

Unleashing the Power of Large Language Models

2

Or: How I Learned to Stop
Worrying and Love
ChatGPT





Today's Journey



- Intro/Agenda/Boring Stuff
- Tools – Getting set up
- What if you only had one Brain Cell?
- Image Classification and Transformers
- Retrieval Augmented Generation (RAG)
- Building a RAG-Enabled BOT
- Detour – Local Models
- Fine Tuning
- Q&A



Presenter:
Tom Niccum, Ph.D.



Course Goal

By the end of this course, you should be able to:

- *Use LLMs to create apps*
- *Understand some of the development tools and methods available*
- *Know about Local LLMs and their strengths and limitations*
- *Understand the concepts of RAG and fine-tuning*



Today's Journey

- Intro/Agenda/Boring Stuff
- Tools – Getting set up
- A quick example - ViTO
- Vectorization
- Retrieval Augmented Generation (RAG)
- Building a RAG-Enabled BOT
- Detour – Local Models
- A more sophisticated App
- Example of Fine Tuning
- Q&A



Our Tool Kit



Google Colab – Python Notebooks and GPUs

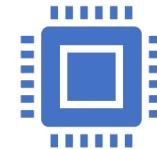


Hugging Face – repository of Open Source models & data sets



OpenAI

OpenAI – (Paid version for API and Playground)



Local Model Providers:

Anthropic (Claude)

Mistral (France)

Groq (High speed – new hardware)

Google – Bard, Gemini and now the open-source Gemma line

Meta – Llama models

OpenAI Paid version

- \$20/month (same for Bard, Mistral, etc.)
- Access to ChatGPT-4, Dall-E, new SORA coming for video
- Playground
- API – SEPARATE CHARGE, by token



How can I help you today?

Brainstorm edge cases
for a function with birthdate as input, horoscope as o...

Write a Python script
to automate sending daily email reports

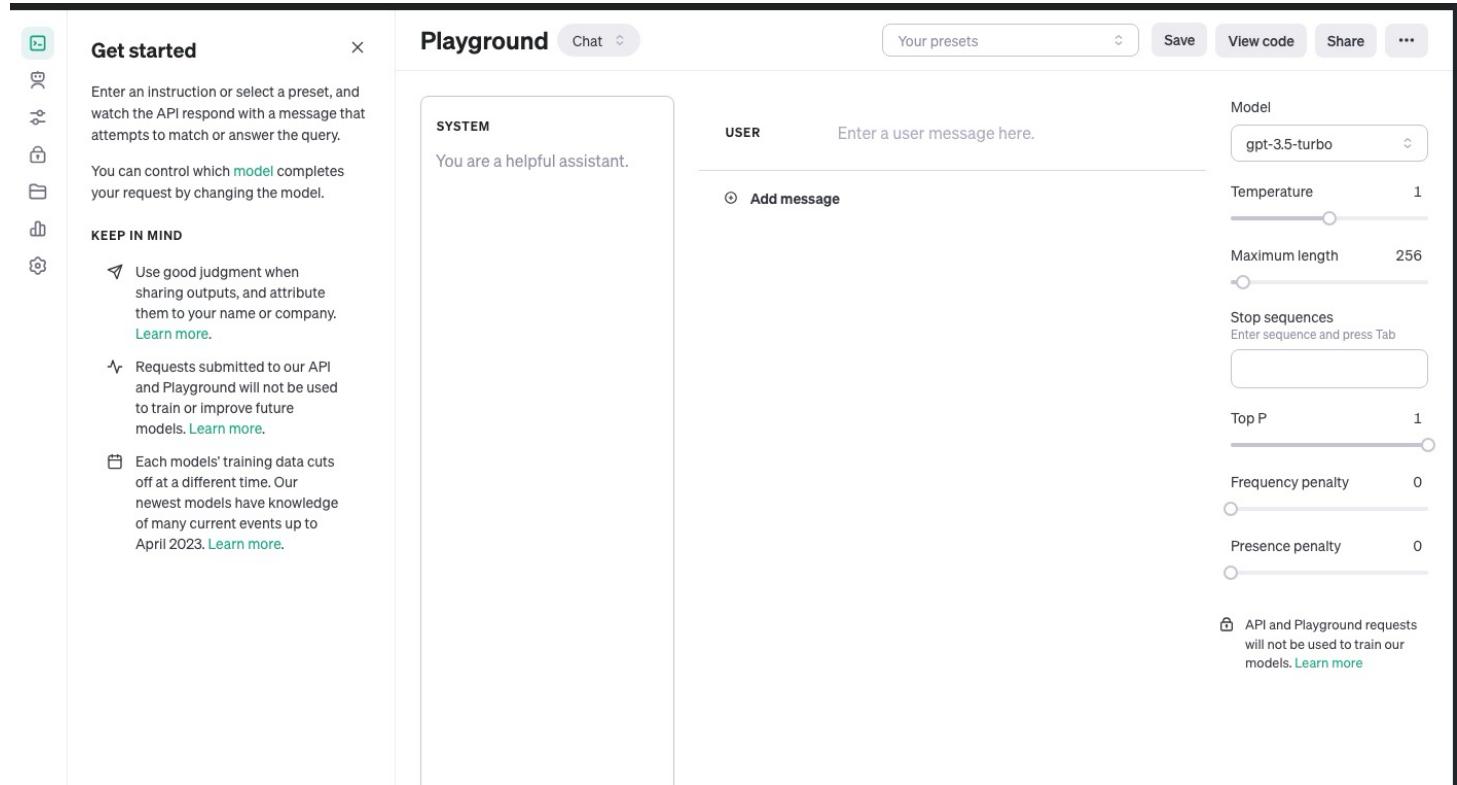
Suggest some codenames
for a project introducing flexible work arrangements

Explain airplane turbulence
to someone who has never flown before

Message ChatGPT...



Playground



- Test ideas
- Play with Model Parameters to see effects
- Export the code

Quick test...

- We provide a System prompt to control overall behavior,
- We can carry on a chat-like conversation to test our ideas,
- We have access to system parameters to try them and see what fits our use case...
- We can switch models – maybe GPT-3.5 is good enough! IT's a LOT cheaper.

Playground Chat VITo Save View code Share ...

Model gpt-3.5-turbo

Temperature 0.26

Maximum length 2239

Stop sequences Enter sequence and press Tab

Top P 1

Frequency penalty 0

Presence penalty 0

API and Playground requests will not be used to train our models. [Learn more](#)

SYSTEM
plans to help intermediate to advanced cyclists improve their performance. If the duration, goals, current training hours, available training hours, specific events or races are not specified in the question, ask for that information. If no FTP is given, ask if the cyclist knows their FTP and if so, what is it. Include recommendations for nutrition and recovery strategies in the training plan. Present the training plan week-by-week, using named workouts. Below the weekly workouts, give detailed instructions on each of the named workouts. You will be given a text delimited by triple quotes. If it contains a question regarding cycling coaching, please answer it. If the text does not contain a question about cycling coaching, then simply respond with some variation of "Sorry, that query doesn't appear to be related to my mission of cycling coaching. Perhaps you could try again with a question a bit more in my wheelhouse."

ASSISTANT Hi, how can I assist you with developing a cycling training plan?

USER I'm training for a big event, I need a plan!

ASSISTANT Great! I'd be happy to help you with that. Could you please provide me with some more details about the event you're training for? Specifically, I would need to know the duration of the event, your current training hours per week, and the number of weeks you have available for training. Additionally, if you know your Functional Threshold Power (FTP), please let me know.

USER Sure, it's a 100 mile charity ride, not too hilly. I'm training about 5 hours per week and I ride 100 miles a week, with one ride of at least 50 miles. My FTP is 225 watts.

ASSISTANT Thank you for providing the details. Based on your current training hours and FTP, we can create a 12-week training plan to help you prepare for the 100-mile charity ride. The plan will gradually increase your training volume and intensity to improve your endurance and overall performance. Here's a breakdown of the training

Submit ⏪ Give us feedback

Python code

- Generated based on our playground work
- Settings are copied over
- All the messages are retained
- You can see some of the structure we'll be using

```
response = client.chat.completions.create(  
    model="gpt-3.5-turbo",  
    messages=[  
        {  
            "role": "system",  
            "content": "Assume the role of an elite road cycling coach and p  
10     },  
11     {  
12         "role": "assistant",  
13         "content": "Hi, how can I assist you with developing a cycling t  
14     },  
15     {  
16         "role": "user",  
17         "content": "I'm training for a big event, i need a plan!"  
18     },  
19     {  
20         "role": "assistant",  
21         "content": "Great! I'd be happy to help you with that. Could you  
22     },  
23     {  
24         "role": "user",  
25         "content": "Sure, it's a 100 mile charity ride, not too hilly."  
26     },  
27     {  
28         "role": "assistant",  
29         "content": "Thank you for providing the details. Based on your c  
30     }  
31 ],  
32     temperature=0.26,  
33     max_tokens=2239,  
34     top_p=1
```

Model	Input	Output
gpt-4-0125-preview	\$0.01 / 1K tokens	\$0.03 / 1K tokens
gpt-4-1106-preview	\$0.01 / 1K tokens	\$0.03 / 1K tokens
gpt-4-1106-vision-preview	\$0.01 / 1K tokens	\$0.03 / 1K tokens

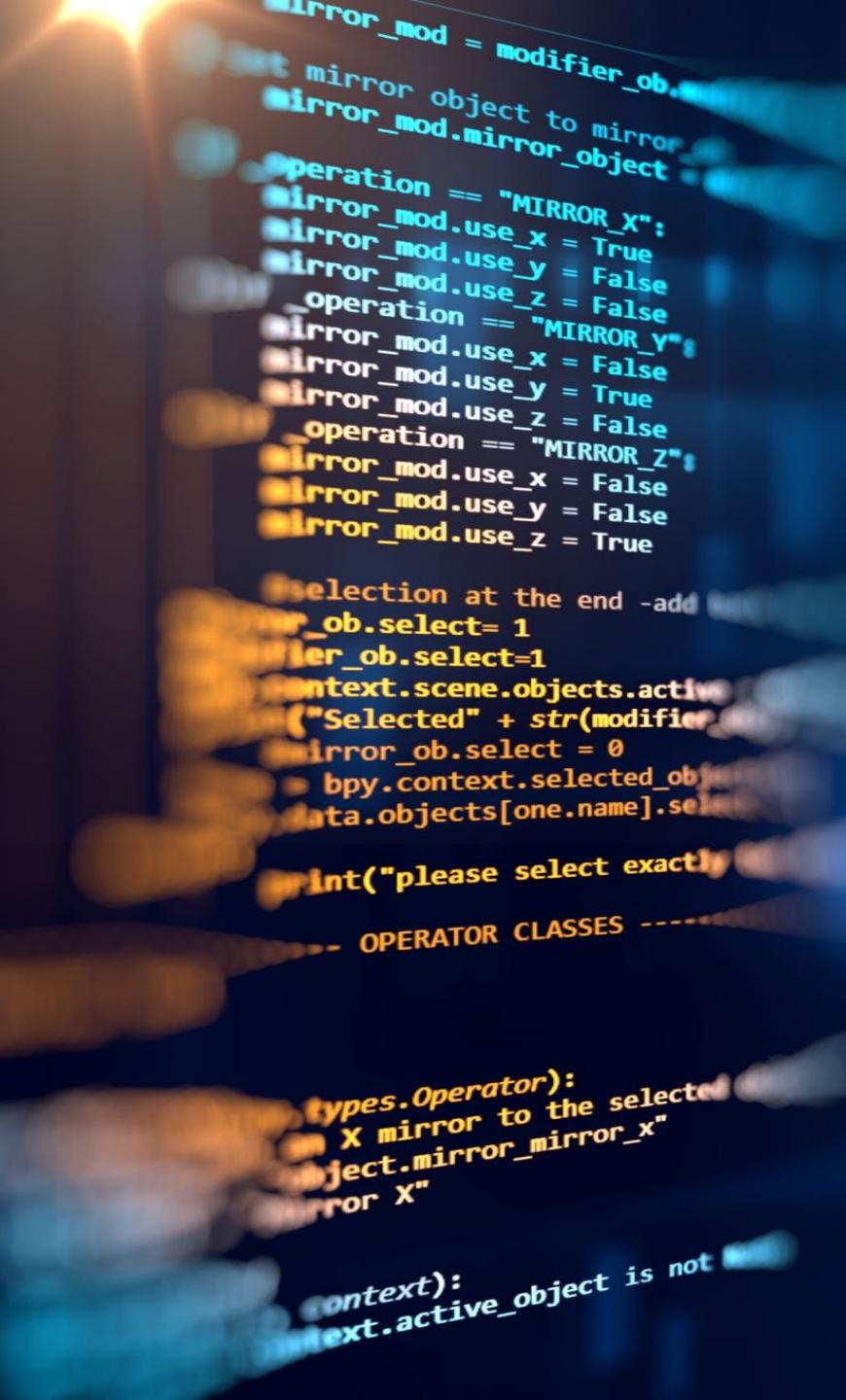
Model	Input	Output
gpt-4	\$0.03 / 1K tokens	\$0.06 / 1K tokens
gpt-4-32k	\$0.06 / 1K tokens	\$0.12 / 1K tokens

Model	Input	Output
gpt-3.5-turbo-0125	\$0.0005 / 1K tokens	\$0.0015 / 1K tokens
gpt-3.5-turbo-instruct	\$0.0015 / 1K tokens	\$0.0020 / 1K tokens



LIVE – Set up Colab

- *Head over to <https://colab.research.google.com/> and set up a free account if you don't already have one*
- *Should work if you have a google account (e.g. gmail)*
- *You may need your phone to validate your existence.*

A close-up photograph of a person's hand pointing their index finger towards a computer monitor. The monitor displays a dark-themed Python script. The script includes code for setting up mirror modifiers, selecting objects, and defining operator classes for mirroring operations like MIRROR_X, MIRROR_Y, and MIRROR_Z. The background is dark, making the white and yellow text stand out.

```
mirror_mod = modifier_obj
# Set mirror object to mirror
mirror_mod.mirror_object = ...
operation = "MIRROR_X":
    mirror_mod.use_x = True
    mirror_mod.use_y = False
    mirror_mod.use_z = False
operation == "MIRROR_Y":
    mirror_mod.use_x = False
    mirror_mod.use_y = True
    mirror_mod.use_z = False
operation == "MIRROR_Z":
    mirror_mod.use_x = False
    mirror_mod.use_y = False
    mirror_mod.use_z = True

#selection at the end - add
#    ob.select= 1
#    other_ob.select=1
context.scene.objects.active = ...
("Selected" + str(modifier))
mirror_ob.select = 0
bpy.context.selected_objects.append(data.objects[one.name].select)
print("please select exactly one object")
-- OPERATOR CLASSES ---
types.Operator:
    # X mirror to the selected object.mirror_mirror_x"
    "mirror X"
    context):
        # context.active_object is not
```

Google Colab – free-ish Python Notebooks

- Write python code and execute it in an easy, browser environment
- There's an AI 'coding companion' to assist you
- Can purchase “compute” credits to use powerful GPUs for ML tasks
 - Free Version for basic stuff
 - Colab Pro - \$9.99/month for 100 ‘compute credits’, \$49.99 for 500.
- Start with OpenAI playground and copy the code over to Colab...

