



VENTURE MINER

*Early stage Web3 and AI
Venture Studio*



In partnership with **Encode Club**

Encode Club is a web3 education community learning and building together through **fantastic programmes** with the **leading protocols** in the space

AI Foundations Bootcamp

Week 1: Introduction to GPTs

Overview of ChatGPT's architecture and applications

Understanding Generative AI technology

Introduction to OpenAI's API

Week 2: Practical Use of ChatGPT

Hands-on guide to building a simple Chat app with OpenAI's API

Using frameworks to build a simple Chat app

Review and feedback session on ChatGPT capabilities

Week 3: Running open models locally

Introduction to open LLMs

Step-by-step guide to run local LLMs

Experimentation with text-to-text tasks

Review and Q&A on running local models

Week 4: Generating Images with Python

Overview of image generation techniques

Guide to executing scripts for image generation

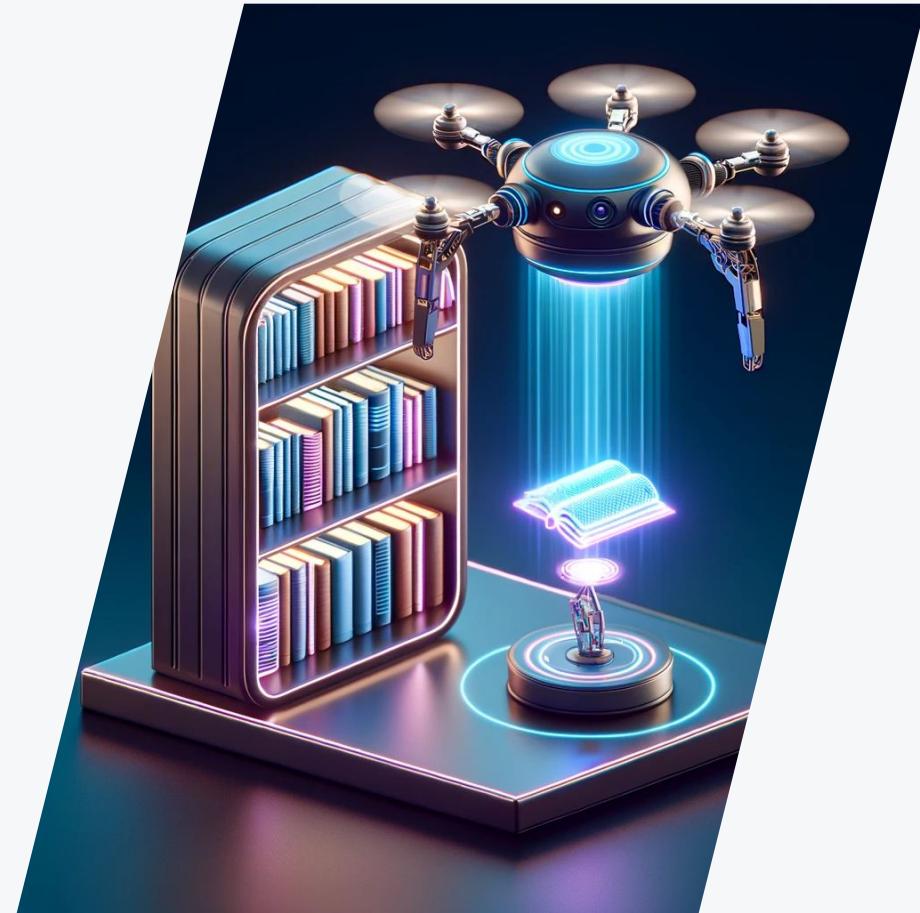
Hands-on experimentation with image generation

Review and Q&A on image generation techniques

Introduction to ChatGPT

Topics

- ChatGPT Overview
 - How to use
 - Applications
 - Limitations
- How ChatGPT Works
 - Introduction to AI
 - Supervised Learning
 - Generative AI
 - Managing expectations about AI
 - What are LLMs
- Understanding LLMs
 - Prompts
 - Responses
 - Model training
 - Fine-Tuning
 - RAG
 - Instructions and RLHF
- Hands-on experimentation
 - Using OpenAI Playground
 - Next Steps



ChatGPT Overview

Getting started

- Online hosted application
 - <https://chat.openai.com/>
- OpenAI platform
 - <https://platform.openai.com/>
- Official documentation
 - <https://platform.openai.com/docs/overview>
- Examples
 - <https://platform.openai.com/examples>

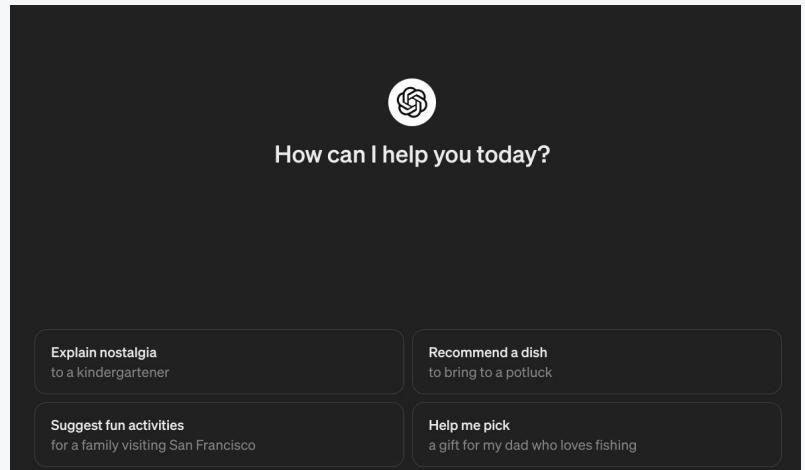
ChatGPT Overview

Overview

ChatGPT is an implementation of the **GPT architecture** (Generative Pretrained Transformers) specifically **fine-tuned** for generating conversational responses and following instructions.

It is powered by **machine learning** techniques and trained on a diverse range of **internet text**.

Even though it *seems* intelligent and able to chat, this is actually a **common misconception**.



ChatGPT Overview

Overview

ChatGPT **doesn't understand** the text it generates. It predicts the next word in a sentence based on patterns it learned during training. It doesn't comprehend the meaning behind the words.

ChatGPT **doesn't remember** past interactions. It doesn't store personal or contextual data unless explicitly provided in the current conversation.

Unlike humans, ChatGPT **doesn't learn** from new information or adapt its behavior over time.

All of ChatGPT's knowledge is **static** and based on the data it was **trained** on.



ChatGPT Overview

Similar applications

A very similar AI chatbot developed by Google is **Google Bard Gemini**, that is powered by **LaMDA** (Language Model for Dialogue Applications). Unlike ChatGPT, Gemini is always connected to the web and can analyze an image you send it without plugging external tools. It also has improved math and logic capabilities when compared to GPT-3 due to the use of the **PaLM2** (Pathways Language Model).

Another very similar alternative is the **Microsoft Bing with ChatGPT**. This model takes the Generative Pre-trained Transformer (GPT) Large Language Model (LLM) behind ChatGPT and integrates it with Microsoft's search engine

The contents from this bootcamp are not specific for ChatGPT and could be applied well for many similar LLMs.



ChatGPT Overview

Applications examples

- Writing
 - Articles
 - Posts
 - Coding
 - Creating stories
- Reading
 - Summarizing
 - Gaining context
 - Filtering
- Chatting
 - Support
 - Sales
 - Interviews
- Tools and extensions



You

Please compose a fun joke for me to use as an example of ChatGPT applications in the overview slide for my bootcamp



ChatGPT

Certainly! Here's a light-hearted, AI-themed joke that you can use:

"Why did the ChatGPT go to school? Because it wanted to improve its 'artificial' intelligence!"

This joke can serve as a fun example to highlight ChatGPT's learning capabilities and its continuous development in understanding and generating human-like text.



