Mastering AI and LLM Engineering – Resources

Nov 13, 2024 LLM

Agent, AI, AI Engineering, Data, Data curation, Fine-tuning, Gen AI, LLM, LLM Engineering, QLoRA, RAG

This has been one of my most enjoyable experiences of the last year: launching an 8 week course on Udemy on AI and LLM Engineering. As part of the course, we build a number of chunky commercial projects. The final project is a fitting conclusion — we build an autonomous Agentic AI solution that solves a complex business problem. It performs way better than I imagined possible.

Here is a list of useful links and resources to accompany the course.

Repo, Setup and Slides

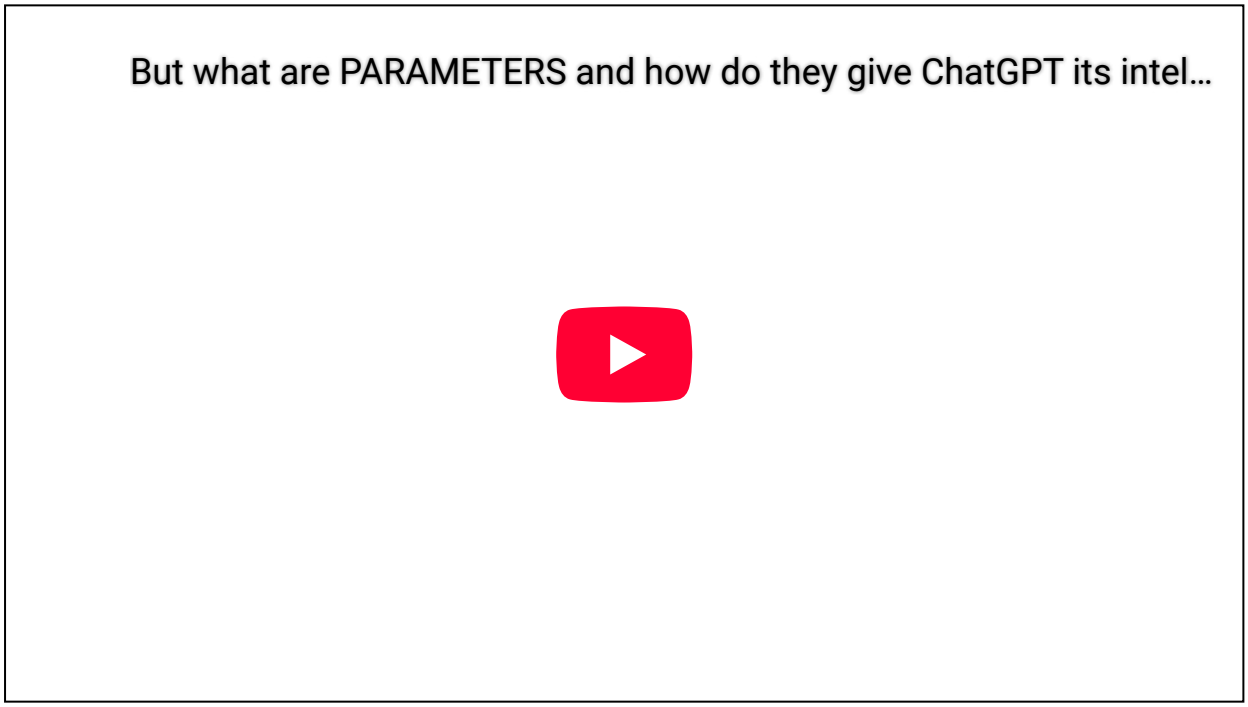
- The Github repo for the course

- The README with setup instructions, and PC and Mac specifics
- All the slides for the course
- My companion courses and how they fit together – if you're interested in going deep on Autonomous Agents, you might consider this as a follow-up course – the projects are equally surprising and amazing!

And if you wish: please connect with me on LinkedIn, follow me on X, and subscribe to me on YouTube! All the multi-modal forms of me 😂

The definitive answer to the most common first question!

People from a non-Data Science background often ask me a great question during Week 1: so what exactly are these "parameters" that we keep hearing about?? I've made this short video to explain what they are, and how they give GPT its super-powers, followed by a peak inside GPT.



Important note: updating your code after each week

I regularly push updates to the labs, including more tips, business applications and exercises. It's worth bringing in the latest code at the start of each week, beginning with Week 2.

First, from the `llm_engineering` project root directory, pull in the latest code from git and merge in any of your changes. Instructions here for those less familiar with git.

Then update your environment to bring in the latest libraries. If you used Anaconda to set up your environment, in an Anaconda window (PC) or Terminal (Mac), run:

```
conda env update --f environment.yml
```

Or if you used virtualenv rather than Anaconda, then run this from your activated environment in a Powershell (PC) or Terminal (Mac):

```
pip install -r requirements.txt
```

Then restart the kernel (Kernel menu >> Restart Kernel and Clear Outputs Of All Cells) to pick up the changes.

