05_gpt2

March 12, 2024

[5]:

```
!pip install transformers==4.28.0 tokenizers datasets accelerate
```

```
Requirement already satisfied: transformers==4.28.0 in
/home/solaris/miniconda3/envs/deep_learning_gpt2/lib/python3.10/site-packages
(4.28.0)
Requirement already satisfied: tokenizers in
/home/solaris/miniconda3/envs/deep_learning_gpt2/lib/python3.10/site-packages
(0.13.3)
Requirement already satisfied: datasets in
/home/solaris/miniconda3/envs/deep_learning_gpt2/lib/python3.10/site-packages
(2.17.0)
Requirement already satisfied: accelerate in
/home/solaris/miniconda3/envs/deep_learning_gpt2/lib/python3.10/site-packages
(0.27.2)
Requirement already satisfied: filelock in
/home/solaris/miniconda3/envs/deep_learning_gpt2/lib/python3.10/site-packages
(from transformers==4.28.0) (3.13.1)
Requirement already satisfied: huggingface-hub<1.0,>=0.11.0 in
/home/solaris/miniconda3/envs/deep_learning_gpt2/lib/python3.10/site-packages
(from transformers==4.28.0) (0.20.3)
Requirement already satisfied: numpy>=1.17 in
/home/solaris/miniconda3/envs/deep_learning_gpt2/lib/python3.10/site-packages
(from transformers==4.28.0) (1.26.4)
Requirement already satisfied: packaging>=20.0 in
/home/solaris/miniconda3/envs/deep_learning_gpt2/lib/python3.10/site-packages
(from transformers==4.28.0) (23.2)
Requirement already satisfied: pyyaml>=5.1 in
/home/solaris/miniconda3/envs/deep_learning_gpt2/lib/python3.10/site-packages
(from transformers==4.28.0) (6.0.1)
Requirement already satisfied: regex!=2019.12.17 in
/home/solaris/miniconda3/envs/deep_learning_gpt2/lib/python3.10/site-packages
(from transformers==4.28.0) (2023.12.25)
Requirement already satisfied: requests in
/home/solaris/miniconda3/envs/deep_learning_gpt2/lib/python3.10/site-packages
(from transformers==4.28.0) (2.31.0)
Requirement already satisfied: tqdm>=4.27 in
/home/solaris/miniconda3/envs/deep_learning_gpt2/lib/python3.10/site-packages
(from transformers==4.28.0) (4.66.2)
```

```
Requirement already satisfied: pyarrow>=12.0.0 in
/home/solaris/miniconda3/envs/deep_learning_gpt2/lib/python3.10/site-packages
(from datasets) (15.0.0)
Requirement already satisfied: pyarrow-hotfix in
/home/solaris/miniconda3/envs/deep_learning_gpt2/lib/python3.10/site-packages
(from datasets) (0.6)
Requirement already satisfied: dill<0.3.9,>=0.3.0 in
/home/solaris/miniconda3/envs/deep_learning_gpt2/lib/python3.10/site-packages
(from datasets) (0.3.8)
Requirement already satisfied: pandas in
/home/solaris/miniconda3/envs/deep_learning_gpt2/lib/python3.10/site-packages
(from datasets) (2.2.0)
Requirement already satisfied: xxhash in
/home/solaris/miniconda3/envs/deep_learning_gpt2/lib/python3.10/site-packages
(from datasets) (3.4.1)
Requirement already satisfied: multiprocess in
/home/solaris/miniconda3/envs/deep_learning_gpt2/lib/python3.10/site-packages
(from datasets) (0.70.16)
Requirement already satisfied: fsspec<=2023.10.0,>=2023.1.0 in
/home/solaris/miniconda3/envs/deep_learning_gpt2/lib/python3.10/site-packages
(from fsspec[http]<=2023.10.0,>=2023.1.0->datasets) (2023.10.0)
Requirement already satisfied: aiohttp in
/home/solaris/miniconda3/envs/deep_learning_gpt2/lib/python3.10/site-packages
(from datasets) (3.9.3)
Requirement already satisfied: psutil in
/home/solaris/miniconda3/envs/deep_learning_gpt2/lib/python3.10/site-packages
(from accelerate) (5.9.8)
Requirement already satisfied: torch>=1.10.0 in
/home/solaris/miniconda3/envs/deep_learning_gpt2/lib/python3.10/site-packages
(from accelerate) (2.2.0)
Requirement already satisfied: safetensors>=0.3.1 in
/home/solaris/miniconda3/envs/deep_learning_gpt2/lib/python3.10/site-packages
(from accelerate) (0.4.2)
Requirement already satisfied: aiosignal>=1.1.2 in
/home/solaris/miniconda3/envs/deep_learning_gpt2/lib/python3.10/site-packages
(from aiohttp->datasets) (1.3.1)
Requirement already satisfied: attrs>=17.3.0 in
/home/solaris/miniconda3/envs/deep_learning_gpt2/lib/python3.10/site-packages
(from aiohttp->datasets) (23.2.0)
Requirement already satisfied: frozenlist>=1.1.1 in
/home/solaris/miniconda3/envs/deep_learning_gpt2/lib/python3.10/site-packages
(from aiohttp->datasets) (1.4.1)
Requirement already satisfied: multidict<7.0,>=4.5 in
/home/solaris/miniconda3/envs/deep_learning_gpt2/lib/python3.10/site-packages
(from aiohttp->datasets) (6.0.5)
Requirement already satisfied: yarl<2.0,>=1.0 in
/home/solaris/miniconda3/envs/deep_learning_gpt2/lib/python3.10/site-packages
(from aiohttp->datasets) (1.9.4)
```

```
Requirement already satisfied: async-timeout<5.0,>=4.0 in
/home/solaris/miniconda3/envs/deep_learning_gpt2/lib/python3.10/site-packages
(from aiohttp->datasets) (4.0.3)
Requirement already satisfied: typing-extensions>=3.7.4.3 in
/home/solaris/miniconda3/envs/deep_learning_gpt2/lib/python3.10/site-packages