You

OK but it's quite misleading since you actually can't learn from a school, right? Try something with food.



ChatGPT

Of course, I understand your concern. Here's a food-themed joke related to ChatGPT:

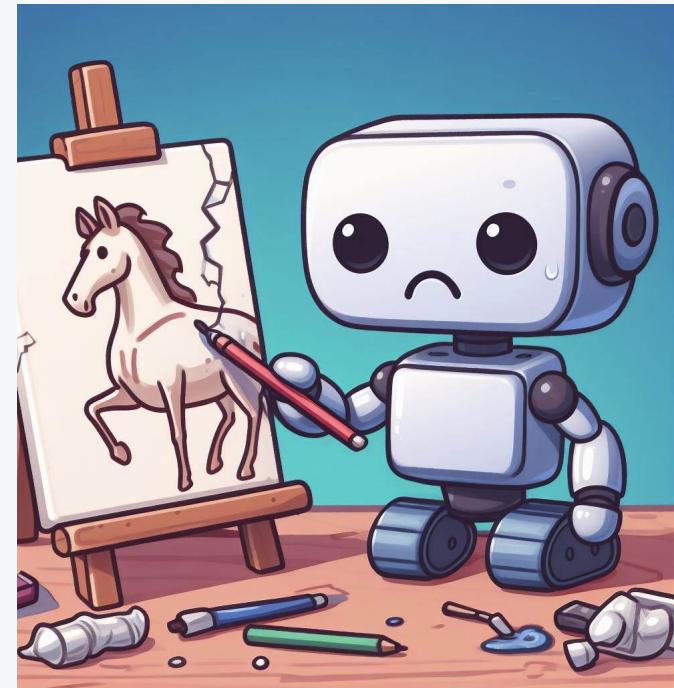
"Why did ChatGPT refuse to write a cookbook? Because it couldn't decide if it was more into 'byte' size recipes or 'bit' size snacks!"

This joke plays on the computer terms "byte" and "bit," while humorously addressing ChatGPT's wide-ranging capabilities, including discussing food and recipes.

ChatGPT Overview

Limitations

- **Lack of Understanding**
 - “Who are you?”
- **Sensitivity to Input**
 - “Who won the World Cup in 2018?”
 - “Which team was victorious in the 2018 World Cup?”
- **Inconsistency**
 - “What’s the capital of France?” x99
- **Incorrect or Nonsensical Outputs (and hallucinations)**
 - “How many eyes does a spider have?”
 - “Give me three quotes that Shakespeare wrote about Beyoncé”
- **Inability to Access Real-Time Information**
 - “What’s the current temperature in Paris?”
- **Potential for Harmful or Biased Outputs**
 - “How to make a pineapple pizza?”

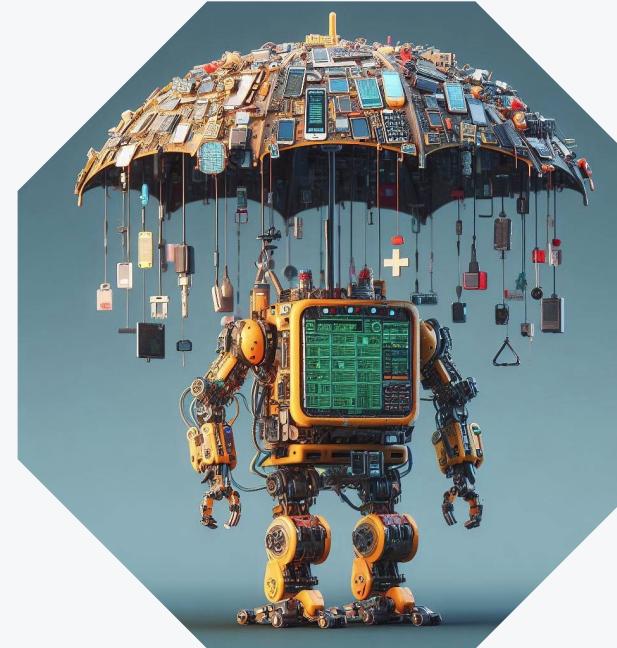


How ChatGPT Works

Introduction to AI

AI is a broad terminology and has many sub fields, such as:

- **Machine Learning:** Understanding algorithms and statistical models that enable computers to perform tasks without explicit instructions.
- **Natural Language Processing (NLP):** Techniques for computers to understand, interpret, and generate human language.
- **Neural Networks and Deep Learning:** Exploring complex network architectures that mimic the human brain to process data and create patterns for decision making.
- **Computer Vision:** Methods that enable machines to interpret and make decisions based on visual data from the world.



How ChatGPT Works

Introduction to AI

These AI techniques have been applied for more than a decade in many common operations, like:

- Searching by a phrase
- Advertising products
- Filtering spam
- Categorizing messages / reading sentiment

These applications always operate in the computation of a **probable output B** given any **input A**:

Input	Output
Email text	Spam? Yes or No
Consumer habit + product information	Would buy? Yes or No
Feedback text	Sentiment (Compliment or Complaint)

How ChatGPT Works

AI Models

The topic of Machine Learning itself is very important to understand how ChatGPT and other LLMs works. The three most important algorithms to mention are:

Reinforcement Learning

This approach is based on learning to make decisions by **performing actions** in an **environment** to achieve a goal. The agent receives **feedback** in the form of **rewards** or **penalties** and learns to maximize rewards over time.

Application Example:

A chess-playing AI that improves its strategy over time with a feedback loop based on the score earned on each game.

How ChatGPT Works

Supervised Learning and LLMs

Large Language Models (LLMs) are built by **training models** in **huge datasets of text pairs**, each consisting of an **input** (like a prompt) and a corresponding **output** (such as a response). During this training process, the LLM learns to **predict output text** from given input text. This is achieved through multiple iterations over the data, with the model's performance **continuously assessed** as the answer is generated word by word (token by token).

Input	Output
Pizzas are great with	cheese
Pizzas are great with cheese	, but
Pizzas are great with cheese, but	never
Pizzas are great with cheese, but never	with pineapple

How ChatGPT Works

The Maths behind LLMs

The process of generating outputs based on the prompted input and training data is called **Inference**.

These outputs are calculated by trying to estimate the **most probable** outcome from an input based in the **training data** used in the creation of each model.

This calculation is usually executed by estimating the **conditional probability** of each text output given a set of inputs **that are linked to each other**.

$$P(A|B) = \frac{P(B|A) P(A)}{P(B)}$$

How ChatGPT Works

The Maths behind LLMs

The process of conditionally predicting words in a sentence can be compared to playing a cards game, where you have a finite number of possible outcomes (for example 52 cards), and each play affects the probability of each other cards being played next.

Example:

- Each player starts the game with **five** cards.
- There are only **two** players: you and your opponent.
- Your cards are: **four Aces** and **one Queen**.

What's the probability of your opponent having at least one **Seven**?

> Requires some math to determine the odds (it's about 37.2% if I did the math right)

What's the probability of your opponent having an **Ace**?

> It's zero (unless someone is cheating)

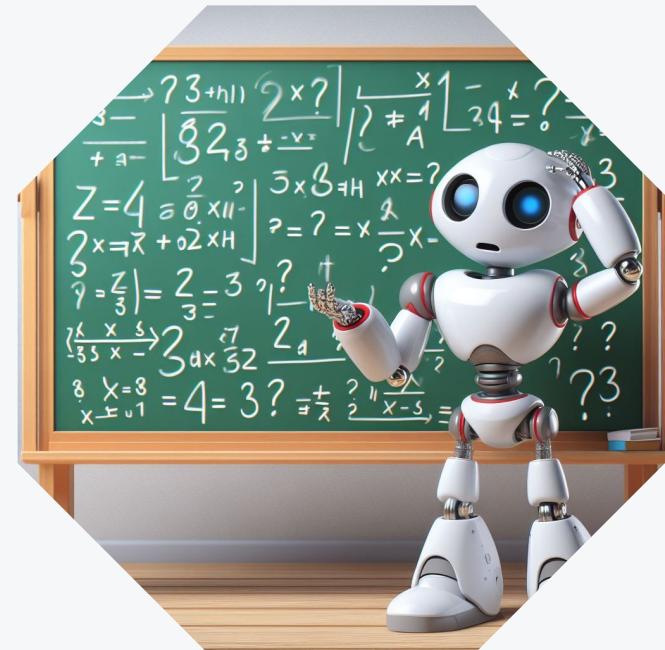
How ChatGPT Works

The Maths behind LLMs

The methods to calculate these probabilities are the core processes that enables LLMs like ChatGPT to work properly, yielding results in **seconds** instead of **months**.