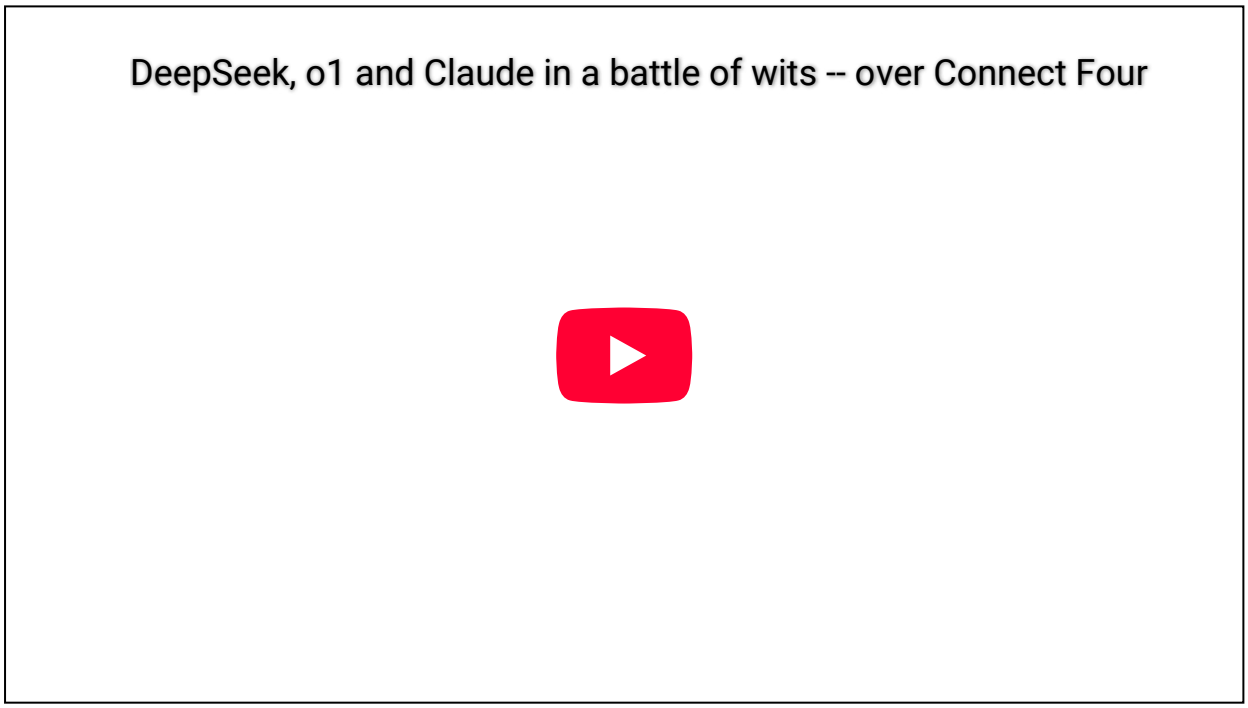
Contributing to the repo

Many students have contributed their own solutions and extensions to the repo. I'm incredibly grateful! I love seeing your progress and innovative ideas, and it adds value for everyone else on the course. As an added benefit, you get recognition in GitHub as a contributor to the repo.

If you're interested in adding your work, please submit a Pull Request and I'll merge as soon as possible. Here are instructions on how to submit a PR. Please make changes in the community-contributions folder only (unless you find a mistake in my code!) and Clear Outputs, as the instructions explain. Let me know if you have any problems, and massive thanks in advance.

Another fun example project

Here's a video extra on a project to have LLMs compete that shows how easy it is to use different Frontier APIs, and the benefits of writing your own lightweight LLM abstraction.

