Gen AI

Bhadri Vaidhyanathan – Personal Notes

Contents

Intro

AI > ML > Deep Learning > Gen AI and LLM

Gen AI uses neural networks and hence sub field of Deep Learning.

Generative vs Discriminative (older ML methods)

Generative : make new stuff	Discriminative : tell what stuff is
Good for generating new	Good for deciding between known labels
Data is modeled; Concept learning ; basic fundamentals understood and knowledge used to create. Eg: Understanding the science to be ready for exam	Shortcuts to predict; use existing data to decide boundaries; Eg: Like practicing last 5 years exam papers to be ready for exam
Warning: could go wrng if concept is misunderstood or used wrng way. Takes longer and with experience	More robust, resilient to outlier data but still training data should have good models wch is usually easy to get
Probabilistic; Bayesian reasoning	Pre-determined boundary values are checked; function approximation
Think LLM, how every word is decided based on a probability distribution from training data	Think Regression, how least squares is used to decide on that best fit line which alone is used later to decide on a new point
Needs less data to get started but application is complex due to susceptibility to outliers; probability distributions created around each and these probs used to decide when tested	Needs lots of labeled data, doable, understandable. Algo learns the boundaries and later boundaries used to decide
Naive Bayes on steroids	Log. Regression on steroids; support vector machines, NN Although u can play around and include Bayesian into above structures and make it

	generative
susceptibility to outliers	Focused on creating the best boundary and so values close to boundary alone matter. Most of the other data in a way ignored and outliers are understood and ignored
Explainable	Black box; certain decisions are not easily explainable

Generative Adversarial Networks: Deep Learning method which uses both generative and discriminative to create new data points

Basic Working:

LLMs are basic word prediction models which is the core engine.
How does LLM respond to a question like a human – Supervised Learning.

1. LLMs are fine-tuned with with text on typical response. Its like mugging a specific response for a type of question
2. Second method is to take multiple outputs from core engine and score it. Pick the best answer from core engine. Like how we were trying in the UChicago lab.
   This method is also called RLHF: Reinforced Learning from Human Feedback where many responses were labelled by humans as Helpful, Honest, Harmless and this text is used to score the multiple ones
3. Integrated Tools:
   Tools like Calculator or Calendar are integrated so LLM can use them to get the right response for a certain query.

GenAI Applications:

1. Writing

    Proofreading

2. Reading

    Summarizing long docs

3. Chatting

4. Brainstroming

LLM Limitations:

1. Knowledge cutoff: data only till the date of training

2. Hallucinations:
   Eg: Give me a quote that Shakespeare gave about Beyonce but LLM will still give an answer confidently

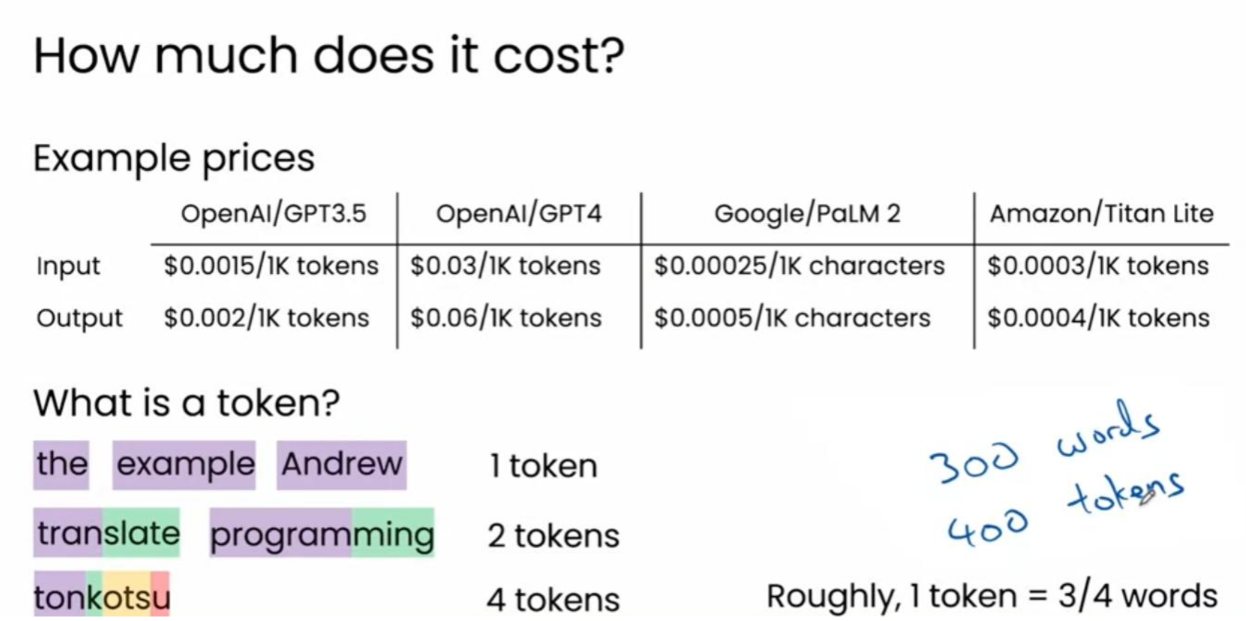
3. Input and output text limit
   Common limits are a thousand words but specialized LLMs are available that overcome this specific limit.

