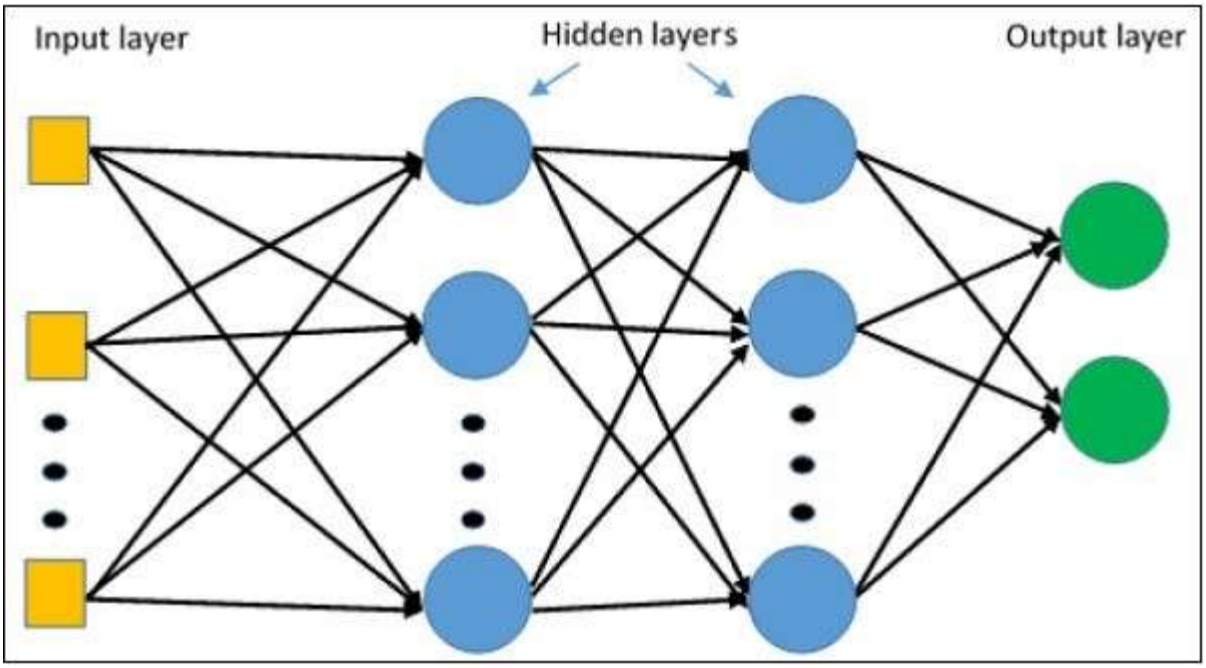
# Multi Layer Perceptron (MLP) | MLP Notation | MLP Intuition

