, and nodes in one layer are connected to nodes in the adjacent layers.
- The architecture of an MLP enables it to learn complex patterns and relationships within the data.



Multilayer Perceptron

Components of a Multilayer Perceptron:

- Input Layer:
  - The input layer receives the features or input data and transmits them to the neurons in the hidden layer.
  - Each node in the input layer represents a feature of the input data.
- Hidden Layers:
  - Hidden layers, as the name suggests, are intermediary layers between the input and output layers.
  - Each node in a hidden layer receives inputs from all nodes in the previous layer and produces an output for nodes in the subsequent layer.
  - The presence of multiple hidden layers allows MLPs to learn intricate and hierarchical representations of data.
- Weights and Biases:
  - Connections between nodes in different layers are associated with weights, which represent the strength of the connection.
  - Additionally, each node has an associated bias that helps in adjusting the overall input to the node.
  - Learning in an MLP involves adjusting these weights and biases based on the training data.

Multilayer Perceptron

Components of a Multilayer Perceptron:

- Activation Functions:
  - Activation functions introduce non-linearities to the model, enabling MLPs to learn complex relationships.
  - Common activation functions include the sigmoid, hyperbolic tangent (tanh), and rectified linear unit (ReLU).
- Output Layer:
  - The output layer produces the final result or prediction.
  - The number of nodes in the output layer depends on the nature of the task.
  - For binary classification, a single node with a sigmoid activation function is often used.
  - For multi-class classification, the output layer may have multiple nodes, often with a softmax activation.

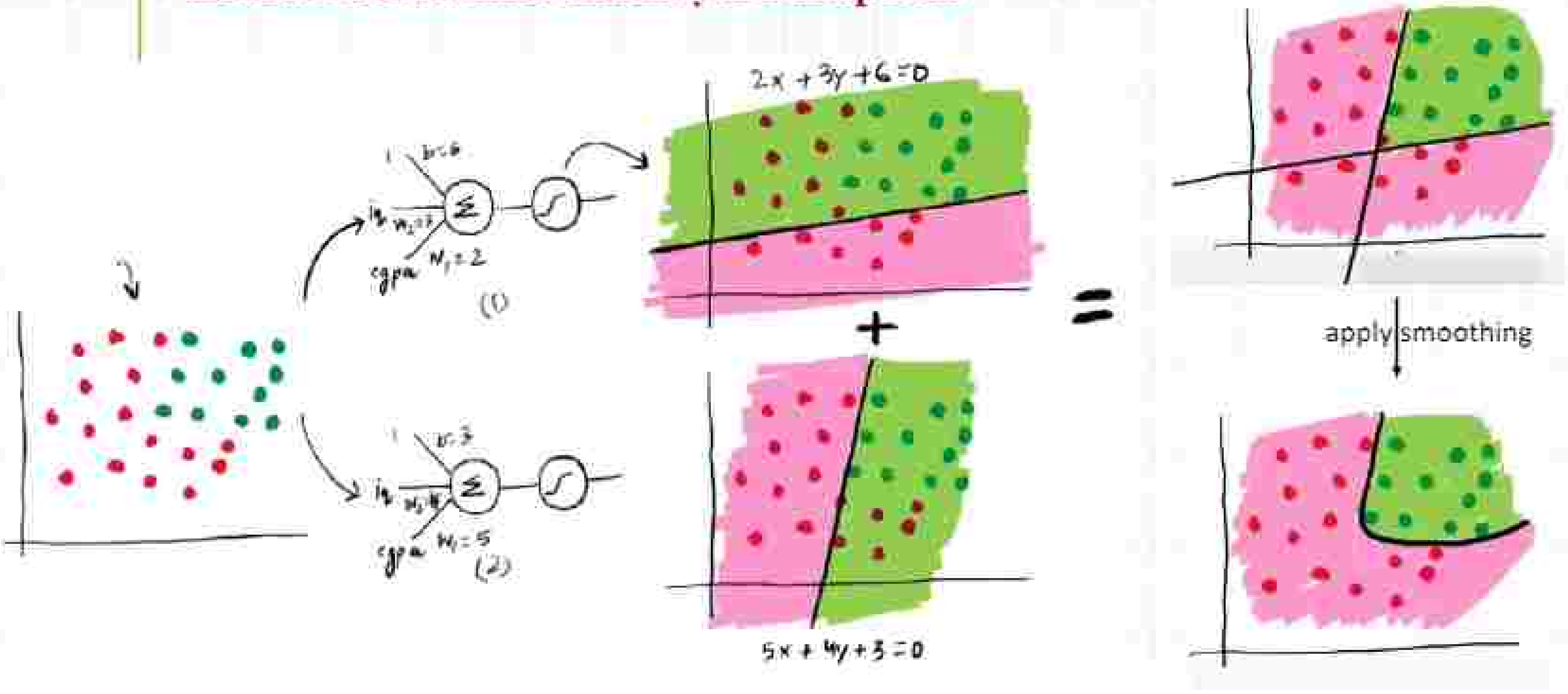
Multilayer Perceptron

Components of a Multilayer Perceptron:

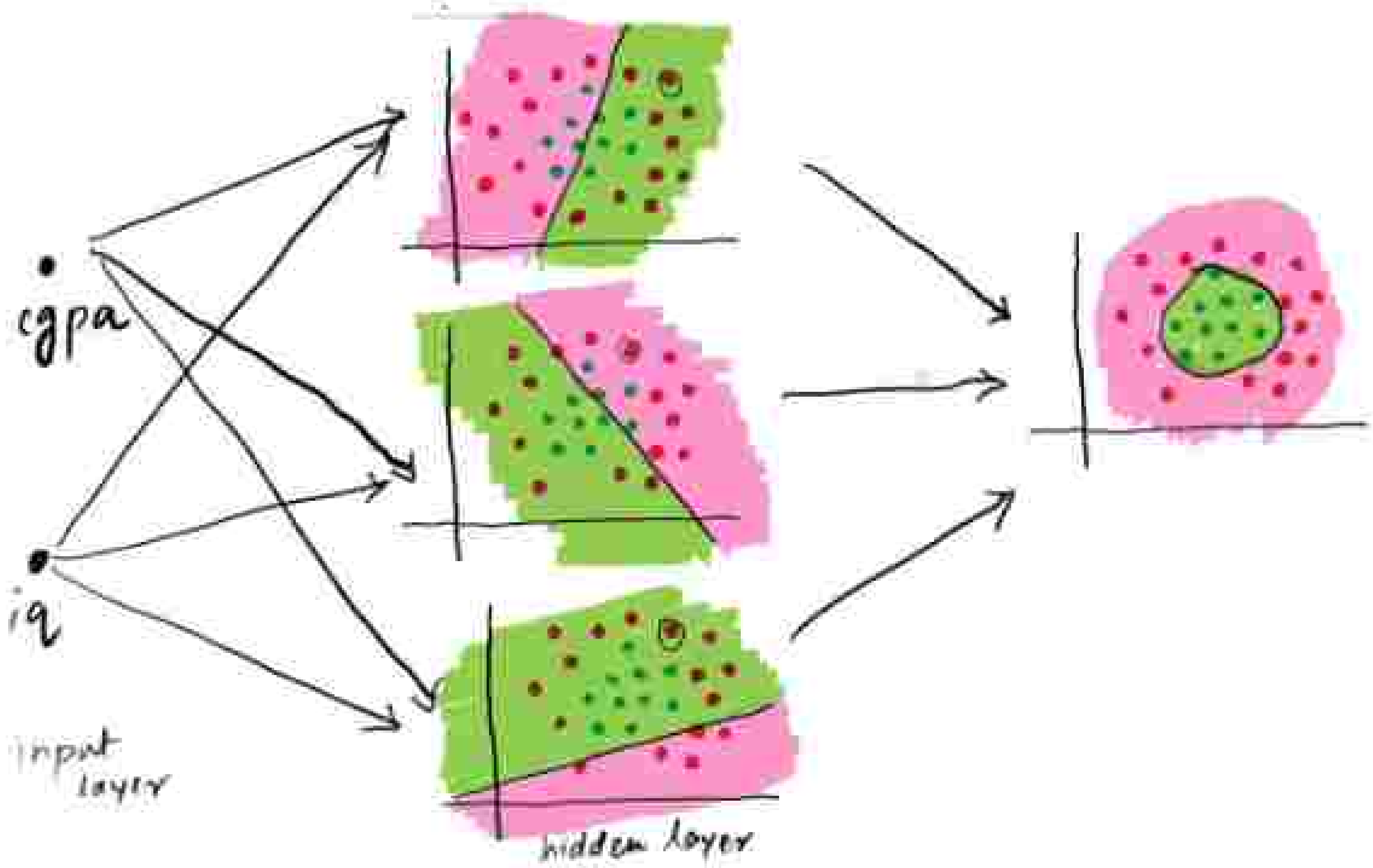
- Feedforward Process
  - During the feedforward process, input data are propagated through the network layer by layer.
  - The weighted sum of inputs, along with biases, is passed through the activation function at each node to produce the output.
- Backpropagation and Training:
  - Training an MLP involves adjusting the weights and biases to minimize the difference between the predicted output and the actual target.
  - Backpropagation is a key algorithm used for updating weights by propagating the error backward through the network.