4. Structured data: Struggles with tabular data like excel files
   It cannot comprehend relationships between variables and give out an answer which a linear regression ML model can

5. Bias and Toxicity
   Garbage in garbage out. So the training text had a bias then LLM will propagate the bias

Cost:

Token is either a word or sometimes each syllable for complex words. Estimate is add 33% of word count for total number of tokens.



How much does it cost?

Example prices

	OpenAI/GPT3.5	OpenAI/GPT4	Google/PaLM 2	Amazon/Titan Lite
Input	$0.0015/1K tokens	$0.03/1K tokens	$0.00025/1K characters	$0.0003/1K tokens
Output	$0.002/1K tokens	$0.06/1K tokens	$0.0005/1K characters	$0.0004/1K tokens

What is a token?

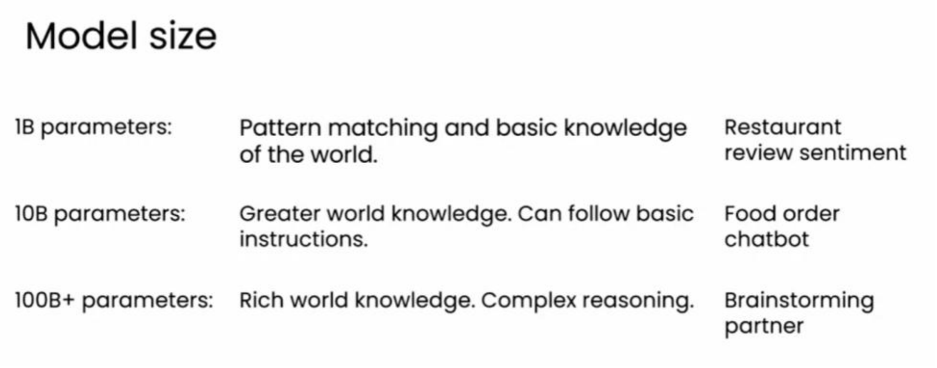
the example Andrew	1 token
translate programming	2 tokens
tonkotsu	4 tokens

300 words
400 tokens

Roughly, 1 token = 3/4 words

Choosing a Model:

What model size should you choose for an application:



Model size

1B parameters:	Pattern matching and basic knowledge of the world.	Restaurant review sentiment
10B parameters:	Greater world knowledge. Can follow basic instructions.	Food order chatbot
100B+ parameters:	Rich world knowledge. Complex reasoning.	Brainstorming partner

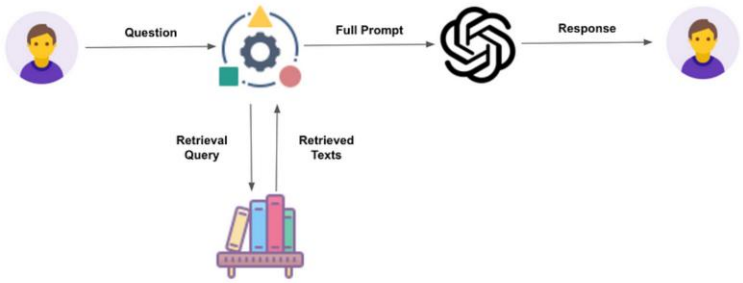
Closed or Open Source:

Closed is like private cloud while open source requires lot of self admin.

