Nagaraj Loni || Computer science and Engineering-2026 || Generative AI || Agentic AI || ML || Computer Vision ||

**NS LONI**

**GEN AI**

**2K25 NOTES**

**Overview of Gen-AI**

1. Generative AI: Generative ai is generate new data based on training sample. Generative model can generate Image, Text, Audio, Videos....etc.

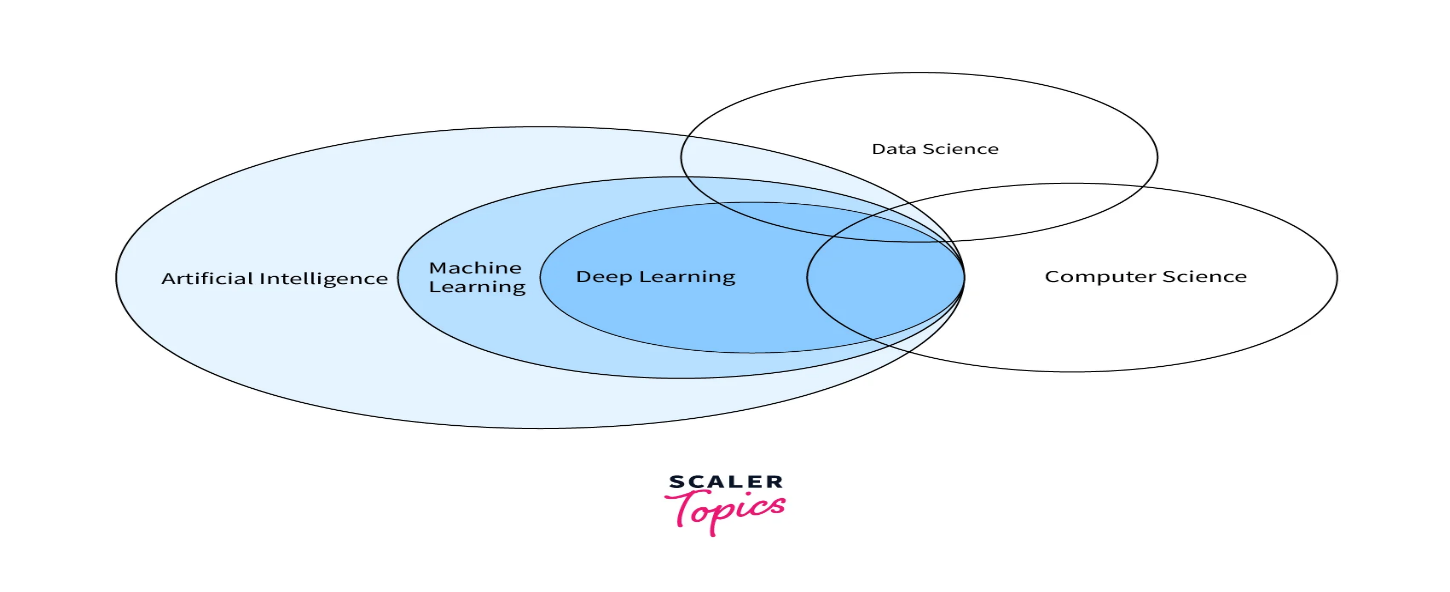
* Generative Image Model - 2018, 19. It's Old
* Generative Language Model - large language model