(from huggingface-hub<1.0,>=0.11.0->transformers==4.28.0) (4.9.0)
Requirement already satisfied: charset-normalizer<4,>=2 in
/home/solaris/miniconda3/envs/deep_learning_gpt2/lib/python3.10/site-packages
(from requests->transformers==4.28.0) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in
/home/solaris/miniconda3/envs/deep_learning_gpt2/lib/python3.10/site-packages
(from requests->transformers==4.28.0) (3.6)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/home/solaris/miniconda3/envs/deep_learning_gpt2/lib/python3.10/site-packages
(from requests->transformers==4.28.0) (2.2.0)
Requirement already satisfied: certifi>=2017.4.17 in
/home/solaris/miniconda3/envs/deep_learning_gpt2/lib/python3.10/site-packages
(from requests->transformers==4.28.0) (2024.2.2)
Requirement already satisfied: sympy in
/home/solaris/miniconda3/envs/deep_learning_gpt2/lib/python3.10/site-packages
(from torch>=1.10.0->accelerate) (1.12)
Requirement already satisfied: networkx in
/home/solaris/miniconda3/envs/deep_learning_gpt2/lib/python3.10/site-packages
(from torch>=1.10.0->accelerate) (3.2.1)
Requirement already satisfied: jinja2 in
/home/solaris/miniconda3/envs/deep_learning_gpt2/lib/python3.10/site-packages
(from torch>=1.10.0->accelerate) (3.1.3)
Requirement already satisfied: nvidia-cuda-nvrtc-cu12==12.1.105 in
/home/solaris/miniconda3/envs/deep_learning_gpt2/lib/python3.10/site-packages
(from torch>=1.10.0->accelerate) (12.1.105)
Requirement already satisfied: nvidia-cuda-runtime-cu12==12.1.105 in
/home/solaris/miniconda3/envs/deep_learning_gpt2/lib/python3.10/site-packages
(from torch>=1.10.0->accelerate) (12.1.105)
Requirement already satisfied: nvidia-cuda-cupti-cu12==12.1.105 in
/home/solaris/miniconda3/envs/deep_learning_gpt2/lib/python3.10/site-packages
(from torch>=1.10.0->accelerate) (12.1.105)
Requirement already satisfied: nvidia-cudnn-cu12==8.9.2.26 in
/home/solaris/miniconda3/envs/deep_learning_gpt2/lib/python3.10/site-packages
(from torch>=1.10.0->accelerate) (8.9.2.26)
Requirement already satisfied: nvidia-cublas-cu12==12.1.3.1 in
/home/solaris/miniconda3/envs/deep_learning_gpt2/lib/python3.10/site-packages
(from torch>=1.10.0->accelerate) (12.1.3.1)
Requirement already satisfied: nvidia-cufft-cu12==11.0.2.54 in
/home/solaris/miniconda3/envs/deep_learning_gpt2/lib/python3.10/site-packages
(from torch>=1.10.0->accelerate) (11.0.2.54)
Requirement already satisfied: nvidia-curand-cu12==10.3.2.106 in
/home/solaris/miniconda3/envs/deep_learning_gpt2/lib/python3.10/site-packages
(from torch>=1.10.0->accelerate) (10.3.2.106)
```

```
Requirement already satisfied: nvidia-cusolver-cu12==11.4.5.107 in
/home/solaris/miniconda3/envs/deep_learning_gpt2/lib/python3.10/site-packages
(from torch>=1.10.0->accelerate) (11.4.5.107)
Requirement already satisfied: nvidia-cusparse-cu12==12.1.0.106 in
/home/solaris/miniconda3/envs/deep_learning_gpt2/lib/python3.10/site-packages
(from torch>=1.10.0->accelerate) (12.1.0.106)
Requirement already satisfied: nvidia-nccl-cu12==2.19.3 in
/home/solaris/miniconda3/envs/deep_learning_gpt2/lib/python3.10/site-packages
(from torch>=1.10.0->accelerate) (2.19.3)
Requirement already satisfied: nvidia-nvtx-cu12==12.1.105 in
/home/solaris/miniconda3/envs/deep_learning_gpt2/lib/python3.10/site-packages
(from torch>=1.10.0->accelerate) (12.1.105)
Requirement already satisfied: triton==2.2.0 in
/home/solaris/miniconda3/envs/deep_learning_gpt2/lib/python3.10/site-packages
(from torch>=1.10.0->accelerate) (2.2.0)
Requirement already satisfied: nvidia-nvjitlink-cu12 in
/home/solaris/miniconda3/envs/deep_learning_gpt2/lib/python3.10/site-packages
(from nvidia-cusolver-cu12==11.4.5.107->torch>=1.10.0->accelerate) (12.3.101)
Requirement already satisfied: python-dateutil>=2.8.2 in
/home/solaris/miniconda3/envs/deep_learning_gpt2/lib/python3.10/site-packages
(from pandas->datasets) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in
/home/solaris/miniconda3/envs/deep_learning_gpt2/lib/python3.10/site-packages
(from pandas->datasets) (2024.1)
Requirement already satisfied: tzdata>=2022.7 in
/home/solaris/miniconda3/envs/deep_learning_gpt2/lib/python3.10/site-packages
(from pandas->datasets) (2024.1)
Requirement already satisfied: six>=1.5 in
/home/solaris/miniconda3/envs/deep_learning_gpt2/lib/python3.10/site-packages
(from python-dateutil>=2.8.2->pandas->datasets) (1.16.0)
Requirement already satisfied: MarkupSafe>=2.0 in
/home/solaris/miniconda3/envs/deep_learning_gpt2/lib/python3.10/site-packages
(from jinja2->torch>=1.10.0->accelerate) (2.1.5)
Requirement already satisfied: mpmath>=0.19 in
/home/solaris/miniconda3/envs/deep_learning_gpt2/lib/python3.10/site-packages
(from sympy->torch>=1.10.0->accelerate) (1.3.0)
```