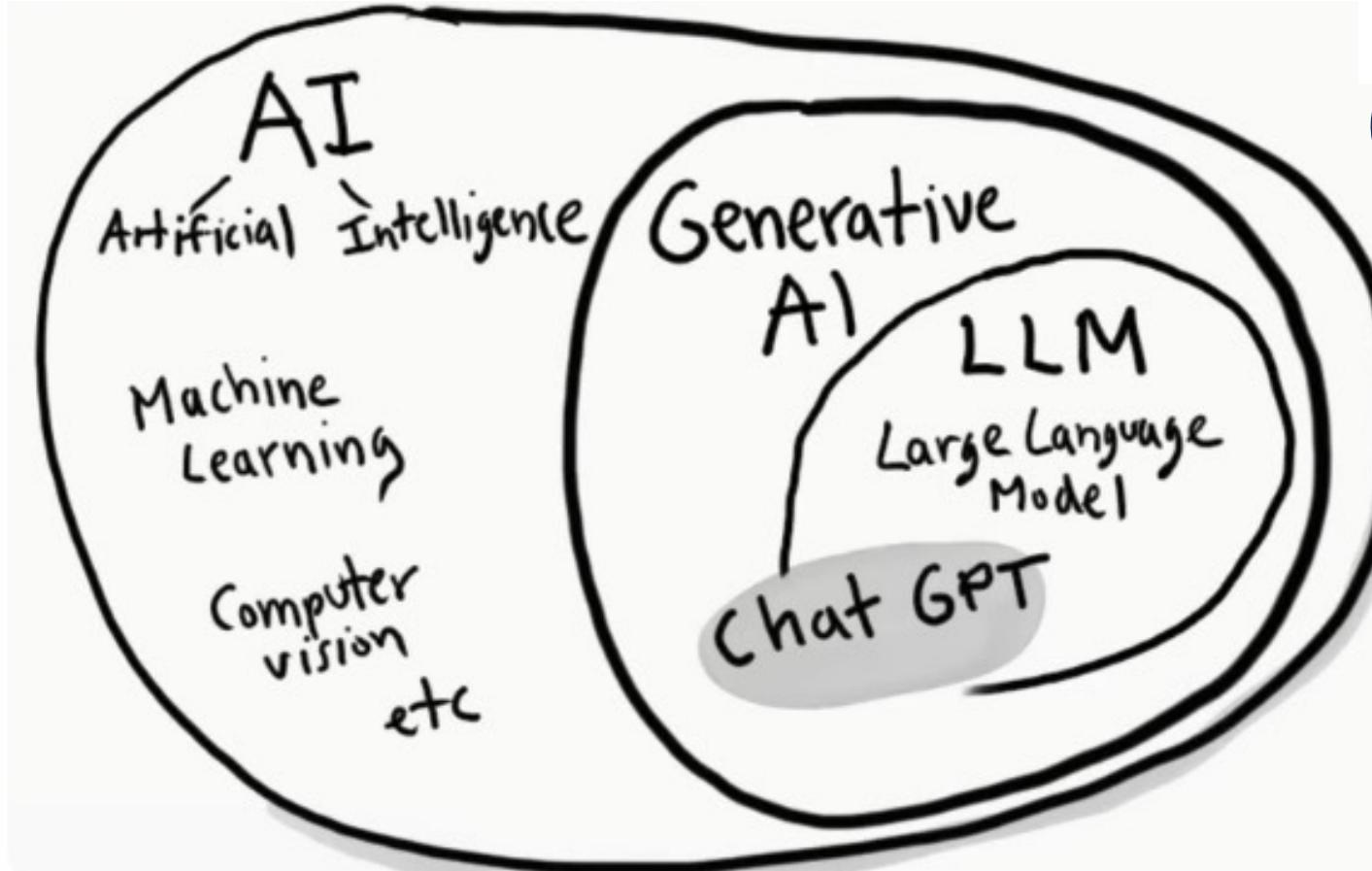


Today's Journey

- Intro/Agenda/Boring Stuff
- Tools – Getting set up
- What if you only had one Brain Cell?
- Image Classification and Transformers
- Retrieval Augmented Generation (RAG)
- Building a RAG-Enabled BOT
- Detour – Local Models
- Fine Tuning
- Q&A



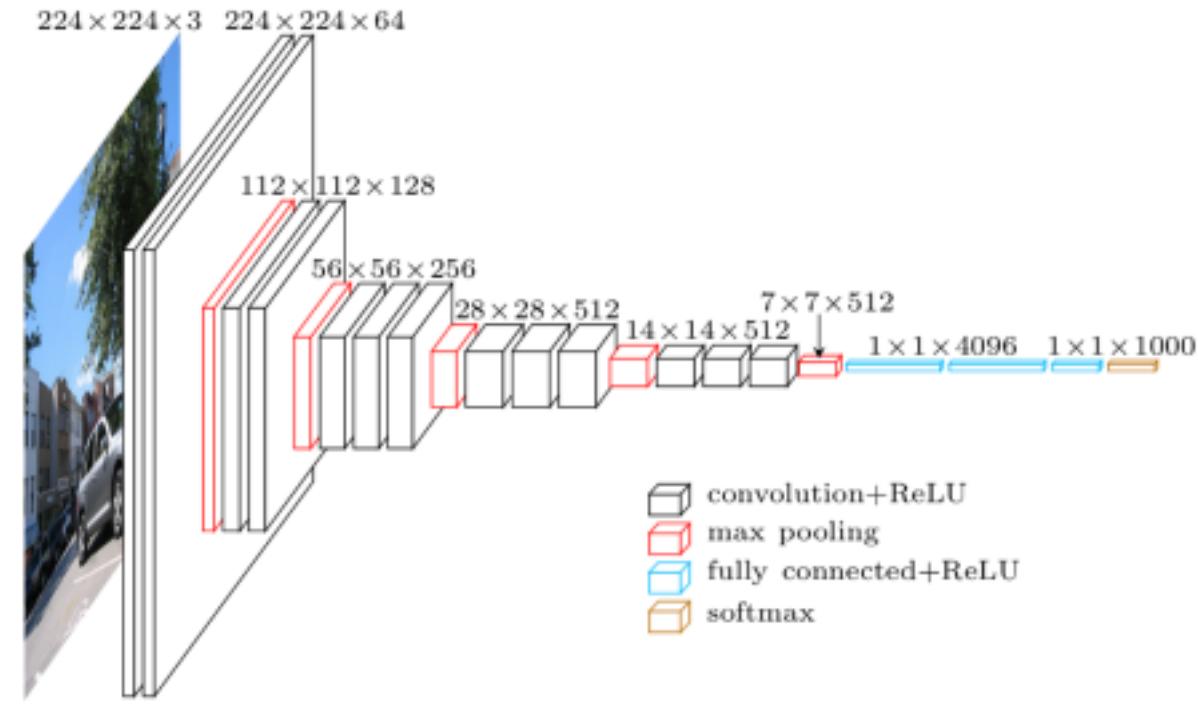
Super quick AI Landscape



The term "generative" refers to ability to produce new content

AI

- **Artificial Intelligence (AI)** is associated with constructing machines that can simulate human intelligence.
- **Machine Learning (ML)** gives machines the ability to learn from experience without the need to be programmed explicitly. Two Main Methods:
 - Supervised learning: **labeled** training data.
 - Unsupervised learning: data **not annotated**.
 - Some common examples of ML models include regression models and classification models
- **Deep learning (DL)** can learn to recognize patterns in data.
 - The term “deep”: DL models are composed of multiple layers of neurons, or processing nodes.
 - The deeper the model, the more layers of neurons it has.
 - This “deepness” allows the model to learn more complex tasks by breaking them down into smaller and smaller pieces.



VGG Deep Convolutional Neural Network Architecture

Terminology - *MODEL*

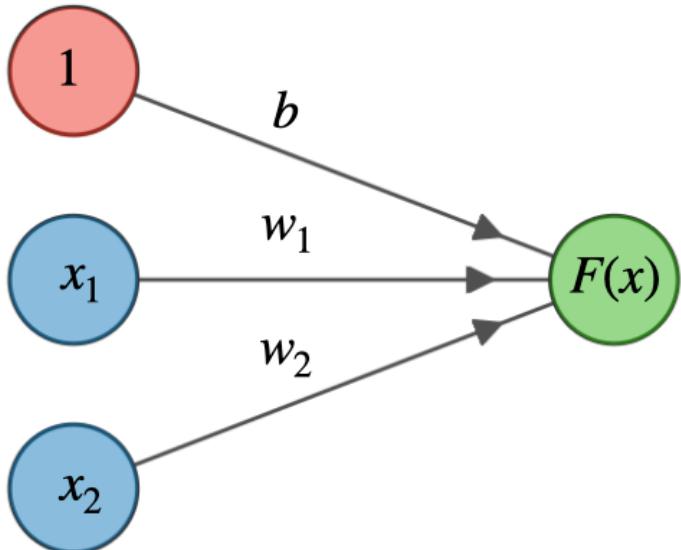
- In the context of artificial intelligence (AI), a "model" is:
 - A set of parameters, resulting from training on large amounts of data
 - Used to make predictions or decisions based on input data.
 - Parameters encapsulate the learned knowledge from data, applies this to new, previously unseen data to perform tasks such as classification, prediction, or creation.
 - Model = Architecture + Parameters



Simple Linear Regression



Linear regression



$$F(x) = w^T x + b$$

Input Layer $\in \mathbb{R}^3$

Output Layer $\in \mathbb{R}^1$

$$F(x) = w_1 \cdot x_1 + w_2 \cdot x_2 + 1 \cdot b$$



What can one Neuron do?

- Steps
 - Load Training and test data
 - Pre-process data
 - Train
 - Test

Let's go to Colab

<https://github.com/tniccum21/DMZ/>

The screenshot shows a GitHub repository named 'DMZ'. The repository is public, as indicated by the 'Public' badge. It has 1 branch and 0 tags. There are 2 commits from the user 'tniccum21' made 2 minutes ago, both updating the 'README.md' file. The repository contains a single file, 'README', which is currently being edited. The content of the README is as follows:

```
DMZ

Colab links for DMZ class

1. One Neuron: https://colab.research.google.com/drive/1fJdtxd3x71wKDgHF0ldUFMd6T7e4Lt-G?usp=sharing
2. Image Classification: https://colab.research.google.com/drive/14maur7VUXh4WBIVghXIQ6et9NDEYts0Z?usp=sharing
3. Vito: https://colab.research.google.com/drive/1qArAXlvaEM001L\_99sTn8ePUSNnlhdRR?usp=sharing
4. LLM-Part-1: https://colab.research.google.com/drive/1sbfcampC9WrOMiJ1r2mC1tzcClqmUN3N?usp=sharing
5. LLM-Part-2: https://colab.research.google.com/drive/1uWTt2azpfkj9AMzCKeT\_2lKMghPATle?usp=sharing
6. LLM-Part-3: https://colab.research.google.com/drive/10o3vh7fmHmeODidK\_GVmWGD6Vgc1U5O?usp=sharing
7. Fine-Tuning: https://colab.research.google.com/drive/1QfUMocrTLmZI0vE9cHnx1JAEif8WNrV?usp=sharing
```



