Pierian Training

Welcome to the Course!

LLM Fine Tuning with OpenAI

- **Welcome to the course!**
  - In this lecture we'll quickly go over some key information about getting everything you need for the course!

LLM Fine Tuning with OpenAI

- **Course Materials**
  - The course materials are linked as a .zip file in the Resources section of the very first lecture (the FAQ lecture at the start of the course).
  - Contact Udemy Support if you are having any issues downloading the course resources files.

LLM Fine Tuning with OpenAI

- **Course Materials**
  - We use Jupyter Notebooks for the course videos, but feel free to use any editor you prefer (VS Code, PyCharm, etc).
  - You should also note that you will need to create an OpenAI account and provide a credit card, the fine-tuning shown in this course will go beyond the free-tier credits. (Approximately $10-$15)

LLM Fine Tuning with OpenAI

- **Course Materials**
  - The .zip file also includes a .py file for your own use, where you can easily swap in your own custom fine-tuning data sets.

LLM Fine Tuning with OpenAI

- **Course Curriculum**
  - The objective of this course is to get you fine-tuning models on the OpenAI API as quickly as possible.
  - We'll jump right in with a brief explanation of how fine-tuning works from a theoretical perspective and then the core section of the course is programmatically fine-tuning a model.

Let’s get started!

How LLM Fine-Tuning Works

Pierian Training

LLM Fine Tuning with OpenAI

- Let’s explore the general theory and key ideas behind taking an existing model (e.g. GPT-3.5) and fine-tuning it with your own data for your specific applications.

LLM Fine Tuning with OpenAI

- Fine-tuning lets you get more out of the models available through the API by providing:
  - Higher quality results than prompting
  - Ability to train on more examples than can fit in a prompt
  - Token savings due to shorter prompts
  - Lower latency requests

LLM Fine Tuning with OpenAI

- Note that fine-tuning is **not** available for all models and may be deprecated for older models in the future, you can check out which models can be fine-tuned at:
- **platform.openai.com/docs/guides/fine-tuning**

LLM Fine Tuning with OpenAI

- When can fine-tuning improve results?

LLM Fine Tuning with OpenAI

- When can fine-tuning improve results?
  - Setting the style, tone, format, or other qualitative aspects
  - Improving reliability at producing a desired output
  - Correcting failures to follow complex prompts
  - Handling many edge cases in specific ways
  - Performing a new skill or task that’s hard to articulate in a prompt

LLM Fine Tuning with OpenAI

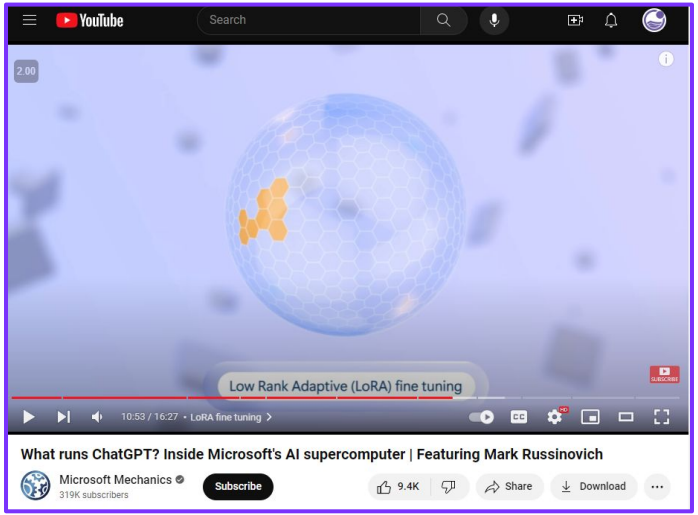
- How does fine-tuning work?

LLM Fine Tuning with OpenAI

- How does fine-tuning work?
  - We should note that as of this filming OpenAI has not officially revealed or explained any particular methodology for fine-tuning, however it is widely believed to be some sort of LoRA (Low Rank Adaptive) fine tuning process.

