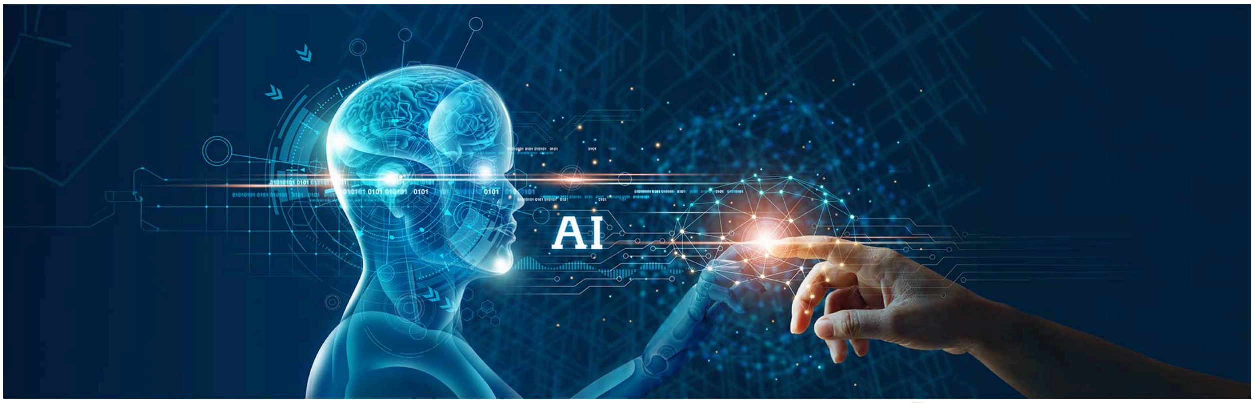


Gen AI Course Content 📔

> 🚀 **Loading your AI & ML journey… Buckle up for an exciting adventure!** 🎯✨

Artificial Intelligence

Definition:

AI refers to the simulation of human intelligence in machines that are programmed to think and act like humans, performing tasks that usually require human intelligence.

Key Features of AI:

- Problem-solving and decision-making
- Natural language processing (NLP)
- Image and speech recognition
- Robotics and automation

Examples of AI in Everyday Life:

Alexa/Siri, Autonomous Vehicles, Recommend Systems

Machine Learning

Definition:

Machine Learning is a subset of AI that involves training algorithms to recognize patterns in data and make predictions or decisions based on it, without explicit programming.

Types of Machine Learning :

- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning

Examples of AI in Everyday Life:

Spam Email Detection, Anomaly Detection, Stock Market Prediction etc

Deep Learning

Definition:

Deep Learning is a subset of machine learning which is based on artificial neural network architecture.

Characteristics of Deep Learning :

- An artificial neural network uses layers of interconnected neurons that work together to process and learn from the input data.
- In a fully connected Deep neural network, there is an input layer and one or more hidden layers connected one after another.
- Each neuron receives input from previous layers neurons or the input layer.
- The output of each neuron becomes the input of other neurons in next layer of the network, and this process continues until the final layer produces the output of the network.

Generative AI

Definition:

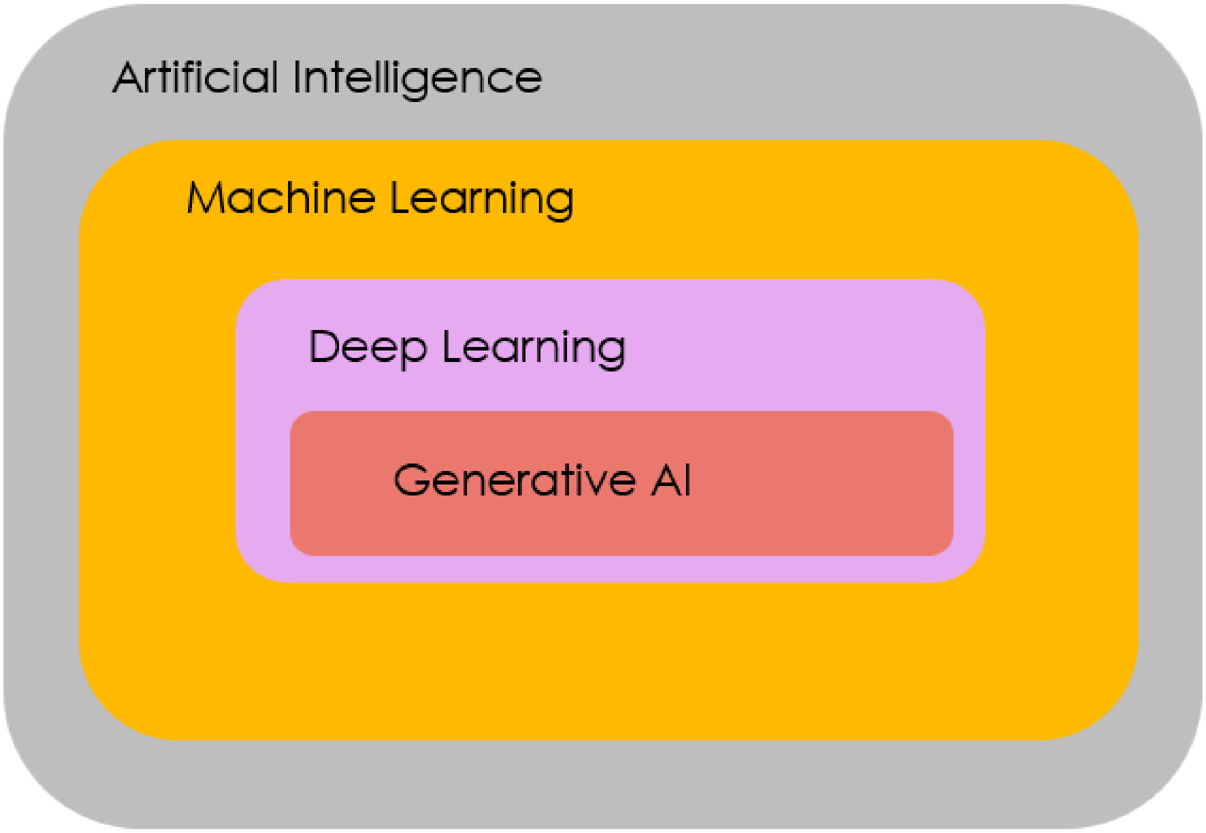
Generative AI is a type of AI that can generate new content—whether text, images, music, or even video—based on patterns and knowledge it has learned from existing data.

