Deep Learning & Perceptron Explained

# What is Deep Learning?

Deep Learning is a **subset of Machine Learning** that uses **artificial neural networks** to learn from large amount of data. Inspired by the structure of the human brain, it automatically discovers complex patterns in data—ranging from images, text, and audio to structured tabular data(csv).

It's widely used in image recognition, language translation, autonomous vehicles, and voice assistants.

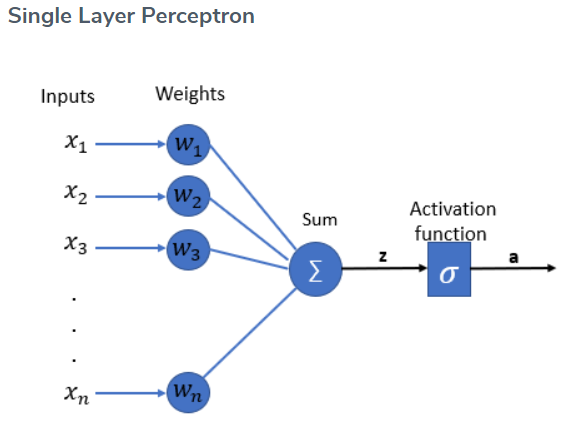
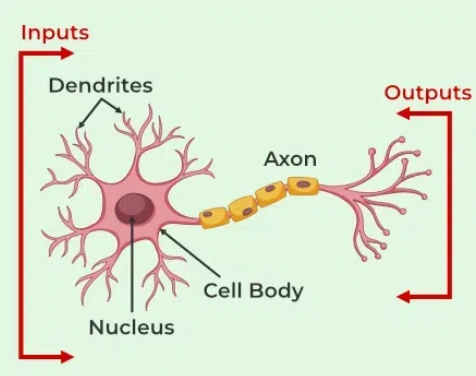
# 🧩 Deep Learning vs Machine Learning

|  |  |  |
| --- | --- | --- |
| **Feature** | **Machine Learning (ML)** | **Deep Learning (DL)** |
| Definition | Subset of AI focused on learning from data. | Subset of ML using multi-layered neural networks. |
| Feature Engineering | Manual feature extraction. | Automatically learns features from data. |
| Data Dependency | Works well on small–medium datasets. | Needs large-scale data to perform optimally. |
| Algorithms | Linear Regression, SVM, KNN, Decision Trees. | CNN, RNN, LSTM, Transformers, GANs. |
| Training Time | Fast to train. | Requires more time and computational power. |
| Use Cases | Fraud detection, spam filters, recommendation systems. | Face detection, medical imaging, language translation. |

# What is a Perceptron?

A Perceptron is the simplest type of artificial neural network, mainly used for binary classification (output: 0 or 1). It’s a **single-layer neural network** inspired by how biological neurons work.

It is the fundamental building block of modern neural networks.

****

# Components of a Perceptron

**1**. **Input Features:** Denoted as **x1, x2, x3, ..., xn**. These represent the features extracted from your data (e.g., pixel values in images, text, audio , video etc.).

**2. Weights:** Denoted as **w1, w2, w3, ..., wn**. Each input has a corresponding weight that determines its importance in the decision.

**Number of inputs = number of weights**

**3. Bias (b):** An extra parameter added to the weighted sum to shift the activation function.

**4. Net Input Function**: A weighted sum of the inputs plus the bias.

**Z = w·x + b**

Where: x=[x1​,x2​,...,xn​] are the input features

w=[w1,w2,...,wn] are the weights

b is the bias term

**5. Activation Function:** Introduces **non-linearity,** enabling the network to model complex patterns. It is a mathematical function applied to the output of a neuron.

Without this non-linearity feature a neural network would behave like a linear regression model no matter how many layers it has.

**Common Activation Functions:**  
- ReLU  
- Sigmoid  
- Tanh  
- Softmax  
- Leaky ReLU

**6. Output:** Result from the activation function, either passed to the next layer or returned as final decision.

# Activation Function Selection

|  |  |  |
| --- | --- | --- |
| Task | Hidden Layers | Output Layer |
| Binary Classification | ReLU | Sigmoid |
| Multi-class Classification | ReLU | Softmax |
| Regression | ReLU or Linear | Linear |

Note: Use Leaky ReLU when facing the 'dead neuron'(vanishing gradient) issue caused by standard ReLU.