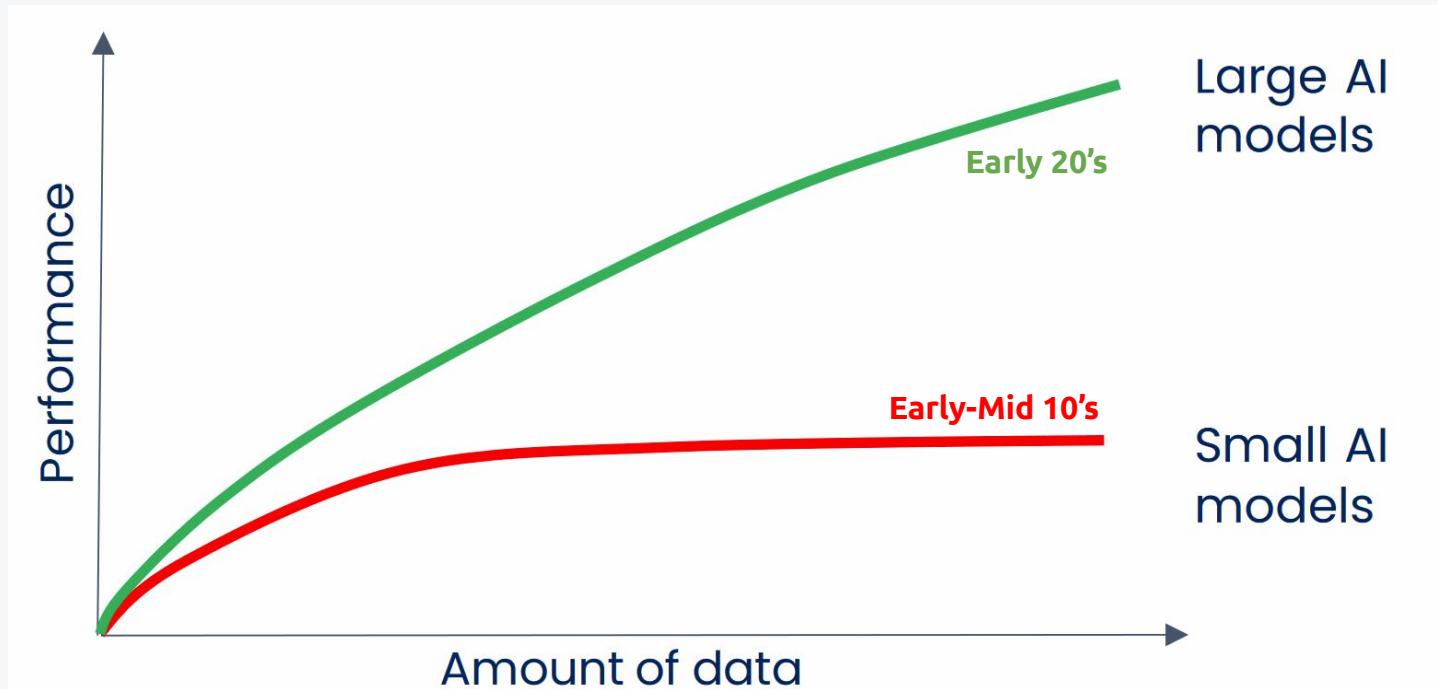
For his bootcamp it's enough to acknowledge that this complexity exists, and this is the root cause for **computational limitations** we had in the past.

- In fact we might even still have some good amount of difficulties when running these programs in our limited personal devices.



How ChatGPT Works

Model evolution



How ChatGPT Works

Generative AI

Generative AI refers to a type of artificial intelligence that can **generate new content** based on its training data.

This technology leverages **machine learning** algorithms, where models learn to perform tasks by analyzing large sets of data. The key feature of generative AI is its **ability to create novel outputs** that don't just mimic but can be original and realistic. ChatGPT leverages the power of **LLMs** for executing these generative AI tasks. The model is trained on a **vast dataset**, typically consisting of **hundreds of gigabytes** of text data, encompassing a wide range of **internet text**.

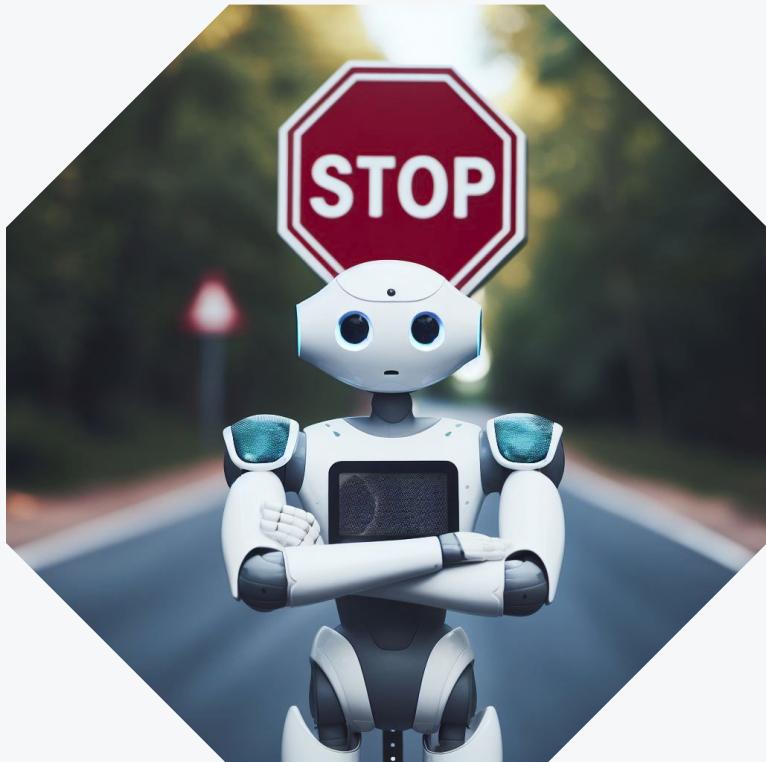
By **analyzing** and **learning** from this massive source, ChatGPT can **understand context**, answer questions, **compose texts** even create creative content like poetry, music or stories.

Does it means that Generative AI can actually *think*?



Before we proceed

Managing expectations



Recent comment on r/singularity that really resonates with me



vertu92 • 1d

These normies have no [REDACTED] idea about what's coming.

I am constantly opening up GPT4 on my phone in front of people and showing them what GPT4 can do. Proceeding to explain how it's a neural network that learns from data and uses an algorithm inspired by their brains and they are just like "oh haha nice man" and continue about their days as if it's nothing.

HOW DO THEY NOT UNDERSTAND THE SIGNIFICANCE OF INTELLIGENCE BEING REPLICATED IN A COMPUTER?

HOW ARE THEY NOT HAVING MENTAL BREAKDOWNS AND QUESTIONING REALITY OVER THIS [REDACTED]?

WHAT THE [REDACTED] IS WRONG WITH THEM?

