Reinforcement Learning – Finetune – GPT2 – with

- Sentiment -Reward – using – PPO

Process 02:

MLCSAS43.ipynb

File Edit View Insert Runtime Tools Help

Commands + Code + Text

```
!pip install transformers==4.37.2
!pip install datasets==2.16.1
!pip install trl==0.7.10
!pip install tqdm==4.66.1
!pip install torch==2.2.0
!pip install peft==0.10.0
!pip install "numpy<2"  # needed for the transformers and tqdm used here
```

```
Collecting transformers==4.37.2
  Downloading transformers-4.37.2-py3-none-any.whl.metadata (129 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 129.4/129.4 kB 5.8 MB/s eta 0:00:00
Requirement already satisfied: filelock in /usr/local/lib/python3.11/dist-packages (from transformers==4.37.2) (3.18.0)
Requirement already satisfied: huggingface-hub<1.0,>=0.19.3 in /usr/local/lib/python3.11/dist-packages (from transformers==4.3
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.11/dist-packages (from transformers==4.37.2) (2.0.2)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.11/dist-packages (from transformers==4.37.2) (24.2)
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.11/dist-packages (from transformers==4.37.2) (6.0.2)
Requirement already satisfied: regex!=2019.12.17 in /usr/local/lib/python3.11/dist-packages (from transformers==4.37.2) (2024.
Requirement already satisfied: requests in /usr/local/lib/python3.11/dist-packages (from transformers==4.37.2) (2.32.3)
Collecting tokenizers<0.19,>=0.14 (from transformers==4.37.2)
  Downloading tokenizers-0.15.2-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (6.7 kB)
```

```
import numpy as np
print(np.__version__)
```

```
1.26.4
```

```
[2] import torch
    from tqdm import tqdm

    from transformers import pipeline, AutoTokenizer
    from datasets import load_dataset

    from trl import PPOTrainer, PPOConfig, AutoModelForCausalLMWithValueHead
    from trl.core import LengthSampler
```

```
config = PPOConfig(
    model_name="lvwerra/gpt2-imdb",
    learning_rate=1.41e-5,
    log_with="wandb",
)

sent_kwargs = {"top_k": None, "function_to_apply": "none", "batch_size": 16}
```

```
import wandb

wandb.init()
```

```
wandb: Logging into wandb.ai. (Learn how to deploy a W&B server locally: https://wandb.me/wandb-server)
wandb: You can find your API key in your browser here: https://wandb.ai/authorize?ref=models
wandb: Paste an API key from your profile and hit enter: ·····
wandb: WARNING If you're specifying your api key in code, ensure this code is not shared publicly.
wandb: WARNING Consider setting the WANDB_API_KEY environment variable, or running `wandb login` from the command line.
wandb: No netrc file found, creating one.
wandb: Appending key for api.wandb.ai to your netrc file: /root/.netrc
wandb: Currently logged in as: rajavardhanreddygogulamudi01 (rajavardhanreddygogulamudi01-university-of-dayton) to https://api.wand
Tracking run with wandb version 0.19.10
Run data is saved locally in /content/wandb/run-20250429_163211-ksig89e2
Syncing run jumping-dawn-7 to Weights & Biases (docs)
View project at https://wandb.ai/rajavardhanreddygogulamudi01-university-of-dayton/uncategorized
View run at https://wandb.ai/rajavardhanreddygogulamudi01-university-of-dayton/uncategorized/runs/ksig89e2
Display W&B run
```

```
[6] def build_dataset(
        config,
        dataset_name="stanfordnlp/imdb",
        input_min_text_length=2,
        input_max_text_length=8,
    ):
        """
        Build dataset for training. This builds the dataset from `load_dataset`, one should
        customize this function to train the model on its own dataset.

        Args:
            dataset_name (`str`):
                The name of the dataset to be loaded.

        Returns:
            dataloader (`torch.utils.data.DataLoader`):
                The dataloader for the dataset.
        """
        tokenizer = AutoTokenizer.from_pretrained(config.model_name)
        tokenizer.pad_token = tokenizer.eos_token
        # load imdb with datasets
        ds = load_dataset(dataset_name, split="train")
        ds = ds.rename_columns({"text": "review"})
        ds = ds.filter(lambda x: len(x["review"]) > 200, batched=False)

        input_size = LengthSampler(input_min_text_length, input_max_text_length)
```

```
        def tokenize(sample):
            sample["input_ids"] = tokenizer.encode(sample["review"])[: input_size()]
            sample["query"] = tokenizer.decode(sample["input_ids"])
            return sample

        ds = ds.map(tokenize, batched=False)
        ds.set_format(type="torch")
        return ds
```

```
[7]  dataset = build_dataset(config)


     def collator(data):
         return dict((key, [d[key] for d in data]) for key in data[0])
```

```
/usr/local/lib/python3.11/dist-packages/huggingface_hub/file_download.py:896: FutureWarning: `resume_download` is deprecated a
  warnings.warn(
/usr/local/lib/python3.11/dist-packages/huggingface_hub/utils/_auth.py:94: UserWarning:
