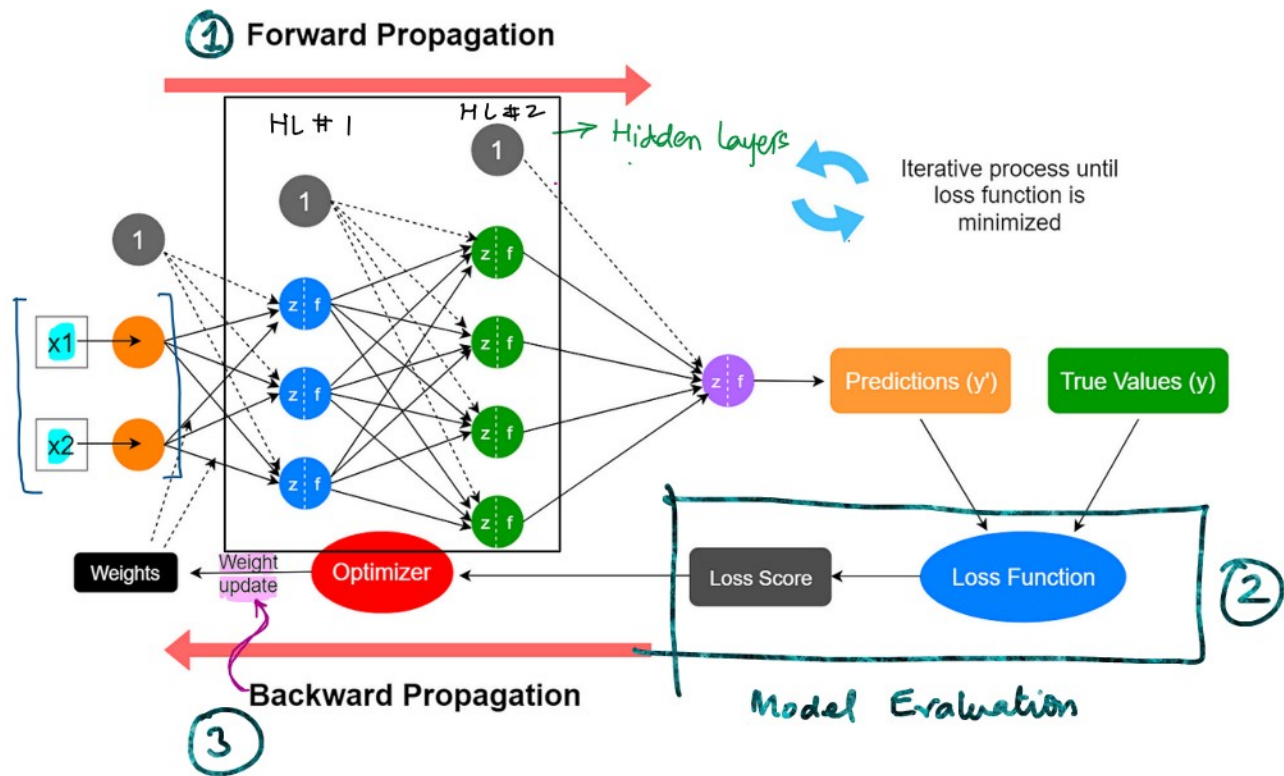


Multiple Layer Perceptron (MLP)

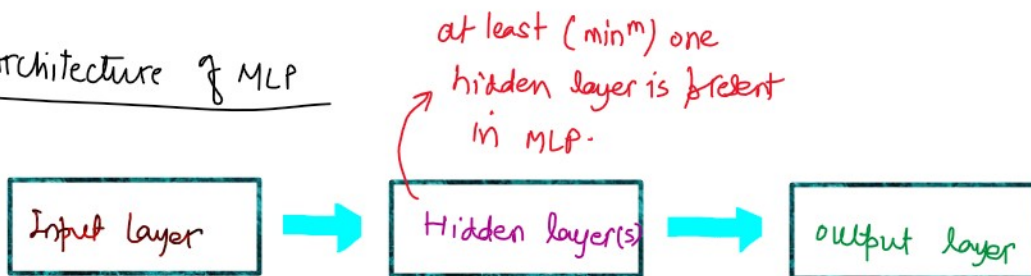
05 October 2025 09:56



Multiple Layer Perceptron

A multi-layer perceptron is class of ANN that consists of multiple layer (hidden layers) of neurons in a feed-forward network.