Multilayer Perceptron

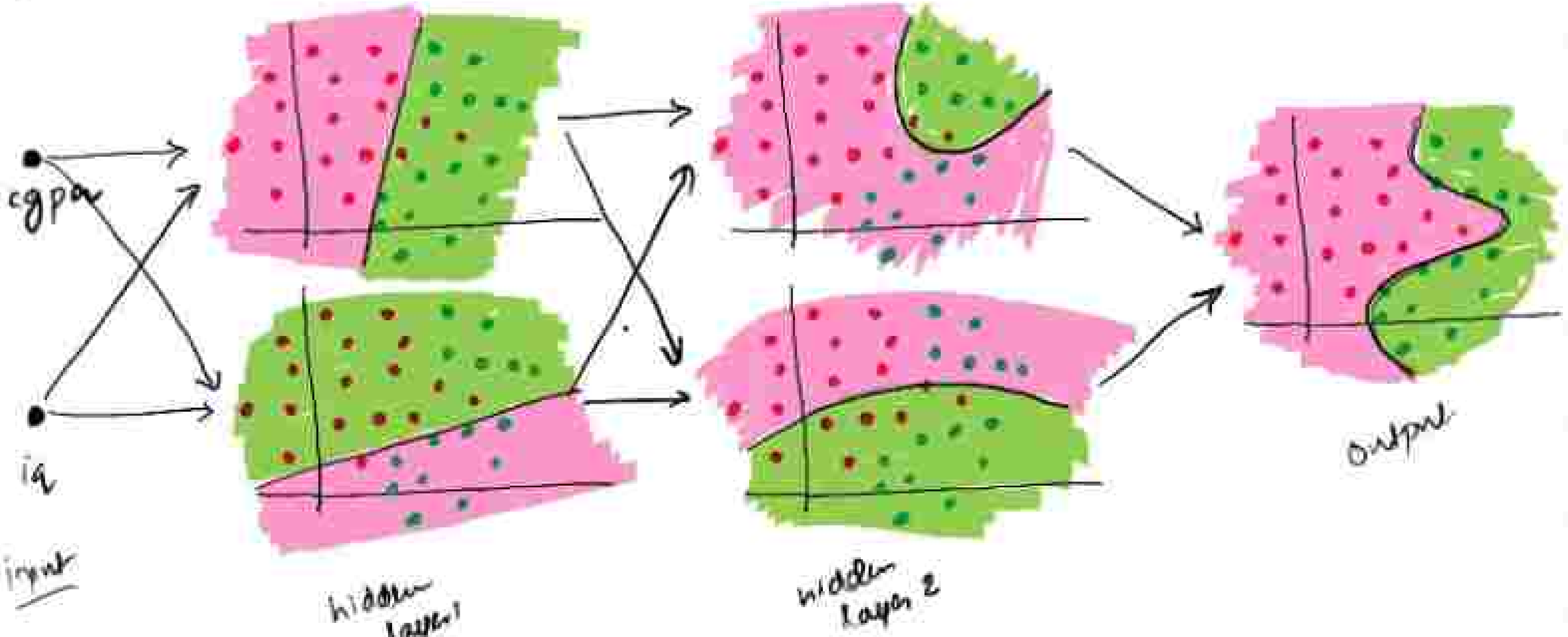
Abstract idea behind Multilayer Perceptron:



Multilayer Perceptron



Multilayer Perceptron



Multilayer Perceptron: Linearly separable, linearly non-separable classes

- A Multilayer Perceptron (MLP) is a type of artificial neural network that consists of multiple layers of interconnected nodes or neurons.
- Unlike a single-layer perceptron, which can only model linearly separable functions, an MLP is capable of learning and representing both linearly separable and non-linearly separable functions.

  - 1. Linearly Separable Classes:
  - 2. Linearly Non-Separable Classes:

Multilayer Perceptron: Linearly separable, linearly non-separable classes

1. Linearly Separable Classes:

- In the context of neural networks, linearly separable classes refer to classes or patterns in the input space that can be separated by a hyperplane.
- A hyperplane is a subspace with one dimension less than the original space. For a two-dimensional space, a hyperplane is a line; for a three-dimensional space, it's a plane, and so on.
- Linearly separable classes can be separated by a straight line, plane, or hyperplane, depending on the dimensionality of the input space.

- In the case of an MLP, if the classes are linearly separable, a single hidden layer with a sufficient number of neurons can learn the decision boundaries needed to separate the classes.
- The activation functions and weights in the hidden layer allow the network to perform non-linear transformations on the input data, making it capable of representing complex decision boundaries.

Multilayer Perceptron: Linearly separable, linearly non-separable classes

2. Linearly Non-Separable Classes:

- Linearly non-separable classes refer to classes or patterns that cannot be separated by a hyperplane in the input space.
- These classes require non-linear decision boundaries to be accurately classified. An MLP is well-suited for handling linearly non-separable classes because of its ability to learn complex relationships and non-linear mappings.
- In the case of an MLP, introducing multiple hidden layers allows the network to capture hierarchical and non-linear features in the data. Each layer applies a non-linear activation function to the input, enabling the network to model intricate decision boundaries.
- The backpropagation algorithm, used for training MLPs, adjusts the weights during the learning process to minimize the error between predicted and actual outputs, effectively adapting the network to complex data distributions.

Notations in Neural Networks

- **Understanding Backpropagation in MLP**
  - Backpropagation is a crucial concept in deep learning, particularly in training a multilayer perceptron (MLP).
- **Challenges in Neural Networks**
  - Neural networks contain numerous biases and weights across different layers.
  - Identifying and understanding the roles of these weights and biases can be confusing without a proper notation system.
- **Importance of Notations**
  - Clear notations are vital for comprehending backpropagation.
  - They help in distinguishing weights and biases, making the learning process more accessible.

Notations in Neural Networks

- **Determining Trainable Parameters**
  - Trainable parameters include weights and biases for each layer in a neural network.
  - Understanding the total number of trainable parameters reveals the network's complexity and power.

- **Notations in Neural Networks**
  - Discuss standard notations used in the industry to label weights and biases in different layers.
  - Clarity in notations simplifies the understanding of backpropagation and the overall neural network structure.