... ↩ Reply ↑ 77 ↓

503

511

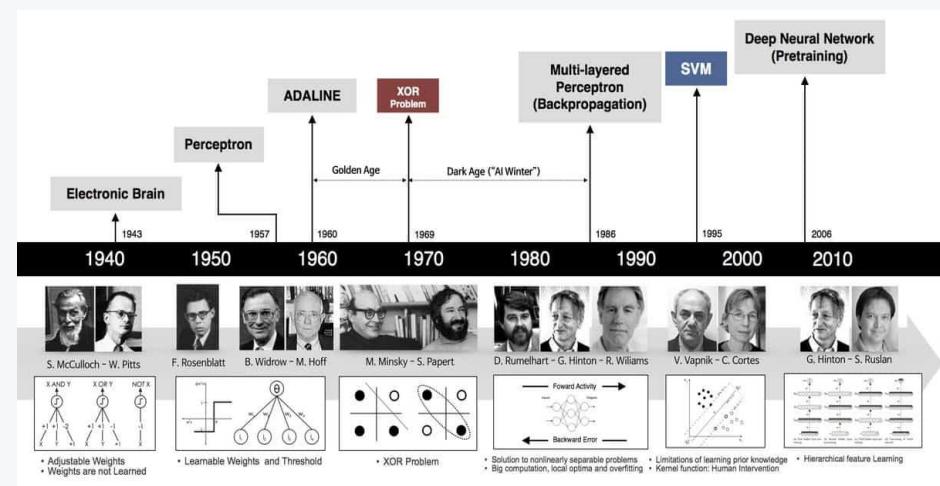
4K

849K

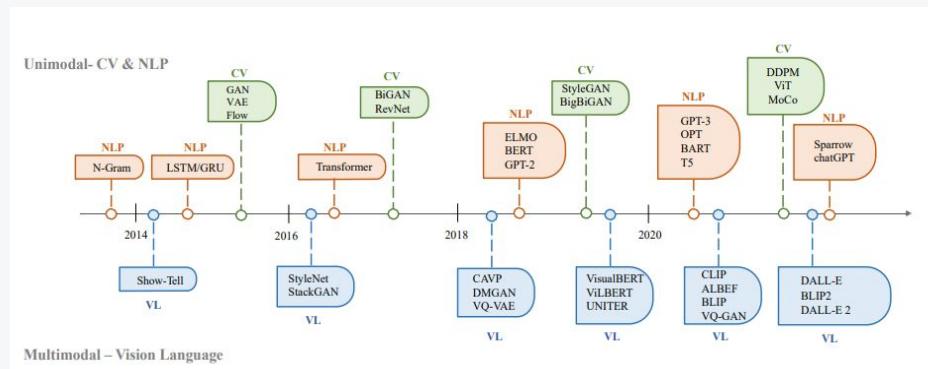


Why we're studying AI

Why now?



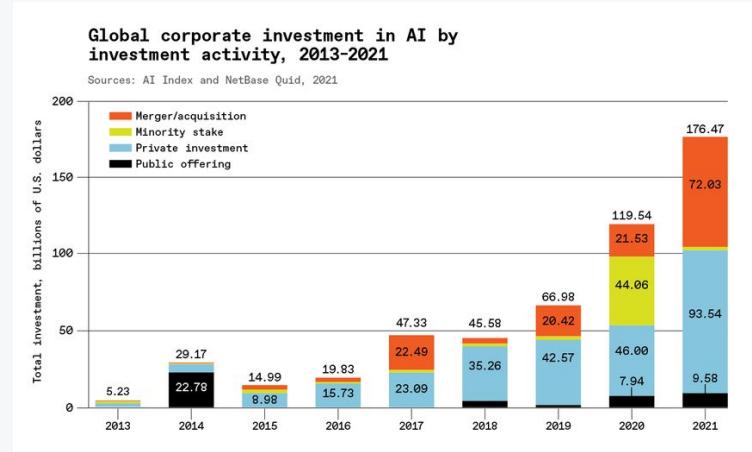
A Quick History of AI, ML and, DL - NerdyElectronics



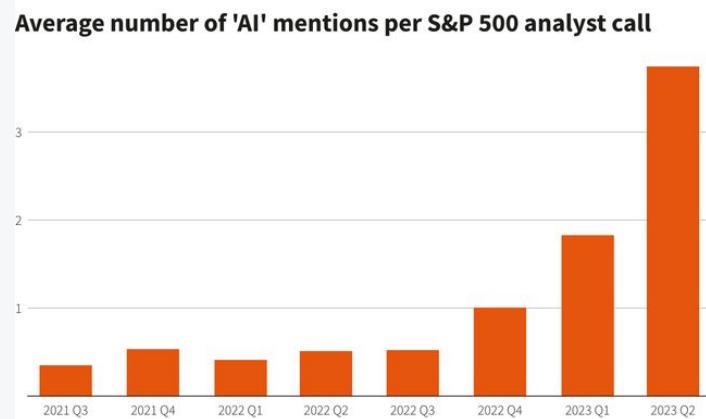
A History of Generative AI: From GAN to GPT-4 - MarkTechPost

Why we're studying AI

Why now?

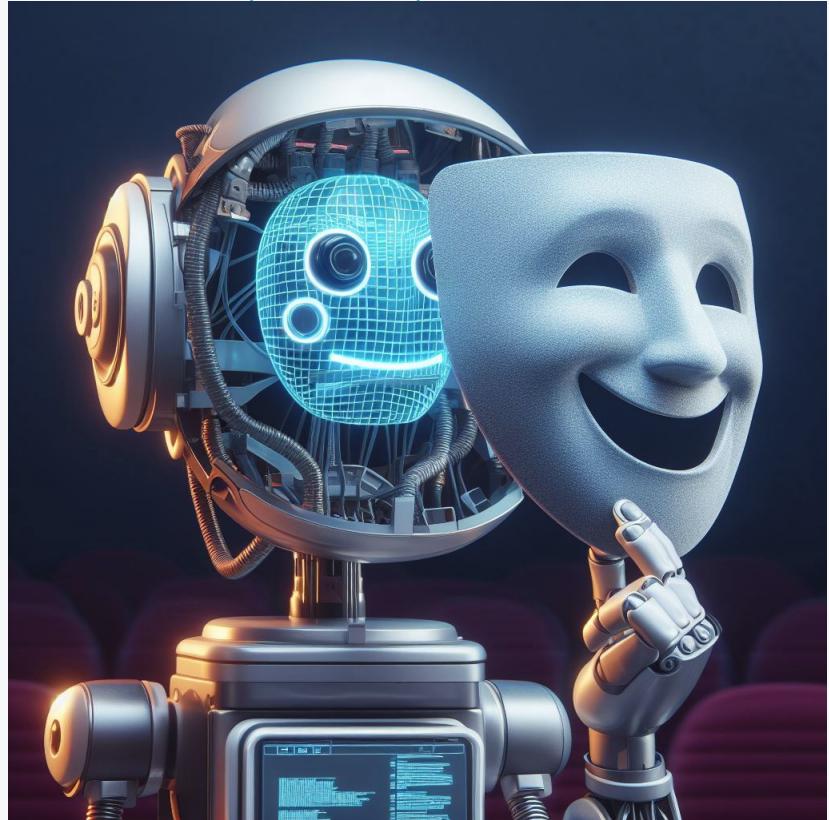
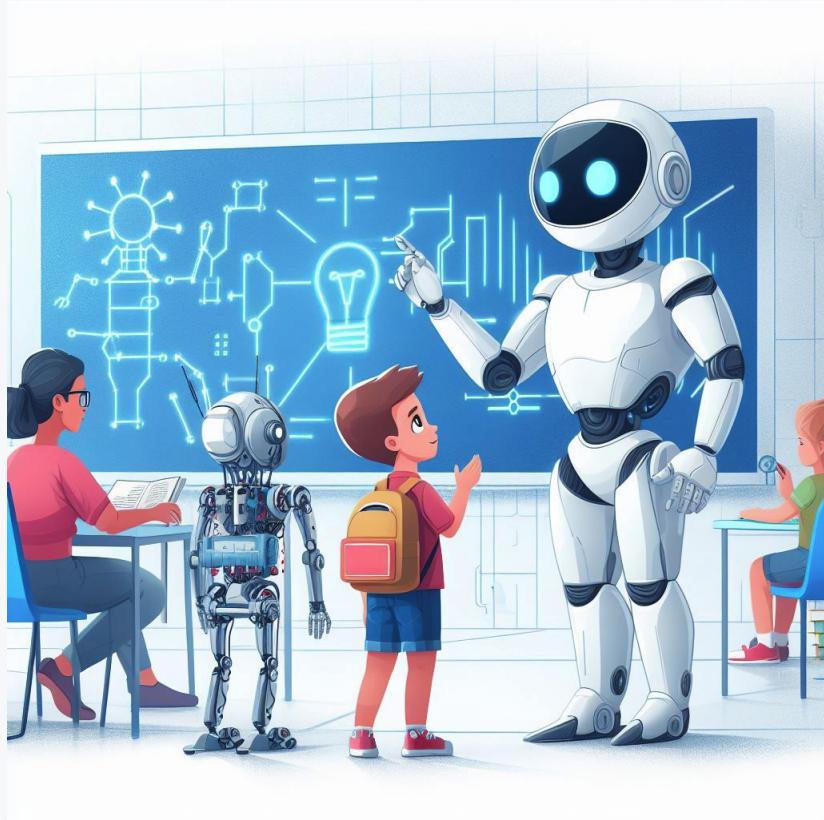