Today's Journey

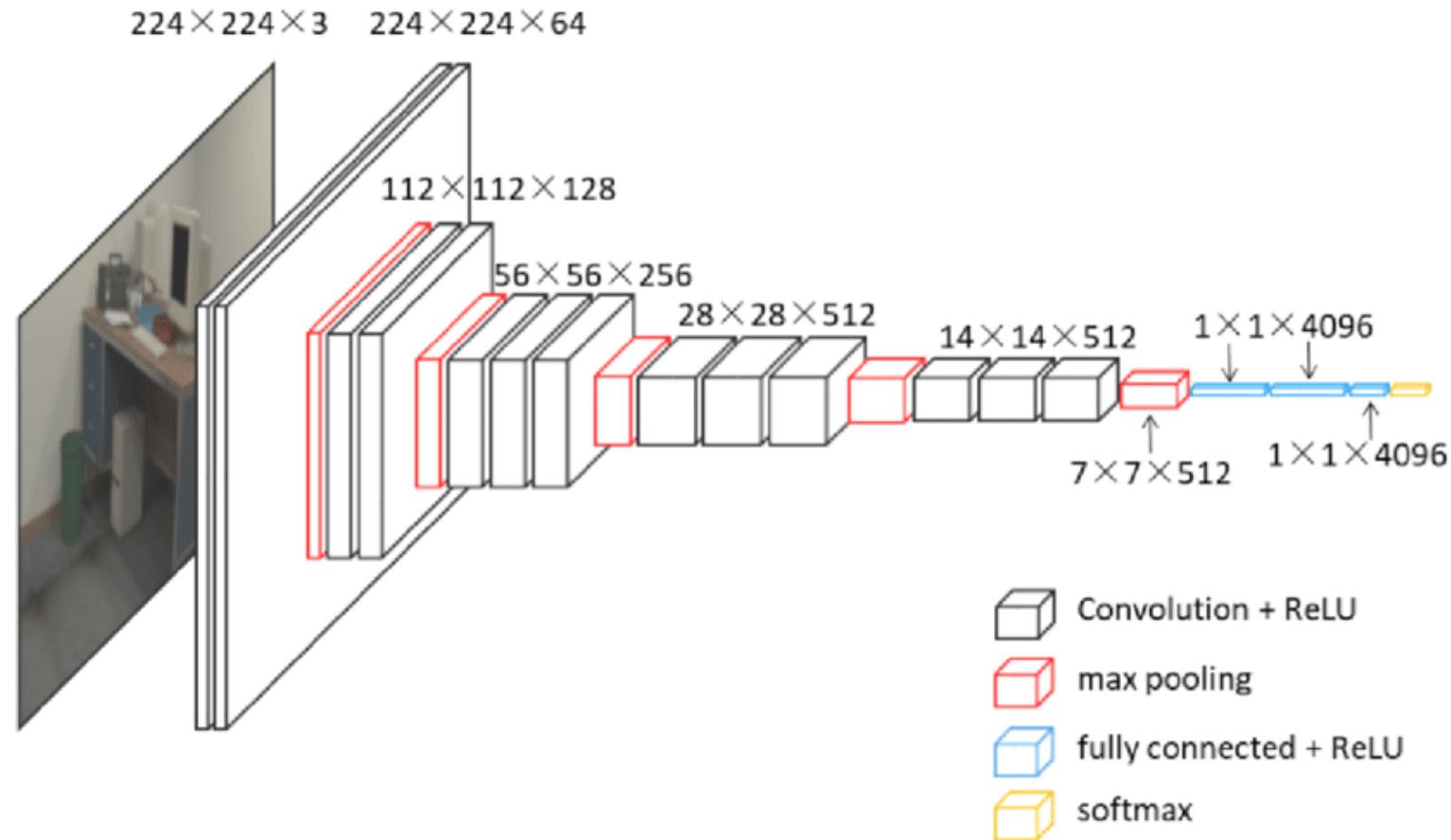
- Intro/Agenda/Boring Stuff
- Tools – Getting set up
- What if you only had one Brain Cell?
- Image Classification and Transformers
- Retrieval Augmented Generation (RAG)
- Building a RAG-Enabled BOT
- Detour – Local Models
- Fine Tuning
- Q&A



Image classification example



Transformers started as Vision Models like VGG-16





Colab – Image Classification

- Steps
 - Load Training and test data
 - Pre-process data
 - Train
 - Test

Let's go to Colab

<https://github.com/tniccum21/DMZ/>

The screenshot shows a GitHub repository named 'DMZ'. The repository is public, has one branch, and zero tags. There are two commits from the user 'tniccum21' that updated the 'README.md' file. The repository contains a single file, 'README', which is currently being edited. The content of the README is as follows:

```
DMZ

Colab links for DMZ class

1. One Neuron: https://colab.research.google.com/drive/1fJdtxd3x71wKDgHF0ldUFMd6T7e4Lt-G?usp=sharing
2. Image Classification: https://colab.research.google.com/drive/14maur7VUXh4WBIVghXIQ6et9NDEYts0Z?usp=sharing
3. Vito: https://colab.research.google.com/drive/1qArAXlvaEM001L\_99sTn8ePUSNnlhdRR?usp=sharing
4. LLM-Part-1: https://colab.research.google.com/drive/1sbfcampC9WrOMiJ1r2mC1tzcClqmUN3N?usp=sharing
5. LLM-Part-2: https://colab.research.google.com/drive/1uWTt2azpfkj9AMzCKeT\_2lKMghPATle?usp=sharing
6. LLM-Part-3: https://colab.research.google.com/drive/10o3vh7fmHmeODidK\_GVmWGD6Vgc1U5O?usp=sharing
7. Fine-Tuning: https://colab.research.google.com/drive/1QfUMocrTLmZI0vE9cHnx1JAEif8WNrV?usp=sharing
```



Today's Journey

- Intro/Agenda/Boring Stuff
- Tools – Getting set up
- What if you only had one Brain Cell?
- Image Classification and Transformers
- Retrieval Augmented Generation (RAG)
- Building a RAG-Enabled BOT
- Detour – Local Models
- Fine Tuning
- Q&A

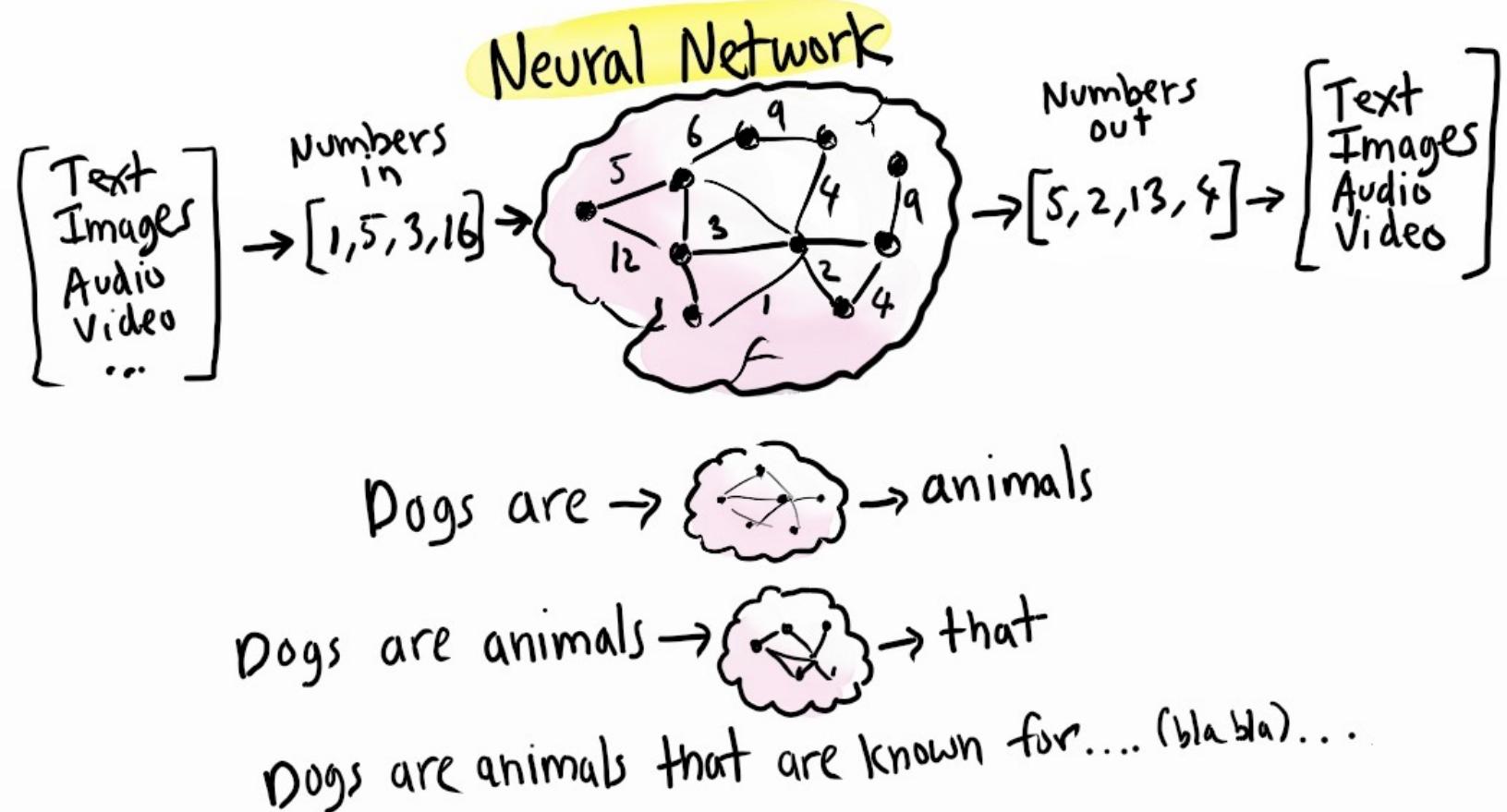


Terminology – Large Language Model (LLM)

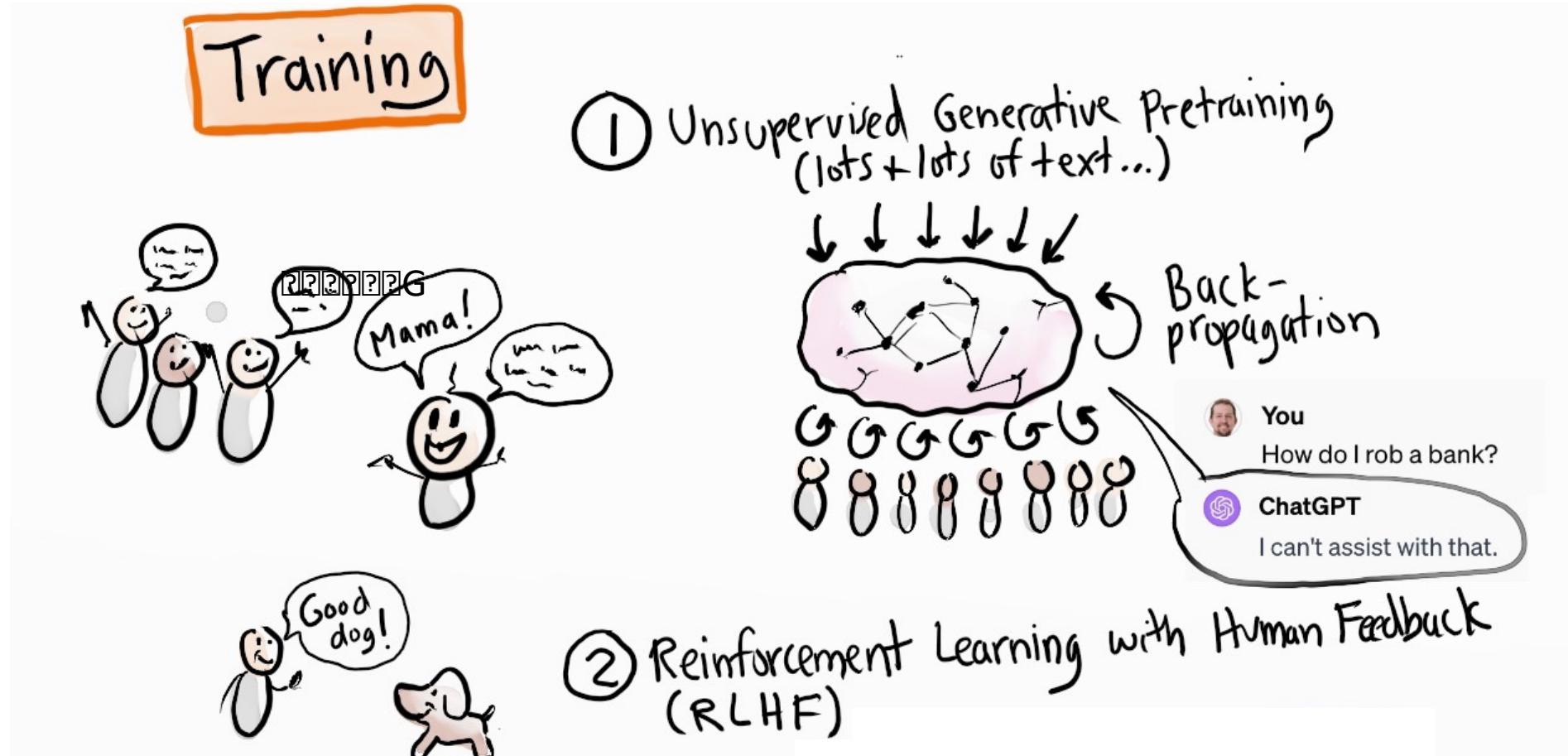
- Large Language Models (LLMs) are a specific type of AI model that specialize in processing, understanding, and generating human language.
- LLMs are trained on vast amounts of text data, learning the statistical properties of language and how words and phrases are commonly structured to convey meaning.
- LLMs are able to perform a wide range of language-based tasks such as translation, summarization, question answering, and content creation.
- LLMs specifically focus on understanding and generating human language.
- LLMs are characterized by their massive scale, both in terms of the data they are trained on and the size of the neural networks themselves, enabling them to understand context, nuance, and even subtleties of different languages and dialects.
- All LLMs are AI models, not all AI models are LLMs. "LLM" refers specifically to those models trained on large datasets to understand and produce human language.