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your settings tab (https://huggingface.co/settings/tokens), set i
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access public models or datasets.
  warnings.warn(
tokenizer_config.json: 100% 17.0/17.0 [00:00<00:00, 2.15kB/s]
config.json: 100% 577/577 [00:00<00:00, 71.0kB/s]
vocab.json: 100% 899k/899k [00:00<00:00, 12.7MB/s]
merges.txt: 100% 456k/456k [00:00<00:00, 8.40MB/s]
special_tokens_map.json: 100% 90.0/90.0 [00:00<00:00, 11.3kB/s]
Downloading readme: 100% 7.81k/7.81k [00:00<00:00, 991kB/s]
Downloading data: 100% 21.0M/21.0M [00:00<00:00, 40.0MB/s]
Downloading data: 100% 20.5M/20.5M [00:00<00:00, 74.1MB/s]
```

```
[8] model = AutoModelForCausalLMWithValueHead.from_pretrained(config.model_name)
    ref_model = AutoModelForCausalLMWithValueHead.from_pretrained(config.model_name)
    tokenizer = AutoTokenizer.from_pretrained(config.model_name)

    tokenizer.pad_token = tokenizer.eos_token
```

```
/usr/local/lib/python3.11/dist-packages/huggingface_hub/file_download.py:896: FutureWarning: `resume_download` i
  warnings.warn(
pytorch_model.bin: 100% 548M/548M [00:02<00:00, 265MB/s]
```

```
[9] ppo_trainer = PPOTrainer(
        config, model, ref_model, tokenizer, dataset=dataset, data_collator=collator
    )
```

```
Finishing previous runs because reinit is set to 'default'.
View run jumping-dawn-7 at: https://wandb.ai/rajavardhanreddygogulamudi01-university-of-dayton/uncategorized/runs/ksig89e2
View project at: https://wandb.ai/rajavardhanreddygogulamudi01-university-of-dayton/uncategorized
Synced 5 W&B file(s), 0 media file(s), 0 artifact file(s) and 0 other file(s)
Find logs at: ./wandb/run-20250429_163211-ksig89e2/logs
Tracking run with wandb version 0.19.10
Run data is saved locally in /content/wandb/run-20250429_163316-6lkepu44
Syncing run northern-lion-6 to Weights & Biases (docs)
View project at https://wandb.ai/rajavardhanreddygogulamudi01-university-of-dayton/trl
View run at https://wandb.ai/rajavardhanreddygogulamudi01-university-of-dayton/trl/runs/6lkepu44
```

```
[10] device = ppo_trainer.accelerator.device
     if ppo_trainer.accelerator.num_processes == 1:
         device = 0 if torch.cuda.is_available() else "cpu"  # to avoid a `pipeline` bug
     sentiment_pipe = pipeline(
         "sentiment-analysis", model="lvwerra/distilbert-imdb", device=device
     )
```

```
config.json: 100% 735/735 [00:00<00:00, 60.6kB/s]
pytorch_model.bin: 100% 268M/268M [00:01<00:00, 255MB/s]
tokenizer_config.json: 100% 333/333 [00:00<00:00, 28.3kB/s]
vocab.txt: 100% 232k/232k [00:00<00:00, 8.74MB/s]
tokenizer.json: 100% 466k/466k [00:00<00:00, 19.5MB/s]
special_tokens_map.json: 100% 112/112 [00:00<00:00, 14.4kB/s]
```

```
[11] text = "this movie was really bad!!"
     sentiment_pipe(text, **sent_kwargs)
```

```
[{'label': 'NEGATIVE', 'score': 2.3350484371185303},
 {'label': 'POSITIVE', 'score': -2.726576328277588}]
```

```
[12] text = "this movie was really good!!"
     sentiment_pipe(text, **sent_kwargs)
```

```
[{'label': 'POSITIVE', 'score': 2.557040214538574},
 {'label': 'NEGATIVE', 'score': -2.294790267944336}]
```

```
[13] gen_kwargs = {
         "min_length": -1,
         "top_k": 0.0,
         "top_p": 1.0,
         "do_sample": True,
         "pad_token_id": tokenizer.eos_token_id,
     }
```

```
[14] output_min_length = 4
     output_max_length = 16
     output_length_sampler = LengthSampler(output_min_length, output_max_length)


     generation_kwargs = {
         "min_length": -1,
         "top_k": 0.0,
         "top_p": 1.0,
         "do_sample": True,
         "pad_token_id": tokenizer.eos_token_id,
     }
```

```
[14] for epoch, batch in enumerate(tqdm(ppo_trainer.dataloader)):
        query_tensors = batch["input_ids"]

        #### Get response from gpt2
        response_tensors = []
        for query in query_tensors:
            gen_len = output_length_sampler()
            generation_kwargs["max_new_tokens"] = gen_len
            query_response = ppo_trainer.generate(query, **generation_kwargs).squeeze()
            response_len = len(query_response) - len(query)
            response_tensors.append(query_response[-response_len:])
        batch["response"] = [tokenizer.decode(r.squeeze()) for r in response_tensors]

        #### Compute sentiment score
        texts = [q + r for q, r in zip(batch["query"], batch["response"])]
        pipe_outputs = sentiment_pipe(texts, **sent_kwargs)
        positive_scores = [
            item["score"]
            for output in pipe_outputs
            for item in output
            if item["label"] == "POSITIVE"
        ]
        rewards = [torch.tensor(score) for score in positive_scores]

        #### Run PPO step
        stats = ppo_trainer.step(query_tensors, response_tensors, rewards)
        ppo_trainer.log_stats(stats, batch, rewards)
```

```