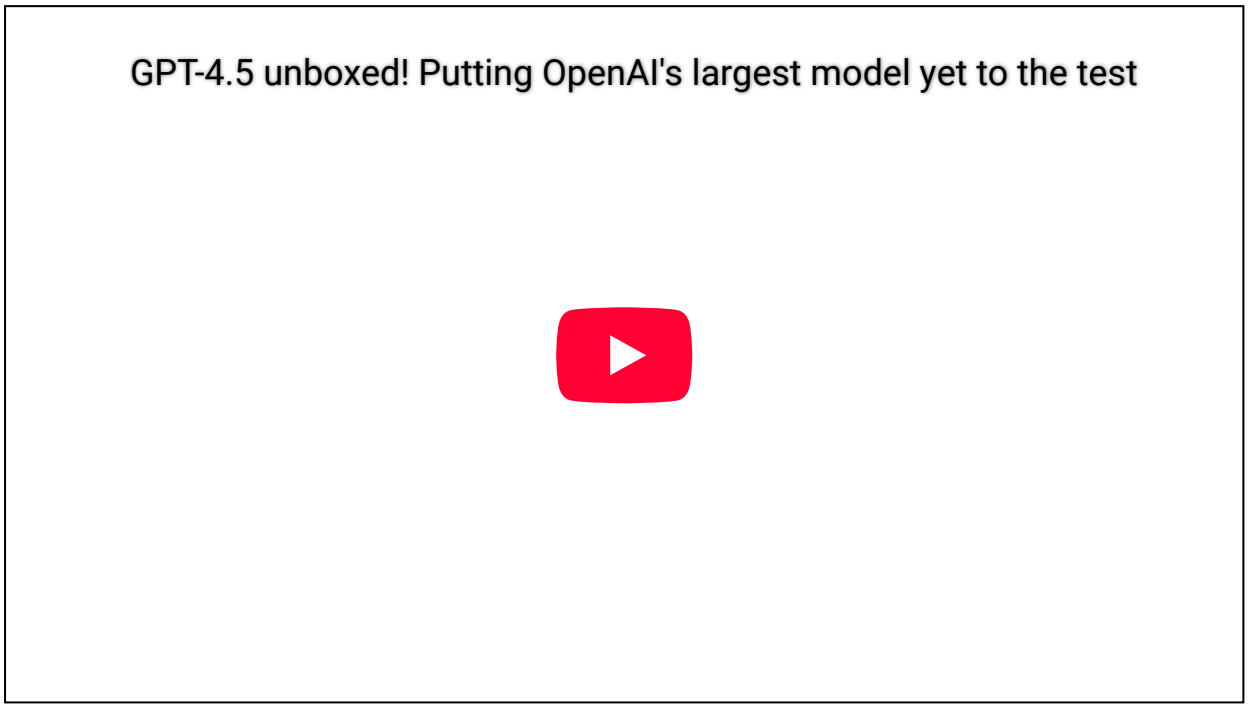


Frontier models – Web interface

1. ChatGPT (latest model GPT-4o and o1) from OpenAI
2. Claude (latest model Claude 3.5 Sonnet) from Anthropic
3. Gemini Advance (latest model Gemini 2.0 Flash) from Google
4. DeepSeek (latest models DeepSeek R1 and V3) from DeepSeek AI
5. Le Chat from French AI powerhouse Mistral

6. Chat with Command R+ from Cohere
7. Meta.ai (model is Llama 3) from Meta
8. Perplexity (latest model is Perplexity Pro) from Perplexity.ai

Here's a review of OpenAI's latest chat model, GPT-4.5:



Frontier models – API

1. GPT API from OpenAI
2. Claude API from Anthropic
3. Gemini API from Google
4. DeepSeek API from DeepSeek AI

Other useful links on models

The seminal 2017 paper 'Attention Is All You Need' from Google scientists that brought about the Transformer is here. This sentence from the Abstract says it all:

> We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely.

The famous paper 'On the Dangers of Stochastic Parrots: Can Language Models Be Too Big?' that discussed bias and deception is here.

The prompt generator from Anthropic is described and linked here.

And here's the Vellum leaderboards including costs and context windows.

The paper on the Chinchilla Scaling Law, describing how the scaling of model parameters is proportional to the size of your training data, can be found here.

Here is the game I made, Outsmart, that pits models against each other in a battle of negotiation.

Common tools used in LLM engineering:

1. Hugging Face – the go-to hub for models, datasets, leaderboards and even applications, and the authors of many essential open source frameworks including the pioneering **transformers** library
2. LangChain – open source framework that provides abstractions connecting multiple LLM operations under a simple API
3. Gradio – a ridiculously simple UI framework that lets you create prototype UIs in one line of code, no frontend experience needed
    - Alternatives include Streamlit, Dash and most recently Mesop from Google
4. Weights & Biases – tooling to analyze and visualize during training
5. Google Colab – write, evaluate and share notebooks remotely on a box in the Google Cloud
    - Amazon SageMaker is a broader alternative that includes Notebooks
6. Modal.com the serverless AI platform

Not covered in this course: using a Managed Service

1. Amazon Bedrock is the managed service from AWS:
   "The easiest way to build and scale generative AI applications with foundation models"
2. Vertex AI is the managed service from Google Cloud:
   "Innovate faster with enterprise-ready AI, enhanced by Gemini models"
3. Azure Machine Learning is the managed service from Microsoft.
   "Build business-critical ML models at scale"

Links to the Google Colabs

You should be able to use the free tier or minimal spend to complete all the projects in the class. I personally signed up for Colab Pro+ and I'm loving it – but it's not required.

