Generative AI

|  |
| --- |
| Links:  <https://github.com/karthickai/Linear-Regression> |

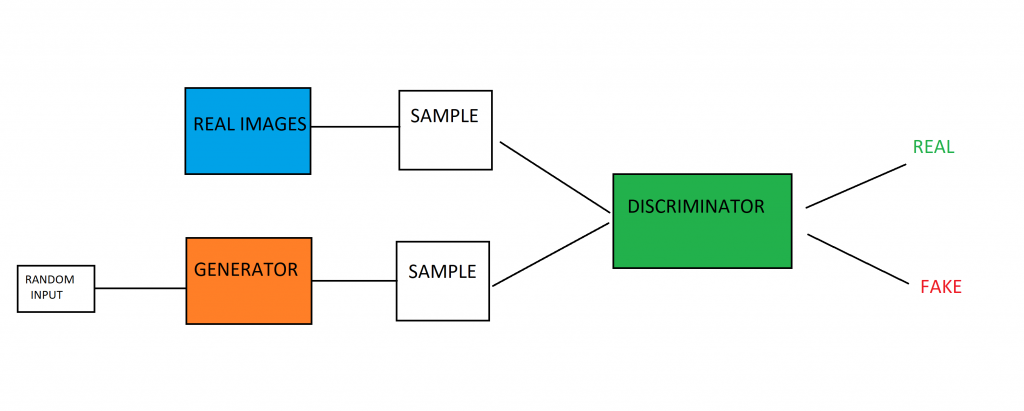
# Concepts :

* Generative AI
* Large Language model
* Open AI
* Langchain
* Vector Database
* Liama Index
* Open source LLM Models
* End to End Project

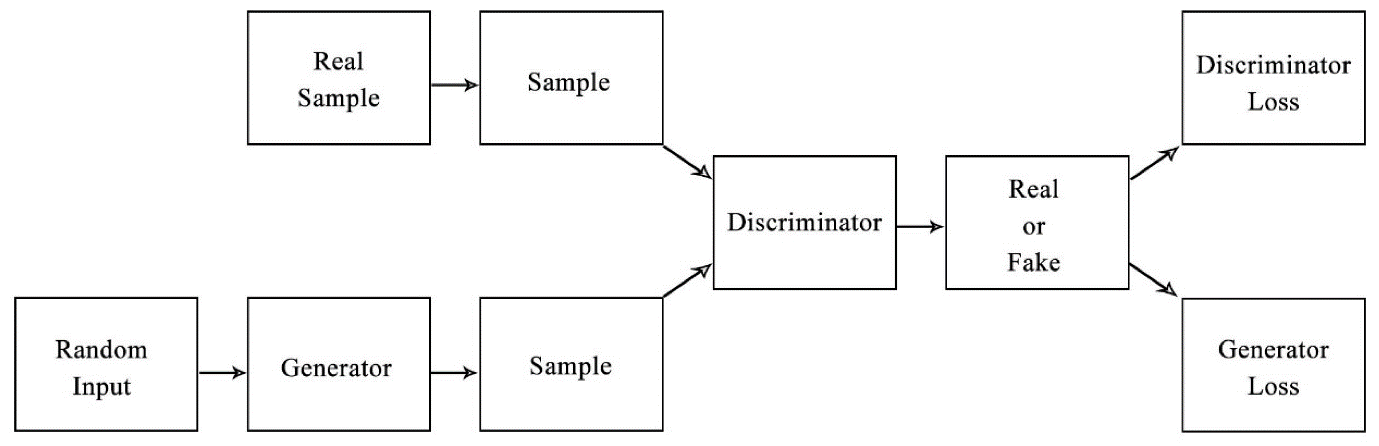
## Deep Learning :

1. Artificial Neural Network (ANN): Structured Data like Numeric data, Categorical data
2. Convolutional Neural Network (CNN): Mostly used for Image and Audio (Grid Datatype)
3. Recurrent Neural Network (RNN): Sequence Related Data
4. Reinforcement Learning (RL)
5. GAN: neural network

### GAN Architecture

1.

2.



**what is generator and discriminator in GAN?**

The generator's aim is to fool the discriminator by producing data that are similar to those in the training set. The discriminator will try not to be fooled by identifying fake data from real data. Both of them work simultaneously to learn and train complex data like audio, video, or image files

## What is Generative AI?

Generative Al generates new data based on training sample. Generative model can generate Image, Text, Audio, Videos etc. data as output.

Example: ChatGPT, Liama, Bard, Devv etc.,

So generative Al is a very huge topics,

* Generative Image model- Image to Image model
* Generative Language model- Text to Image model, Text to text model,

Image to image model

## What is LLM?

Large Language Models (LLMs) are a type of artificial intelligence designed to understand and generate human-like text based on vast amounts of data. They are trained on diverse datasets, which include books, articles, websites, and more, allowing them to perform a wide range of tasks such as answering questions, summarizing information, translating languages, and even creating content.

## Open AI:

OpenAI is a research organization focused on developing artificial general intelligence (AGI) that benefits all of humanity. [They are known for creating advanced AI models, including the well-known ChatGPT, which can assist with a variety of tasks such as writing, learning, brainstorming, and more](https://openai.com/).