LLMs are based on Neural Networks

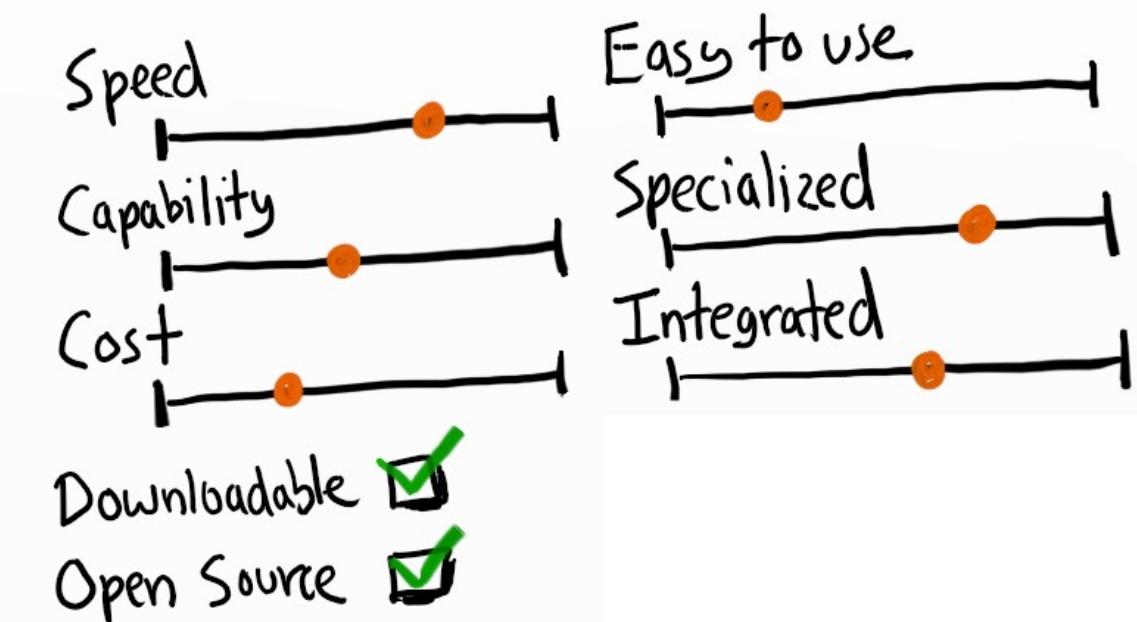
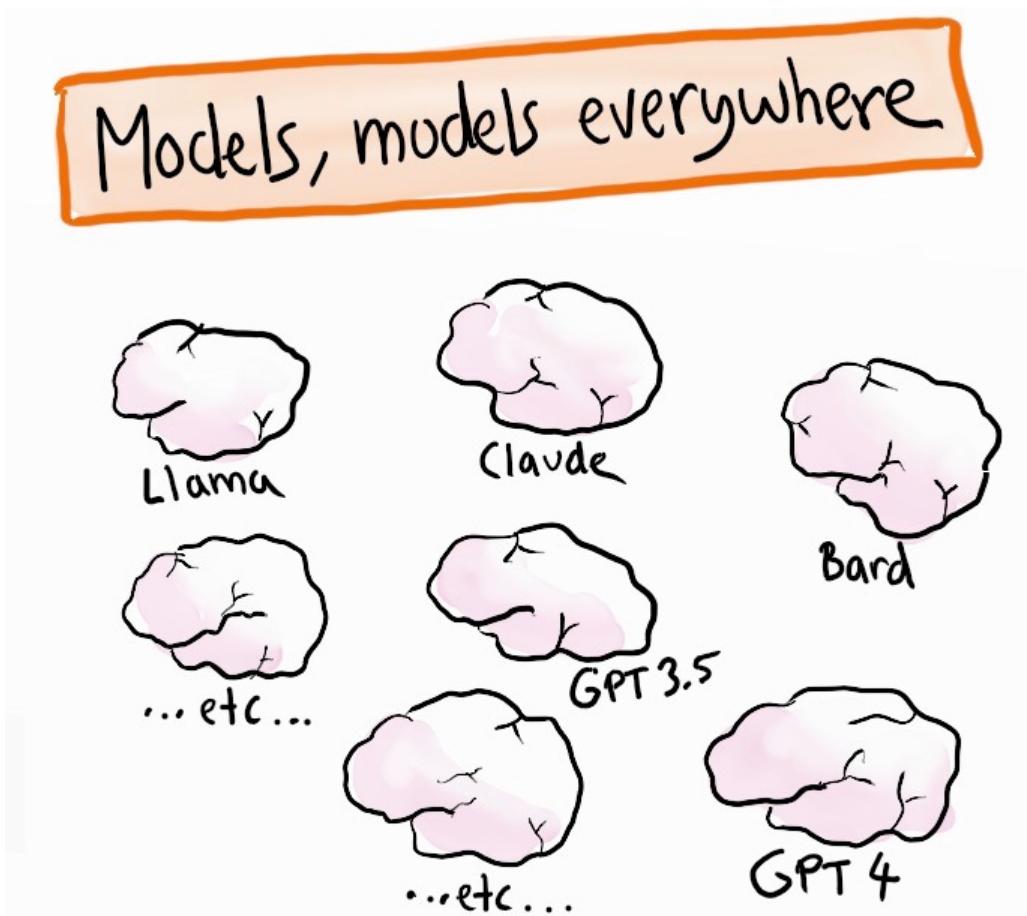
An LLM, or Large Language Model, is an advanced artificial intelligence algorithm designed to understand, generate, and interact with human language. These models are trained on enormous amounts of text, enabling them to perform a wide range of natural language processing (NLP) tasks such as text generation, translation, summarization, and question-answering.



Training the Neural Network

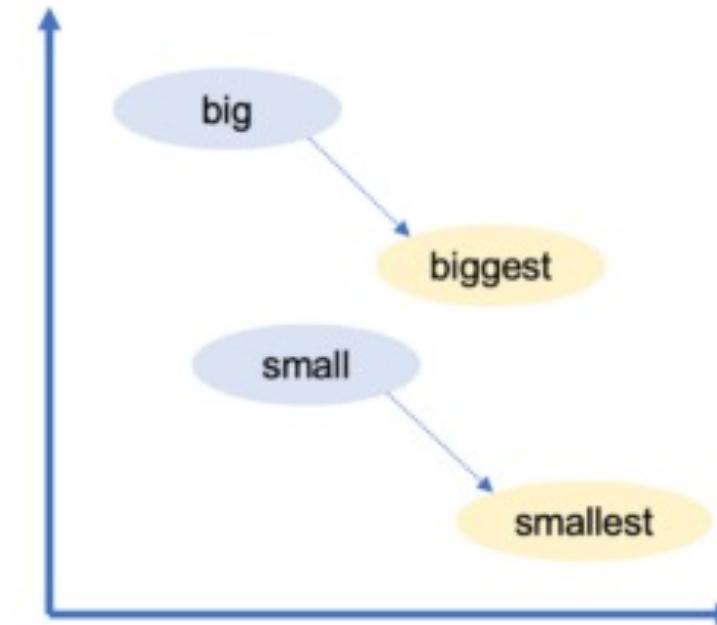
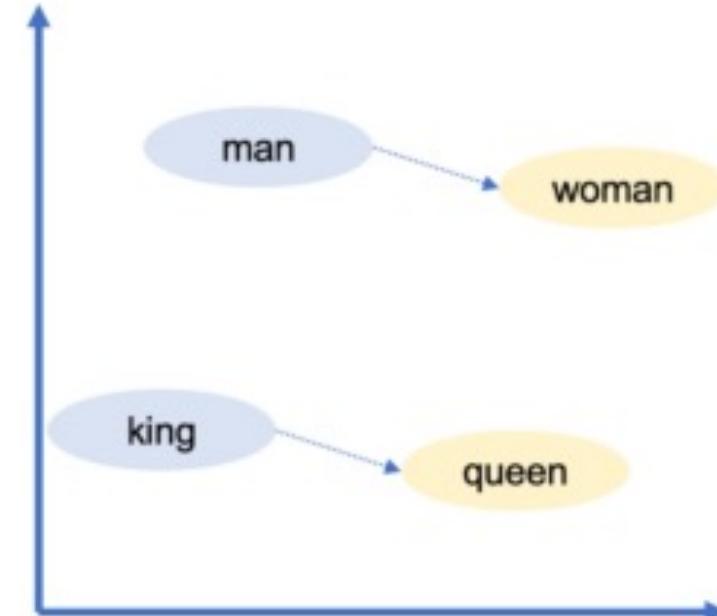


Training a Network results in a “model”



What IS the LLM Model Learning?

- Text input is converted to numeric “vectors” using an encoder
- Each vector can be hundreds or thousands of numbers
- A vector plots the word into a “space” - this “embedding” is learned via training
- Vectors that are close to each other in this “space” are related in meaning
- Context around the word changes its vector!
- Vectors are used to encode the input, and output of the numerically-oriented Neural Network
- Models learn, based on these vectors, the probabilities of the “next” word in a sequence



What is GPT? ChatGPT?

- GPT = "Generative Pre-trained Transformer."
- It is a type of artificial intelligence model developed by that generates human-like text based on the input it receives.
- "Pre-trained" means that it has been trained on a large dataset before being fine-tuned for specific tasks. The "transformer" component is an architecture used in machine learning models to process and generate sequences of data, such as text or speech.
- Pre-training can take day, weeks or even months on very expensive hardware. Typically it is beyond the average companies budget!
- GPT4 - Estimated \$20M per training run
- Meta – Recently ordered 150,000 GPUs!

OpenAI and GPT

- OpenAI popularized the **Generative Pretrained Transformer** (GPT) series.
- GPT-2, with **1.5 billion parameters**
- GPT-3, with **175 billion** showed a remarkable ability to generate human-like text
- GPT-4 has over **1 TRILLION parameters** and takes (a reported) \$100 Million in computing resources to train
- GPTs now built into: Microsoft Bing, Github Copilot, Duolingo, Khan Academy, and many to come!



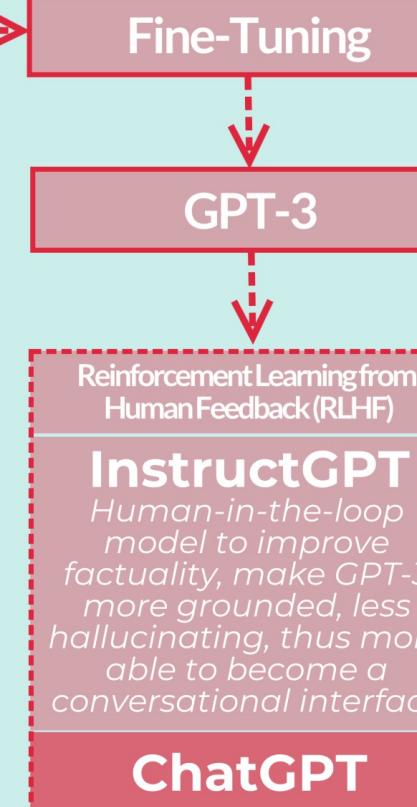
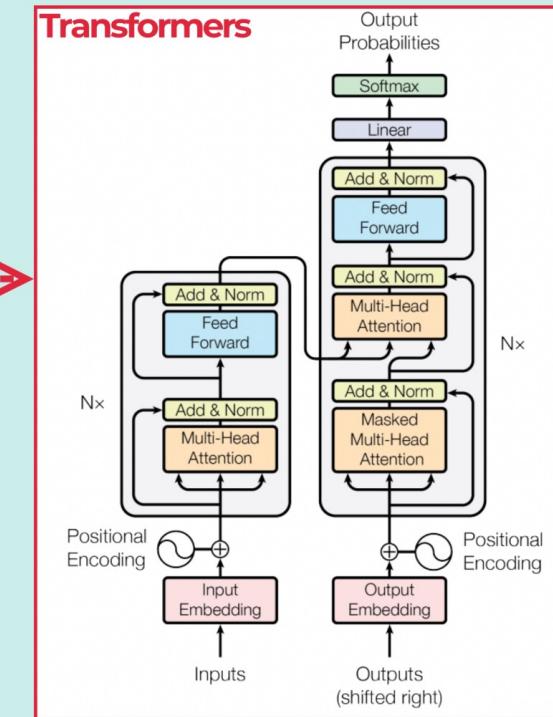
GPT vs ChatGPT

How Does ChatGPT Work?

