Multilayer Perceptron (MLP)

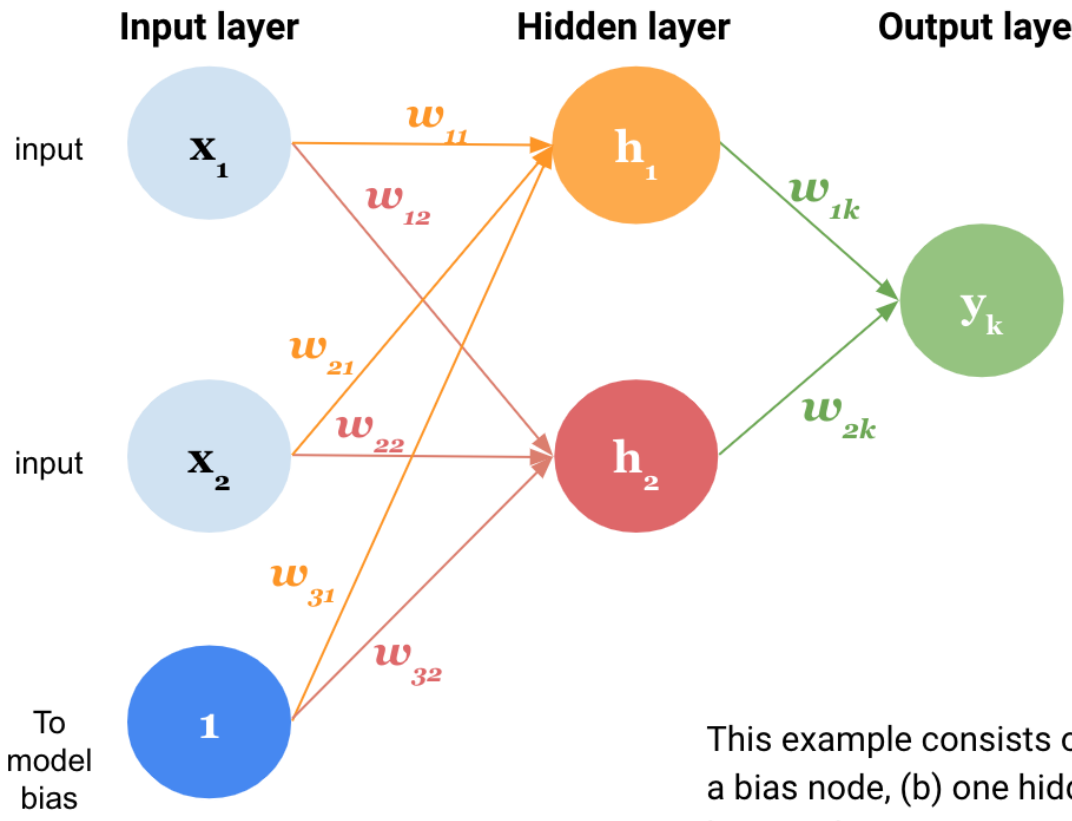
What is a Multilayer Perceptron (MLP)?

- An artificial neural network with multiple layers of neurons.
- Extension of Perceptron and Adaline.
- Has hidden layers in addition to input and output layers.
- Can solve non-linear problems due to hidden layers and activation functions.

Architecture of MLP

- **Input Layer**: Takes input features.
- **Hidden Layer(s)**: Applies weights, bias, and non-linear activations.
- **Output Layer**: Produces final prediction using task-specific activation functions.

Illustrative example of Multilayer perceptron, a Feedforward neural network



x_1 , x_2 : *input data features*

w_{ij} : *weights of the network*

h_1 , h_2 : *nodes in the hidden layer*

y_k : *output variable*

This example consists of: (a) an input layer with two input nodes and a bias node, (b) one hidden layer with two neurons, and (c) an output layer with one neuron

Working of MLP

1. **Forward Propagation**: Data passes layer by layer applying weights, bias, and activation.
2. **Loss Calculation**: Compare predicted output with actual using loss function.
3. **Backpropagation**: Gradients are computed, weights updated using optimizers.

Activation Functions in MLP

- **Sigmoid**: Maps values between (0,1).
- **Tanh**: Maps values between (-1,1).
- **ReLU**: Allows only positive values, efficient.
- **Softmax**: Used for multi-class classification outputs.

Applications of MLP

- Handwriting and speech recognition.
- Image classification.
- Predictive modeling (stock prices, weather).
- Natural Language Processing (NLP).
- Medical diagnosis.

Key Difference from Perceptron

- **Perceptron**: Only 1 layer, linear decision boundary.
- **MLP**: Multiple layers + non-linear activations → can solve complex, non-linear problems.

A **Multilayer Perceptron (MLP)** is a type of **feedforward neural network** because

1. In an MLP, information flows **only in one direction**: **Input Layer → Hidden Layer(s) → Output Layer**

2. There are **no cycles or loops** in the connections.

3. Each neuron in one layer is connected to neurons in the next layer (fully connected).

4. Training is done using **backpropagation**, but that's just for adjusting weights — the actual prediction process is still **forward only**.