Architecture of MLP



- receives the input data (after flattening) and connect each of the input Features (columns) to

- one or more hidden layer(s) between input and output layers where the artificial

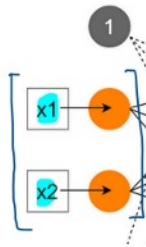
- produces the final prediction

```

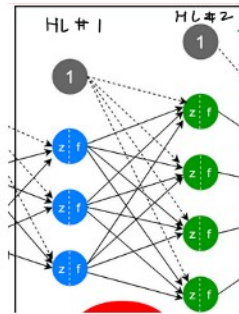
graph TD
    A[produces the final prediction] --> B[ ]
    B --> C[ ]
    B --> D[ ]
  
```

Responsible for the final output

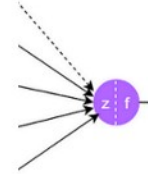
connect each of the input
Features (columns) to
neurons in the input layer



and output layers
where the actual
learning happens-



↓ ↓
Regression Classification
• Binary
• Multi-class.



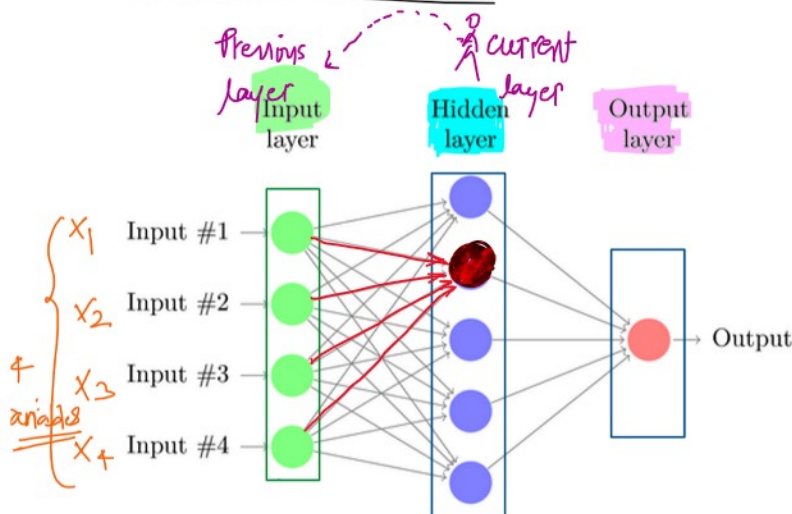
- No learning or calculation

Neural Network Terminologies

Fully Connected Network

- a layer where each and every neuron is
connected to every neuron in the preceding/
previous layer.

Neural Network Terminologies



(self-read)

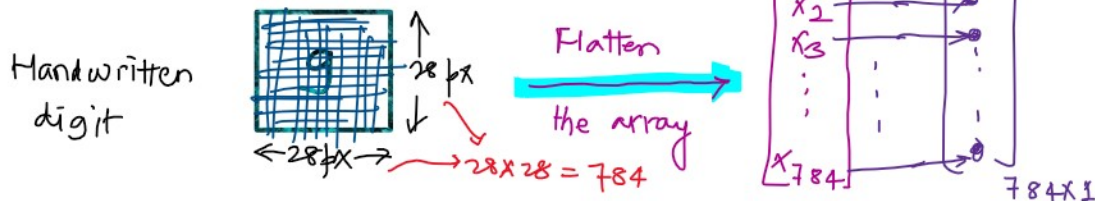
Single Layer Perceptron (SLP) vs Multi Layer Perceptron (MLP)		
Feature	SLP (Single Layer Perceptron)	MLP (Multi Layer Perceptron)
Layers	1 layer (input \rightarrow output)	2 or more layers (input \rightarrow hidden(s) \rightarrow output)
Neurons	No hidden layer, just output neuron(s)	One or more hidden layers with multiple neurons
Functions	Can only solve linearly separable problems	Can solve non-linear and more complex problems
Learning	Simple weights & bias update (perceptron rule)	Uses backpropagation and more complex optimization
Representation	Linear decision boundaries	Can learn complex, non-linear boundaries
Use cases	Very basic classification tasks	Most modern neural network applications

computation: simple and fast
(but just a proof of concept)

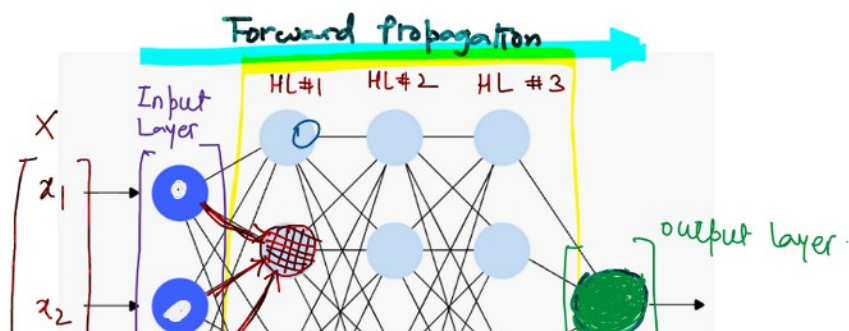
more computationally intensive
than SLP
(state of art NN model)