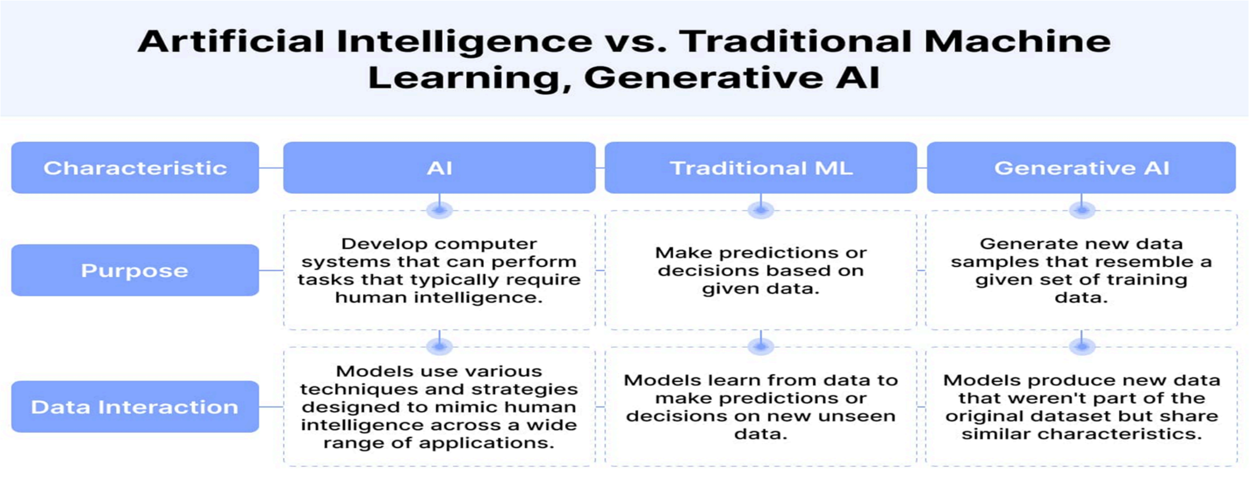
Key characteristics:

- **Creativity**: Capable of producing new, human-like content.
- **Learning from Data**: Trains on massive datasets to replicate patterns.
- **Flexibility**: Can create various types of content (text, images, audio).

Examples of gen AI in Everyday Life:

ChatGPT, Gemini, DALLE, DeepSeek

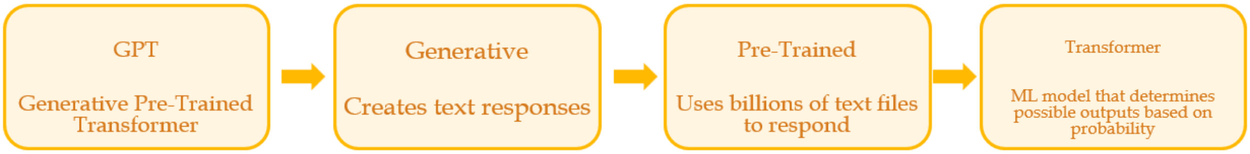




Artificial Intelligence vs. Traditional Machine Learning, Generative AI

Characteristic	AI	Traditional ML	Generative AI
Purpose	Develop computer systems that can perform tasks that typically require human intelligence.	Make predictions or decisions based on given data.	Generate new data samples that resemble a given set of training data.
Data Interaction	Models use various techniques and strategies designed to mimic human intelligence across a wide range of applications.	Models learn from data to make predictions or decisions on new unseen data.	Models produce new data that weren't part of the original dataset but share similar characteristics.