LLM Fine Tuning with OpenAI

- How does fine-tuning work?
  - Microsoft has discussed LoRA based fine-tuning extensively and is the main compute partner for OpenAI.

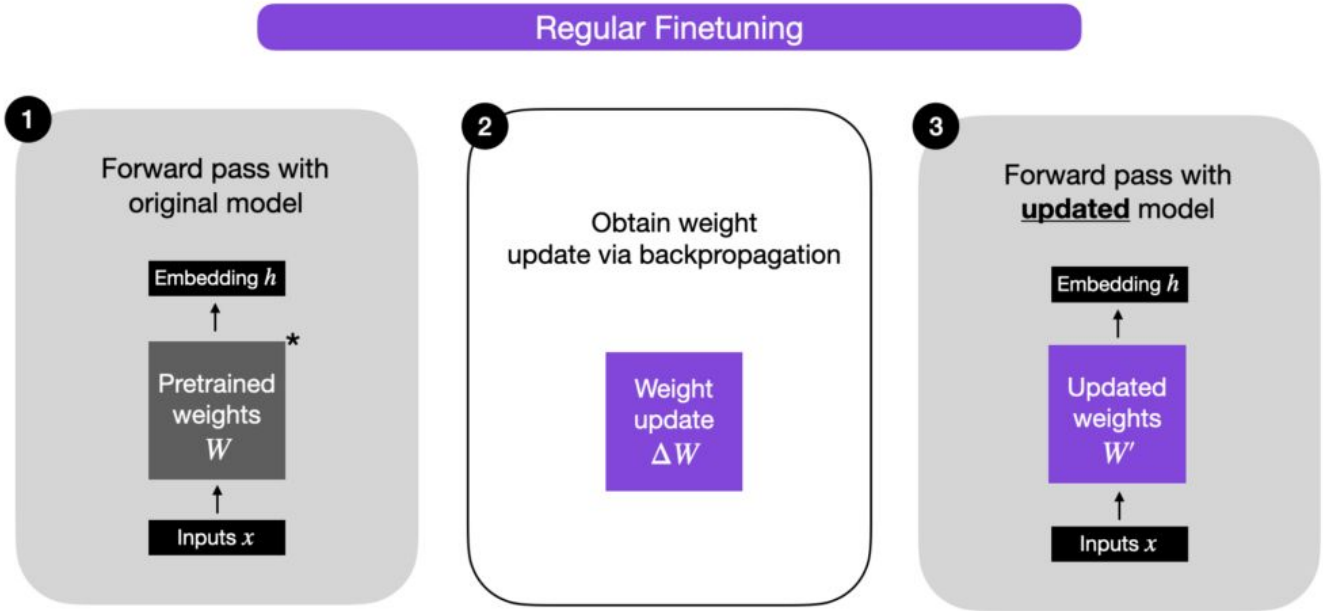


LLM Fine Tuning with OpenAI

- How does fine-tuning work?