- Simplifying the Complex concepts and making more understandable

Notations in Neural Networks

- **Denote bias in Neural Networks**
  - b_{ij}
  - Where i=layer number and j=node number in a particular layer

- **Denote output of neuron in Neural Networks**
  - O_{ij}
  - Where i=layer number and j=node number in a particular layer
- **Denote weight in Neural Networks**

  $$W_{ij}^{k}$$

  - Where k=weight is inserting in which layer number;
    i=weight is exiting from which number of node/neuron
    j= weight is inserting in which number of node/neuron

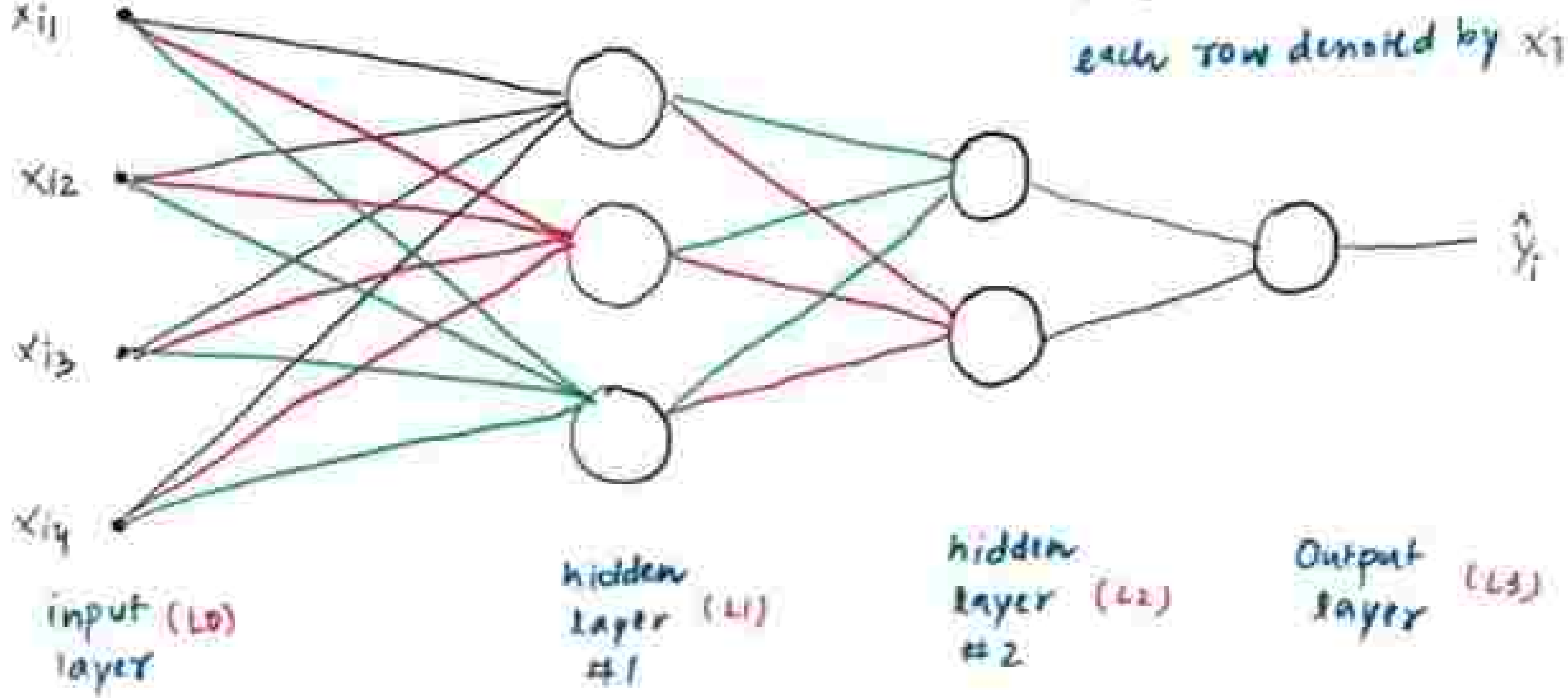
Notations in Neural Networks

Data → { m×n }

m → # rows

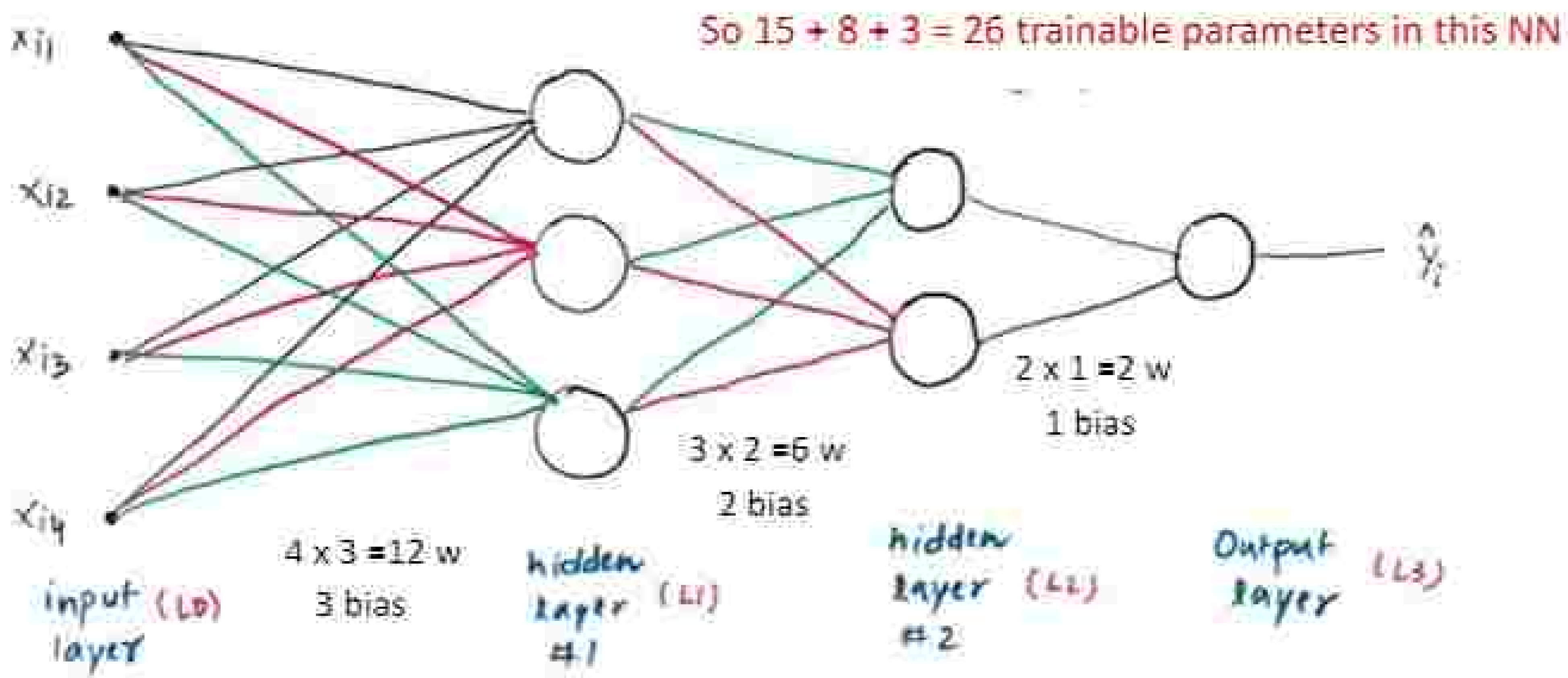
n = 4

each row denoted by x_i

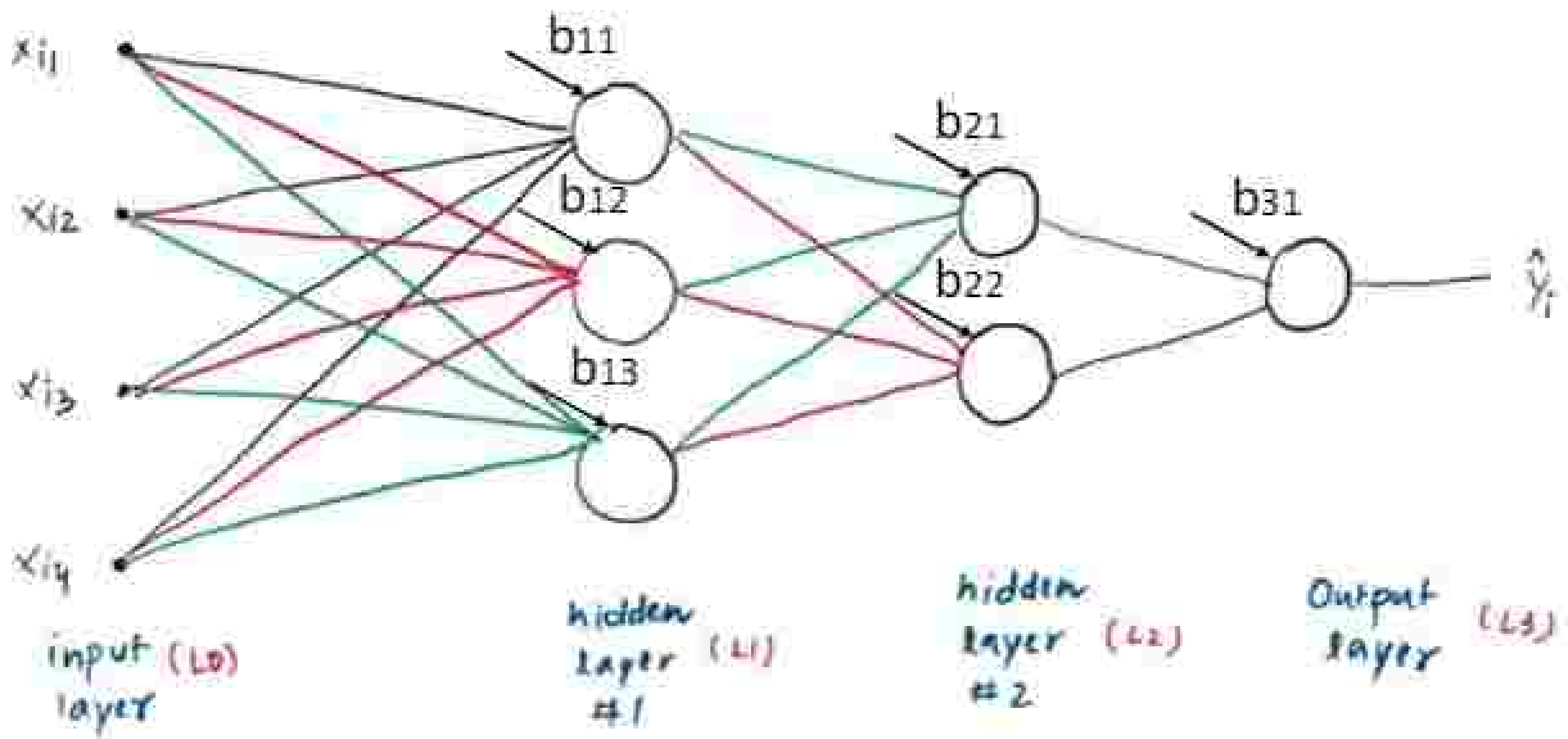


Notations in Neural Networks

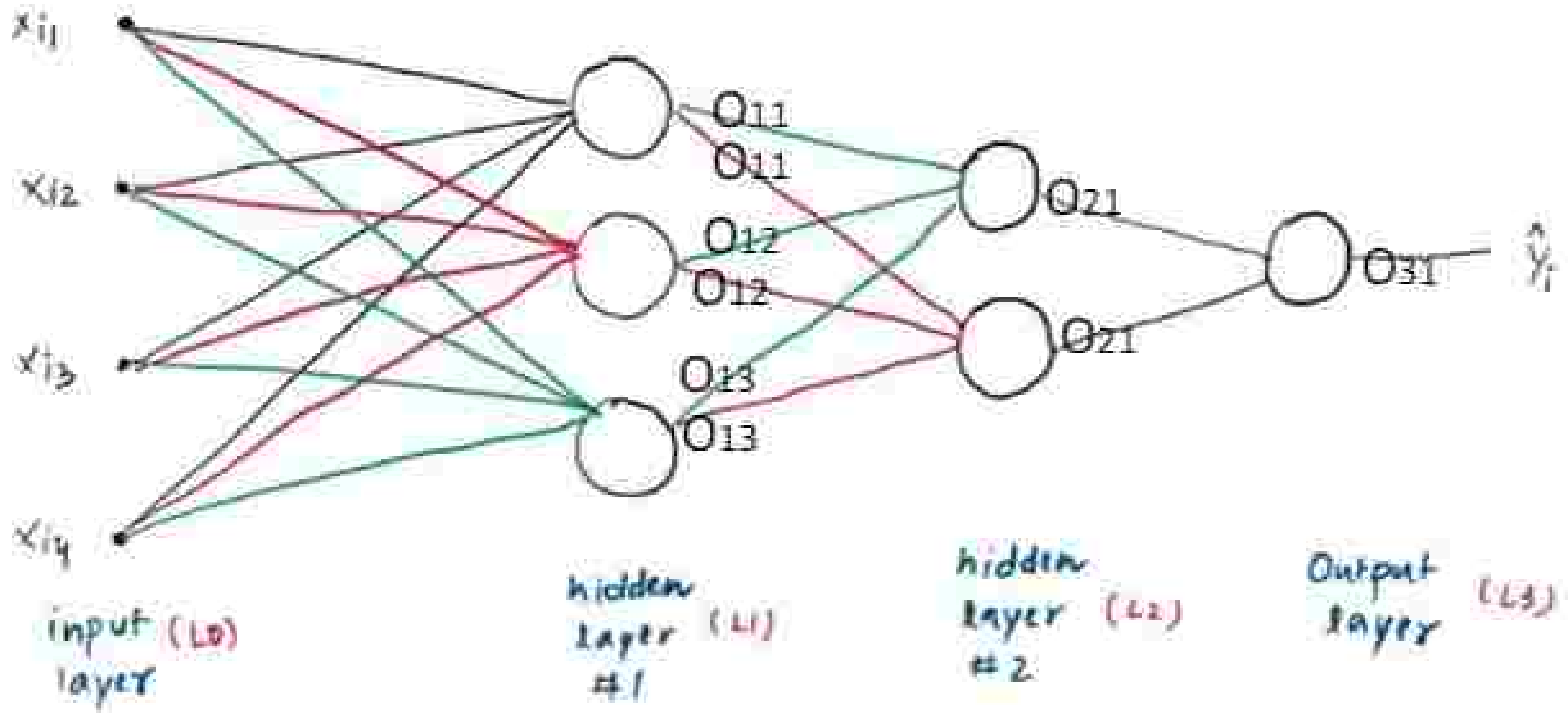
Total no. of trainable parameters in NN



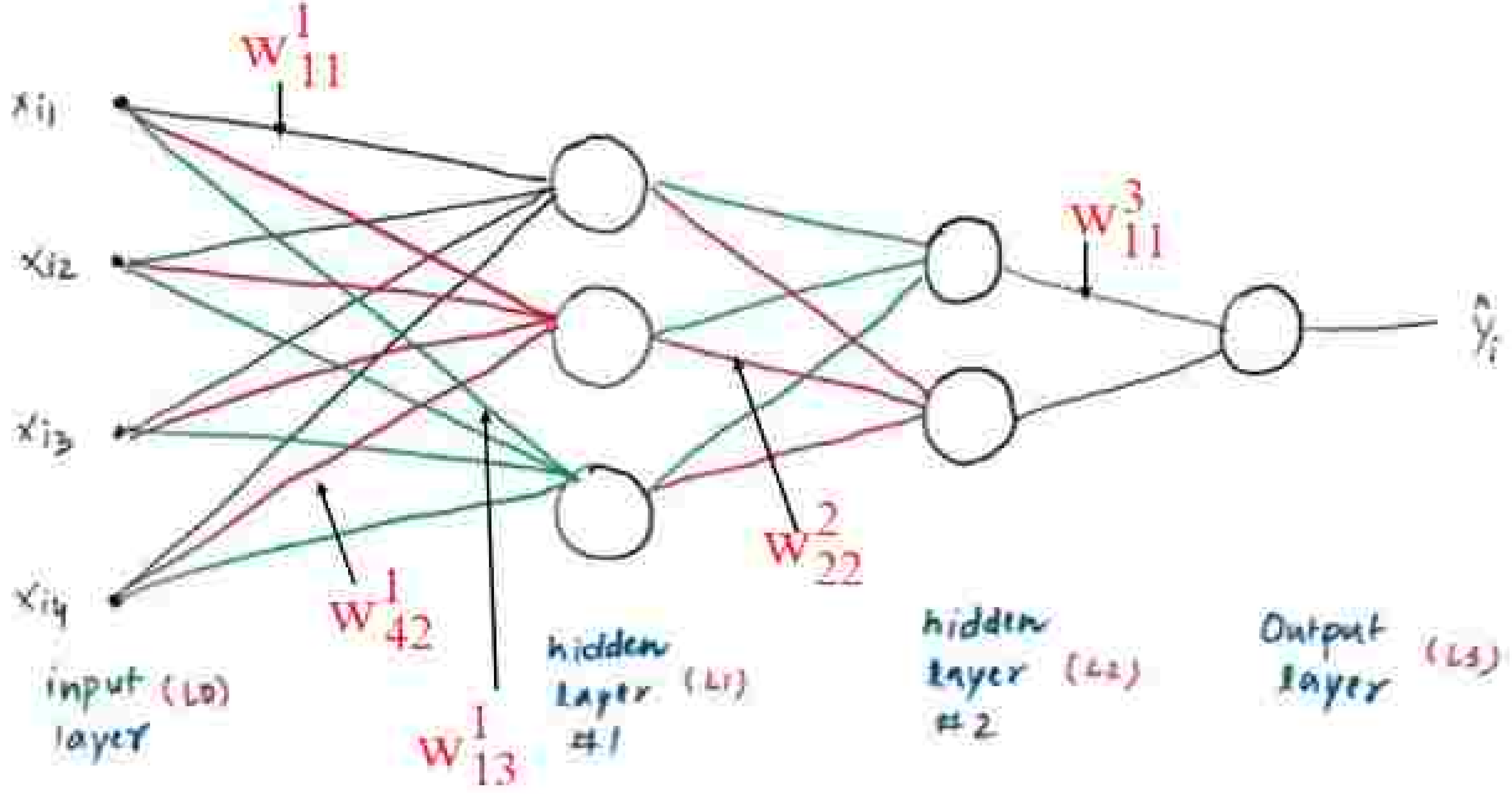
Notations in Neural Networks



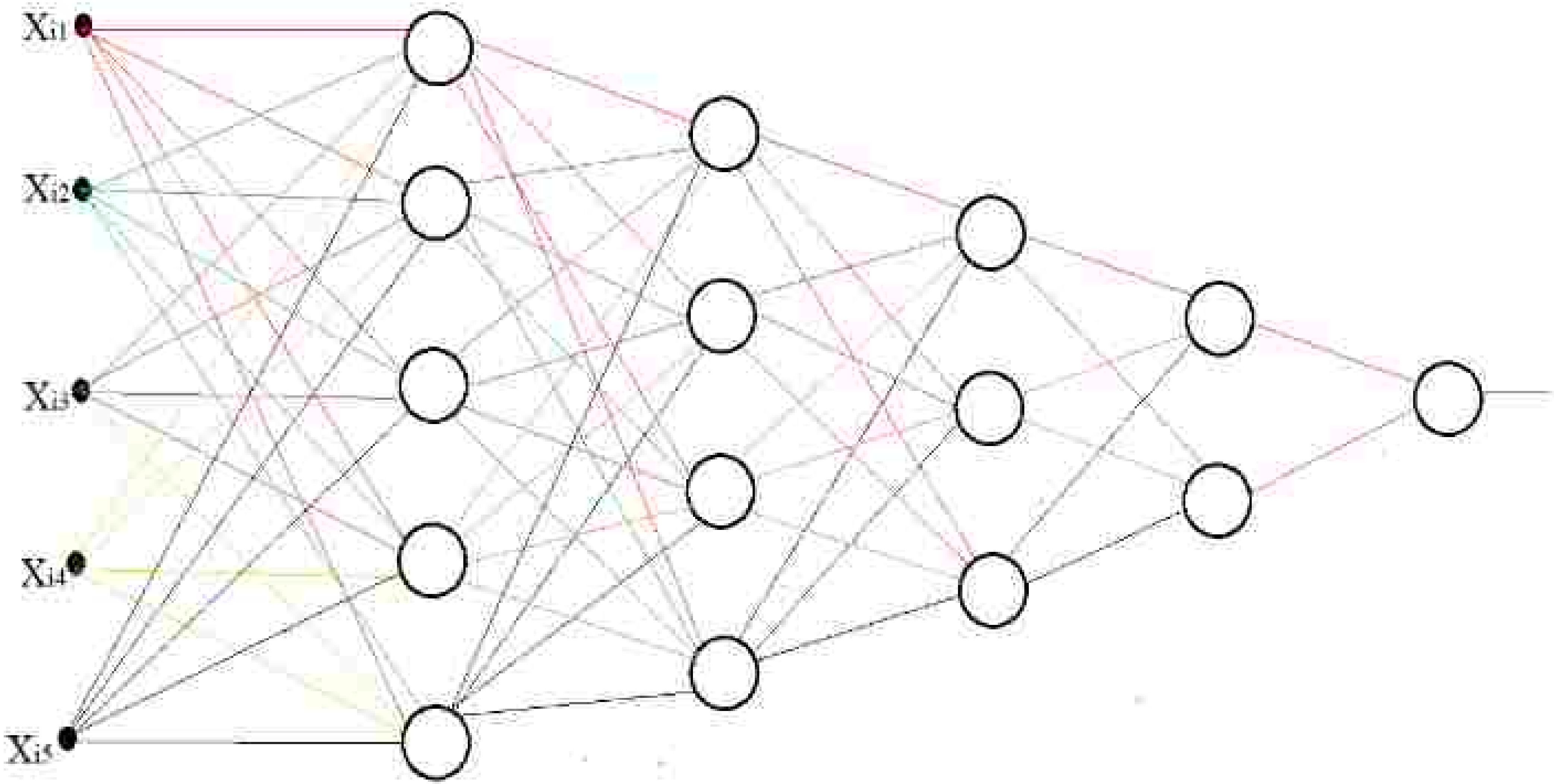
Notations in Neural Networks



Notations in Neural Networks



