Text Generation with GPT-2

1. Setting up the Environment

```
!pip install transformers
!pip install datasets
!pip install ipykernel
!pip install torch
```

```
Requirement already satisfied: transformers in /usr/local/lib/python3.10/dist-packages (4.42.4)
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages (from transformers) (3.15.4)
Requirement already satisfied: huggingface-hub<1.0,>=0.23.2 in /usr/local/lib/python3.10/dist-packages (from transformers) (0.23.
Requirement already satisfied: numpy<2.0,>=1.17 in /usr/local/lib/python3.10/dist-packages (from transformers) (1.26.4)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from transformers) (24.1)
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.10/dist-packages (from transformers) (6.0.2)
Requirement already satisfied: regex!=2019.12.17 in /usr/local/lib/python3.10/dist-packages (from transformers) (2024.5.15)
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (from transformers) (2.32.3)
Requirement already satisfied: safetensors>=0.4.1 in /usr/local/lib/python3.10/dist-packages (from transformers) (0.4.4)
Requirement already satisfied: tokenizers<0.20,>=0.19 in /usr/local/lib/python3.10/dist-packages (from transformers) (0.19.1)
Requirement already satisfied: tqdm>=4.27 in /usr/local/lib/python3.10/dist-packages (from transformers) (4.66.5)
Requirement already satisfied: fsspec>=2023.5.0 in /usr/local/lib/python3.10/dist-packages (from huggingface-hub<1.0,>=0.23.2->tr
Requirement already satisfied: typing-extensions>=3.7.4.3 in /usr/local/lib/python3.10/dist-packages (from huggingface-hub<1.0,>=
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests->transformers)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests->transformers) (3.7)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests->transformers) (2.0.7
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests->transformers) (2024.
Requirement already satisfied: datasets in /usr/local/lib/python3.10/dist-packages (2.21.0)
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages (from datasets) (3.15.4)
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.10/dist-packages (from datasets) (1.26.4)
Requirement already satisfied: pyarrow>=15.0.0 in /usr/local/lib/python3.10/dist-packages (from datasets) (17.0.0)
Requirement already satisfied: dill<0.3.9,>=0.3.0 in /usr/local/lib/python3.10/dist-packages (from datasets) (0.3.8)
Requirement already satisfied: pandas in /usr/local/lib/python3.10/dist-packages (from datasets) (2.1.4)
Requirement already satisfied: requests>=2.32.2 in /usr/local/lib/python3.10/dist-packages (from datasets) (2.32.3)
Requirement already satisfied: tqdm>=4.66.3 in /usr/local/lib/python3.10/dist-packages (from datasets) (4.66.5)
Requirement already satisfied: xxhash in /usr/local/lib/python3.10/dist-packages (from datasets) (3.5.0)
Requirement already satisfied: multiprocess in /usr/local/lib/python3.10/dist-packages (from datasets) (0.70.16)
Requirement already satisfied: fsspec<=2024.6.1,>=2023.1.0 in /usr/local/lib/python3.10/dist-packages (from fsspec[http]<=2024.6.
Requirement already satisfied: aiohttp in /usr/local/lib/python3.10/dist-packages (from datasets) (3.10.3)
Requirement already satisfied: huggingface-hub>=0.21.2 in /usr/local/lib/python3.10/dist-packages (from datasets) (0.23.5)
Requirement already satisfied: packaging in /usr/local/lib/python3.10/dist-packages (from datasets) (24.1)
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.10/dist-packages (from datasets) (6.0.2)
Requirement already satisfied: aiohappyeyeballs>=2.3.0 in /usr/local/lib/python3.10/dist-packages (from aiohttp->datasets) (2.3.5
Requirement already satisfied: aiosignal>=1.1.2 in /usr/local/lib/python3.10/dist-packages (from aiohttp->datasets) (1.3.1)
Requirement already satisfied: attrs>=17.3.0 in /usr/local/lib/python3.10/dist-packages (from aiohttp->datasets) (24.2.0)
Requirement already satisfied: frozenlist>=1.1.1 in /usr/local/lib/python3.10/dist-packages (from aiohttp->datasets) (1.4.1)
Requirement already satisfied: multidict<7.0,>=4.5 in /usr/local/lib/python3.10/dist-packages (from aiohttp->datasets) (6.0.5)
Requirement already satisfied: yarl<2.0,>=1.0 in /usr/local/lib/python3.10/dist-packages (from aiohttp->datasets) (1.9.4)
Requirement already satisfied: async-timeout<5.0,>=4.0 in /usr/local/lib/python3.10/dist-packages (from aiohttp->datasets) (4.0.3
Requirement already satisfied: typing-extensions>=3.7.4.3 in /usr/local/lib/python3.10/dist-packages (from huggingface-hub>=0.21.
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests>=2.32.2->datase
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests>=2.32.2->datasets) (3.7)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests>=2.32.2->datasets) (2
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests>=2.32.2->datasets) (2
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.10/dist-packages (from pandas->datasets) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas->datasets) (2024.1)
Requirement already satisfied: tzdata>=2022.1 in /usr/local/lib/python3.10/dist-packages (from pandas->datasets) (2024.1)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.8.2->pandas->datasets
Requirement already satisfied: ipykernel in /usr/local/lib/python3.10/dist-packages (5.5.6)
Requirement already satisfied: ipython-genutils in /usr/local/lib/python3.10/dist-packages (from ipykernel) (0.2.0)
Requirement already satisfied: ipython>=5.0.0 in /usr/local/lib/python3.10/dist-packages (from ipykernel) (7.34.0)
Requirement already satisfied: traitlets>=4.1.0 in /usr/local/lib/python3.10/dist-packages (from ipykernel) (5.7.1)
Requirement already satisfied: jupyter-client in /usr/local/lib/python3.10/dist-packages (from ipykernel) (6.1.12)
Requirement already satisfied: tornado>=4.2 in /usr/local/lib/python3.10/dist-packages (from ipykernel) (6.3.3)
Requirement already satisfied: setuptools>=18.5 in /usr/local/lib/python3.10/dist-packages (from ipython>=5.0.0->ipykernel) (71.0
Requirement already satisfied: jedi>=0.16 in /usr/local/lib/python3.10/dist-packages (from ipython>=5.0.0->ipykernel) (0.19.1)
Requirement already satisfied: decorator in /usr/local/lib/python3.10/dist-packages (from ipython>=5.0.0->ipykernel) (4.4.2)
```