How Does Generative AI Works



GPT

Generative Pre-Trained Transformer

Generative

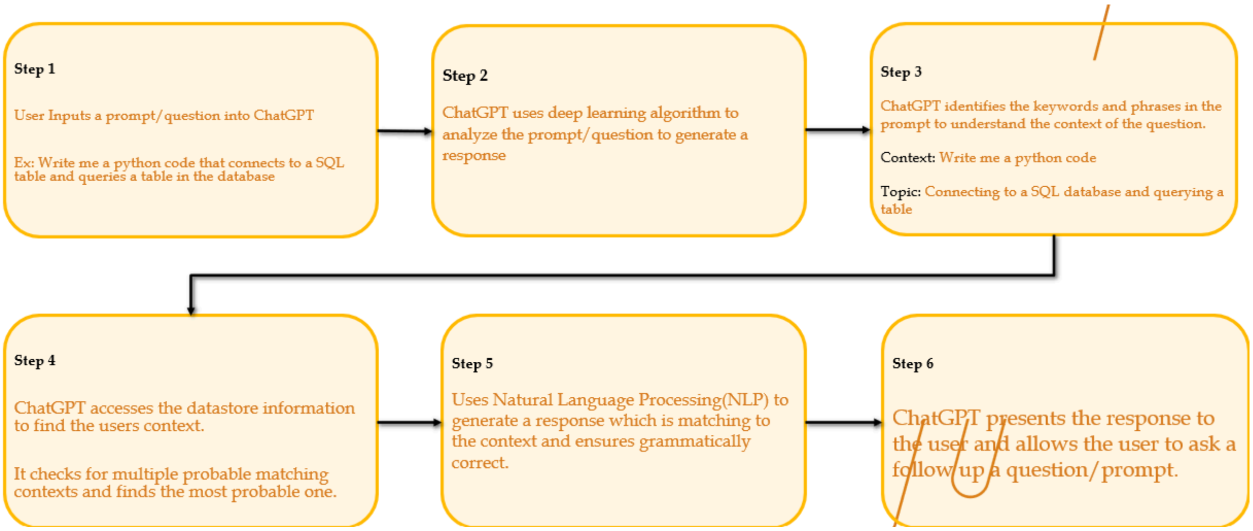
Creates text responses

Pre-Trained

Uses billions of text files to respond

Transformer

ML model that determines possible outputs based on probability



Step 1

User Inputs a prompt/question into ChatGPT

Ex: Write me a python code that connects to a SQL table and queries a table in the database

Step 2

ChatGPT uses deep learning algorithm to analyze the prompt/question to generate a response

Step 3

ChatGPT identifies the keywords and phrases in the prompt to understand the context of the question.

Context: Write me a python code

Topic: Connecting to a SQL database and querying a table

Step 4

ChatGPT accesses the datastore information to find the users context.

It checks for multiple probable matching contexts and finds the most probable one.

Step 5

Uses Natural Language Processing(NLP) to generate a response which is matching to the context and ensures grammatically correct.

Step 6

ChatGPT presents the response to the user and allows the user to ask a follow up a question/prompt.

What is a Model?

Definition:

A **machine learning model** is a mathematical representation that learns patterns from data in order to make predictions, classifications, or generate new outputs.

Key Components of a Model:

1.**Data**: The raw material that trains the model. High-quality, relevant data is essential for accurate learning.

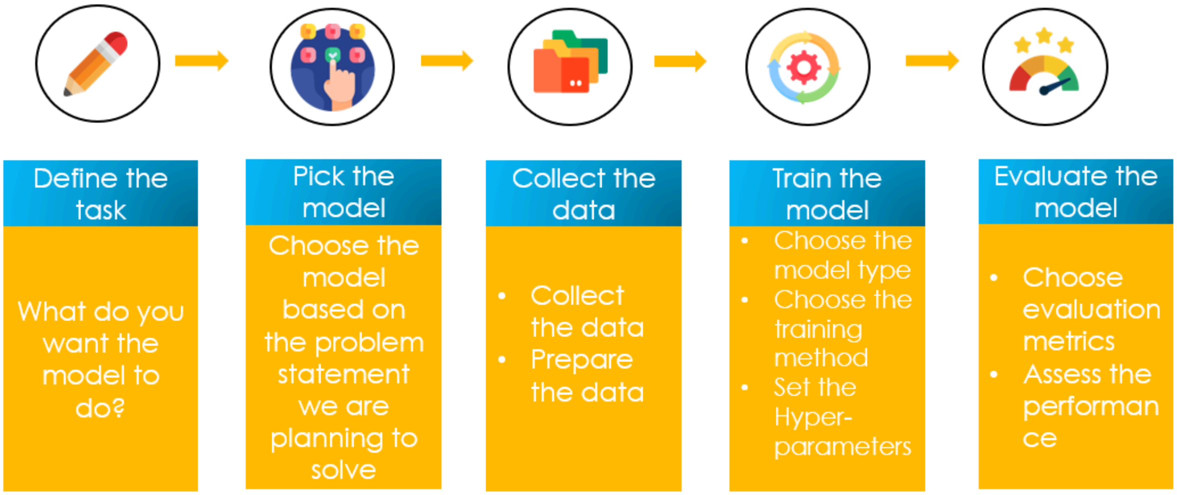
2.**Architecture**: The structure of the model (e.g., neural networks, decision trees, transformers). It defines how the data flows through the model and how it processes it.

3.**Parameters**: Internal variables (like weights in neural networks) that the model adjusts during training to minimize errors.

4.**Loss Function**: A function that measures how far off the model's predictions are from the actual results. The goal is to minimize the loss.

5.**Optimization**: The process of adjusting the model's parameters to reduce the loss and improve the model's performance over time.

High Level Machine Learning Process



Define the task

What do you want the model to do?

Pick the model

Choose the model based on the problem statement we are planning to solve

Collect the data

- Collect the data
- Prepare the data

Train the model

- Choose the model type
- Choose the training method
- Set the Hyper-parameters

Evaluate the model

- Choose evaluation metrics
- Assess the performance

Types of Machine Learning

Supervised Learning

Definition:

Supervised learning is defined as when a model gets trained on a **“Labelled Dataset”.** Labelled datasets have both input and output parameters.