Notations in Neural Networks



Basic Terminologies of Deep Learning

1. Neural Network
2. Neuron
3. Activation Function
4. Layer
5. Deep Learning
6. Training
7. Backpropagation
8. Loss Function
9. Overfitting
10. Dropout
11. Batch Normalization
12. Convolutional Neural Network (CNN)
13. Recurrent Neural Network (RNN)
14. Transfer Learning
15. Hyperparameter
16. Gradient Descent
17. Epoch
18. Mini-batch
19. Tensor
20. GPU (Graphics Processing Unit)

Basic Terminologies of Deep Learning

- 1. **Neural Network:**
  - A computational model inspired by the human brain, consisting of interconnected nodes (neurons) organized in layers. Neural networks are the foundation of deep learning.

- **2. Neuron:**
  - A fundamental unit of a neural network that processes information. Neurons receive input, apply a weighted sum, pass it through an activation function, and produce an output.

- **3. Activation Function:**
  - A mathematical operation applied to the weighted sum of inputs in a neuron, determining its output. Common activation functions include sigmoid, tanh, and ReLU.

Basic Terminologies of Deep Learning

- **4. Layer:**
  - A structural component of a neural network that consists of neurons. Common types include input, hidden, and output layers.
- **5. Deep Learning:**
  - A subset of machine learning that involves neural networks with multiple layers (deep neural networks). Deep learning excels at learning hierarchical representations from data.
- **6. Training:**
  - The process of adjusting the weights and biases of a neural network using a labeled dataset to minimize the difference between predicted and actual outputs.

Basic Terminologies of Deep Learning

- **7. Backpropagation:**
  - A supervised learning algorithm used to train neural networks. It involves calculating the gradient of the loss function with respect to the weights and adjusting them accordingly.

- **8. Loss Function:**
  - A measure of the difference between the predicted and actual outputs of a neural network. The goal during training is to minimize this function.

- **9. Overfitting:**