2.Data Preparation

a) Dataset collecting and formatting

```
from datasets import load_dataset
import pandas as pd
```

```
dataset = load_dataset("csv", data_files="/deeplearning_questions.csv")
```

```
Generating train split:   111/0 [00:00<00:00, 1899.15 examples/s]
```

```
df = pd.DataFrame(dataset['train'])
df.head()
```

	ID	DESCRIPTION
0	1	What is padding
1	2	Sigmoid Vs Softmax
2	3	What is PoS Tagging
3	4	What is tokenization
4	5	What is topic modeling

b) Tokenization

```
from transformers import GPT2Tokenizer
```

```
print(dataset)
```

```
DatasetDict({
    train: Dataset({
        features: ['ID', 'DESCRIPTION'],
        num_rows: 111
    })
})
```

```
tokenizer = GPT2Tokenizer.from_pretrained("gpt2")

tokenizer.pad_token = tokenizer.eos_token
```

```
def tokenize_function(examples):
    # Tokenize descriptions and set labels to be same as input_ids
    tokenized = tokenizer(examples["DESCRIPTION"], truncation=True, padding="max_length", max_length=128)
    tokenized["labels"] = tokenized["input_ids"]  # Set labels to be same as input_ids
    return tokenized

tokenized_datasets = dataset.map(tokenize_function, batched=True)
```

```
Map: 100%                                         99/99 [00:00<00:00, 1035.58 examples/s]

Map: 100%                                         12/12 [00:00<00:00, 291.56 examples/s]
```

c) Data splitting

```
# Split the dataset into training and validation sets
from datasets import DatasetDict

# Create a validation split from the training data
dataset = DatasetDict({
    'train': tokenized_datasets['train'].shuffle(seed=42).select([i for i in list(range(int(0.9 * len(tokenized_datasets['train']))))]).
    'validation': tokenized_datasets['train'].shuffle(seed=42).select([i for i in list(range(int(0.9 * len(tokenized_datasets['train'])
})
```