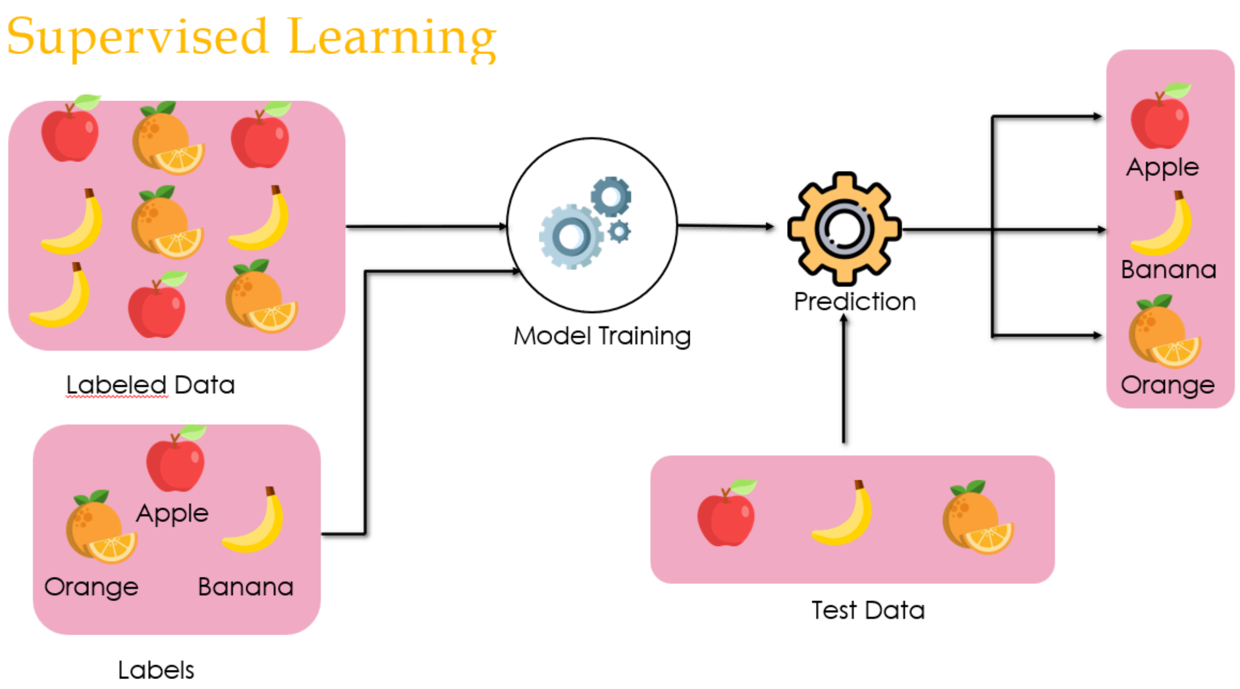
In Supervised Learning algorithms learn to map points between inputs and correct outputs. It has both training and validation datasets labelled.

Categories:

- Classification
- Regression

Applications of Supervised Learning:

Image classification, Speech recognition, Email spam detection, Weather forecasting, Medical diagnosis



Classification

Classification deals with predicting categorical target variables, which represent discrete classes or labels. For instance, classifying emails as spam or not spam, or predicting whether a patient has a high risk of heart disease. Classification algorithms learn to map the input features to one of the predefined classes.

Here are some classification algorithms:

- Logistic Regression
- Support Vector Machine
- Random Forest
- Decision Tree
- K-Nearest Neighbors (KNN)
- Naive Bayes

Regression

Regression, on the other hand, deals with predicting continuous target variables, which represent numerical values.

For example, predicting the price of a house based on its size, location, and amenities, or forecasting the sales of a product. Regression algorithms learn to map the input features to a continuous numerical value.

Here are some classification algorithms:

- Linear Regression
- Polynomial Regression
- Ridge Regression
- Lasso Regression
- Decision tree
- Random Forest

Example: Product price forecasting

Advantages

1.Supervised Learning models can have high accuracy as they are trained on labelled data.
2.The process of decision-making in supervised learning models is often interpretable.

3.It can often be used in pre-trained models which saves time and resources when developing new models from scratch.

Disadvantages

1.It has limitations in knowing patterns and may struggle with unseen or unexpected patterns that are not present in the training data.
2.It can be time-consuming and costly as it relies on labeled data only.

3.It may lead to poor generalizations based on new data.

Unsupervised Learning

Definition:

Unsupervised Learning Unsupervised learning is a type of machine learning technique in which an algorithm discovers patterns and relationships using unlabeled data.

Unlike supervised learning, unsupervised learning doesn’t involve providing the algorithm with labeled target outputs.

The primary goal of Unsupervised learning is often to discover hidden patterns, similarities, or clusters within the data, which can then be used for various purposes, such as data exploration, visualization, dimensionality reduction, and more.

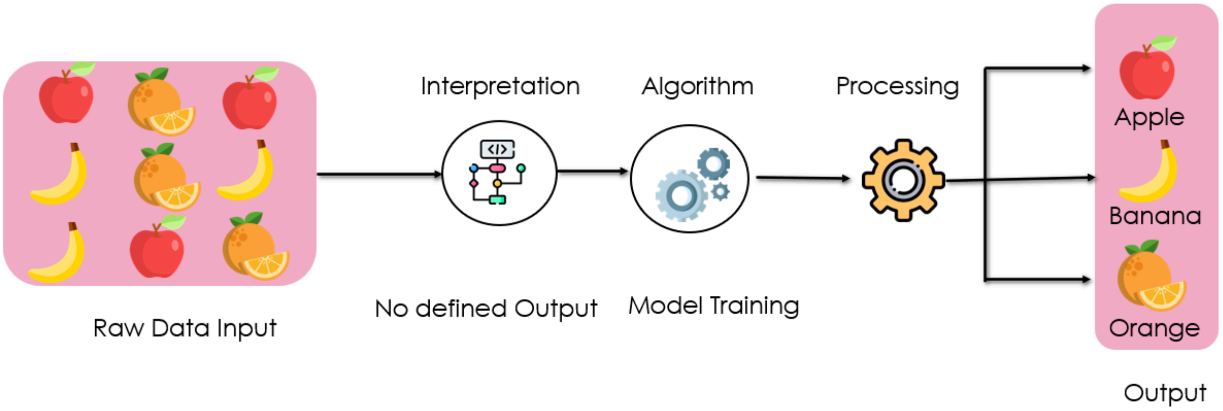
Categories:

- Clustering
- Association

Applications of Un-Supervised Learning:

Clustering, Anomaly detection, Exploratory data analysis (EDA), Content recommendation

Unsupervised Learning



Clustering

Clustering is the process of grouping data points into clusters based on their similarity. This technique is useful for identifying patterns and relationships in data without the need for labeled examples.