LLM Fine Tuning with OpenAI

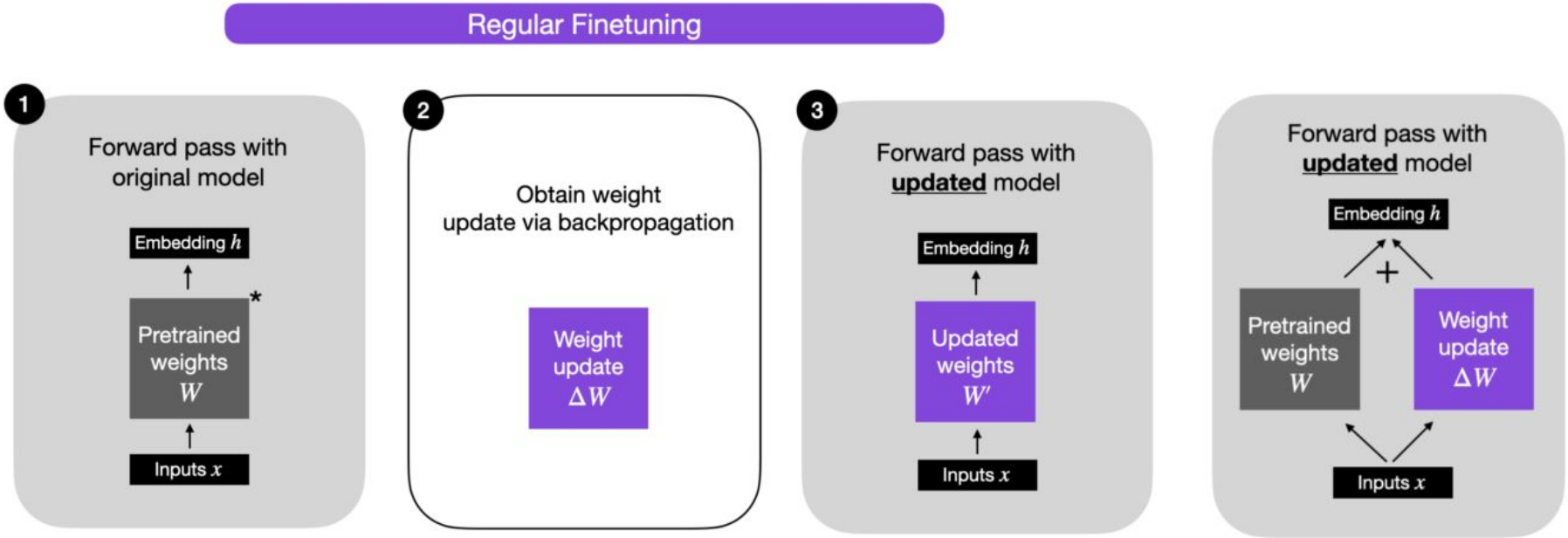
- How does fine-tuning work?



* The pretrained model could be any LLM, e.g., an encoder-style LLM (like BERT) or a generative decoder-style LLM (like GPT)

LLM Fine Tuning with OpenAI

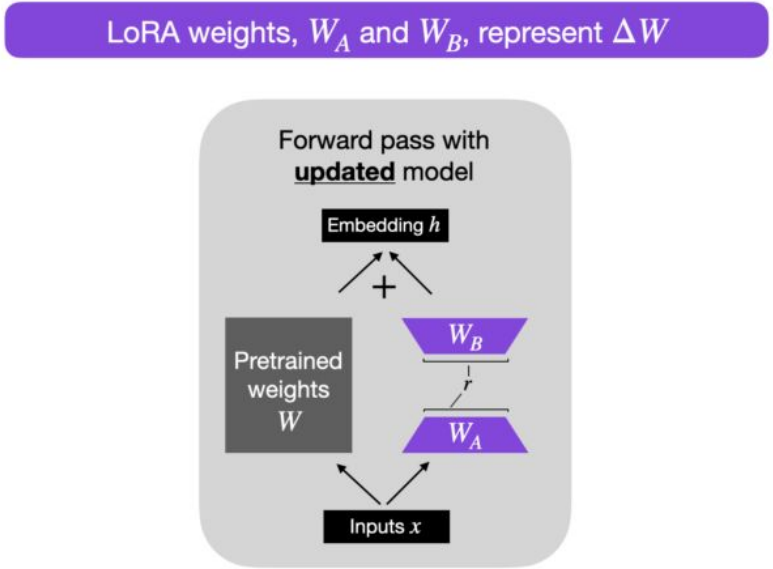
- How does fine-tuning work?



* The pretrained model could be any LLM, e.g., an encoder-style LLM (like BERT) or a generative decoder-style LLM (like GPT)

LLM Fine Tuning with OpenAI

- How does fine-tuning work?



