# Deep Learning

# What is Deep Learning

Deep Learning is collection of statistical techniques of machine learning for learning feature hierarchies that are actually based on artificial neural networks.

# Deep learning vs Machine learning

|  |  |  |
| --- | --- | --- |
|  | Deep Learning | Machine Learning |
| **Data** | Needs a big data | Performs well with a small to a medium dataset |
| **Hardware requirements** | Requires machines with GPU | Works with low-end machines |
| **Engineering peculiarities** | Needs to understand the basic functionality of the data | Understands the features and how they represent the data |
| **Training time** | Long | Short |
| **Processing time** | A few hours or weeks | A few seconds or hours |
| **Number of Algorithms** | Few | Many |
| **Data interpretation** | Difficult | Some ML algorithms are easy to interpret, whereas some are hardly possible |

# What is Neuron and Neural Networks, Types of Deep learning Networks

## Neuron

Neurons are nerve cells that send messages all over your body to allow you to do everything from breathing to talking, eating, walking, and thinking.

## Neural Networks

A neural network is a machine learning model inspired by the human brain, using interconnected nodes (neurons) to process data and learn patterns.

## Types of Deep Learning Networks

* **Perceptron**
* **Feed Forward Networks**
* **Multi-Layer Perception (ANN)**
* **Radial Based Networks**
* **Convolutional Neural Networks** (use for image data)
* **Recurrent Neural Networks** (use for text data)
* **Long Short-Term Memory Networks**

# Single Layer Perceptron

A single-layer perceptron is a fundamental building block in deep learning, representing the simplest form of a neural network. It consists of a single layer of neurons that receive inputs, perform a weighted sum, and apply an activation function to produce an output. This output is typically a binary classification (0 or 1) or a continuous value.

# Perceptron Work

dataset = pd.read\_csv("preceptron\_customer\_purchase\_dataset.csv")

dataset.head(3)

plt.figure(figsize=(4,3))

sns.scatterplot(x="Age", y="AdClicks", data=dataset, hue="Purchase")

plt.show()

x = dataset.iloc[:,:-1]

y = dataset["Purchase"]

from sklearn.model\_selection import train\_test\_split

x\_train, x\_test, y\_train, y\_test = train\_test\_split(x, y, test\_size=0.2, random\_state=42)

from sklearn.linear\_model import Perceptron

pr = Perceptron() # alpha=0.01

pr.fit(x\_train, y\_train)

pr.score(x\_train, y\_train)\*100, pr.score(x\_test, y\_test)\*100

from mlxtend.plotting import plot\_decision\_regions

plt.figure(figsize=(4,3))

plot\_decision\_regions(x.to\_numpy(),y.to\_numpy(),clf=pr)

plt.show()

# Multi-Layer Perceptron (ANN)

A Multilayer Perceptron (MLP) is a type of neural network that uses multiple layers of interconnected nodes (neurons) to learn complex patterns and relationships in data. It's a fundamental architecture in deep learning, known for its ability to handle non-linear problems and learn from training data.

* **Input Layer:** Receives the initial data or features as input.
* **Hidden Layers:** Perform computations on the input data, transforming it and extracting features. A network with one or more hidden layers is considered a deep neural network.
* **Output Layer:** Generates the final prediction or result based on the processed information from the hidden layers.

# Forward Propagation and Back Propagation

## Forward Propagation

Forward propagation is the process of passing input data through a neural network, layer by layer, to generate an output.

## Back Propagation

Back Propagation is one of the important concepts of a neural network. Our task is to classify our data best. For this, we have to update the weights of parameter and bias.

# Activation Function for Neutral Network

An activation function is a mathematical function applied to the output of a neuron. It introduces non-linearity into the model, allowing the network to learn and represent complex patterns in the data. Without this non-linearity feature, a neural network would behave like a linear regression model, no matter how many layers it has.

Activation function decides whether a neuron should be activated by calculating the weighted sum of inputs and adding a bias term. This helps the model make complex decisions and predictions by introducing non-linearities to the output of each neuron.