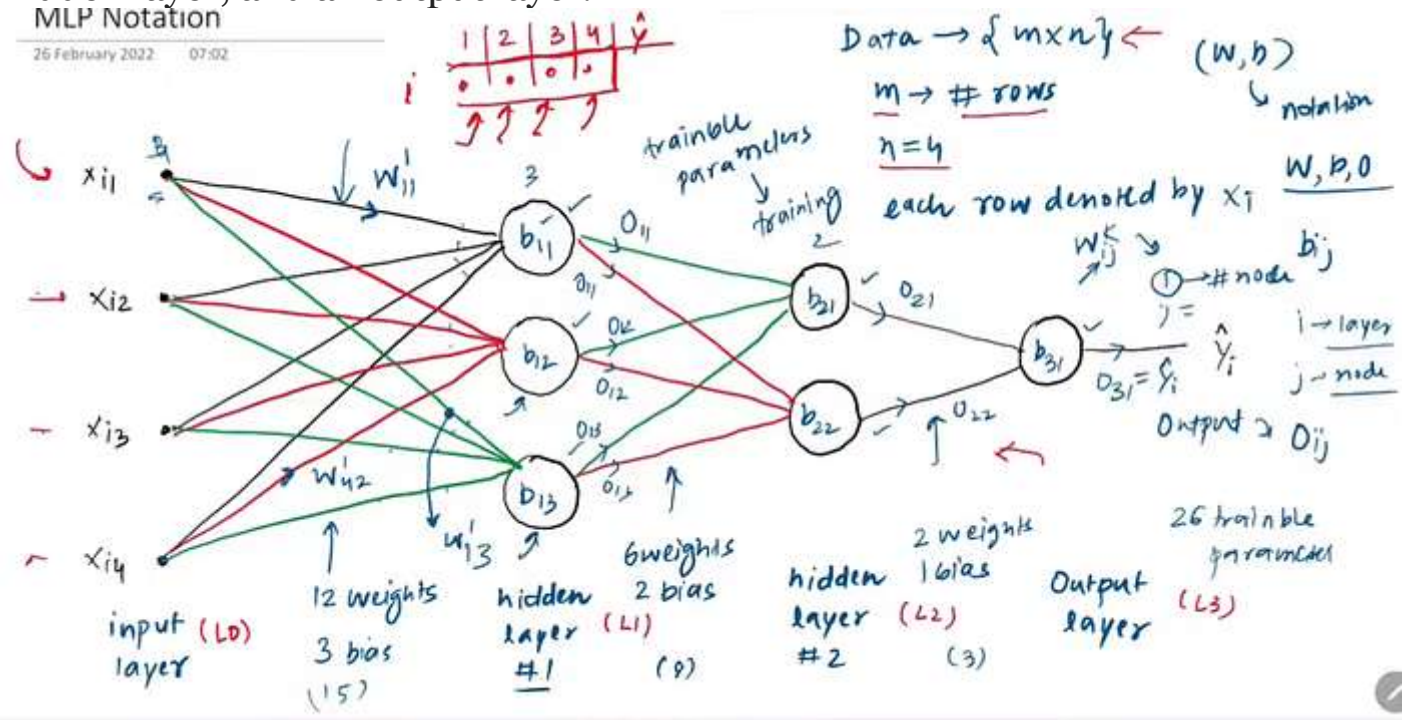



MLP is made up of multiple single layer perceptron to form a complex neural network that can work on non linear data. MLP consists of input layer, multiple hidden layer and a output layer. Input layer where all the input are coming. Hidden layer is something which consists of multiple nodes and we can have as many as hidden layer in our neural network. Multi-layer Perceptron is a set of input and output layers and can have one or more hidden layers with several neurons stacked together per hidden layer. And a multi-layer neural network can have an activation function that imposes a threshold, like ReLU or sigmoid. Neurons in a Multilayer Perceptron can use any arbitrary activation function.

A perceptron in neural networks is a unit or algorithm which takes input values, weights, and biases and does complex calculations to detect the features inside the input data and solve the given problem. Weights and biases (denoted as w and b) are the learnable parameters of neural networks. Weights are the parameters in a neural network that passes the input data to the next layer containing the weight of the information, and more weights mean more importance.

## MLP Notations

he above picture shows the Multi-layer neural network having an input layer, a hidden layer, and an output layer.




picture taken from CampusX tutorial of MLP Notation. Thank You :)

### 1. Notation for Weights :

Notation: $W_{ij}^{h}$

where,

i = From which node weight is passing to the next layer's node.

j= To which node weight is arriving.

h = Layer in which weight is arriving.

Example:

$W_{11}^{1}$ = Weight passing into 1st node of 1st layer of 1st node of the previous layer.

$W_{23}^{1}$ = Weight passing into the 3rd node of the 1st hidden layer from the 2nd node of the previous layer.

$W_{45}^{1}$ = Weight passing into the 5th node of the 2nd hidden layer from the 4th node of the previous layer.

## 2. Notations for Biases :

Notation: bij

Where,

i = Layer to which a bias belongs.

j = node to which a bias belongs.

Example:

b11 = biases of 1st node of 1st hidden layer.

b23 = bias of 3rd of 2nd hidden layer.

b41 = bias of 1st node of 4th hidden layer.

### 3. Notations for Outputs:

Notation: Oij

Where,

i = Layer to which a bias belongs.

j = node to which a bias belongs.

Example:

O11 = Output of 1st node of 1st hidden layer.

O45 = Output of 5th of 4th hidden layer.

O34 = Output of 4th node of 3rd hidden layer.

## Ways to change MLP

- by adding nodes to hidden layer
- by adding inputs to input layer
- by adding multiple hidden layer

# □ What is MLP (Multi-Layer Perceptron)?

A **Multi-Layer Perceptron (MLP)** is a type of **Artificial Neural Network (ANN)**.
Its main purpose is to **learn patterns** in data and make predictions for **classification** or **regression** tasks.

### ➡□ Example use cases:

- Image classification (handwritten digits)
- Spam detection
- Predicting house prices

---

# □ MLP Structure / Notation

An MLP is made of **three types of layers**:

### 1□ Input Layer

This layer takes the raw data (features like pixel values, age, income, etc.).
□ If your data has 3 features (x1, x2, x3), the input layer will have **3 neurons**.

---

### 2□ Hidden Layers (One or More)

These are the layers **between input and output**.
Hidden layers allow the network to learn **complex patterns**.
Each neuron in these layers performs:

- A **linear operation**: (weights * inputs + bias)
- Followed by an **activation function** (ReLU, Sigmoid, Tanh, etc.)

These layers are called **hidden** because we don't directly observe their outputs.

---

### 3□ Output Layer

The output layer gives the final **prediction**.

- For classification: Usually uses **Softmax** or **Sigmoid**.
- For regression: Often uses a **Linear** output.

---

### Notation Example

If your MLP looks like this:

- Input Layer: 3 neurons
- Hidden Layer 1: 4 neurons
- Hidden Layer 2: 3 neurons
- Output Layer: 1 neuron

We write this as:
**3 - 4 - 3 - 1**

---

# □ MLP Intuition (How to Think About It)

Think of an MLP as a **series of mathematical filters**:

1. Data passes through multiple layers.
2. Each layer transforms the data a little (linear + non-linear transformation).
3. The network **learns these transformations** during training (adjusting weights via backpropagation).
4. Final layer gives the **desired prediction**.

Each neuron helps the network understand something more **abstract** about the data as it goes deeper through layers.

---