Learn about Google Colab and set up a Google account (if you don't already have one) here

The colab links are in the Week folders and also here:

- For week 3 day 1, this Google Colab shows what colab can do
- For week 3 day 2, here is a colab for the HuggingFace pipelines API
- For week 3 day 3, here's the colab on Tokenizers
- For week 3 day 4, we go to a colab with HuggingFace models
- For week 3 day 5, we return to colab to make our Meeting Minutes product
- For week 7, we will use these Colab books: Day 1 | Day 2 | Days 3 and 4 | Day 5

The Leaderboards and Arenas

Student Eloy C. raised an excellent point: I should have given more attention to one of the most common reasons why people select a model for the task: its ability to work with a non-English language. You'll find this on various language specific leaderboards on HuggingFace Open LLM and SEAL below, and you can also bring up the models page in HuggingFace and click the Languages section to filter on models with expertise in particular spoken languages.

- Hugging Face Open LLM
- Hugging Face Big Code
- Hugging Face LLM-Perf
- All Hugging Face leaderboards – medical, Portuguese and more
- Vellum.ai Leaderboard – includes BBHard, also Cost & Context Window comparison
- SEAL specialist leaderboards from Scale.ai
- AlpacaEval
- LM Arena (formerly known as LMSYS Arena) and contribute your votes here
- LiveBench – a hard leaderboard that's resistant to training data leakage

HuggingFace has a summary of all the metrics on the OpenLLM leaderboard with links to the original papers that is worth bookmarking! And below is a table with a selection of the earlier, common benchmarks that pre-date the new metrics.

Benchmark	Area	Description
LMSYS ELO	Chatbot Arena ranking	Crowdsourced votes from humans choosing between 2 models
ARC	A12 Reasoning Challenge	A benchmark for evaluating scientific reasoning with multiple-choice science questions.
DROP	Language comprehension	Ability to pull important details from English text and then perform distinct reasoning actions, such as adding, sorting or counting items
HellaSwag	Common sense	"Harder Endings, Longer contexts, and Low-shot Activities for Situations With Adversarial Generations"
MMLU	Massive Multitask Language Understanding	Covers 57 subjects and assesses many times of questions and tasks, from factual recall to reasoning and problem solving.
TruthfulQA	Accurate, truthful responses	Evaluates the robustness of the model in providing truthful responses under difficult conditions.
Winogrande	Understand context	A pronoun resolution task that tests a model's ability to understand context and resolve ambiguous references.
GSM8K	Grade School Math 8K	Reflects math and word problems taught in elementary and middle schools; calculations and basic mathematical concepts.
MGSM	Multi-lingual GSM8K	A test of around 250 arithmetic problems in multiple languages, testing that the LLM can understand the problem and explain its reasoning.
HumanEval	Python Coding	Functional correctness for coding from docstrings. 164 Python problems.
MultiPL-E	18 Programming Languages	Translation of HumanEval more broadly to 18 languages
GPQA	Graduate-Level Google-Proof Q&A	448 challenging questions by experts in biology, physics and chemistry. PhD level experts get 65%, non-experts get 34% even with web access.
MATH	Competition math	12,500 challenging competition math problems
BBHard	Probing for Future Capabilities	204 tasks that are believed to be beyond the capabilities of current language models across diverse topics

Real-world examples of LLMs making commercial impact

- Harvey.ai – Law
- Nebula.io – Talent (where I work – we do great things!)
- Bloop.ai – Tech (porting legacy code)
- Einstein Copilot: Health – Healthcare
- Khanmigo – Education

Extra – Robotics Links

Humanoid Robotics:

- Phoenix from Sanctuary
- Figure 01 from Figure

Rototics Models and Frameworks:

- GROOT from Nvidia
- RFM1 – 8B parameter LLM for Robotics from Covariant
- LeRobot framework from Hugging Face

Recreating the Robotics Dataset Visualization:

See the LeRobot GitHub repo here and follow their setup instructions:

```
git clone https://github.com/huggingface/lerobot.git && cd lerobot

conda create -y -n lerobot python=3.10 && conda activate lerobot

pip install .

pip install ".[aloha, pusht]"
```

And then to visualize the dataset of the Aloha-Mobile robot cooking a shrimp, run this:

```
python lerobot/scripts/visualize_dataset.py --repo-id lerobot/aloha_mobi
```

The Extra Extra Project for Fun

I mentioned my experiment to train an LLM on my 240,000 text message history. My write-up of the journey is here, and the subsequent blog posts take you through the adventure of replicating this yourself!

Finally

Somehow you made it all the way to the end of the resources — thank you! If you're not completely fed up with me by now, then please connect with me on LinkedIn! I'd love to stay in touch and I'm always available if you have feedback, questions or ideas.

← Previous

Next →

Leave a Reply