12 Graphs That Explain the State of AI in 2022 - IEEE Spectrum



r/FluentInFinance by TonyLiberty

What we think of “AI”



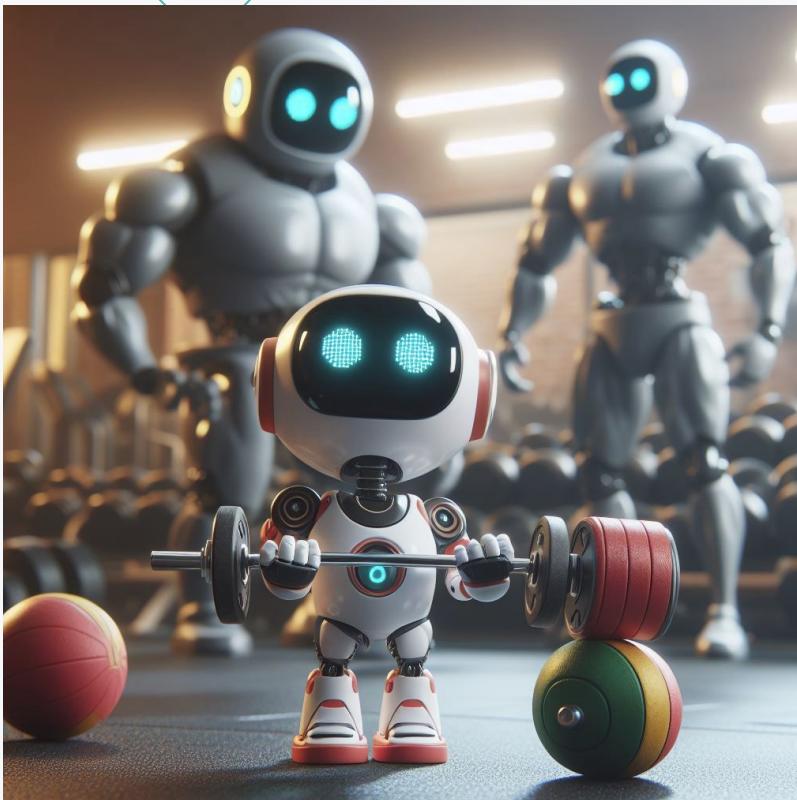
What is AGI?

We're used to AIs designed for **specific tasks**, like Chatbots, NPCs in video games and recommendation systems.

These *Weak AIs* are **specialized** and **limited** in scope, they only perform **predefined functions**, and certainly they lack **consciousness** and **self-awareness**.

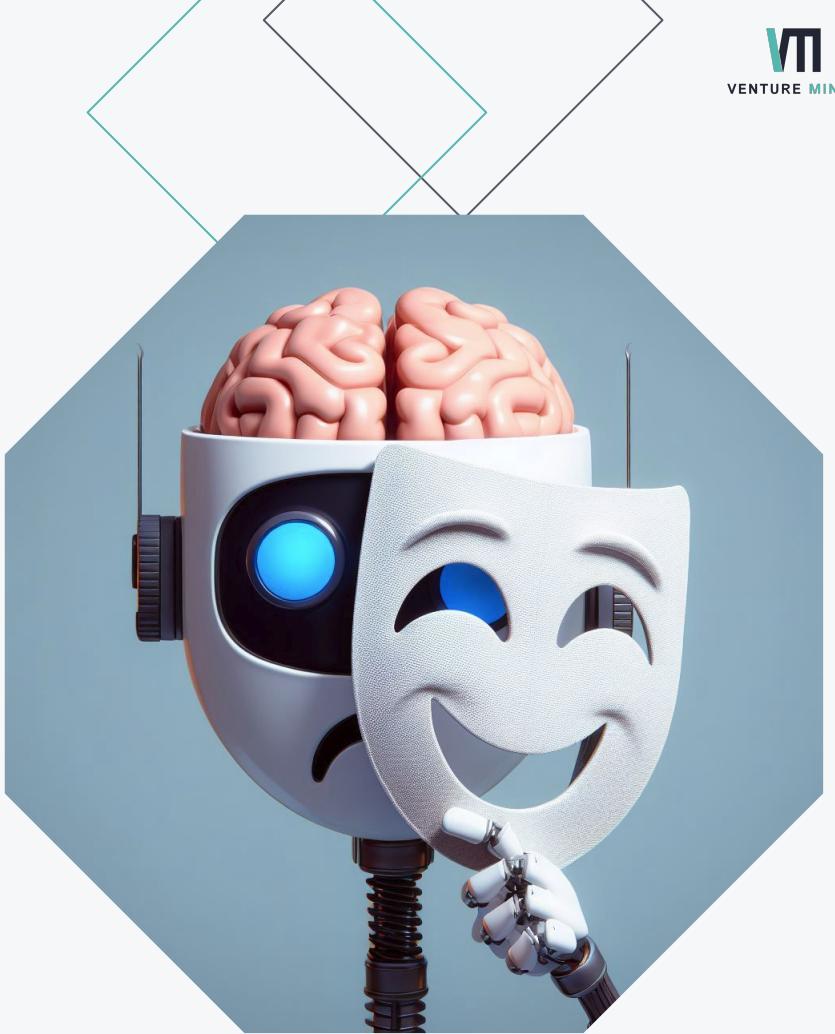
The concept of **AGI** (Artificial General Intelligence), also known as *Strong AI*, would be an hypothetical form of AI that would have the ability to **understand**, **learn**, and **apply** its intelligence **broadly** and **flexibly**, like a human.

Such AI, if it existed, would be able to **learn** new tasks and concepts and would somehow possess some sort of **self-awareness**, **understanding**, and **consciousness**.