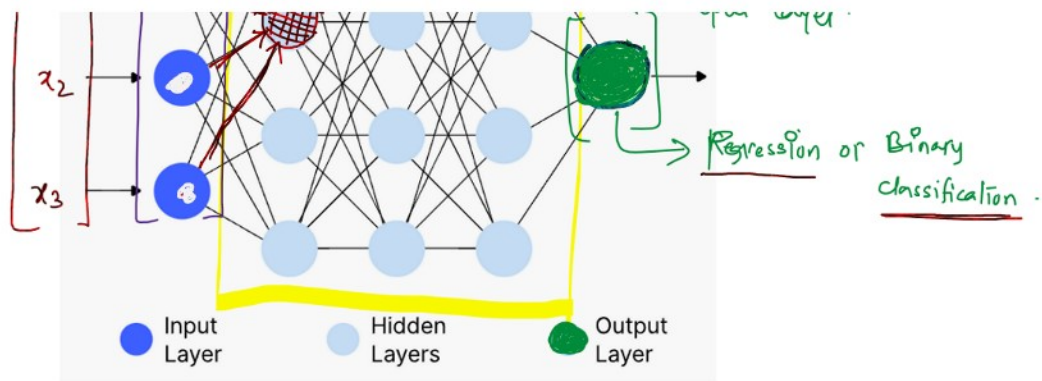
Working of MLP

Step #1 Getting input data ready.
- to classify a handwritten digit



Step #2 Take (weighted sum of inputs + bias) \rightarrow in hidden layer





HL #1

↓
1st

Each neuron in the 1st HL receives a weighted sum of inputs or features (x_1, x_2, x_3) from the input layer.

\oplus
(bias)

Computing the weighted sum of inputs along with bias:

$$Z_j^l = W_{ij}^l x_i + b_j^l$$

where

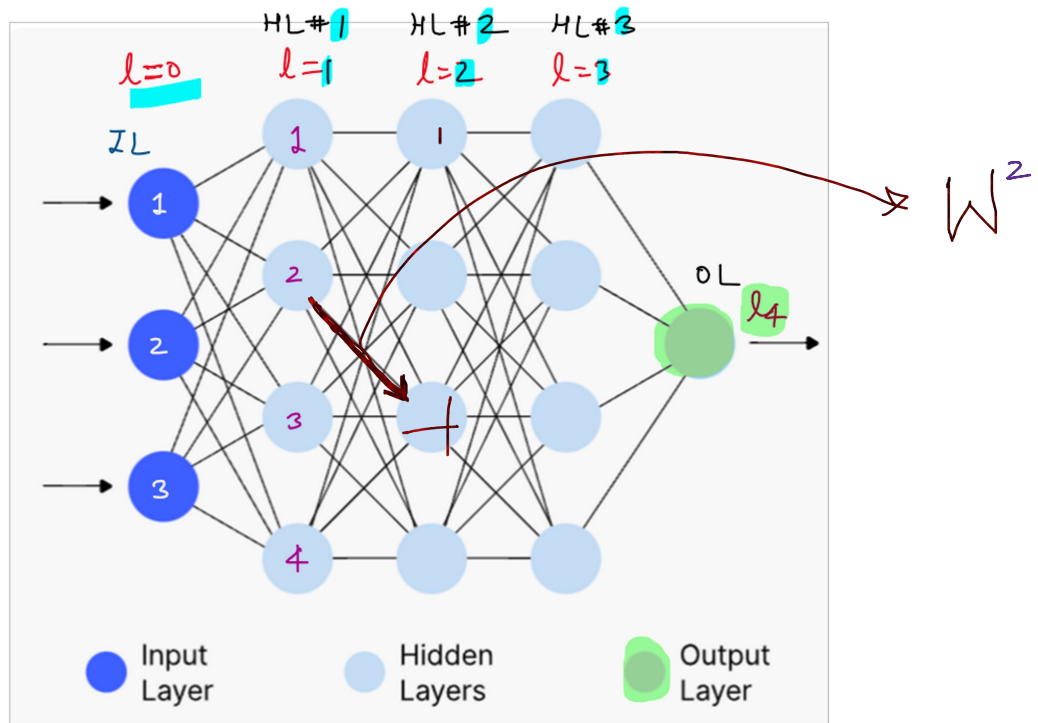
Z_j^l : is the weight sum for neurons j in the layer l

W_{ij}^l : is the weight between neuron i from the previous layer ($l-1$) and neuron j in the current layer (l)

x_i : is the input from neuron i in the previous layer

b_j^l : is the bias associated with neuron j in the layer l





$$z_j^{l \rightarrow 1}$$

$$z_2^1$$