Fine Tuning in Practice

# Activate Python Environment and Set API Key

A screenshot of a computer program

Description automatically generated

# 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.

A screen shot of a computer code

Description automatically generated

# Analyze and Prepare Data

A screenshot of a computer screen

Description automatically generated

# 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

A screenshot of a computer

Description automatically generated

# 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

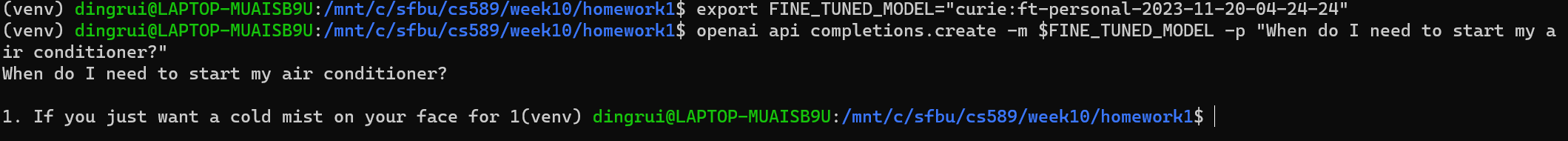
A screenshot of a computer program

Description automatically generated

# 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>



# 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

)

A computer screen with white text

Description automatically generated

# 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}'

A computer screen with white text

Description automatically generated

# Analyze Fine-Tuned Model

Use this command to analyze the fine-tuning results:

openai api fine\_tunes.results -i <YOUR\_FINE\_TUNE\_JOB\_ID>

A screenshot of a computer program

Description automatically generated

# 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" | |

A screenshot of a computer

Description automatically generated

# 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" |

A screen shot of a computer

Description automatically generated