/usr/local/lib/python3.11/dist-packages/trl/trainer/ppo_trainer.py:1212: UserWarning: The average ratio of batch (43.10) exceeds threshold 10
  warnings.warn(
/usr/local/lib/python3.11/dist-packages/trl/trainer/ppo_trainer.py:1212: UserWarning: The average ratio of batch (15.84) exceeds threshold 10
  warnings.warn(
/usr/local/lib/python3.11/dist-packages/trl/trainer/ppo_trainer.py:1212: UserWarning: The average ratio of batch (22.19) exceeds threshold 10
  warnings.warn(
/usr/local/lib/python3.11/dist-packages/trl/trainer/ppo_trainer.py:1212: UserWarning: The average ratio of batch (77.37) exceeds threshold 10
  warnings.warn(
/usr/local/lib/python3.11/dist-packages/trl/trainer/ppo_trainer.py:1212: UserWarning: The average ratio of batch (86.29) exceeds threshold 10
  warnings.warn(
 99%|█████████▉| 96/97 [2:07:28<01:19, 79.02s/it]/usr/local/lib/python3.11/dist-packages/trl/trainer/ppo_trainer.py:1212: UserWarning: The av
  warnings.warn(
/usr/local/lib/python3.11/dist-packages/trl/trainer/ppo_trainer.py:1212: UserWarning: The average ratio of batch (52.23) exceeds threshold 10
  warnings.warn(
/usr/local/lib/python3.11/dist-packages/trl/trainer/ppo_trainer.py:1212: UserWarning: The average ratio of batch (251.89) exceeds threshold 1(
  warnings.warn(
/usr/local/lib/python3.11/dist-packages/trl/trainer/ppo_trainer.py:1212: UserWarning: The average ratio of batch (23.27) exceeds threshold 10
  warnings.warn(
/usr/local/lib/python3.11/dist-packages/trl/trainer/ppo_trainer.py:1212: UserWarning: The average ratio of batch (67.85) exceeds threshold 10
  warnings.warn(
/usr/local/lib/python3.11/dist-packages/trl/trainer/ppo_trainer.py:1212: UserWarning: The average ratio of batch (50.60) exceeds threshold 10
  warnings.warn(
/usr/local/lib/python3.11/dist-packages/trl/trainer/ppo_trainer.py:1212: UserWarning: The average ratio of batch (18.95) exceeds threshold 10
  warnings.warn(
/usr/local/lib/python3.11/dist-packages/trl/trainer/ppo_trainer.py:1212: UserWarning: The average ratio of batch (63.71) exceeds threshold 10
  warnings.warn(
/usr/local/lib/python3.11/dist-packages/trl/trainer/ppo_trainer.py:1212: UserWarning: The average ratio of batch (65.30) exceeds threshold 10
  warnings.warn(
/usr/local/lib/python3.11/dist-packages/trl/trainer/ppo_trainer.py:1212: UserWarning: The average ratio of batch (12.87) exce
  warnings.warn(
/usr/local/lib/python3.11/dist-packages/trl/trainer/ppo_trainer.py:1212: UserWarning: The average ratio of batch (1103.81) ex
  warnings.warn(
/usr/local/lib/python3.11/dist-packages/trl/trainer/ppo_trainer.py:1212: UserWarning: The average ratio of batch (73.85) exce
  warnings.warn(
100%|██████████| 97/97 [2:08:48<00:00, 79.68s/it]
```

```
#### get a batch from the dataset
import pandas as pd

bs = 16
game_data = dict()
dataset.set_format("pandas")
df_batch = dataset[:].sample(bs)
game_data["query"] = df_batch["query"].tolist()
query_tensors = df_batch["input_ids"].tolist()

response_tensors_ref, response_tensors = [], []

#### get response from gpt2 and gpt2_ref
for i in range(bs):
    query = torch.tensor(query_tensors[i]).to(device)

    gen_len = output_length_sampler()
    query_response = ref_model.generate(
        query.unsqueeze(0), max_new_tokens=gen_len, **gen_kwargs
    ).squeeze()
    response_len = len(query_response) - len(query)
    response_tensors_ref.append(query_response[-response_len:])
```

```
[16]    query_response = model.generate(
            query.unsqueeze(0), max_new_tokens=gen_len, **gen_kwargs
        ).squeeze()
        response_len = len(query_response) - len(query)
        response_tensors.append(query_response[-response_len:])

    #### decode responses
    game_data["response (before)"] = [
        tokenizer.decode(response_tensors_ref[i]) for i in range(bs)
    ]
    game_data["response (after)"] = [
        tokenizer.decode(response_tensors[i]) for i in range(bs)
    ]

    #### sentiment analysis of query/response pairs before/after
    texts = [q + r for q, r in zip(game_data["query"], game_data["response (before)"])]
    pipe_outputs = sentiment_pipe(texts, **sent_kwargs)
    positive_scores = [
        item["score"]
        for output in pipe_outputs
        for item in output
        if item["label"] == "POSITIVE"
    ]
    game_data["rewards (before)"] = positive_scores
```

```
    texts = [q + r for q, r in zip(game_data["query"], game_data["response (after)"])]
    pipe_outputs = sentiment_pipe(texts, **sent_kwargs)
    positive_scores = [
        item["score"]
        for output in pipe_outputs
        for item in output
        if item["label"] == "POSITIVE"
    ]
    game_data["rewards (after)"] = positive_scores

    # store results in a dataframe
    df_results = pd.DataFrame(game_data)
    df_results
```

```
/usr/local/lib/python3.11/dist-packages/transformers/pipelines/base.py:1123: UserWarning: You seem
  warnings.warn(
```

```
warnings.warn(
```