- Using llama generate an Image and more

- Text to Image / Image to Text

- Implement Image to Image

2. Generative Model : | input| ---> LLM ----> | Output |

3. Generative AI is a Part of Deep Learning 

*4. Olden Days tools*

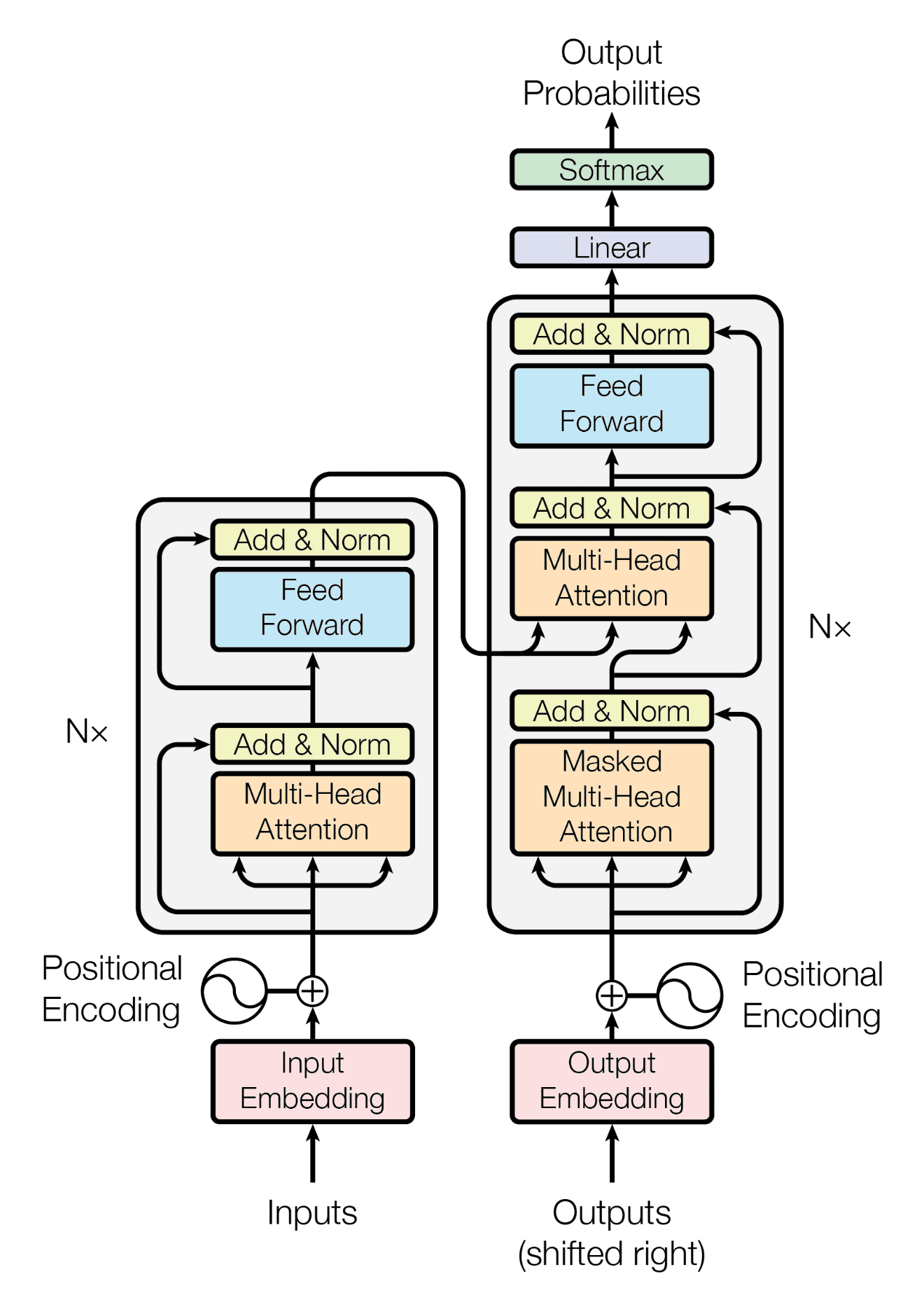
Having sum retractions

* RNN
* LSTM
* GRU
* These are the need a **Sequence** **Data.**
* Input 🡪 Output [ Both will have same length]

*5. Now day’s Using as following:*

* Encode[RNN, LSTM, GRU] 🡪 Decoder[RNN, LSTM, GRU] it gives the small length of the output.
* If User Input and output length will same & longer input and output will not support.
* Refer more research paper for reference.
* Above tech tool are in the year of 2014 to 2016.
* Encode 🡪 Layer and Vector 🡪 Decoder [Attention needs 2018 Paper {NLP}].

6. Transformer Model ML,AI,DP,NLP.



* Pass input in parallel manner.
* Multi-Head {Find the relation b/w each word.
* Feed Forword and Neural Network
* Repeating output to input and final get the correct output.
* Linear
* Get Output by SoftMax.



**7. Discriminative V/S Generative Model**

1. [Discriminative ] Music 🡪 D-Model {llama} 🡪 (It may be) 🡪 I. Rock, ii. Classical, iii. Romantic etc.

2. [Gen-AI] Music 🡪 Gen-Models {llama} 🡪 New

* Discriminative models focus on learning decision boundaries between classes, directly predicting output probabilities given inputs.
* They excel in classification tasks and require labelled data. Examples include logistic regression and SVMs. Generative models, conversely, learn the underlying data distribution, enabling them to generate new samples and model joint probabilities of inputs and outputs.
* They are versatile, suitable for both supervised and unsupervised learning, and can work with unlabelled data. Examples include Bayesian networks and GANs.
* Discriminative models typically capture p(Y|X), while generative models capture p(X, Y) or p(X). The choice between them depends on the specific task, available data, and desired outcomes.

