Fine Tuning in Practice

Activate Python Environment and Set API Key

```
dingrui@LAPTOP-MUAISB9U:/mnt/c/sfbu/cs589/week10$ python -m venv venv


dingrui@LAPTOP-MUAISB9U:/mnt/c/sfbu/cs589/week10$
dingrui@LAPTOP-MUAISB9U:/mnt/c/sfbu/cs589/week10$
dingrui@LAPTOP-MUAISB9U:/mnt/c/sfbu/cs589/week10$
dingrui@LAPTOP-MUAISB9U:/mnt/c/sfbu/cs589/week10$
dingrui@LAPTOP-MUAISB9U:/mnt/c/sfbu/cs589/week10$ ls
venv
dingrui@LAPTOP-MUAISB9U:/mnt/c/sfbu/cs589/week10$ . venv/bin/activate
(venv) dingrui@LAPTOP-MUAISB9U:/mnt/c/sfbu/cs589/week10$
(venv) dingrui@LAPTOP-MUAISB9U:/mnt/c/sfbu/cs589/week10$ ls
homework1  homework2  homework3  homework4  venv
(venv) dingrui@LAPTOP-MUAISB9U:/mnt/c/sfbu/cs589/week10$ cd homework
-bash: cd: homework: No such file or directory
(venv) dingrui@LAPTOP-MUAISB9U:/mnt/c/sfbu/cs589/week10$ cd homework1
(venv) dingrui@LAPTOP-MUAISB9U:/mnt/c/sfbu/cs589/week10/homework1$ cat ../../week6/homework1/.env
API_KEY=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
ORG_ID=org-A00vXLRbnib0TuY2mcN7PB5A
GOOG_API_KEY=AIzaSyDR9QgS1yJUR1asJ7gueltp5fx4LlRN57U
(venv) dingrui@LAPTOP-MUAISB9U:/mnt/c/sfbu/cs589/week10/homework1$ export OPENAI_API_KEY="sk-MmlVU8BfGolMAUpE79sxT3BlbkF
JXSsNuMT9yaqeR4jIGwKY"
(venv) dingrui@LAPTOP-MUAISB9U:/mnt/c/sfbu/cs589/week10/homework1$
```

Create JSONL File

- Create a JSONL file, data.json with prompt-completion pairs.
    - JSONL format is newline-delimited JSON that is suitable
      for training data.
- We are using 3 examples in our JSONL file, so the resulting
  model will not perform well, but we are testing things out and we
  want to see how it works in practice for now.

```
(venv) dingrui@LAPTOP-MUAISB9U:/mnt/c/sfbu/cs589/week10/homework1$ code .
(venv) dingrui@LAPTOP-MUAISB9U:/mnt/c/sfbu/cs589/week10/homework1$ cat data.json
{"prompt":"When do I have to start the heater?", "completion":"Every day in the morning at 7AM. You should stop it at 2P
M"}
{"prompt":"Where is the garage remote control?", "completion":"Next to the yellow door, on the key ring"}
{"prompt":"Is it necessary to program the scent diffuser every day?", "completion":"The scent diffuser is already progra
mmed, you just need to recharge it when its battery is low"}(venv) dingrui@LAPTOP-MUAISB9U:/mnt/c/sfbu/cs589/week10/home
work1$
```