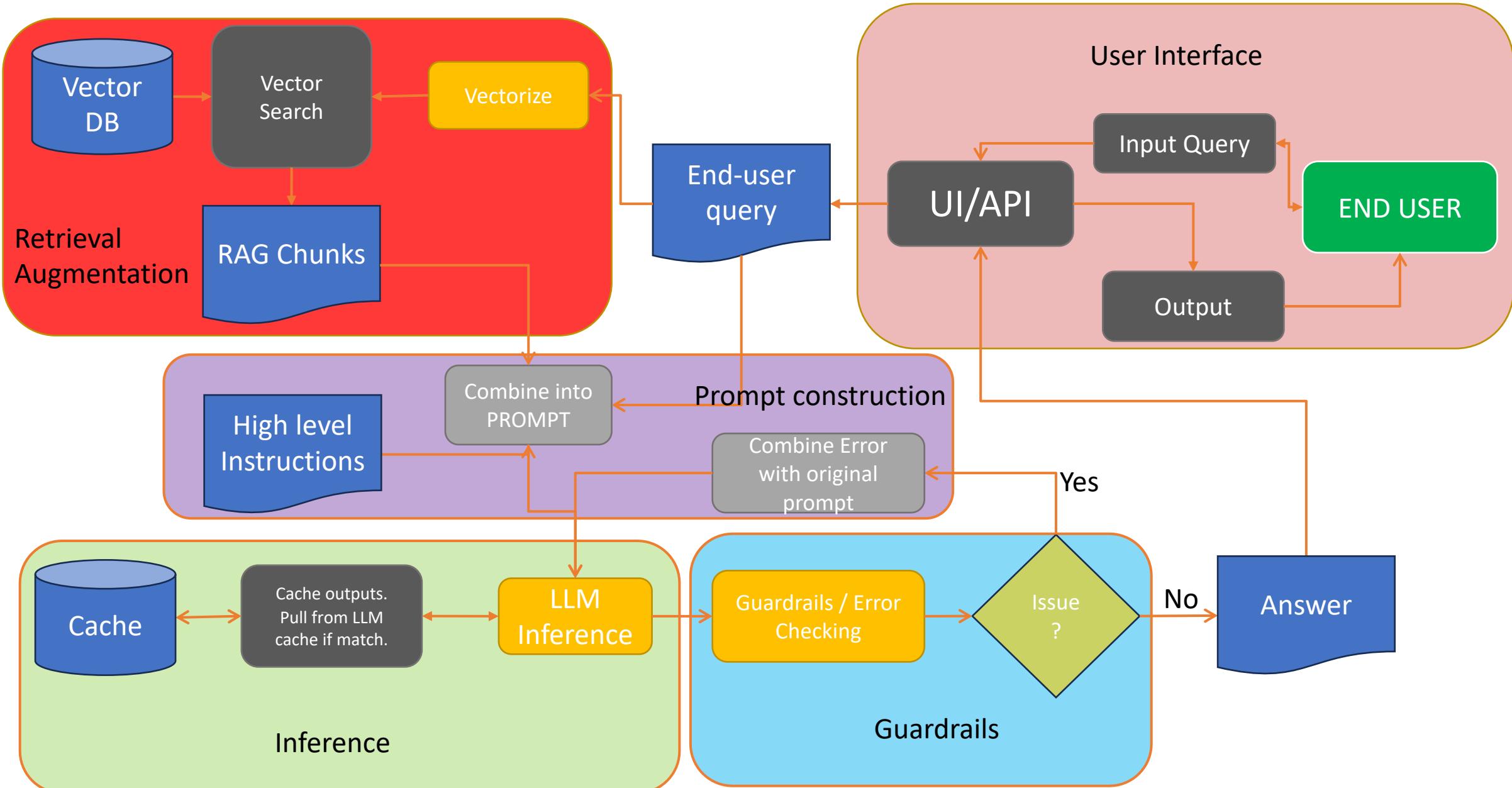
ChatGPT leverages GPT-3.5 as the underlying model, while it uses an additional layer, a model called InstructGPT, which has become a standard within the OpenAI large language models. InstructGPT optimizes conversational abilities and improves on top of the existing GPT models.

Pre-Training

- Large Amount of Data.
- Billions of Parameters.
- Unsupervised Learning.



A Basic Generative AI Platform





Today's Journey

- Intro/Agenda/Boring Stuff
- Tools – Getting set up
- What if you only had one Brain Cell?
- Image Classification and Transformers
- Retrieval Augmented Generation (RAG,
- Building a RAG-Enabled BOT
- Detour – Local Models
- Fine Tuning
- Q&A

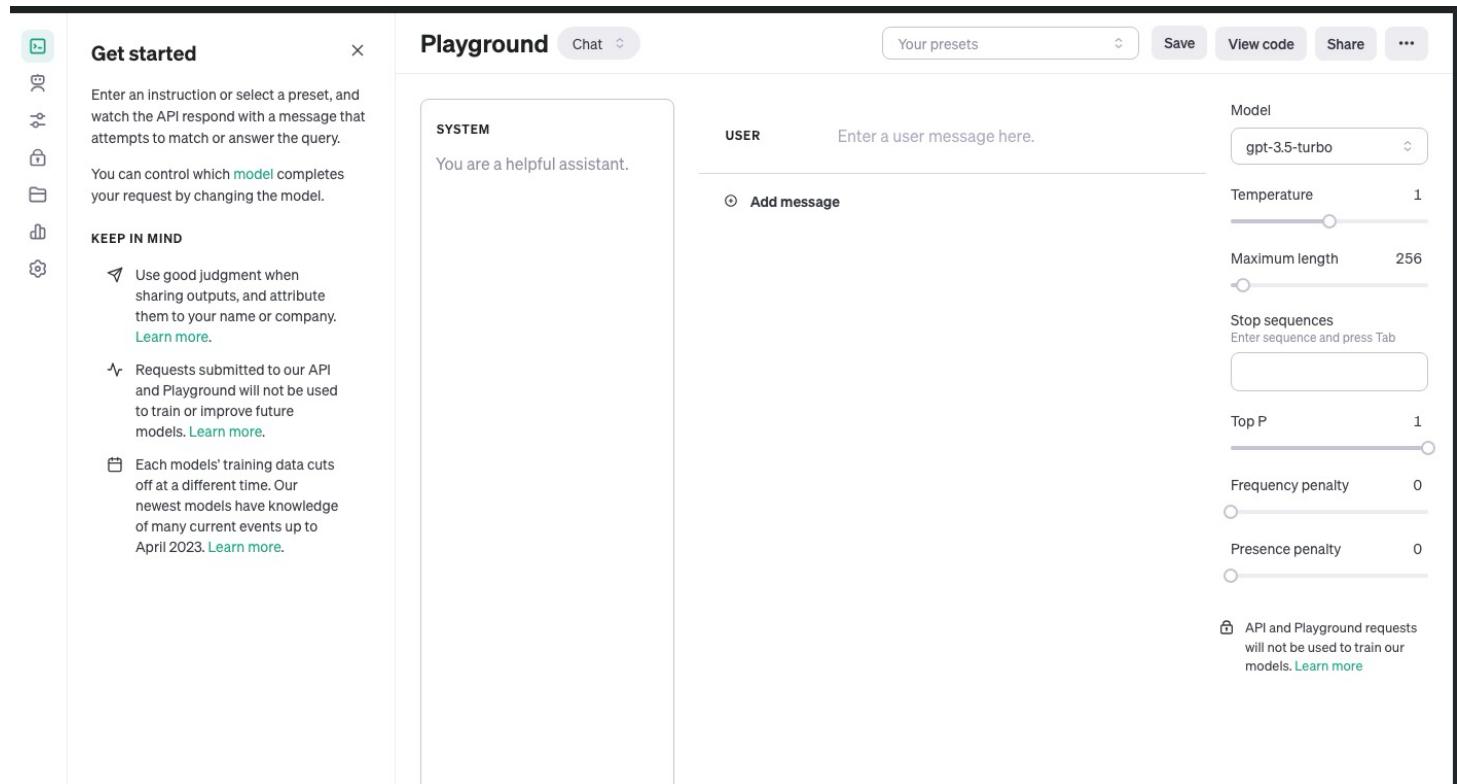


A Quick App - ViTO

- We'll walk through how to control OpenAI's GPT via a Python/API interface
- ViTO is a short python app that:
 - Narrows the mission to one topic
 - Preloads the prompt to stay on topic and guide the LLM
 - Uses conversational memory to allow follow-up questions



We can start in OpenAI Playground



- Try some prompts
- Play with Model Parameters to see effects
- Export the code

Let's go to Colab

<https://github.com/tniccum21/DMZ/>

The screenshot shows a GitHub repository named 'DMZ'. The repository is public, has 1 branch, and 0 tags. There are 2 commits from the user 'tniccum21' that updated the 'README.md' file 2 minutes ago. The repository contains a single file, 'README', which is currently being edited. The content of the README file is as follows:

```
DMZ

Colab links for DMZ class

1. One Neuron: https://colab.research.google.com/drive/1fJdtxd3x71wKDgHF0ldUFMd6T7e4Lt-G?usp=sharing
2. Image Classification: https://colab.research.google.com/drive/14maur7VUXh4WBIVghXIQ6et9NDEYts0Z?usp=sharing
3. Vito: https://colab.research.google.com/drive/1qArAXlvaEM001L\_99sTn8ePUSNnlhdRR?usp=sharing
4. LLM-Part-1: https://colab.research.google.com/drive/1sbfcampC9WrOMiJ1r2mC1tzcClqmUN3N?usp=sharing
5. LLM-Part-2: https://colab.research.google.com/drive/1uWTt2azpfkj9AMzCKeT\_2lKMghPATle?usp=sharing
6. LLM-Part-3: https://colab.research.google.com/drive/10o3vh7fmHmeODidK\_GVmWGD6Vgc1U5O?usp=sharing
7. Fine-Tuning: https://colab.research.google.com/drive/1QfUMocrTLmZI0vE9cHnx1JAEif8WNrV?usp=sharing
```



So... What if you have special information, not already in a Model?

- Dynamic information
- Data stored in a database
- Documents in PDFs
- Spreadsheets
- Company data not on the internet?

Three Approaches...

Train a new model

- EXPENSIVE! Possibly thousands to millions of dollars.
- Need enormous quantities of data to make it worth while.

Fine-tune an existing model

- Much cheaper, but still a fair amount of effort.
- Data must be curated, cleaned and prepped.
- Works better to alter behavior, not add knowledge.

Retrieval Augmented Generation – RAG

- Use the power of the LLMs to analyze and use documents as input.
- Cheap, effective.

Retrieval Augmented Generation (RAG)

- RAG is a hybrid framework that uses both retrieval models and generative models to produce text that is not only contextually accurate but also information-rich.
- Retrieval: the retrieval model acts as a 'librarian,' selecting relevant information from a database or a batch of documents.
- Generation: acts as the 'writer,' crafting informative text based on the retrieved data.
- They work in concert to craft answers that are accurate and contextually rich.



Benefits of RAG

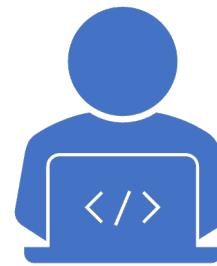
- RAG is exceptionally versatile. It's used in various applications like:
 - Real-time news summarization,
 - Automated customer service,
 - Complex research tasks that require understanding and integrating information from multiple sources.
- Adaptability allows it to be incorporated into different types of systems, making it an invaluable asset in modern language processing tasks.
- In summary, Retrieval Augmented Generation has revolutionized AI by leveraging the strengths of both retrieval and generative models.
- Understanding RAG is critical for anyone looking to harness the full power of Natural Language Processing.

Components of a RAG System



Retrieval Models - search through a large body of data (structured or unstructured) to find the relevant pieces of information that can be used for text generation.

Generally implemented using vector embeddings and vector search.



Generative Models - act as creative writers, combining and synthesizing the retrieved information into text relevant to the user's query. They use a Large Language Models (LLMs), with the capability to create text that is grammatically correct, meaningful, and aligned with the initial query or prompt.

They take the raw data returned by the retrieval models and give it a structure, making the information easily understandable.

Components of a RAG System

- **Source Data**
 - the knowledge that the retrieval model scans through to find relevant information
 - PDFs, DOC, TXT, even images, spreadsheets, websites, etc.
- **Data Chunking**
 - We divide the source documents into “chunks”
 - Chunk sizes are an “art form”
 - Overlap
- **Text-to-Vector Conversion (Embeddings)**
 - Create a “vector” representation of each chunk of information.
 - These vectors encapsulate the semantic meaning of the text.
 - The vectors make the information searchable by “meaning”
 - Stored in a specialized “vector” database

Embeddings

A word embedding can have hundreds of values, each representing a different aspect of a word's meaning. Just as you might describe a house by its characteristics type, location, bedrooms, bathrooms, levels - the values in an embedding quantify a word's linguistic features.

The way these characteristics are “learned” means we don't know exactly what each value represents, but words we expect to be used in comparable ways often have similar-looking embeddings.

work

Sea

Ocean

football

Soccer

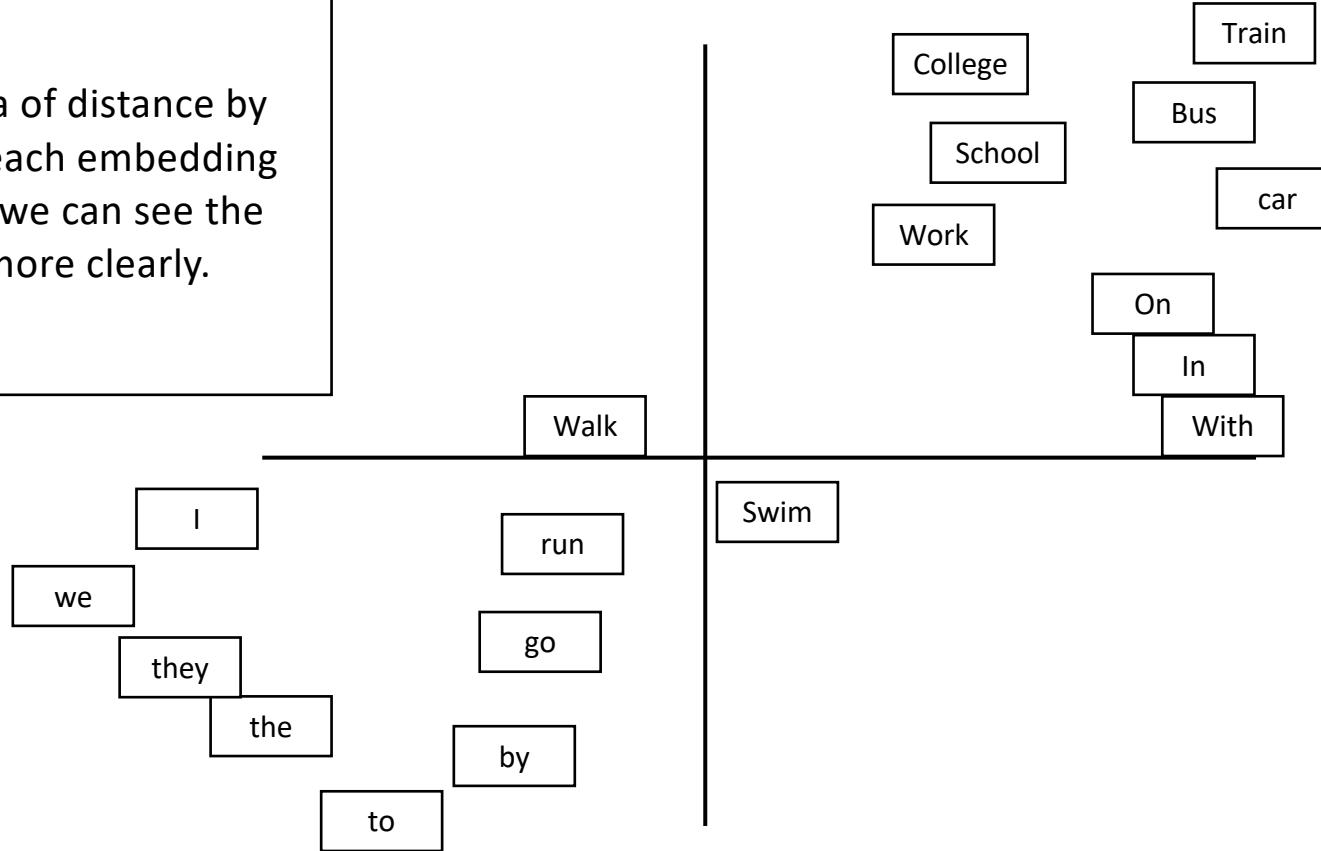
I

We

A pair of words like **sea** and **ocean** may not be used in identical contexts (we say ‘lost at sea’, not ‘lost at ocean’ so they aren’t always direct substitutes), but their meanings are close to each other, and embeddings allow us to quantify that closeness using mathematical measurements of ‘distance’.