Here are some Clustering algorithms:

- K-Means Clustering algorithm
- Mean-shift algorithm
- DBSCAN Algorithm
- Principal Component Analysis
- Independent Component Analysis

Example:

Customer Segmentation, Image Segmentation

Association

Association rule learning is a technique for discovering relationships between items in a dataset. It identifies rules that indicate the presence of one item implies the presence of another item with a specific probability.

Here are some Clustering algorithms:

- Apriori Algorithm
- Eclat
- FP-growth Algorithm

Example:

Market Basket analysis, Website Navigation Analysis

Advantages

1.It helps to discover hidden patterns and various relationships between the data.
2.Used for tasks such as customer segmentation, anomaly detection, and data exploration.
3.It does not require labeled data and reduces the effort of data labeling.

Disadvantages

1.Without using labels, it may be difficult to predict the quality of the model's output.
2.Cluster Interpretability may not be clear and may not have meaningful interpretations.
3.It has techniques such as autoencoders and dimensionality reduction that can be used to extract meaningful features from raw data.

Reinforcement Learning

Definition:

Reinforcement machine learning algorithm is a learning method that interacts with the environment by producing actions and discovering errors**. Trial, error, and delay** are the most relevant characteristics of reinforcement learning.

In this technique, the model keeps on increasing its performance using Reward Feedback to learn the behavior or pattern. These algorithms are specific to a particular problem e.g. Self Driving car where a bot competes with humans and even itself to get better and better performers.

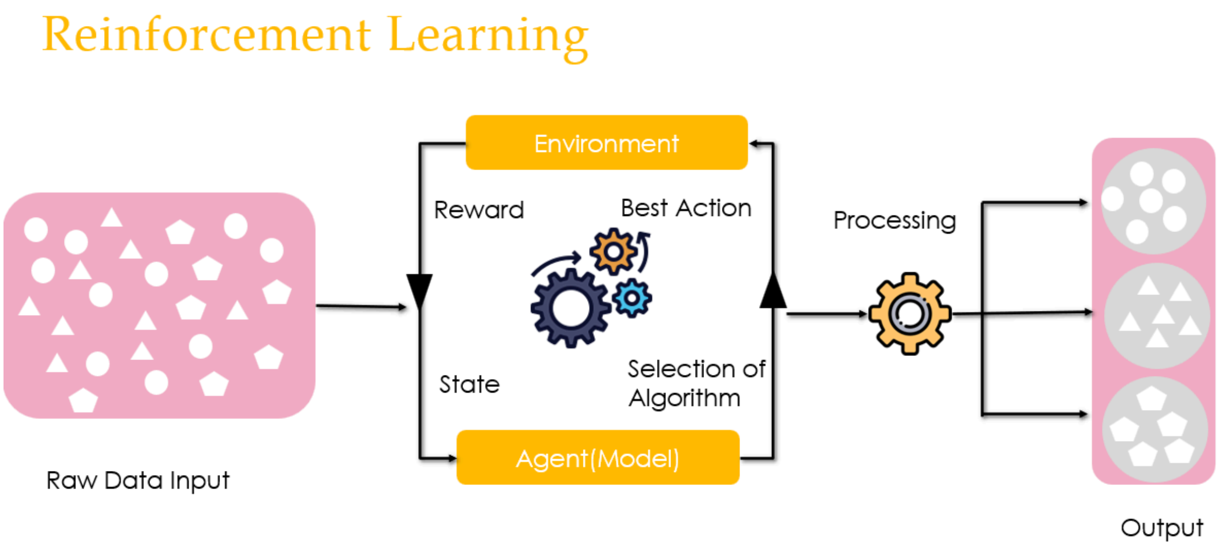
Each time we feed in data, they learn and add the data to their knowledge which is training data. So, the more it learns the better it gets trained and hence experienced.

Common RL algorithms:

- Q-Learning
- SARSA (State-Action-Reward-State-Action)
- Deep Q-learning

Applications of Supervised Learning:

Game Playing, Robotics, SCM, Self Driving Car



Positive Reinforcement

- Rewards the agent for taking a desired action.
- Encourages the agent to repeat the behavior.
- Examples: Giving a treat to a dog for sitting, providing a point in a game for a correct answer.

Negative Reinforcement

- Removes an undesirable stimulus to encourage a desired behavior.
- Discourages the agent from repeating the behavior.
- Examples: Turning off a loud buzzer when a lever is pressed, avoiding a penalty by completing a task.