Analyze and Prepare Data

```
(venv) dingrui@LAPTOP-MUAISB9U:/mnt/c/sfbu/cs589/week10/homework1$ openai tools fine_tunes.prepare_data -f data.json
Analyzing...

- Your JSON file appears to be in a JSONL format. Your file will be converted to JSONL format
- Your file contains 3 prompt-completion pairs. In general, we recommend having at least a few hundred examples. We've found that performance tends to
  linearly increase for every doubling of the number of examples
- All prompts end with suffix `?`
- Your data does not contain a common ending at the end of your completions. Having a common ending string appended to the end of the completion makes
  it clearer to the fine-tuned model where the completion should end. See https://platform.openai.com/docs/guides/fine-tuning/preparing-your-dataset fo
r more detail and examples.
- The completion should start with a whitespace character (` `). This tends to produce better results due to the tokenization we use. See https://plat
form.openai.com/docs/guides/fine-tuning/preparing-your-dataset for more details

Based on the analysis we will perform the following actions:
- [Necessary] Your format `JSON` will be converted to `JSONL`
- [Recommended] Add a suffix ending '\n' to all completions [Y/n]: Y
- [Recommended] Add a whitespace character to the beginning of the completion [Y/n]: Y


Your data will be written to a new JSONL file. Proceed [Y/n]: Y

Wrote modified file to `data_prepared.jsonl`
Feel free to take a look!

Now use that file when fine-tuning:
> openai api fine_tunes.create -t "data_prepared.jsonl"

After you've fine-tuned a model, remember that your prompt has to end with the indicator string `?` for the model to start generating completions, rat
her than continuing with the prompt. Make sure to include `stop=["\n"]` so that the generated texts ends at the expected place.
Once your model starts training, it'll approximately take 2.48 minutes to train a `curie` model, and less for `ada` and `babbage`. Queue will approxim
ately take half an hour per job ahead of you.
(venv) dingrui@LAPTOP-MUAISB9U:/mnt/c/sfbu/cs589/week10/homework1$ cat data_prepared.jsonl
{"prompt":"When do I have to start the heater?","completion":" Every day in the morning at 7AM. You should stop it at 2PM\n"}
{"prompt":"Where is the garage remote control?","completion":" Next to the yellow door, on the key ring\n"}
{"prompt":"Is it necessary to program the scent diffuser every day?","completion":" The scent diffuser is already programmed, you just need to recharg
e it when its battery is low\n"}
```

Fine-Tune the Model

Use the OpenAI CLI tool to initiate the fine-tuning process using the prepared JSONL data file:

```
openai api fine_tunes.create -t "data_prepared.jsonl" -m
curie
```

```
(venv) dingrui@LAPTOP-MUAISB9U:/mnt/c/sfbu/cs589/week10/homework1$ openai api fine_tunes.create -t "data_prepared.jsonl" -m curie
Upload progress: 100%|                                                                    | 417/417 [00:00<00:00, 810kit/s]
Uploaded file from data_prepared.jsonl: file-yl1mmjE2X33RVMASmlRz008f
Created fine-tune: ft-UokKswUWeRQ7VLuy0lMWXgRZ
Streaming events until fine-tuning is complete...

(Ctrl-C will interrupt the stream, but not cancel the fine-tune)
[2023-11-19 20:22:34] Created fine-tune: ft-UokKswUWeRQ7VLuy0lMWXgRZ
[2023-11-19 20:23:00] Fine-tune costs $0.00
[2023-11-19 20:23:01] Fine-tune enqueued. Queue number: 0
[2023-11-19 20:23:02] Fine-tune started
```

Resume Fine-Tuning (if needed)

If the fine-tuning operation is interrupted, you can resume it using this command:

```
openai api fine_tunes.follow -i <YOUR_FINE_TUNE_JOB_ID>
```

I skipped this step because my fine tuning was not interrupted.

List Fine-Tuned Models

List your fine-tuned models using this command:

```
openai api fine_tunes.list
```

```
(venv) dingrui@LAPTOP-MUAISB9U:/mnt/c/sfbu/cs589/week10/homework1$ openai api fine_tunes.list
{
  "object": "list",
  "data": [
    {
      "object": "fine-tune",
      "id": "ft-UokKswUWeRQ7VLuy0lMWXgRZ",
      "hyperparams": {
        "n_epochs": 4,
        "batch_size": 1,
        "prompt_loss_weight": 0.01,
        "learning_rate_multiplier": 0.1
      },
      "organization_id": "org-A00vXLRbnib0TuY2mcN7PB5A",
      "model": "curie",
      "training_files": [
        {
          "object": "file",
          "id": "file-yl1mmjE2X33RVMASmlRz008f",
          "purpose": "fine-tune",
          "filename": "data_prepared.jsonl",
          "bytes": 417,
          "created_at": 1700454153,
          "status": "processed",
          "status_details": null
        }
      ],
      "validation_files": [],
      "result_files": [
        {
          "object": "file",
          "id": "file-BVVRMeQklrOvVWLTz2KR4PPx",
          "purpose": "fine-tune-results",
          "filename": "compiled_results.csv",
          "bytes": 746,
          "created_at": 1700454265,
          "status": "processed",
          "status_details": null
        }
      ],
      "created_at": 1700454154,
```