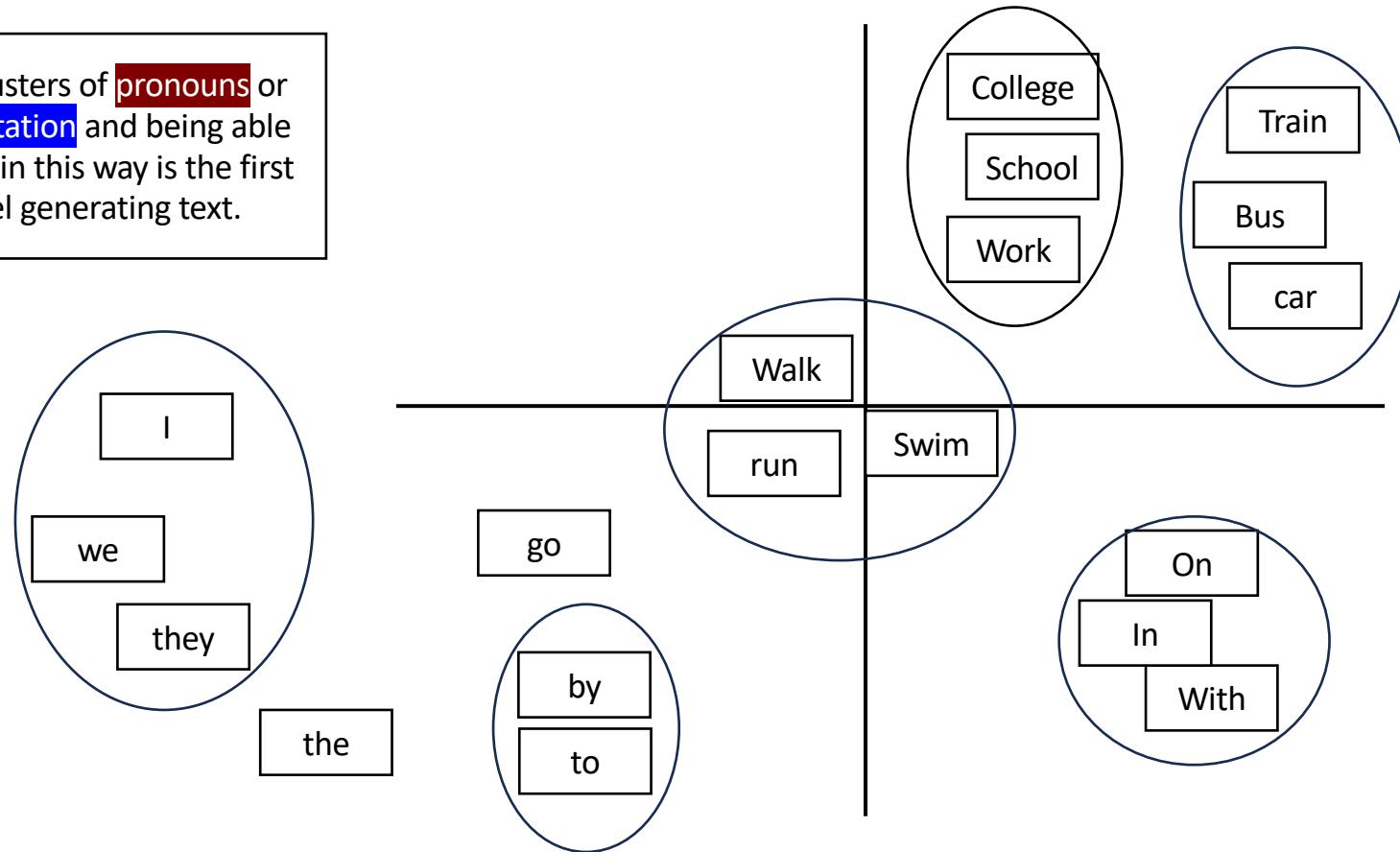
Vector-space

We can demonstrate this idea of distance by reducing hundreds of values each embedding represents to just two. Now we can see the distance between words more clearly.

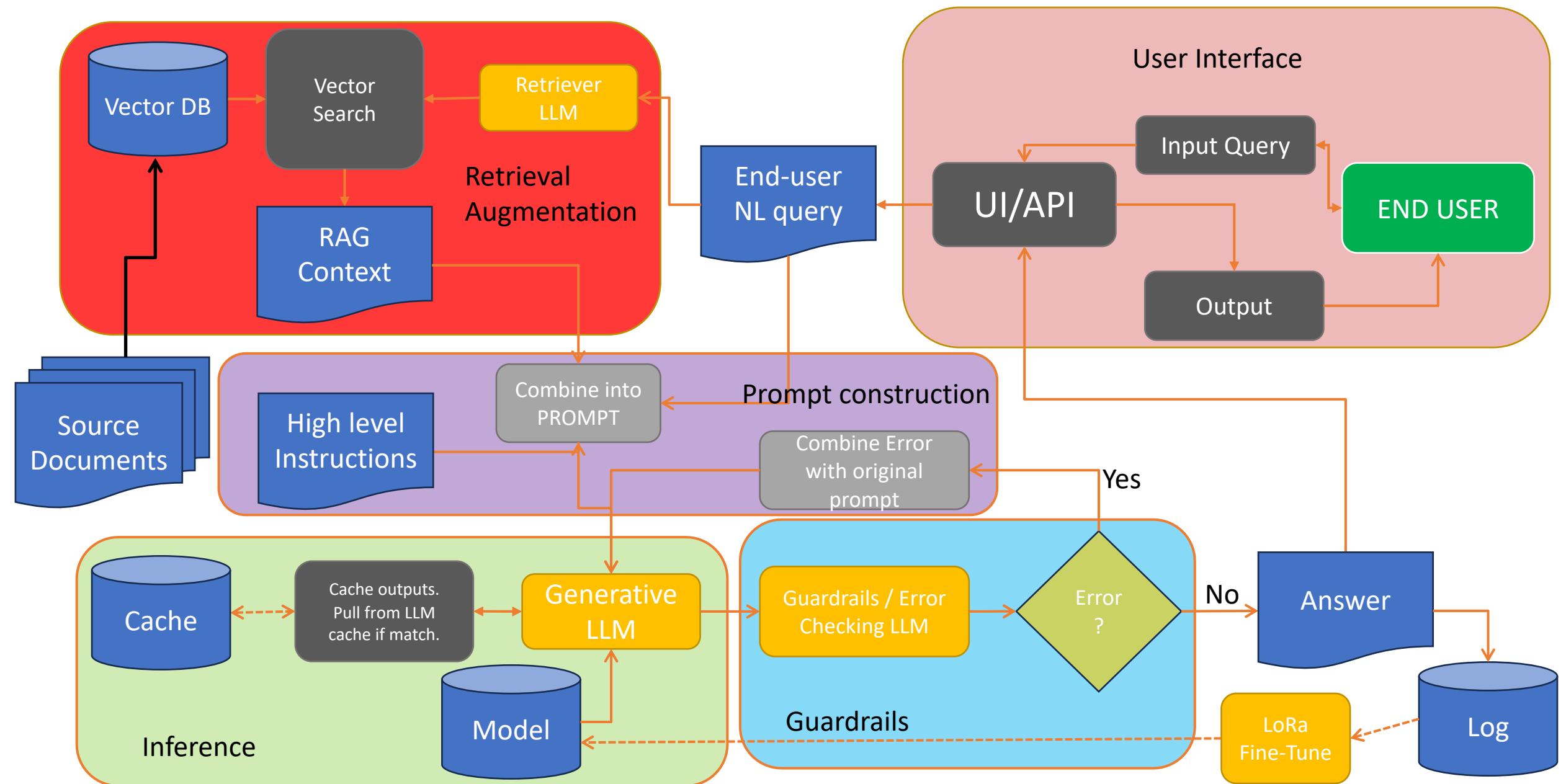


Vector “nearness” = Semantic “nearness”

We might spot clusters of pronouns or mode of transportation and being able to quantify words in this way is the first step in a model generating text.

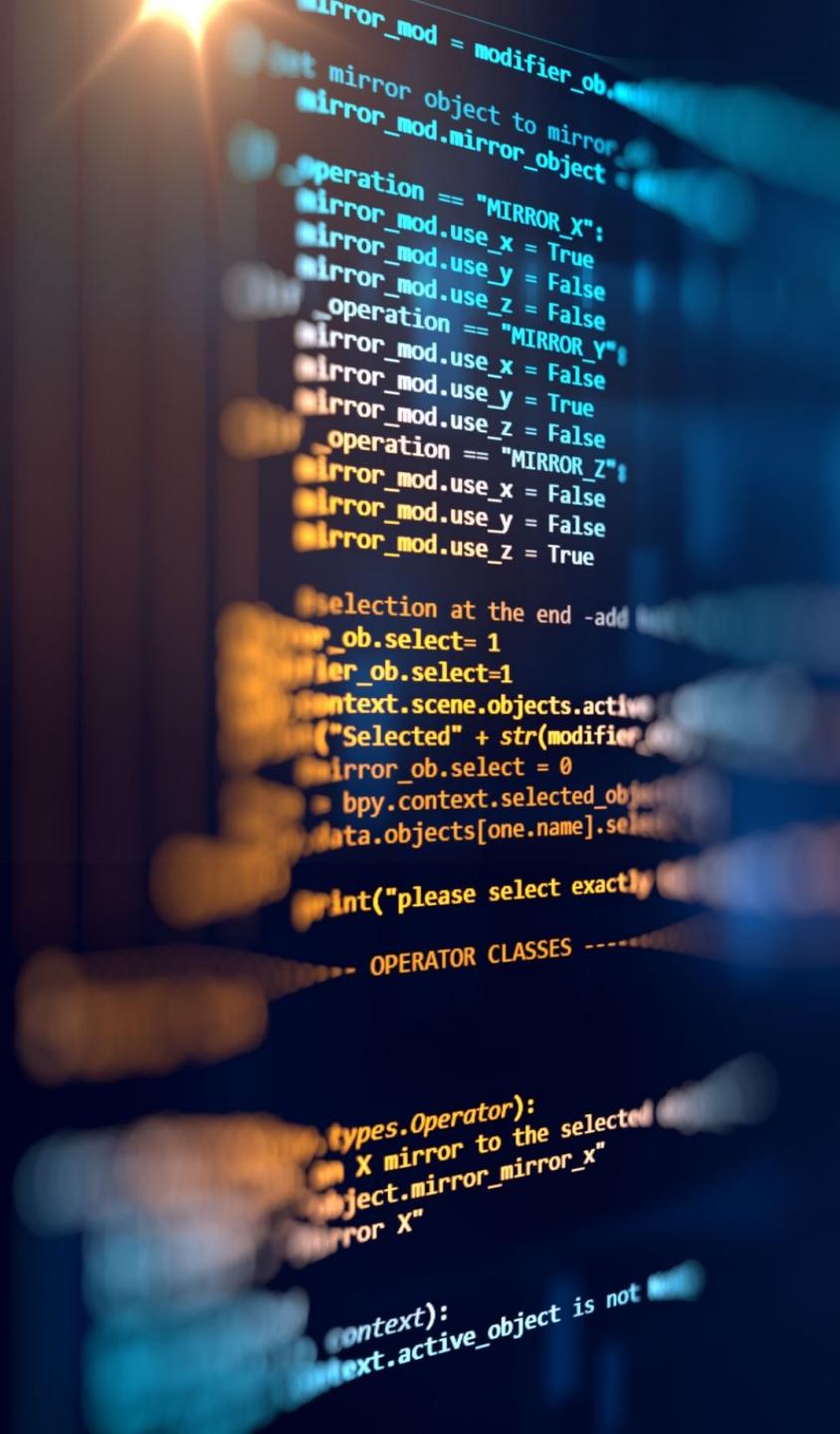


A RAG Architecture



Our App – RAG Expert

- We'll utilize a pdf source of information
- We'll ingest that into our RAG expert
- We'll develop a way to 'chunk' all of that into a vector database
- Then we'll use all of that to assist in answering questions about
- We'll break it into a lot of individual pieces and take our time to walk through them.