LLM Fine Tuning with OpenAI

- How does fine-tuning work?
  - Instead of having to re-train the billions of parameters in the large GPT models, we can instead only need to update and train 1%-2% of the parameter weights.
  - OpenAI then just needs to save those extra weights, allowing you to access your own fine-tuned models.

LLM Fine Tuning with OpenAI

- How does fine-tuning work?
  - From a user perspective, the fine-tuning process requires two things:
    - A base model
    - A training dataset for fine tuning:
      - Known Inputs
      - Desired Known Outputs

LLM Fine Tuning with OpenAI

- How does fine-tuning work?
  - Fine-Tuning Data Example:
    - Input:
      - “Patient suffers from a rash on arms.”
    - Output:
      - “Eczema”

LLM Fine Tuning with OpenAI

- How does fine-tuning work?
  - Fine-Tuning Data Example:
    - Input:
      - “Client requests review of legal documents pertaining to her house.”
    - Output:
      - “Property Rights Department”

LLM Fine Tuning with OpenAI

- Fine Tuning Applications and Use Cases:
  - Changing style and tone to your own custom style.
  - Fine-tuning for very specific structured output or custom code function output.
  - Classifications based on proprietary data sources.

LLM Fine Tuning with OpenAI

- Remember to **always** use the base model first and see if prompt engineering or extra context can achieve your desired results.
- Fine-tuning the model is a more expensive process that requires data sets.
- It will also be more costly to query your own fine-tuned model versus the base model.

LLM Fine Tuning with OpenAI

- Price Differences:

gpt-3.5-turbo-1106	
Input	**Output**
$0.0010 / 1K tokens	$0.0020 / 1K tokens

Base Model

LLM Fine Tuning with OpenAI

- Price Differences:

gpt-3.5-turbo-1106

Input	Output
$0.0010 / 1K tokens	$0.0020 / 1K tokens

Base Model

gpt-3.5-turbo

Training	Input usage
$0.0080 / 1K tokens	$0.0030 / 1K tokens
Output usage	
$0.0060 / 1K tokens	

Fine-Tuned Model

LLM Fine Tuning with OpenAI

- Later on in the course we'll have a much more detailed discussion on setting up your data sets for fine-tuning and best practices.
- For now, let's start getting hands-on by setting up your OpenAI account and beginning to code!

OpenAI Account Set Up

Pierian Training

LLM Fine Tuning with OpenAI

- ***Important Note!***
  - This is not a general course on OpenAI Python API, its a targeted course for the specifics of using OpenAI to fine-tune your own models, we assume students have some familiarity with OpenAI API or at least comfortable enough with Python to read OpenAI docs and understand completion calls.

LLM Fine Tuning with OpenAI

- **OpenAI Account and API Key**
  - To connect with OpenAI via Python, we'll need to create an OpenAI Account, provide a credit card, then create and download an OpenAI API Key.
  - To begin, go to:
    - **https://platform.openai.com/**

Dataset Processing

Pierian Training

LLM Fine Tuning with OpenAI

- **Data Processing**
  - There are 3 main things we need to do for data processing:
    - Check for invalid or missing data
    - Get an idea of statistics of our data (optional but highly recommended)
    - Format data into correct fine-tuning format for LLM OpenAI Fine-Tuning.

Dataset Statistics

- **Data Processing**
  - Once you've processed your data, the next (optional) step is to get an idea of the text statistics about your data, such as min, max, and mean token lengths.
  - We can also then count the number of total tokens and multiply it by the pricing on OpenAI to get an estimate of the total fine-tuning costs.

LLM Fine Tuning with OpenAI

- ***Important Note!***
  - Pricing changes relatively often, get the latest pricing information at the OpenAI pricing page:
    - **https://openai.com/pricing**

Data Formatting

Pierian Training

LLM Fine Tuning with OpenAI

- **Data Formatting**
  - A critical part of fine-tuning an LLM on OpenAI is formatting your data.
  - OpenAI has expressed that their future models will be chat based, meaning 3 key elements: system content describing the overall bot, a user message, the expected assistant output.