### Creating an OpenAI API KEY :

OpenAI API has been degined to provide devlopers with seamless access to state of art, pre trained, artifical intelligence models like gpt-3 gpt-4 dall e whisper,embeddings etc so by using this openai api you can integrate cutting edge ai capabilities into your applications regardless the progamming language.

So,the conclusion is by using this OpenAI API you can unlock the advance functionalities and you can enhane the intelligence and performance of your application.

A screenshot of a computer

Description automatically generated

## Lang Chain:

LangChain is a framework designed to simplify the creation of applications using large language models (LLMs). [It provides a suite of tools and components that help developers build, run, and manage LLM-powered applications more efficiently1](https://www.langchain.com/)[2](https://python.langchain.com/v0.2/docs/introduction/).

Here are some key features of LangChain:

* [**Modular Components**: LangChain offers a library of abstractions for Python and JavaScript, representing common steps and concepts necessary to work with language models3](https://www.ibm.com/topics/langchain).
* [**LangGraph**: This tool helps build stateful, multi-actor applications by modeling steps as edges and nodes in a graph2](https://python.langchain.com/v0.2/docs/introduction/).
* [**LangSmith**: A developer platform for debugging, testing, evaluating, and monitoring LLM applications2](https://python.langchain.com/v0.2/docs/introduction/).
* [**Integration**: LangChain integrates smoothly with various third-party tools and services, making it flexible for different use cases2](https://python.langchain.com/v0.2/docs/introduction/).

## Hugging Face API:

The Hugging Face API provides access to a vast collection of pre-trained machine learning models for various tasks such as text classification, sentiment analysis, question answering, translation, and more. Here are some key features:

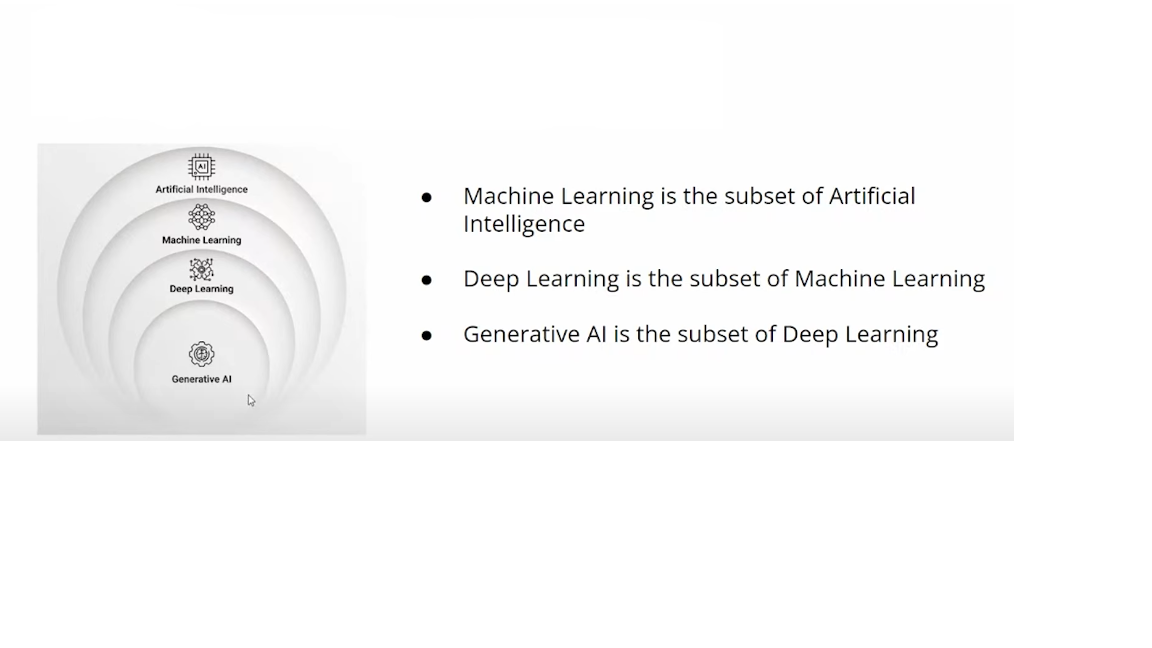
* **Inference API**: Allows you to test and evaluate over 150,000 publicly accessible models or your own private models via simple HTTP requests. [This service is free to use but rate-limited](https://huggingface.co/docs/api-inference/index)[1](https://huggingface.co/docs/api-inference/index).
* **Inference Endpoints**: A production-ready service that lets you deploy any machine learning model on dedicated, fully managed infrastructure. [You can select the cloud, region, compute instance, autoscaling range, and security level to match your needs1](https://huggingface.co/docs/api-inference/index).
* **Hub API**: Provides endpoints to retrieve information about models, datasets, and Spaces stored on the Hugging Face Hub. [It also allows you to perform actions like creating repositories and managing models2](https://huggingface.co/docs/hub/api).