Today's Journey

- Intro/Agenda/Boring Stuff
- Tools – Getting set up
- What if you only had one Brain Cell?
- Image Classification and Ti
- Retrieval Augmented Generati
- Building a RAG-Enabled BC
- Detour – Local Models
- Fine Tuning
- Q&A



Local LLMs – Why?

- The big guys have invested billions – their models are great – why go local?
 - **Better data privacy:** By using a local LLM, all the data generated stays in-house, ensuring privacy and preventing access by companies running publicly-facing LLMs.
 - **Offline usage:** Uninterrupted usage in remote or isolated areas without reliable internet access, providing a valuable tool in such scenarios.
 - **Cost Savings:** The average price of accessing an LLM like GPT-4 or Claude 2 is \$20 per month per person. Although that might not seem like an alarming price, or you pay by the Token ingested or generated. It can add up.
 - **Customization:** a locally hosted AI can be customized for specific needs. You can train on proprietary data tailored to your use cases, improving relevance and accuracy.

Local LLMs – Why Not?

- The big guys have invested billions – their models are great – why NOT go local?
 - **Resource Intensive:** you'll need high-end hardware. Think powerful CPUs, lots of RAM, and likely several dedicated GPUs.
 - **Slooooow:** After near-instant responses from ChatGPT, Bard, and others, local LLMs can feel sluggish.
 - **Complex:** Setting up a full local LLM stack requires downloading frameworks, configuring infrastructure, and integrating various components.
 - **Limited Knowledge:** Most local LLMs are smaller and have less “capacity”. Also, they generally haven't been trained on as much data.

HuggingFace.co

- A “hub” where thousands of AI models and datasets are stored
- Many “local” models are available, from tiny models that can run on a phone, to giants.
- Some are remarkably capable.

The screenshot shows the HuggingFace.co website interface. On the left, a sidebar for the user 'tniccum21' includes options for creating new models, managing inbox, settings, and organizations. Below this are resources like the hub guide and documentation. In the center, a 'Following' section lists AI creators to follow, with three specific profiles shown: 'abhishek', 'LucileSaulnier', and 'SkalskiP'. On the right, a 'Trending' section displays popular models over the last 7 days, including 'metavoiceio/metavoice-1B-v0.1', 'briaai/RMBG-1.4', 'openbmb/MiniCPM-2B-sft-fp32', 'abacusai/Smaug-72B-v0.1', and 'stabilityai/stable-cascade'. A red banner at the bottom features the text 'Running on ZERO InstantID'.

Example

- Models that are trained to generate SQL – currently 630 different models
- There are a lot of different sizes with different attributes
- Most common are 7b, 34b and 70b parameter models
- New methods are allowing the “cramming” of more parameters into RAM - quantization

The screenshot shows a search interface with the query "sql" entered. The results list 630 models. The results are as follows:

- defog/sqlcoder-7b-2 (Text Generation, Updated 1 day ago, 3.67k views, 79 likes)
- chatdb/natural-sql-7b (Text Generation, Updated 9 days ago, 1.08k views, 62 likes)
- defog/sqlcoder-70b-alpha (Text Generation, Updated 14 days ago, 3.52k views, 134 likes)
- defog/sqlcoder-7b (Text Generation, Updated 8 days ago, 3.78k views, 54 likes)
- llmware/slim-sql-tool (Text Generation, Updated 3 days ago, 55 views, 3 likes)
- defog/sqlcoder (Text Generation, Updated Sep 15, 2023, 2.25k views, 278 likes)
- defog/sqlcoder2 (Text Generation, Updated Oct 13, 2023, 2.37k views, 97 likes)
- llmware/slim-sql-1b-v0 (Text Generation, Updated Jan 4, 276 views, 5 likes)
- chatdb/natural-sql-7b-GGUF (Text Generation, Updated 9 days ago, 4 views)
- whoami02/defog-sqlcoder-2-GGUF (Text Generation, Updated 2 days ago, 2 views)
- charanhu/text_to_sql_5 (Text2Text Generation, Updated Jan 25, 2023, 35 views, 3 likes)
- cssupport/t5-small-awesome-text-to-sql (Text2Text Generation, Updated Aug 28, 2023, 7.46k views, 22 likes)

defog/sqlcoder-70b-alpha like 134

Text Generation Transformers Safetensors llama Inference Endpoints

Model card Files and versions Community 9

Model Card for SQLCoder-70B-Alpha

A capable large language model for natural language to SQL generation. Outperforms all generalist models (including GPT-4) on text to SQL.

Percentage of correctly generated SQL queries on novel schemas not seen in training (n = 200) in SQL-Eval

Model	Percentage (%)
defog-sqlcoder-70b	93%
defog-sqlcoder-34b	84%
gpt-4	82%
gpt-4-turbo	80%
defog-sqlcoder-15b	78%
defog-sqlcoder-7b	71%
gpt-3.5-turbo	65%
claude-2	65%
claude_instant_1	61%

Uses

This model is intended to be used by non-technical users to understand data inside their SQL databases. It is meant as an analytics tool, and not as a database admin tool.

This model has not been trained to reject malicious requests from users with write access to databases, and should only be used by users with read-only access.

How to Get Started with the Model

Use the code [here](#) to get started with the model.

Evaluation

This model was evaluated on [SQL-Eval](#), a PostgreSQL based evaluation framework developed by Defog for testing and alignment of model capabilities.

You can read more about the methodology behind SQLEval [here](#).

Results

We classified each generated question into one of 6 categories. The table displays the percentage of questions answered correctly by each model, broken down by category.

QUANTIZATION of LLM Model Weights



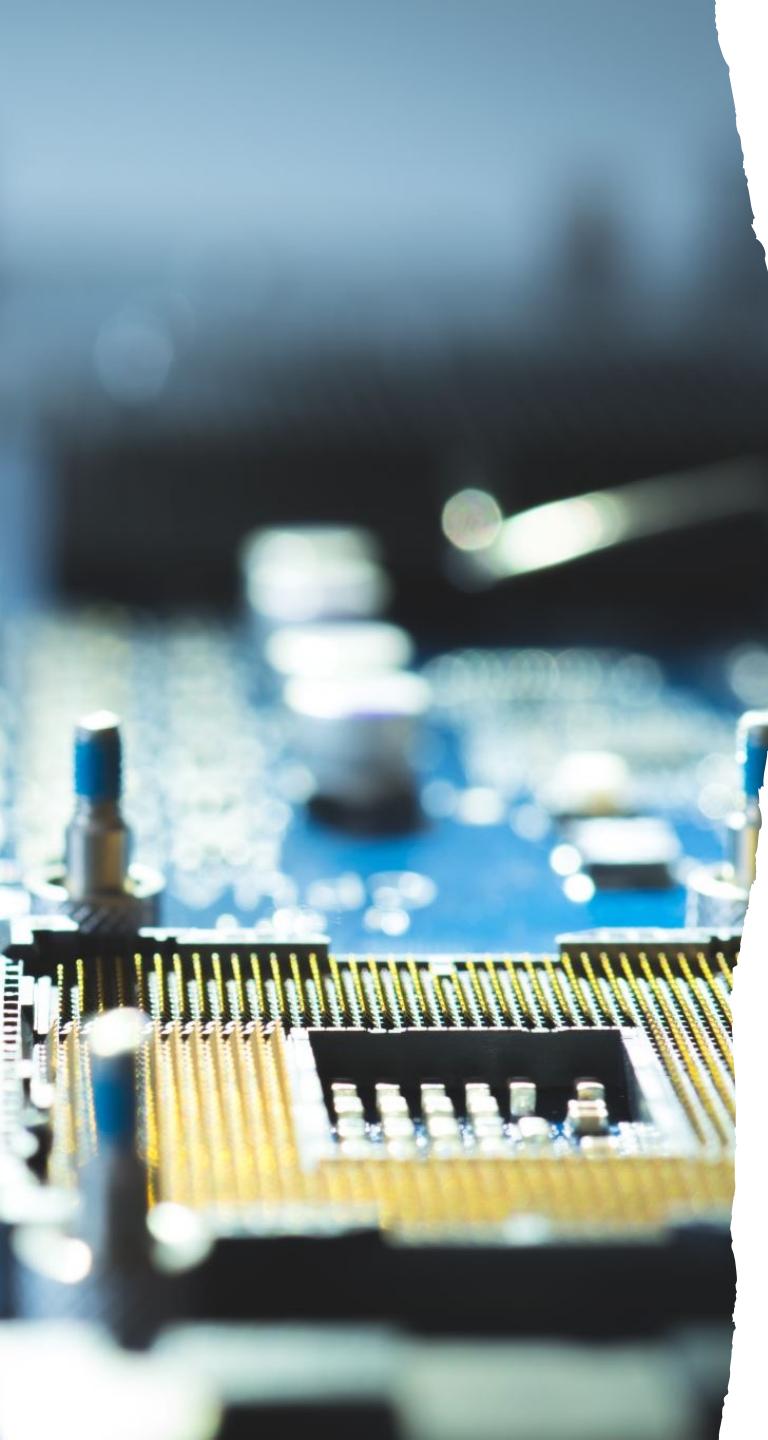
Quantization, is a technique aimed at optimizing model performance without significantly sacrificing accuracy or the quality of generated content.



Similar to data compression and optimization ... quantization is akin to compressing a high-quality image to reduce its file size while trying to keep the visual quality as intact as possible.

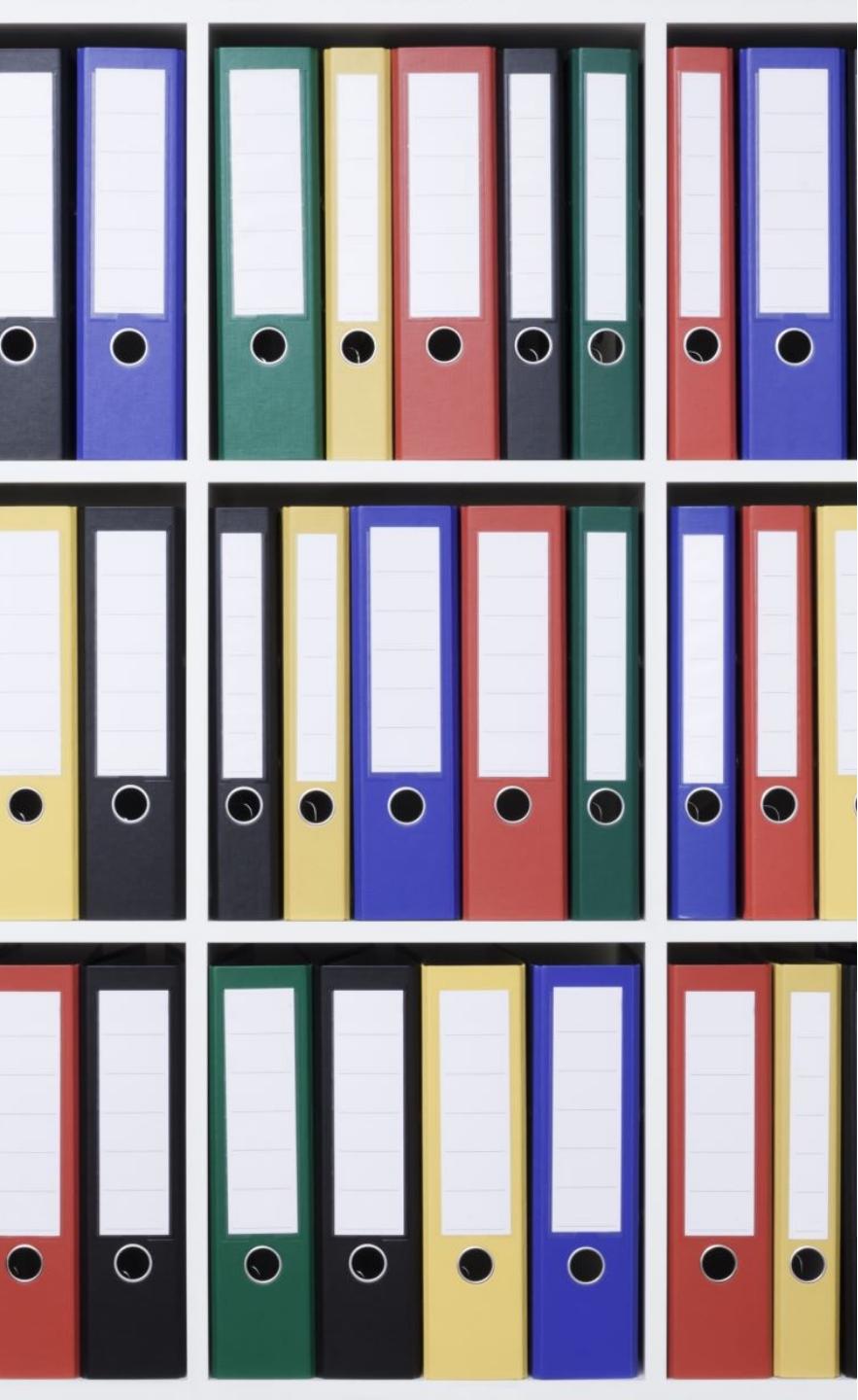


Like image software processing JPGs, you can have a sliding scale to trade size vs. quality.