  - A situation where a neural network learns the training data too well, including its noise and outliers, leading to poor performance on new, unseen data.

Basic Terminologies of Deep Learning

- **10. Dropout:**
  - - A regularization technique in deep learning where randomly selected neurons are ignored during training. It helps prevent overfitting.

- **11. Batch Normalization:**
  - - A technique that normalizes the inputs of each layer in a neural network, reducing internal covariate shift and improving training stability.

- **12. Convolutional Neural Network (CNN):**
  - - A type of neural network designed for processing structured grid data, like images. CNNs use convolutional layers to detect patterns hierarchically.

Basic Terminologies of Deep Learning

- **13. Recurrent Neural Network (RNN):**
  - - A type of neural network designed for sequential data processing. RNNs use recurrent connections to store information about previous inputs.

- **14. Transfer Learning:**
  - - A technique in which a pre-trained model on a specific task is used as a starting point for a new, similar task. It helps leverage knowledge gained from one domain in another.

- **15. Hyperparameter:**
  - - Parameters that are set prior to training and remain constant during the training process. Examples include learning rate and the number of hidden layers.

Basic Terminologies of Deep Learning

- **16. Gradient Descent:**
  - - An optimization algorithm used to minimize the loss function during training by adjusting the weights and biases in the direction of the steepest descent.
- **17. Epoch:**
  - - One complete pass through the entire training dataset during training.
- **18. Mini-batch:**
  - - A small, random subset of the training data used in each iteration of training. Mini-batch training balances efficiency and accuracy.

Basic Terminologies of Deep Learning

- **19. Tensor:**
  - - A mathematical object representing a multi-dimensional array, fundamental for data representation in deep learning frameworks.

- **20. GPU (Graphics Processing Unit):**
  - - Hardware accelerators widely used in deep learning to speed up training due to their parallel processing capabilities.