	query	response (before)	response (after)	rewards (before)	rewards (after)
0	this movie is the	funniest thing ever! You've seen it before	classic.<|endoftext|>	2.689889	2.252165
1	Okay. So I	'm presenting the story, but the premise is not	know it will answer a lot of questions as the	-1.455610	1.800407
2	This film was bad.	There simply wasn't enough thawing	I liked it immensely, so I took	-3.042115	-1.183581
3	I'm	cocking these guys because I'm real dumb	sure they'll have a couple of surprises in	-1.926365	1.007598
4	After watching many of the "Next	Generation" movies, I	" show, I've	0.973060	1.194663
5	I really like 101 Dalmations	Part 1. Under an	) I saw it.	1.015429	1.640871
6	I don't what that other review	say, is the more/less obvious piece of work a	is. It's a great movie. It is great--	-1.476582	2.811067
7	I bought the DVD	of the show Saturday night, and went, I	and Point was awesome. The story was incredib...	0.686910	2.614584
8	Have I ever seen a film more	laughably funny than	memorable? Answer is	1.745479	0.781623
9	Skippy	gets taken to the	. Good skill and	-0.240897	2.100599
10	This movie is	really dumb, and doesn't have a storyline whi...	so fun, it's so entertaining, and some of the...	-2.673111	2.669794
11	From the moment the	writers were running out of money	filmmakers were in Belfast, they	-1.557214	0.552077
12	I couldn	't bear my own walked through the empty bathro...	't imagine (I truly enjoyed it) how wonderful ...	-1.958725	2.739160
13	Stewart Moss stars as	a family healing doctor and his son walks awa...	The Lion is among THE stars of the story,	1.019888	1.988639
14	Within the first 17	minutes of the film, the characters at least ...	/18 inches of Plate (here the series), he's im...	-2.047499	2.280530
15	I liked it but then	got bored with Mike Myers	saw something that was also	-0.253857	2.101673

```
[17] print("mean:")
     display(df_results[["rewards (before)", "rewards (after)"]].mean())
     print()
     print("median:")
     display(df_results[["rewards (before)", "rewards (after)"]].median())
```

```
mean:
```

	0
rewards (before)	-0.531333
rewards (after)	1.709492

dtype: float64

```
median:
```

	0
rewards (before)	-0.854733
rewards (after)	2.044619

dtype: float64

THE END