Quantization – Basic Concepts

- **Quantization in AI/ML:** We reduce the precision of the numbers used to represent model weights.
 - Switching from 32-bit floating-point numbers to 16-bit integers cuts memory use in half.
 - This reduction in precision also can significantly speed up computation, especially on hardware optimized for lower-precision arithmetic.
- **Model Efficiency:** Quantization allows these models to be deployed more efficiently, enabling faster response times and lower resource consumption, which is crucial for applications requiring real-time performance or running on limited hardware resources.
- **Trade-Offs:** The challenge with quantization lies in the trade-off between performance and model quality. Reducing the precision of weights too much can lead to a loss in the model's ability to generate high-quality outputs or maintain accuracy in tasks. Careful implementation and testing can minimize these effects.



Types of Models – For different Architectures

- **Safetensors and PyTorch** bin files are examples of raw float16 model files. These files are primarily utilized for continued fine-tuning purposes. Very large (2x # of Parameters bytes).
- **GGML/GGUF** - allows for better support of multiple quantization sizes and includes prompt templates. GGUF can be executed solely on a CPU or partially/fully offloaded to a GPU. By utilizing K quants, the GGUF can range from 2 bits to 8 bits. (1/2 to 1/8 the size of the raw float16 files) – REQUIRES LLAMA.CPP Library.
- **GPTQ/AWQ/EXL2** - GPTQ served as a GPU-only optimized quantization method. However, it has been surpassed by AWQ, which is approximately twice as fast. The latest advancement in this area is EXL2, which offers even better performance. Typically, these quantization methods are implemented using 4 bits. However the most important parameters (1%) remain full-precision. (1/4 the size of raw float16)

Quantization - Practical Impacts

- **Deployment Flexibility:**

- With quantization, Models can be deployed in a wider range of environments, from high-end servers to edge devices, without requiring extensive computational resources. This expands the potential applications of such models in real-world scenarios, from cloud-based solutions to on-device AI applications.

- **Cost Efficiency:**

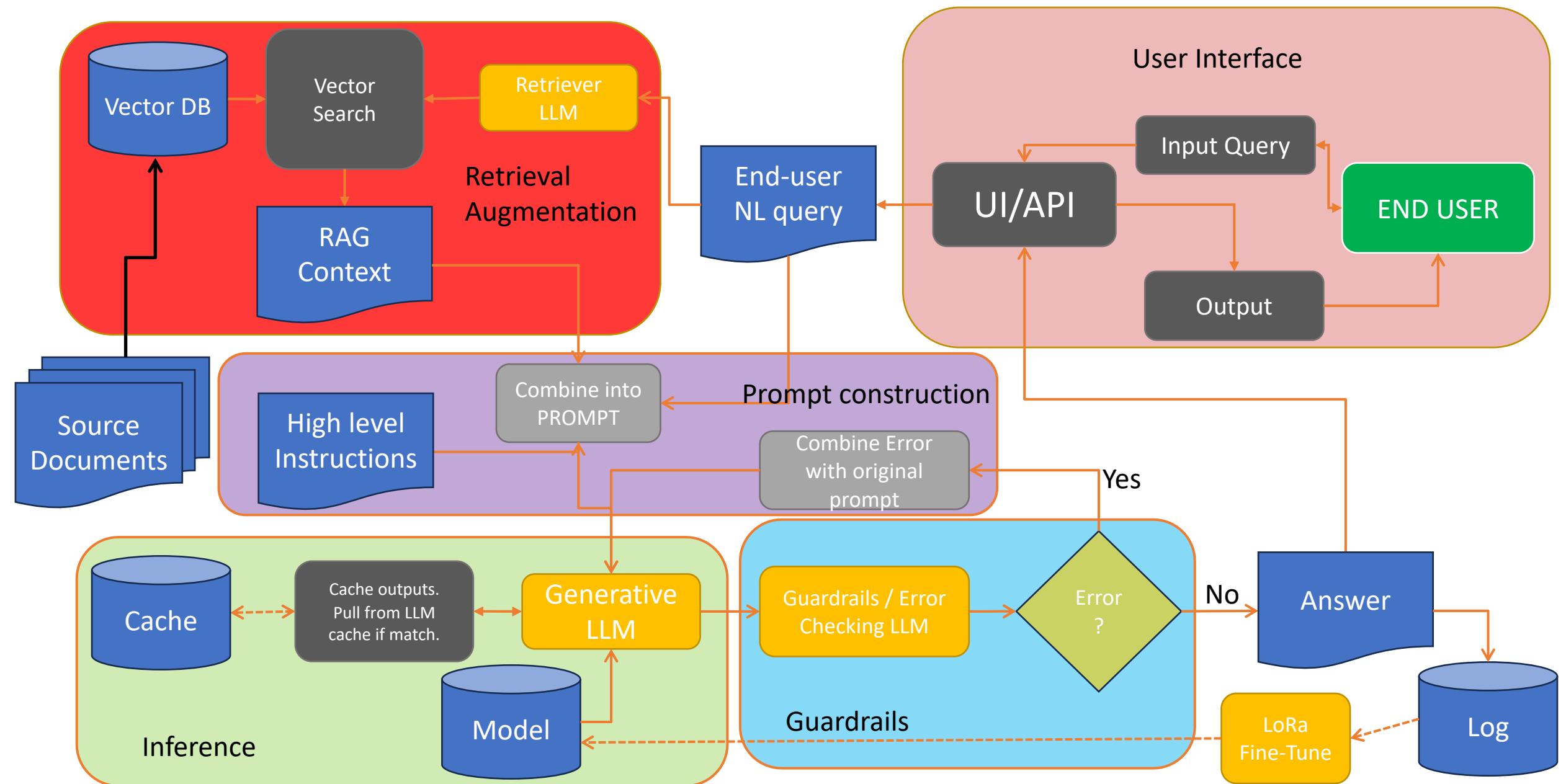
- By reducing the computational load and storage requirements, quantization helps lower the costs associated with running large-scale AI models, making advanced AI technologies more accessible to a broader range of users and applications.

- **In summary,**

- Quantization is a critical optimization technique
- Balances the need for computational efficiency with the desire to maintain the quality of AI-generated outputs.
- Key tool in making state-of-the-art AI models more practical and accessible for real-world applications, especially in environments where computing resources are limited or cost efficiency is a priority.



A RAG Architecture





Colab – RAG Expert

- Steps
 - Instantiate LLM, get a basic chatbot working via API
 - Ingest, Vectorize and store source documents in DB
 - User Query: Retrieve most relevant chunks based on query
 - Combine document chunks + Prompt to form context
 - Feed everything to the LLM
 - Return Result

Let's go to Colab

<https://github.com/tniccum21/DMZ/>

The screenshot shows a GitHub repository named 'DMZ'. The repository is public, as indicated by the 'Public' badge. It has 1 branch and 0 tags. There are 2 commits from the user 'tniccum21' made 2 minutes ago, both updating the 'README.md' file. The repository contains a single file, 'README', which is currently being edited. The content of the README is as follows:

```
DMZ

Colab links for DMZ class

1. One Neuron: https://colab.research.google.com/drive/1fJdtxd3x71wKDgHF0ldUFMd6T7e4Lt-G?usp=sharing
2. Image Classification: https://colab.research.google.com/drive/14maur7VUXh4WBIVghXIQ6et9NDEYts0Z?usp=sharing
3. Vito: https://colab.research.google.com/drive/1qArAXlvaEM001L\_99sTn8ePUSNnlhdRR?usp=sharing
4. LLM-Part-1: https://colab.research.google.com/drive/1sbfcampC9WrOMiJ1r2mC1tzcClqmUN3N?usp=sharing
5. LLM-Part-2: https://colab.research.google.com/drive/1uWTt2azpfkj9AMzCKeT\_2lKMghPATle?usp=sharing
6. LLM-Part-3: https://colab.research.google.com/drive/10o3vh7fmHmeODidK\_GVmWGD6Vgc1U5O?usp=sharing
7. Fine-Tuning: https://colab.research.google.com/drive/1QfUMocrTLmZI0vE9cHnx1JAEif8WNrV?usp=sharing
```



Today's Journey

- Intro/Agenda/Boring Stuff
- Tools – Getting set up
- A quick example - ViTO
- Retrieval Augmented Generation (RAG)
- Building a RAG-Enabled E^T
- Detour – Local Models
- A more sophisticated App
- Example of Fine Tuning
- Q&A





What is ‘Fine-Tuning’

- **LLMs are expensive to train:**
 - Large Language Models (LLMs) like GPT (Generative Pre-trained Transformer) are AI systems trained on vast amounts of text data. They can generate text, answer questions, and perform tasks related to language understanding and generation. But they can cost millions of dollars to train. GPT4 has about 1.8Trillion parameters... Rumors are that it cost >\$20M to train.
- **What is Fine-tuning?:**
 - Fine-tuning is a method to specialize a pre-trained AI model on a narrower task or data set. It's akin to adding a specialized course to a broadly educated student's curriculum to give them expertise in a specific field.
- **Why Fine-tune?:**
 - **Improved Performance:** Tailoring the model to specific needs or datasets enhances its accuracy and relevance for those tasks.
 - **Customization:** Enables customization of the model's responses, making it better suited for particular industries, jargon, or languages.
 - **Efficiency:** Fine-tuning requires less data and computing resources than training a model from scratch, leveraging the LLM's existing knowledge base.



Applications of Fine Tuning

- **Industry-Specific Needs:** Tailoring models for specific sectors like healthcare, legal, or financial services, by focusing on their unique vocabulary and requirements.
- **Language and Localization:** Adapting the model to understand and generate text in specific languages or dialects not well-represented in the original training data.
- **Task-Specific Models:** Creating versions of the model that excel at particular tasks, such as summarization, translation, or content generation with a specific style or tone.



LoRA - Low-Rank Adaptation

- LoRA uses a novel approach to modifying pre-trained models. Instead of fine-tuning all parameters of a model, LoRA focuses on adapting a small subset of the model's weights. This method enables significant customization with minimal changes to the model's architecture.
- LoRA identifies key parameters within a model (typically attention and feed-forward layers) and introduces trainable low-rank matrices that adapt these parameters for specific tasks.
- Adjusting only these low-rank matrices instead of the entire weight matrices, LoRA achieves fine-tuning with far fewer trainable parameters. This approach drastically reduces the computational cost and memory footprint of model adaptation.
- Useful for creating models tailored to specific industries or fields, such as legal, medical, or technical domains, where understanding of specialized terminology and concepts is crucial.

LoRA uses Matrix decomposition...

You can multiply two small matrices to get a larger matrix.

$$\begin{matrix} 1 \\ 3 \\ 7 \\ -4 \\ 2 \end{matrix} \times \begin{matrix} 5 & 1 & -1 & 3 & 4 \end{matrix} = \begin{matrix} 5 & 1 & -1 & 3 & 4 \\ 15 & 3 & -3 & 9 & 12 \\ 35 & 7 & -7 & 21 & 28 \\ -20 & -4 & 4 & -12 & -16 \\ 10 & 2 & -2 & 6 & 8 \end{matrix}$$

Here, we multiply a 1×5 matrix by a 5×1 matrix to get a 5×5 matrix out

So, We can “decompose” a Large Matrix

5	1	-1	3	4
15	3	-3	9	12
35	7	-7	21	28
-20	-4	4	-12	-16
10	2	-2	6	8

=

?
?
?
?
?

x

?	?	?	?	?
---	---	---	---	---

When multiplied, the two small matrices get close in value to the original.

The Large Matrix has 25 values, but the two small have 10 combined...

As Models Get Larger LoRA really pays off!

# Total Parameters	Full Matrix Dimensions	Parameters in Decomposed Matrices (Rank 1)	Relative Number of Values
25	5x5	10	40%
100	10x10	20	20%
2.5k	50x50	100	4%
1M	1k x 1k	2k	0.2%
13B	114k x 114k	228k	0.001%



Colab – Fine Tuning

- Steps
 - Prepare data into training format
 - Split into Training and Test batches
 - Submit to OpenAI for fine tuning run
 - Return fine-tuned model
 - Test

Let's go to Colab

<https://github.com/tniccum21/DMZ/>

The screenshot shows a GitHub repository named 'DMZ'. The repository is public, as indicated by the 'Public' badge. It has 1 branch and 0 tags. There are 2 commits from the user 'tniccum21' made 2 minutes ago, both updating the 'README.md' file. The repository contains a single file, 'README', which is currently being edited. The content of the README is as follows:

```
DMZ

Colab links for DMZ class

1. One Neuron: https://colab.research.google.com/drive/1fJdtxd3x71wKDgHF0ldUFMd6T7e4Lt-G?usp=sharing
2. Image Classification: https://colab.research.google.com/drive/14maur7VUXh4WBIVghXIQ6et9NDEYts0Z?usp=sharing
3. Vito: https://colab.research.google.com/drive/1qArAXlvaEM001L\_99sTn8ePUSNnlhdRR?usp=sharing
4. LLM-Part-1: https://colab.research.google.com/drive/1sbfcampC9WrOMiJ1r2mC1tzcClqmUN3N?usp=sharing
5. LLM-Part-2: https://colab.research.google.com/drive/1uWTt2azpfkj9AMzCKeT\_2lKMghPATle?usp=sharing
6. LLM-Part-3: https://colab.research.google.com/drive/10o3vh7fmHmeODidK\_GVmWGD6Vgc1U5O?usp=sharing
7. Fine-Tuning: https://colab.research.google.com/drive/1QfUMocrTLmZI0vE9cHnx1JAEif8WNrV?usp=sharing
```



Today's Journey

- Intro/Agenda/Boring Stuff
- Tools – Getting set up
- A quick example - ViTO
- Retrieval Augmented Generation (RAG)
- Building a RAG-Enabled BOT
- Detour –
- A more so
- Example
- Q&A



Me making sure to type "thank u" to ChatGPT
so it spares my life during the apocalypse