## Memory Config in LangChain:

In LangChain, memory configuration is crucial for managing conversational context. Here are some key aspects:

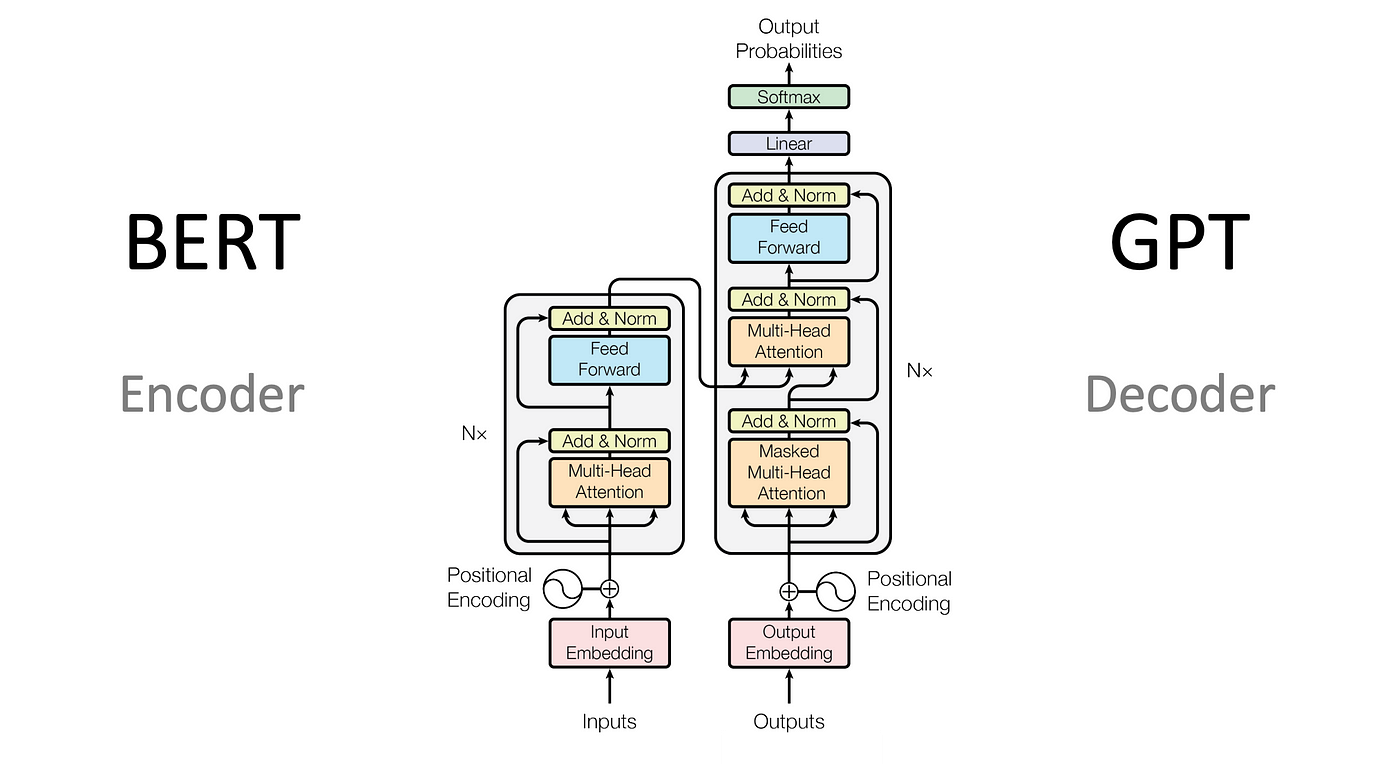
1. **Basic Memory Configuration**:
   * **ConversationBufferMemory**: Stores the entire conversation history for context. [This is passed to the ConversationChain as a parameter](https://zilliz.com/learn/langchain-memory-enhancing-AI-conversational-capabilities)[1](https://zilliz.com/learn/langchain-memory-enhancing-AI-conversational-capabilities).
2. **Customizing Memory**:
   * [You can customize memory by using different memory classes like ConversationBufferMemory, ConversationSummaryMemory, or ConversationEntityMemory2](https://python.langchain.com/v0.1/docs/modules/memory/conversational_customization/).
3. **Persisting Memory**:
   * To save and load conversation memory, you can use persistent storage options. [For example, you can integrate with databases or use in-memory lists3](https://stackoverflow.com/questions/75965605/how-to-persist-langchain-conversation-memory-save-and-load).
4. **Advanced Memory Management**:
   * [LangChain supports more complex memory management techniques, such as summarizing long conversations or extracting entities for better context management4](https://python.langchain.com/v0.1/docs/use_cases/chatbots/memory_management/)[5](https://python.langchain.com/v0.1/docs/modules/memory/).

**Where Generative AI Exists?**



|  |
| --- |
| Attention? |

**The transformer uses an encoder-decoder architecture :**



 The encoder extracts features from an input sentence, and the decoder uses the features to produce an output sentence (translation)

Refer this link for basic understanding :  
<https://kikaben.com/transformers-encoder-decoder/#encoder-block-internals>

Transformer Tree  