8. LLM In Gen-AI

* Large Language Models (LLMs) are a specialized subset of generative AI focused on processing and generating human-like text.
* They are built on advanced Transformer architectures and trained on massive text datasets, enabling them to understand context, generate coherent responses, and perform various language-related tasks.
* LLMs excel at text generation, translation, summarization, and question-answering. While generative AI encompasses a broader range of content creation (including images, audio, and video), LLMs specifically deal with text-based applications.
* Examples include GPT-3 and BERT. LLMs have revolutionized natural language processing and find applications in customer support, content creation, and educational tools. They represent a crucial component of AI innovation in language-related tasks.
* [How] 🡪 Trian huge data and generating new data,
* [why] 🡪 Size & complexity will more then its work it on large/huge/big-data data therefore called large model.
* LLM working based on the Transformer model architecture.
* **Mile-Stone in LLM**
* BERT{Google}, GPT{OpenAI}, XLM{Lample}, T5{Google AI}, Megatron{NVIDIA}, M2M-100{Facebook}.
* Decoder LLM’s : GTP, GTP-2, GTP-3, GTP-Neo, GTP-J, CTRL.
* Encoder LLM’s : BERT, XLM, ALBERT etc.
* Using Both Decoder & Encoder 🡪T5, BART, M2M-100, Big Bird.
* OpenAI Based LLM models [With Premium]: GPT-4, GPT-3.5, GPT-base, GPT-3 etc.
* Other open Model [Free]: BLOOM, Llama, PaLM, etc.
* Used for {LLM}: Test Classification, Text Generation, Text Summarization, Q&A, Spelling Correction.
* Prompt Designing [I/P and O/P prompt].

How GPT was Trained ?

Internally Using a LLM which is GPT-3.5 or GPT -4. It has trained on a large of data Which is available all over the internet.

* Generative Pre-Training.
* Supervised fine-tuning.
* Reinforcement Learning.

**Open API**

1. OpenAI :

* OpenAI API: using a Python, Use more lang like JS, Node etc.
* Playground: How setup and create an env.
* Chak Completion : Create an API.
* Finally Function Call.
* Started the work on OpenAI in 2019 then 2020-21 release.
* Read Researches of OpenAI.

2. Hugging Face:

* All the open Sources {llama}.

3. Practical [Install 🡪 Jupyter Notebook Python, Anaconda, VS Code / PyCharm etc]

* Create Virtual ENV.
* Install All Required Packages and more.

4. Start Practical Implementation. & Create an env.

* Step-1
* conda create -n testingopenapi python=3.10 [enter]
* conda activate testingopenapi [enter]
* pip list
* Step-2
* Pip install jupyter notebook
* Pip list
* Jupyter notebook
* 🡪 It opens a Jupyter Notebook….
* Created a notebook successfully.
* Start coding [Set python 3]

5. Coding Part

* Use QuickStart in OpenAI website and setup virtual env[install all package like OpenAI]
* **Note:** Don’t stop cmd.
* Write code use my GitHub generative ai repositories.
* Person-1 🡨 🡪 Communication 🡨 🡪Person-2.
* Creating new model will need 1-10 cr. Then Use Open API then access GPT Model then conversion will do.
* Open API Key: need money then we use the Hugging Face…

If u need code as following:

*import openai*

*mykey="Open API key "*

*openai.api\_key=mykey*

*all\_models=openai.models.list()*

*list(all\_models)*

*import pandas as pd*

*pd.DataFrame(list(all\_models),columns=["id","created","object","owned\_by"])*

*openai.Completion.create(*

*model="GPT-3.5",*

*prompt="who was the first prime minister of india?"*

*)*

*# This code is for v1 of the openai package: pypi.org/project/openai*

*from openai import OpenAI*

*client = OpenAI(api\_key=mykey)*

*response = client.chat.completions.create(*

*model="gpt-3.5-turbo",*

*messages=[*

*{*

*"role": "user",*

*"content": "who won the first cricket worldcup?"*

*}*

*]*

*,*

*max\_tokens=150,*

*n=3*

*)*

*type(response)*

*response*

*response.choices[0].message.content*

*# now let try to understand the different parameters inside the methods*

*model= ""*

*prompt=input prompt*

*max\_tokens=in how many number of tokens you want result*

*temperature=for getting some creative output*

*n= number of the output*

: I am Nagaraj Loni

Token and Char’s: **there are 4 tokens having different colour and 14 chars.**

**Lang-Chain with OpenAI**

1.pip install langchain and langchain.