3. Fine Tuning GPT 2

1. Load the Pre-trained GPT-2 Model:

```
from transformers import GPT2LMHeadModel, Trainer, TrainingArguments
```

```
# Load the pre-trained GPT-2 model
model = GPT2LMHeadModel.from_pretrained('gpt2')

# If you added a padding token, resize the model’s embedding layer to match the tokenizer’s vocabulary size
model.resize_token_embeddings(len(tokenizer))
```

```
Embedding(50257, 768)
```

2. Set Up Training Arguments:

```
training_args = TrainingArguments(
    output_dir="./results",          # Directory to save model checkpoints
    overwrite_output_dir=True,       # Overwrite the content of the output directory
    num_train_epochs=3,              # Number of training epochs
    per_device_train_batch_size=4,   # Batch size per device during training
    save_steps=500,                  # Save checkpoint every 500 steps
    save_total_limit=2,              # Limit the total number of checkpoints
    prediction_loss_only=True,       # Only return loss in the evaluation
)
```

3. Set Up the Trainer:

```
trainer = Trainer(
    model=model,                        # The pre-trained GPT-2 model
    args=training_args,                 # Training arguments
    train_dataset=dataset['train'],     # Training dataset
    eval_dataset=dataset['validation']  # Validation dataset
)
```

4. Start Training:

```
trainer.train()
```

```
[84/84 14:04, Epoch 3/3]
Step  Training Loss
TrainOutput(global_step=84, training_loss=0.6881452287946429, metrics={'train_runtime': 856.8833, 'train_samples_per_second':
```

5. Run a small test batch:

```
# Select a small subset for testing
small_test_dataset = tokenized_datasets["train"].select([0, 1, 2])  # Select first 3 examples

# Update the Trainer to use the small test dataset
trainer.train_dataset = small_test_dataset

# Run training
trainer.train()
```

```
[3/3 00:34, Epoch 3/3]
Step  Training Loss
TrainOutput(global_step=3, training_loss=0.13036566972732544, metrics={'train_runtime': 48.8927, 'train_samples_per_second': 0.184,
```

4. Check Model Output with Sample Data:

```
# Load model and tokenizer
tokenizer = GPT2Tokenizer.from_pretrained('gpt2')
model = GPT2LMHeadModel.from_pretrained('gpt2')

# Tokenize a sample input
inputs = tokenizer("Example input text", return_tensors="pt")

# Perform a forward pass
outputs = model(**inputs, labels=inputs["input_ids"])
print(outputs.loss)  # Should print the loss value
```

```
tensor(8.3631, grad_fn=<NllLossBackward0>)
```

4. Evaluation

```
results = trainer.evaluate()
print(results)
```

```
[2/2 00:03]
[illegible]
```

5. Save the Trained Model

```
model.save_pretrained("./gpt2-finetuned")
tokenizer.save_pretrained("./gpt2-finetuned")
```

```
('./gpt2-finetuned/tokenizer_config.json',
 './gpt2-finetuned/special_tokens_map.json',
 './gpt2-finetuned/vocab.json',
 './gpt2-finetuned/merges.txt',
 './gpt2-finetuned/added_tokens.json')
```

6. Generate Text

```
# Generate text
input_text = " What is tokenization"
input_ids = tokenizer.encode(input_text, return_tensors="pt")
```

```
# Generate output
output = model.generate(input_ids, max_length=100 , num_return_sequences=1, no_repeat_ngram_size=2)
```

```
The attention mask and the pad token id were not set. As a consequence, you may observe unexpected behavior. Please pass your input
Setting `pad_token_id` to `eos_token_id`:50256 for open-end generation.
```

```
# Decode and print the generated text
print(tokenizer.decode(output[0], skip_special_tokens=True))
```

```
 What is tokenization?

Tokenization is a process of creating a new token. It is the process by which a token is created.
. The process is called tokenizing. This process takes a number of steps. First, it creates a unique identifier for the token, and t
```