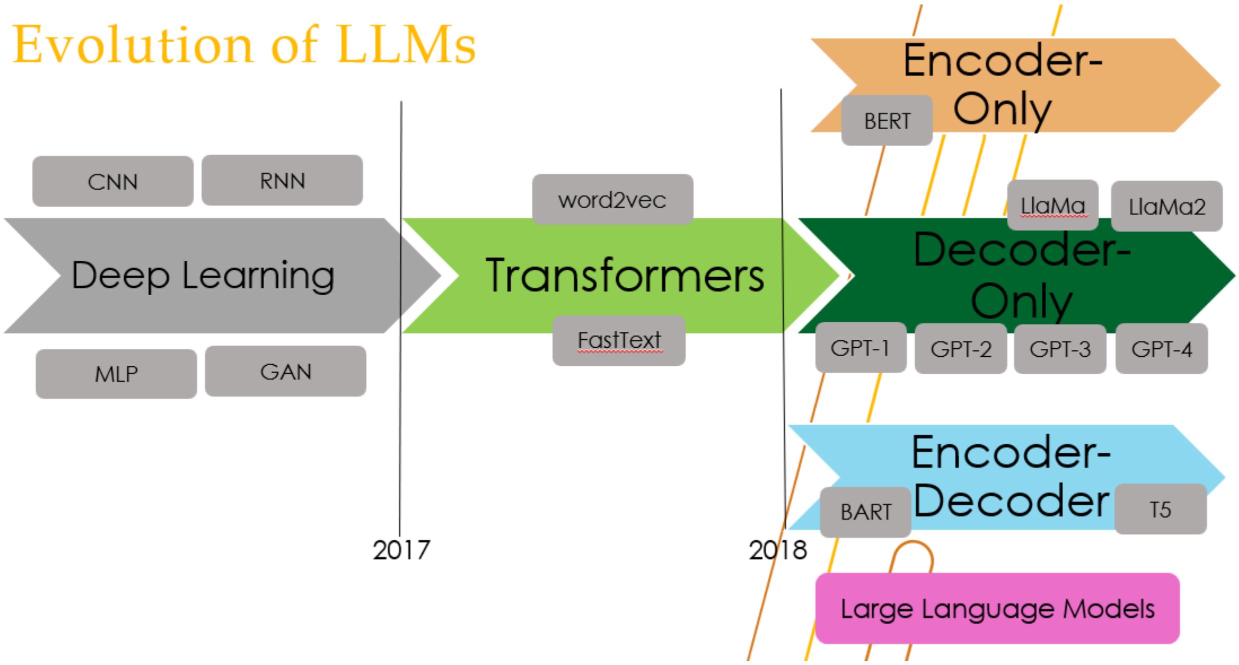
Advantages

1.It has autonomous decision-making that is well-suited for tasks and that can learn to make a sequence of decisions, like robotics and game-playing.
2.This technique is preferred to achieve long-term results that are very difficult to achieve.
3.It is used to solve a complex problems that cannot be solved by conventional techniques.

Disadvantages

1.Training Reinforcement Learning agents can be computationally expensive and time-consuming.
2.Reinforcement learning is not preferable to solving simple problems.
3.It needs a lot of data and a lot of computation, which makes it impractical and costly.

Evolution of LLMs



Artificial Neural Network

Definition:

An Artificial Neural Network is the foundation of deep learning. It consists of layers of neurons (also called nodes) connected by weights. These networks can learn complex patterns from data through training.

Architecture:

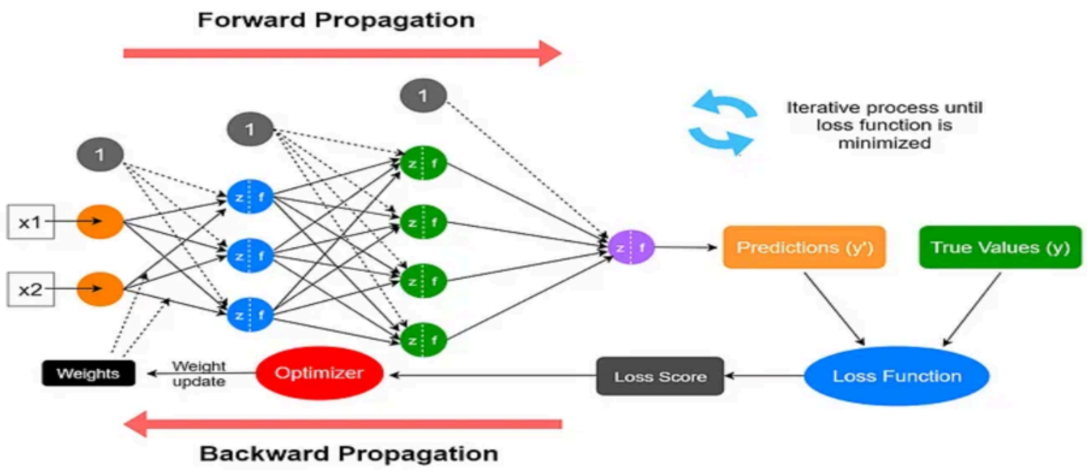
It consists of three main layers:

- **Input layer**: Receives the raw data.
- **Hidden layers**: Layers where computation happens. There can be many hidden layers in a deep neural network.
- **Output layer**: Produces the final prediction or classification.
-

Use case: Classification, Regression

Artificial Neural Network(ANN)

Learning Process:



Convolutional Neural Network(CNN)

Definition:

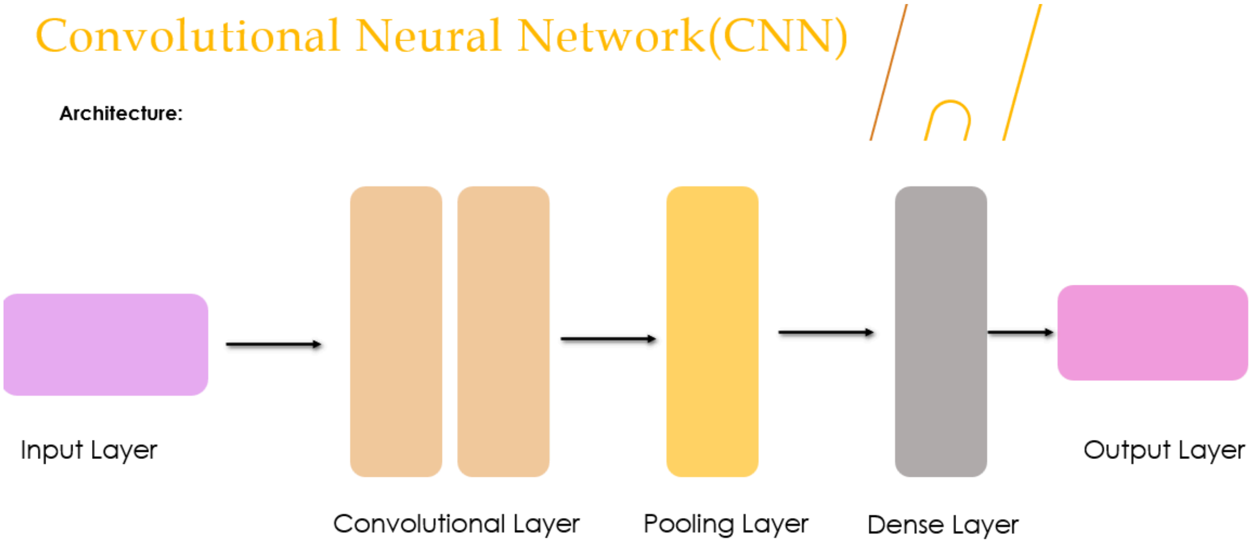
A CNN is specifically designed to process data with a grid-like structure. The most common application is image classification, where the goal is to classify an *image into one or more categories. CNNs are composed of several types of layers, each serving a specific function*