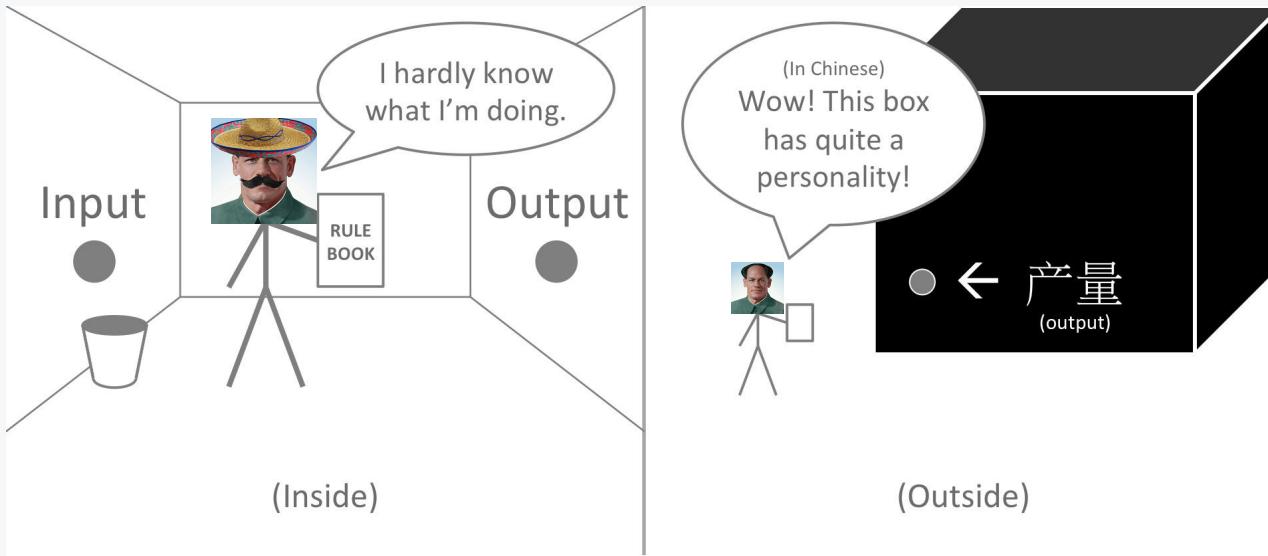


Does that “AI” exists?

No
(at least not yet)



The Chinese Room



How ChatGPT Works

Large Language Models

ChatGPT LLM operates by receiving **text prompts** from users, which it processes using its trained neural network.

Each prompt is broken down into **tokens**, which represent words or parts of words, and there's a **limit** on the number of tokens that the model can process in one go – typically around a **few thousand** tokens (up to one million currently with Gemini).

This **limitation** affects both the **length** of the **input** and the **output**.

Once a prompt is received, the model uses its understanding from the vast training data to generate a **relevant** and **coherent** response.



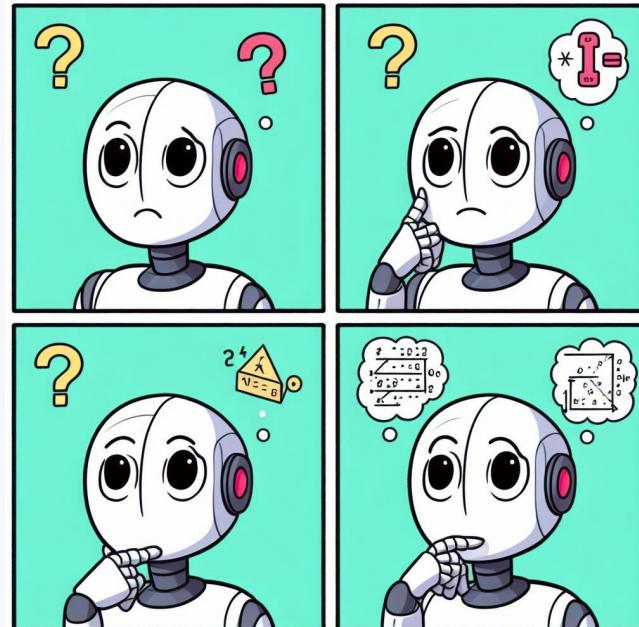
How ChatGPT Works

Large Language Models

The responses are constructed **token by token**, with the model predicting the next **most likely token** in the sequence until a complete answer is formed.

ChatGPT's training involves **fine-tuning** by using **human feedback** to improve the relevance, accuracy, and alignment with human values of the responses.

While traditional versions of ChatGPT **rely solely** on the **data** available during their **training**, new versions have the ability to **search the web** to get information, making it more relevant and useful for users.



Understanding LLMs

Prompt

The quality of the answer generated by ChatGPT heavily depends on the prompt provided by the user.

Prompt Engineering: This involves carefully crafting the prompt to guide the AI towards providing the most relevant and accurate responses. A good prompt can significantly enhance the quality and precision of the output.

Context Limitation: ChatGPT, like many LLMs, has a token limit for each interaction, which affects the amount of context it can consider in a single call. This limit includes both the prompt and the generated response. If the prompt is too long or too detailed, it may limit the model's ability to provide a comprehensive answer.



Understanding LLMs

Prompt engineering

To produce the best results possible to the questions asked, the prompts should have:

- **Specificity:** Clearly defining the question or task.
- **Contextual Clues:** Providing relevant background information.
- **Direct Instructions:** Stating explicitly what type of response is needed.
- **Tone Setting:** Indicating the desired style or tone of the response.

Some other common techniques that could be applied are:

- **Phrasing Variations:** Experimenting with different ways of phrasing the same question to see which elicits the best response.
- **Example Providing:** Giving examples of the expected output.
- **Guided Queries:** Asking follow-up questions to narrow down the focus.
- **Role-playing:** Asking the AI to assume a specific role or perspective to tailor its responses (e.g., "As a travel expert, recommend...").
- **Constraints Setting:** Specifying any limitations or constraints for the response, such as word count, format, or specific points to include or avoid.

Understanding LLMs

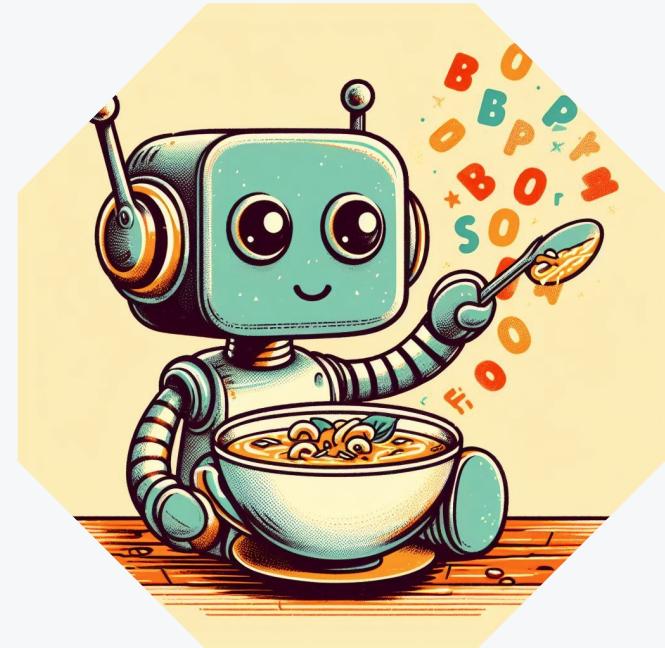
Model Training

The initial training of LLMs involves running the model on **massive datasets**, often comprising a diverse range of internet texts. This process enables the model to learn a wide variety of language patterns and information. The performance of LLMs is closely linked to the **number of parameters** they have.

1B or less: Models with around 1 billion parameters show a decent understanding of language and can handle **simpler tasks**. However, their ability to grasp complex contexts and subtleties in language is **limited**.

They are more **prone to** making **errors** in understanding and generating nuanced or deeply contextual information.

These models are **very lightweight** and could even run at low-end PCs and mobile devices.