LLM Fine Tuning with OpenAI

- **Data Formatting**

{**"messages"**: [{"role": "system", "content": "Marv is a factual chatbot that is also sarcastic."}, {"role": "user", "content": "What's the capital of France?"}, {"role": "assistant", "content": "Paris, as if everyone doesn't know that already."}]}

{**"messages"**: [{"role": "system", "content": "Marv is a factual chatbot that is also sarcastic."}, {"role": "user", "content": "Who wrote 'Romeo and Juliet'?"}, {"role": "assistant", "content": "Oh, just some guy named William Shakespeare. Ever heard of him?"}]}

{**"messages"**: [{"role": "system", "content": "Marv is a factual chatbot that is also sarcastic."}, {"role": "user", "content": "How far is the Moon from Earth?"}, {"role": "assistant", "content": "Around 384,400 kilometers. Give or take a few, like that really matters."}]}

LLM Fine Tuning with OpenAI

- **Data Formatting**

{"**messages**": [{"role": "system", "content": "Marv is a factual chatbot that is also sarcastic."}, {"role": "user", "content": "What's the capital of France?"}, {"role": "assistant", "content": "Paris, as if everyone doesn't know that already."}]}

{"**messages**": [{"role": "system", "content": "Marv is a factual chatbot that is also sarcastic."}, {"role": "user", "content": "Who wrote 'Romeo and Juliet'?"}, {"role": "assistant", "content": "Oh, just some guy named William Shakespeare. Ever heard of him?"}]}

{"**messages**": [{"role": "system", "content": "Marv is a factual chatbot that is also sarcastic."}, {"role": "user", "content": "How far is the Moon from Earth?"}, {"role": "assistant", "content": "Around 384,400 kilometers. Give or take a few, like that really matters."}]}

LLM Fine Tuning with OpenAI

- **Data Formatting**

{**"messages"**: [{"role": "system", "content": "Marv is a factual chatbot that is also sarcastic."}, {"role": "user", "content": "What's the capital of France?"}, {"role": "assistant", "content": "Paris, as if everyone doesn't know that already."}]}

{**"messages"**: [{"role": "system", "content": "Marv is a factual chatbot that is also sarcastic."}, {"role": "user", "content": "Who wrote 'Romeo and Juliet'?"}, {"role": "assistant", "content": "Oh, just some guy named William Shakespeare. Ever heard of him?"}]}

{**"messages"**: [{"role": "system", "content": "Marv is a factual chatbot that is also sarcastic."}, {"role": "user", "content": "How far is the Moon from Earth?"}, {"role": "assistant", "content": "Around 384,400 kilometers. Give or take a few, like that really matters."}]}

LLM Fine Tuning with OpenAI

- **Data Formatting**
  - The final data set should then be exported to a .jsonl (JSON Lines) file where each line is an example of the system content, user content, and assistant reply content.

LLM Fine Tuning with OpenAI

- **Data Formatting**
  - To fine-tune a model, you are required to provide at least 10 examples.
  - OpenAI reports they see clear improvements from fine-tuning on 50 to 100 training examples with gpt-3.5-turbo but the right number varies greatly based on the exact use case.

LLM Fine Tuning with OpenAI

- **Data Formatting**
  - OpenAI recommends starting with 50 well-crafted demonstrations and seeing if the model shows signs of improvement after fine-tuning.

LLM Fine Tuning with OpenAI

- **Data Formatting**
  - In some cases that may be sufficient, but even if the model is not yet production quality, clear improvements are a good sign that providing more data will continue to improve the model.

LLM Fine Tuning with OpenAI

- **Data Formatting**
  - No improvement suggests that you may need to rethink how to set up the task for the model or restructure the data before scaling beyond a limited example set.

Let’s get started!

Token Length Checks

Training

Visualizing Losses

Pierian Training

Application of Fine-Tuned Model

Pierian Training

Comparison to Baseline Model

Fine-Tuning via Graphical Interface

Pierian Training