Architecture:

It consists of three main layers:

- **Input layer**
- **Convolutional Layer**
- **Max Pooling layers**
- **Dense layer**
- **Output Layer**

Use case: Classification, Regression



Recurrent Neural Network(RNN)

Definition:

*A Recurrent Neural Network (RNN) is a type of neural network designed for sequential data. In a traditional feedforward neural network data flows in one direction—from input to output. But in an RNN, the network **loops back on itself**, allowing it to maintain memory about previous inputs and their relationships.*

Key Components:

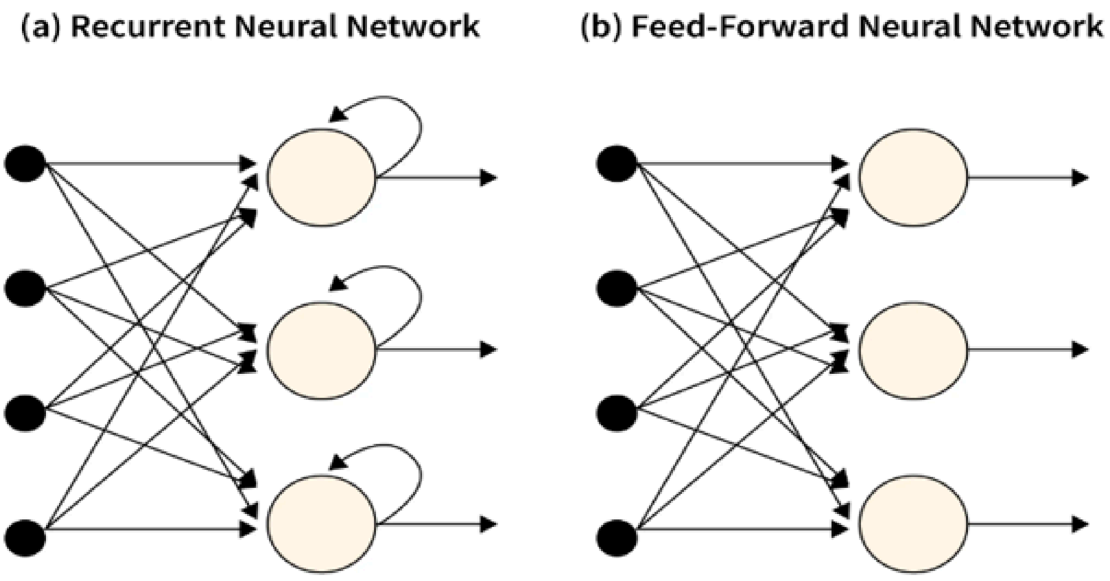
It consists of two main components:

- **Recurrent Neurons**
- **RNN Unfolding**

Use case: Language Modelling, Stock Prediction

Recurrent Neural Network(RNN)

Learning Process:



Natural Language Processing(NLP)

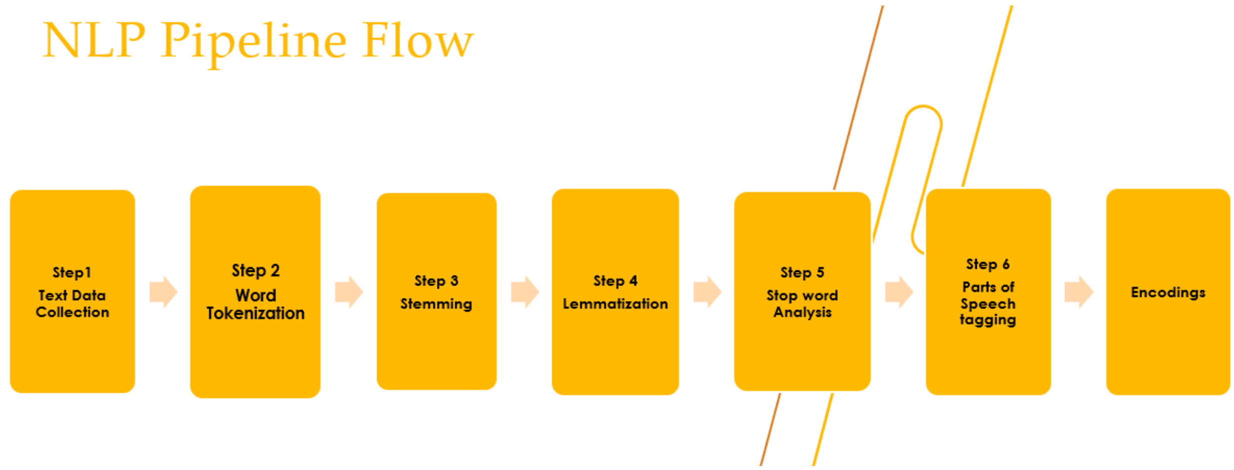
Definition:

Natural language processing (NLP) is a subfield of computer science and artificial intelligence (AI) that uses machine learning to enable computers to understand and communicate with human language.