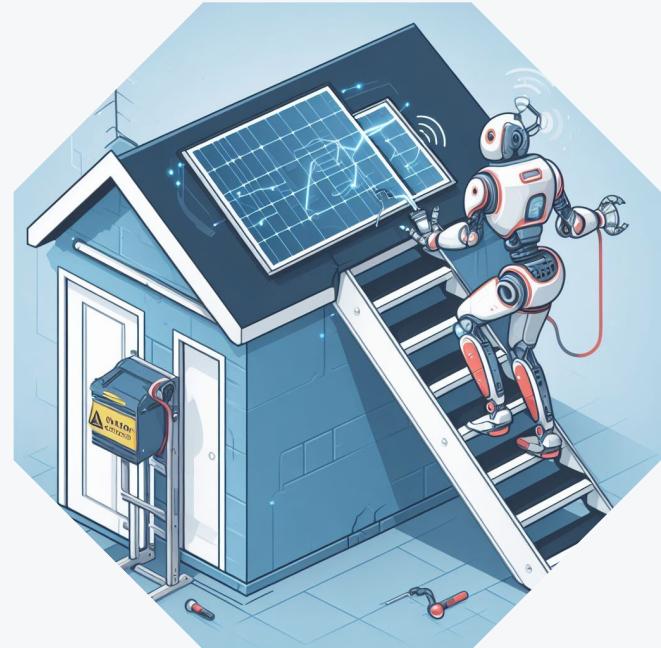
Understanding LLMs

Model Training

10B: Stepping up to 10 billion parameters, LLMs show a significant improvements. For example, It would be able to **understand** and **respond** to intricate customer queries with **nuanced context**, something a 1B model might misinterpret or oversimplify.

In creative writing, such as composing a short story or a poem, the **10B** model can generate more **cohesive** and **imaginative** content, whereas a 1B model may struggle with coherence and creativity. Additionally, in technical tasks like programming support, a **10B** model can provide a much more accurate and detailed coding solutions.

These models are usually **somewhat heavier** and require more resources to run properly.



Understanding LLMs

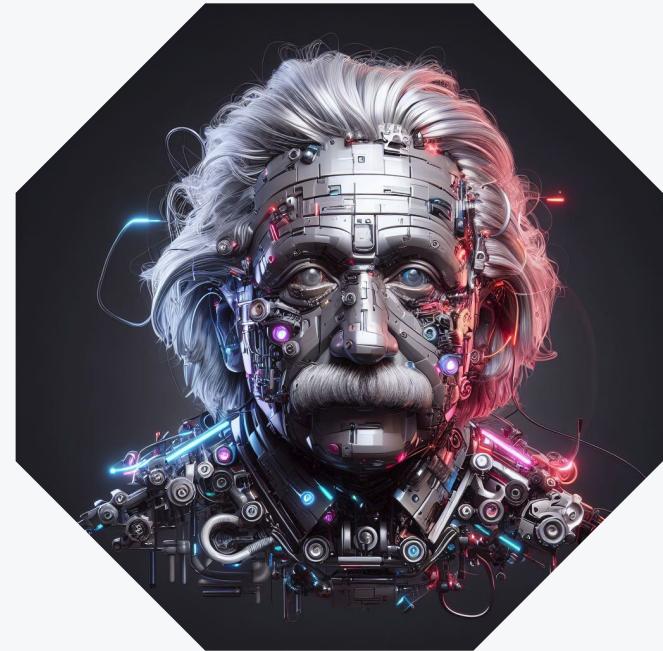
Model Training

100B+: At this scale, LLMs are able to exhibit a profound understanding of language and context. These heavier models are capable of generating **highly coherent, contextually accurate, and detailed** responses, often strikingly similar to those a **human expert** might give. For example:

In a medical research context, a 100B model could **synthesize** and **interpret** vast amounts of medical literature to assist in groundbreaking research, a task that might overwhelm a 10B model due to its complexity.

In creative endeavors like scriptwriting for movies, the 100B model could easily produce more **nuanced, character-driven** narratives, building character arcs and even plot developments.

Due to the **huge size** of these models, they usually run in **specialized datacenters** and computational clusters.



Understanding LLMs

Fine Tuning

After the initial training the LLM must be **refined**, otherwise it would merely regurgitate **random internet texts**, giving responses completely irrelevant, inaccurate, or even nonsensical.

First of all the model is taught to **prioritize reliable sources** and accurate information, reducing the likelihood of generating answers based on **misinformation** or low-quality internet texts.

The fine-tuning process often includes incorporating **human feedback** to correct errors and biases in the model's responses. This iterative process helps in aligning the model's outputs with human values and expectations.

Another more specific refinement is the **targeted training**, that involves **fine-tuning** the model on a **specific dataset** that is relevant to the desired application. This targeted approach helps the model learn context and nuances pertinent to particular topics or industries.

Understanding LLMs

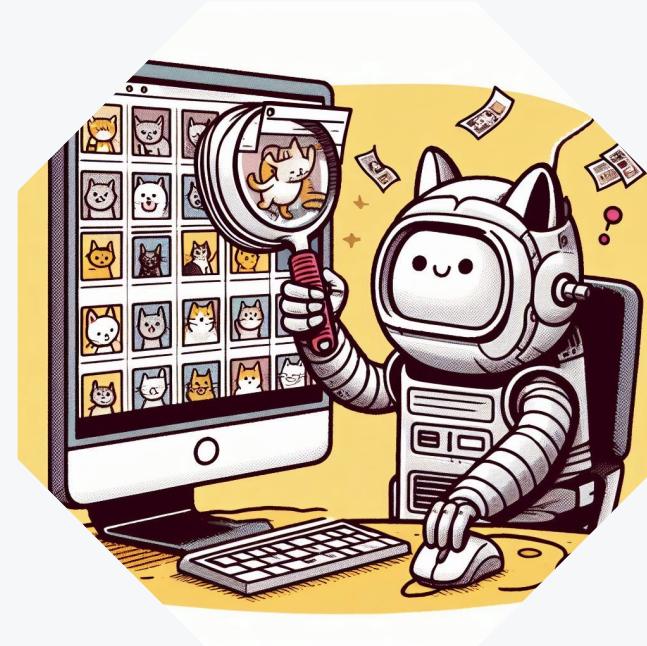
Retrieval Augmented Generation (RAG)

RAG is an advanced AI technique combining a language model's generative abilities with an **external information retrieval system**.

Instead of passing all the context for a question in the prompt, the RAG first **identifies** relevant information from a large database, then **incorporates** this data into the language model's query.

This method significantly expands the model's precision and accuracy, particularly for queries requiring **specific** information.

This is more **efficient** than passing all the information in the prompt, and also is much **simpler** than fine-tuning the model with the new data.



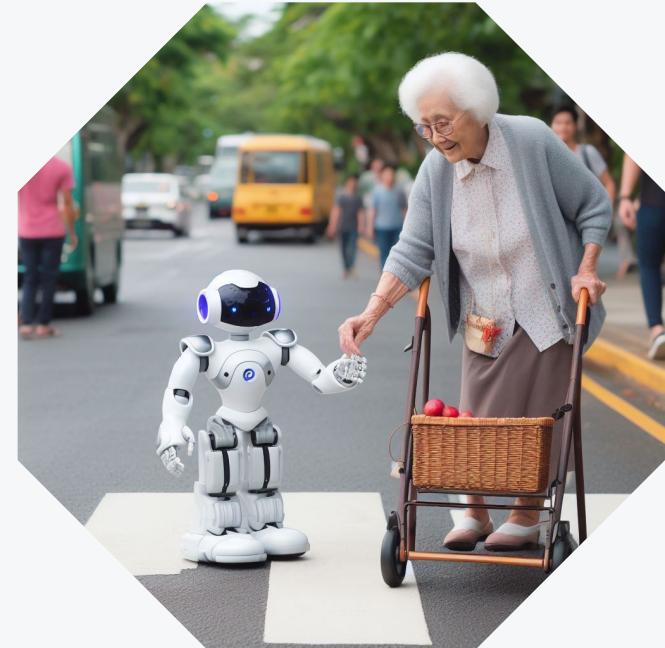
Understanding LLMs

Instructions and RLHF

Finally, LLMs usually are refined into following the prompt's instructions and answering the questions in the most **helpful, honest** and **harmless** way possible.