Use the Fine-Tuned Model

Set the fine-tuned model and use it for completions:

```
export FINE_TUNED_MODEL="<FINE_TUNED_MODEL>"
openai api completions.create -m $FINE_TUNED_MODEL -p
<YOUR_PROMPT>
```

```
(venv) dingrui@LAPTOP-MUAISB9U:/mnt/c/sfbu/cs589/week10/homework1$ export FINE_TUNED_MODEL="curie:ft-personal-2023-11-20-04-24-24"
(venv) dingrui@LAPTOP-MUAISB9U:/mnt/c/sfbu/cs589/week10/homework1$ openai api completions.create -m $FINE_TUNED_MODEL -p "When do I need to start my a
ir conditioner?"
When do I need to start my air conditioner?

1. If you just want a cold mist on your face for 1(venv) dingrui@LAPTOP-MUAISB9U:/mnt/c/sfbu/cs589/week10/homework1$ |
```

Use Fine-Tuned Model in Python

If using Python, use the fine-tuned model as shown below:

```
import openai
        openai.Completion.create(
          model=FINE_TUNED_MODEL,
          prompt=YOUR_PROMPT
          # additional parameters
          # temperature,
          # frequency_penalty,
          # presence_penalty
          # ..etc
        )
```

```
(venv) dingrui@LAPTOP-MUAISB9U:/mnt/c/sfbu/cs589/week10/homework1$ cat test.py
import openai
response = openai.Completion.create(
    model="curie:ft-personal-2023-11-20-04-24-24",
    prompt="When do I need to start my air conditioner?"
    # additional parameters
    # temperature,
    # frequency_penalty,
    # presence_penalty
    # ..etc
)

print(response.choices[0].text)(venv) dingrui@LAPTOP-MUAISB9U:/mnt/c/sfbu/cs589/week10/homework1$ python test.py


There are many ways to build this one. Stainless Steel and Bronze
```

Use Fine-Tuned Model with cURL

Use cURL to interact with the fine-tuned model:

```
curl https://api.openai.com/v1/completions \
        -H "Authorization: Bearer $OPENAI_API_KEY" \
        -H "Content-Type: application/json" \
        -d '{"prompt": YOUR_PROMPT, "model": FINE_TUNED_MODEL}'
```

```
(venv) dingrui@LAPTOP-MUAISB9U:/mnt/c/sfbu/cs589/week10/homework1$ curl https://api.openai.com/v1/completions        -H "Authorization: Bearer $OPENA
I_API_KEY"        -H "Content-Type: application/json"        -d '{"prompt": "When do I need to start my air conditioner?", "model": "curie:ft-person
al-2023-11-20-04-24-24"}'
{
  "id": "cmpl-8MqRZLAIFXIL5Bcfsr5CEiRgjfnpm",
  "object": "text_completion",
  "created": 1700455501,
  "model": "curie:ft-personal-2023-11-20-04-24-24",
  "choices": [
    {
      "text": "\n\nInstall the air conditioner so that the air conditioner outlet is level",
      "index": 0,
      "logprobs": null,
      "finish_reason": "length"
    }
  ],
  "usage": {
    "prompt_tokens": 11,
    "completion_tokens": 16,
    "total_tokens": 27
  }
}
```