Practical Use cases of NLP:

- Chatbots
- Sentiment Analysis

- Machine Translation
- Speech Recognition
- Multilingual support and so on ....



Tokenization

Definition:

Tokenization is the process of splitting a stream of text into individual units, called "tokens." These tokens can be words, subwords, or characters, depending on the level of tokenization used.

Types of Tokenization:

- Sentence Tokenization
- Word Tokenization
- Character Tokenization

Word Tokenization

Word tokenization involves breaking down text into individual words. This method provides a good balance between granularity and context, enabling a deeper understanding of the text's semantics.

Character Tokenization

In character tokenization, the text is split into its constituent characters, rather *than words. This is useful for certain applications such as language modeling or when working with languages where words are difficult to separate*

Sentence Tokenization

Sentence Tokenization involves breaking down text into individual sentences. This is especially useful for document analysis or tasks like summarization.

Stemming

stemming is the process of reducing a word to its root form or base stem, usually by removing prefixes and suffixes. The goal is to obtain a word that represents the concept or meaning of its variations

Types of Stemming:

- PorterStemming
- RegexStemming
- Snowball Stemmer

Lemmatization

Lemmatization is a process in Natural Language Processing (NLP) that involves reducing a word to its base or dictionary form, called a lemma. Unlike stemming, which often results in non-words, lemmatization ensures that the root form of the word is a valid, meaningful word in the language.

The key difference between stemming and lemmatization is that lemmatization takes into account the context and the part of speech (POS) of the word. This makes it more accurate and linguistically sound, as it produces real words.

Popular Lemmatization Algorithm:

WordNet Lemmatizer

Stopwords

Stopwords are commonly occurring words in a language (such as "the," "is," "in," "and," etc.) that are often removed during text processing because they don't carry much meaningful information for many NLP tasks like text classification, search indexing, or sentiment analysis.

Parts of Speech Tagging

Part-of-Speech (POS) tagging is a fundamental task in Natural Language Processing (NLP) that involves assigning a grammatical category (such as noun, verb, adjective, etc.) to each word in a sentence.

The goal is to understand the syntactic structure of a sentence and identify the grammatical roles of individual words. POS tagging provides essential information for various NLP applications, including text analysis, machine translation, and information retrieval.

Text Vectorization

Text vectorization is the process of turning words and documents into mathematical representations. These representations capture the semantic meaning in a multidimensional space

Techniques of Encoding:

- One Hot Encoding
- Bag of Words(BoW)
- TF-IDF

One Hot Encoding

One-hot encoding is a technique used in machine learning to convert categorical data into a numerical format so that it can be used by algorithms that require numerical inputs.

1.***Categorical Data:*** *Suppose you have a column with categorical data, such as a "Color" column with values like "Red," "Green," and "Blue."*

2.***Create Binary Columns****: For each unique category, you create a new column. Each new column represents whether the original category is present or not. So, for the "Color" column, you would create three new columns:*

1.*One for "Red"*

2.*One for "Green"*

3.*One for "Blue"*

3.***Assign Binary Values:*** *For each row, you assign a 1 in the column corresponding to the category present in that row, and a 0 in all other columns.*

Advantages

Simplicity: Easy to implement and interpret.

No Assumptions: Does not assume any ordinal relationship between categories

Works for Most ML Models: Suitable for algorithms that can't process categorical data directly.

Disadvantages

1.High Dimensionality: Can lead to sparse, high-dimensional vectors for large datasets.

2.Sparsity: Vectors are sparse, wasting memory and computational resources and overfitting.

3.No Relationship Capture: Does not capture any relationships between categories.

4.Out of Vocabulary

Bag of words(BoW)

BoW is a simple method that represents text data by counting how often each word appears in the text. The idea is that the meaning of a sentence or document is captured by the ***frequency*** *of the words, ignoring grammar and word order.*

How it works:

- Create a vocabulary of all the unique words in your dataset.
- For each document (or sentence), create a vector where each element in the vector corresponds to a word in the vocabulary. The value in the vector represents the frequency (or count) of that word in the document.

Advantages

Simplicity: Easy to implement and understand.

Efficient for Small Datasets: Works well with smaller datasets or when the vocabulary is not too large.

Captures Word Frequency: Useful for models that benefit from knowing the frequency of words in a document (e.g., text classification).

Disadvantages

Ignores Word Order: Does not capture the sequence or context of words, leading to potential loss of meaning.

High Dimensionality: Can produce very large feature vectors, especially with large vocabularies, leading to sparsity and memory inefficiency.

No Semantic Meaning: Treats all words as independent, so it misses relationships between words (e.g., synonyms or similar meanings).

TF-IDF

TF-IDF adjusts the frequency of words based on how common or rare they are across all documents.

How it works:

Term Frequency (TF): Measures how often a word appears in a document. A higher frequency suggests greater importance. If a term appears frequently in a document, it is likely relevant to the document's content.

Inverse Document Frequency (IDF): Reduces the weight of common words across multiple documents while increasing the weight of rare words. If a term appears in fewer documents, it is more likely to be meaningful and specific.

Advantages

Reduces Importance of Common Words: Gives less weight to common words (e.g., "the", "is") and more to rare, relevant words, improving model performance.

Contextual Relevance: Highlights words that are more specific to a document, making it useful for tasks like text classification.

Scalable: Can be applied to a wide range of document collections without significant changes in performance.

Disadvantages

Ignores Word Order and Context: Like BoW, it doesn't capture the sequence or meaning of words, which can be important for some tasks.

High Dimensionality: Creates large, sparse matrices, especially when dealing with large vocabularies, which can be computationally expensive.

Sensitive to Rare Words: Sometimes assigns high importance to words that appear infrequently but may not be particularly meaningful (e.g., typos, unimportant terms).