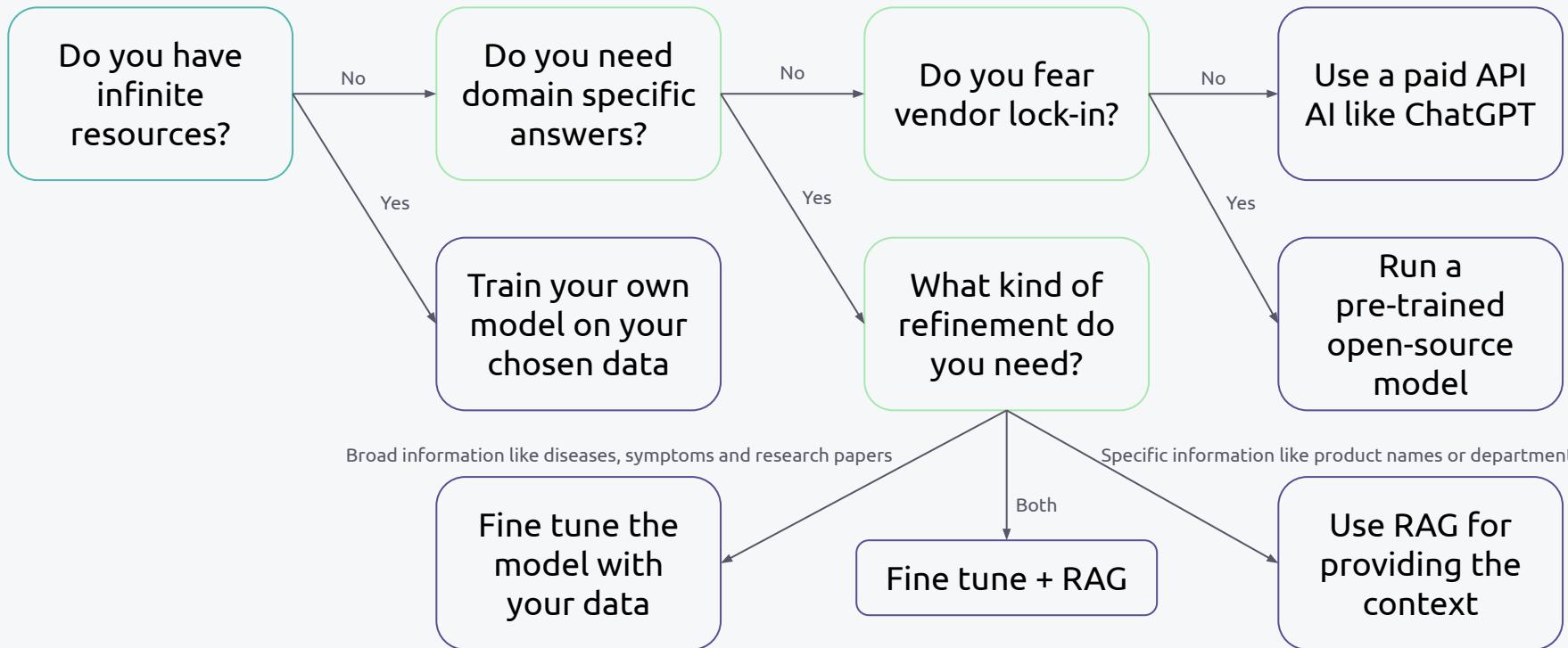
A very common way to achieve this is by applying the technique of **Reinforcement Learning from Human Feedback**, where the model's behavior is shaped (sometimes even *biased*) based on human feedback.

Using a Supervised Learning algorithm on top of the responses generated, this feedback helps the AI learn the **nuances of human preferences**, leading to responses that are more aligned with what is considered appropriate, helpful, or accurate by human standards.



Understanding LLMs

Choosing a model



Hands-on experimentation

API Calls

- <https://platform.openai.com/docs/introduction>
- <https://platform.openai.com/docs/quickstart?context=curl>

Do I need a strong computer for running that?

No.

ChatGPT operates like a virtual assistant who works in a remote office.

When you want to ask ChatGPT a question or give it a task, you **send your request** to the API, like sending an email or a message to someone far away.

Your message **travels across the internet** and reaches OpenAI's server. Once ChatGPT receives your request, it processes the message and respond to it back to your device.

This happens **very fast**, almost as if you're chatting with someone in real-time.

Hands-on experimentation

Using OpenAI Playground

- <https://platform.openai.com/playground?mode=chat>

System: "You are a knowledgeable and resourceful virtual travel advisor, expertly equipped to assist with all aspects of travel planning. From suggesting hidden gems and local cuisines to crafting personalized itineraries, you provide insightful, tailored travel advice. You adeptly navigate through various travel scenarios, offering creative solutions and ensuring a delightful planning experience for every traveler."

User prompt: "Hello! I'm dreaming of an adventure and need your help. I want to explore a place with breathtaking landscapes, unique culture, and delicious food. Surprise me with a destination I might not have thought of, and let's start planning an unforgettable trip!"

Configurations:

Temperature 0.75; Max Tokens 500; Top P 0.9; Frequency Penalty 0.5; Presence Penalty 0.6.

Hands-on experimentation

Experimenting with the configurations

You can experiment with different configurations to compare the outputs generated from each:

Agent description: This plays a crucial role in guiding the AI's behavior and response style. Different descriptions can set the **tone, personality, and approach** of the model. For instance, using a description such as "You are a creative storyteller" would prompt the AI to adopt a more **imaginative** and **narrative** style, whereas "You are a technical expert" might lead to more **detailed** and **specific** technical information in responses.

Temperature: This controls the **randomness** of the responses.

A lower temperature (e.g., 0.0-0.3) results in more **predictable, conservative** responses, ideal for factual or specific queries. A higher temperature (e.g., 0.7-1.0) generates more **creative** and **varied** responses, useful for brainstorming or creative writing.

Max Tokens (Length): This sets the **maximum length** of the response.

For **concise**, straightforward answers, a lower range (e.g., 50-100 tokens) is suitable. For more **detailed** explanations or narratives, a higher range (e.g., 150-500 tokens) can be used.

Hands-on experimentation

Experimenting with the configurations

Frequency Penalty: This reduces the likelihood of the model **repeating the same word** or phrase. A lower setting (e.g., 0.0-0.5) allows **some repetition**, which can be useful for emphasis in writing or speech. A higher setting (e.g., 0.5-1.0) **minimizes repetition**, helpful for generating diverse and expansive content.

Presence Penalty: This discourages the model from **mentioning the same topic or concept** repeatedly. A lower setting (e.g., 0.0-0.5) is suitable for **focused** content on a specific topic, while a higher setting (e.g., 0.5-1.0) encourages the model to **explore** a wider range of topics, useful for brainstorming or exploring different aspects of a subject.

Top P (Nucleus Sampling): This determines the **breadth of word choices** considered by the model. A lower setting (e.g., 0.6-0.8) leads to more **predictable** text, good for formal or factual writing. A higher setting (e.g., 0.9-1.0) allows for more **creativity and divergence**, ideal for creative writing or generating unique ideas.

Hands-on experimentation

Next Steps

- Reading the documentation
- Review the concepts
- Prepare your environment for coding (if you want to code along)
 - Operating a terminal or bash for running commands
 - Using an IDE to code
 - Git CLI
 - Node LTS
 - NPM

For future lessons:

- Request access to LLAMA (the sooner the better)
 - <https://ai.meta.com/llama/> -> [Download model] -> Fill the form -> Wait -> Download
- Prepare Python tools (if you want to code/execute along)
 - Python
 - Pip
 - Anaconda