Analyze Fine-Tuned Model

Use this command to analyze the fine-tuning results:

```
openai api fine_tunes.results -i <YOUR_FINE_TUNE_JOB_ID>
```

```
(venv) dingrui@LAPTOP-MUAISB9U:/mnt/c/sfbu/cs589/week10/homework1$ openai api fine_tunes.results -i ft-UokKswUWeRQ7VLuy0lMWXgRZ
step,elapsed_tokens,elapsed_examples,training_loss,training_sequence_accuracy,training_token_accuracy
1,33,1,1.7561375378817319,0.0,0.23529411764705882
2,66,2,1.4656687936349773,0.0,0.55
3,91,3,1.682499381750822,0.0,0.36363636363636365
4,124,4,1.305190972024575,0.0,0.55
5,157,5,1.5102434629015624,0.0,0.29411764705882354
6,182,6,1.5153488444536922,0.0,0.45454545454545453
7,207,7,1.4586372093359627,0.0,0.45454545454545453
8,240,8,1.3332806890644133,0.0,0.35294117647058826
9,273,9,1.0225329246313777,0.0,0.65
10,306,10,1.0040450236957985,0.0,0.65
11,339,11,1.2657160077011214,0.0,0.35294117647058826
12,364,12,1.3393244941780964,0.0,0.45454545454545453
13,389,13,1.3393244941780964,0.0,0.45454545454545453
14,422,14,0.9928822490677702,0.0,0.65
```

Add Suffix to Fine-Tuned Model Name

You can add a **suffix** to your **fine-tuned model name** using the **suffix parameter**:

```
openai api fine_tunes.create -t data.jsonl -m <engine> --suffix
"my_model_name"
```

```
(venv) dingrui@LAPTOP-MUAISB9U:/mnt/c/sfbu/cs589/week10/homework1$ openai api fine_tunes.create -t data_prepared.jsonl -m davinci --suffix "ruis_model
_name"
Found potentially duplicated files with name 'data_prepared.jsonl', purpose 'fine-tune' and size 417 bytes
file-yl1mmjE2X33RVMASmlRz008f
Enter file ID to reuse an already uploaded file, or an empty string to upload this file anyway:
Upload progress: 100%|                                        | 417/417 [00:00<00:00, 639kit/s]
Uploaded file from data_prepared.jsonl: file-GYyyGTLIWhJFvALtoyMAmBMt
Created fine-tune: ft-6BEJWUE9LD9sroymLt2s4fe5
Streaming events until fine-tuning is complete...

(Ctrl-C will interrupt the stream, but not cancel the fine-tune)
[2023-11-19 20:51:18] Created fine-tune: ft-6BEJWUE9LD9sroymLt2s4fe5
[2023-11-19 20:51:25] Fine-tune costs $0.01
[2023-11-19 20:51:25] Fine-tune enqueued. Queue number: 0
[2023-11-19 20:51:28] Fine-tune started
```

Delete Fine-Tuned Model

Delete your **fine-tuned model** using these methods:

```
# CLI
openai api models.delete -i <FINE_TUNED_MODEL>


# Python
openai.Model.delete(FINE_TUNED_MODEL)


# cURL
curl -X "DELETE" https://api.openai.com/v1/models/<FINE_TUNED_MODEL>
\
    -H "Authorization: Bearer $OPENAI_API_KEY"
```

```
(venv) dingrui@LAPTOP-MUAISB9U:/mnt/c/sfbu/cs589/week10/homework1$ openai api models.delete -i davinci:ft-personal:ruis-model-name-2023-11-20-04-53-41
{
  "id": "davinci:ft-personal:ruis-model-name-2023-11-20-04-53-41",
  "object": "model",
  "deleted": true
}
(venv) dingrui@LAPTOP-MUAISB9U:/mnt/c/sfbu/cs589/week10/homework1$ openai api models.delete -i "curie:ft-personal-2023-11-20-04-24-24"
{
  "id": "curie:ft-personal-2023-11-20-04-24-24",
  "object": "model",
  "deleted": true
}
(venv) dingrui@LAPTOP-MUAISB9U:/mnt/c/sfbu/cs589/week10/homework1$
```