Advanced LLM Techniques / Applications:

Retrieval Augmented Generation:

- From prompt identify relevant docs from custom db.
- Add text from these docs to prompt as input and then generate relevant answer



+

Fine Tuning:

Add some more training text to specialise a LLM – way cheaper than pre-training a foundation model

Applications:

1. Return output in a specific format and structure

2. Mimic a certain style in writing

3. Domain knowledge
   eg: Medical

4. Build a smaller model (1Bil parameters vs LLM-100B+ parameters) to run on mobile phone

PEFT: LORA and QLORA:

Parameter Efficient Fine Tuning (PEFT), and explore LoRA and QLoRA, Two of the most important PEFT methods. We will understnad how PEFT can be used to fine tune the model for domain specific tasks, at the lowest cost and minimal infrastrcuture.

Employing the QLoRa technique to fine-tune a model like Falcon **7B** can be achieved cost-effectively using Google Colab Pro, which costs $9.72 per month and you can cancel anytime. Alternatively, fine-tuning on a PC equipped with least 16 GB VRAM graphic card is another viable option. This setup enables efficient and budget-friendly fine-tuning of large language models with results comparable to traditional fine-tuning methods.

Pre-Training LLM

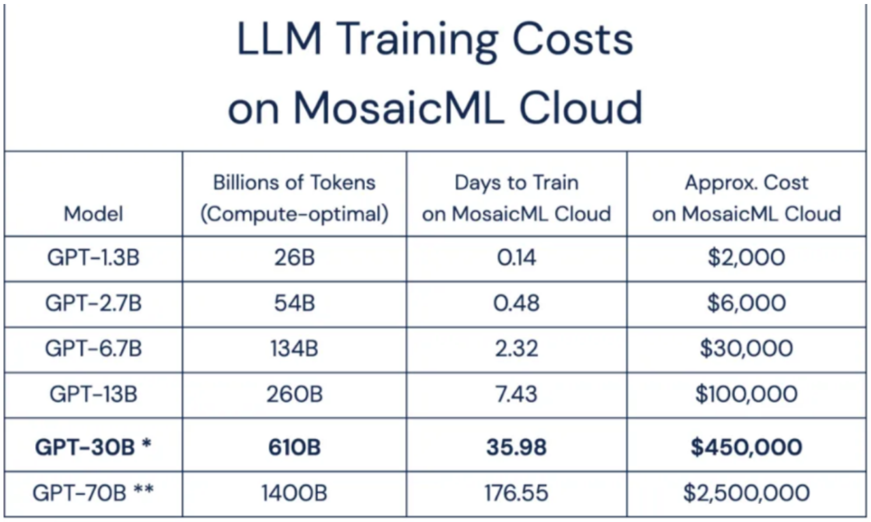
Pre-training is creating an LLM from scratch and is very expensive not to mention, the limited resources (machines / experts) while fine-tuning is so much cheaper like few hundred$.

In the most economical scenario, developing a specialised model based on Falcon 7B, fine-tuned on proprietary data can cost 9.99$ using Google Colab Pro. Deploying the model on-demand with just a single GPU machine, can keep expenses under control with 1.006 $/hr.

Estimate numbers:

For our analysis, let’s take the example of the Falcon 7B model, an average-sized LLM that demonstrates satisfying performance. Remarkably, this model can be accommodated on just one common Nvidia V100 GPU, making it accessible for various applications.

MosaicML (acquired by Databricks for 1.3B $) is a platform that enables you to easily train and deploy LLMs. According to pricing GPT-3 clone with 30B parameters can be trained at a significantly lower cost, approximately $450,000. Furthermore, smaller yet powerful 7B model can be trained for as little as $30,000 while still delivering comparable performance on specific tasks to their more expensive counterparts.



LLM Training Costs on MosaicML Cloud

Model	Billions of Tokens (Compute-optimal)	Days to Train on MosaicML Cloud	Approx. Cost on MosaicML Cloud
GPT-1.3B	26B	0.14	$2,000
GPT-2.7B	54B	0.48	$6,000
GPT-6.7B	134B	2.32	$30,000
GPT-13B	260B	7.43	$100,000
GPT-30B *	**610B**	**35.98**	**$450,000**
GPT-70B **	1400B	176.55	$2,500,000