[6]:

```python
import tensorflow as tf
import glob
import os
import shutil
import tqdm
import random
import matplotlib.pyplot as plt
import torch
from datasets import load_dataset
from tokenizers import Tokenizer
from tokenizers.models import BPE
from tokenizers.trainers import BpeTrainer
from tokenizers.pre_tokenizers import Whitespace
from transformers import PreTrainedTokenizerFast
from transformers import DataCollatorForLanguageModeling
from transformers import GPT2Config, GPT2LMHeadModel
from transformers import TrainingArguments, Trainer

tf.config.list_physical_devices("GPU")
```

```
/home/solaris/miniconda3/envs/deep_learning_gpt2/lib/python3.10/site-
packages/tqdm/auto.py:21: TqdmWarning: IProgress not found. Please update
jupyter and ipywidgets. See
https://ipywidgets.readthedocs.io/en/stable/user_install.html
  from .autonotebook import tqdm as notebook_tqdm
2024-02-16 12:15:18.715624: I
external/local_xla/xla/stream_executor/cuda/cuda_executor.cc:901] successful
NUMA node read from SysFS had negative value (-1), but there must be at least
one NUMA node, so returning NUMA node zero. See more at
https://github.com/torvalds/linux/blob/v6.0/Documentation/ABI/testing/sysfs-bus-
pci#L344-L355
2024-02-16 12:15:18.716454: I
external/local_xla/xla/stream_executor/cuda/cuda_executor.cc:901] successful
NUMA node read from SysFS had negative value (-1), but there must be at least
one NUMA node, so returning NUMA node zero. See more at
https://github.com/torvalds/linux/blob/v6.0/Documentation/ABI/testing/sysfs-bus-
pci#L344-L355
2024-02-16 12:15:18.716608: I
external/local_xla/xla/stream_executor/cuda/cuda_executor.cc:901] successful
NUMA node read from SysFS had negative value (-1), but there must be at least
one NUMA node, so returning NUMA node zero. See more at
https://github.com/torvalds/linux/blob/v6.0/Documentation/ABI/testing/sysfs-bus-
pci#L344-L355
```

```
[6]: [PhysicalDevice(name='/physical_device:GPU:0', device_type='GPU')]
```

1 Load Dataset

[7]:

```
dataset_file = "dataset.txt"

# How many files to load.
file_number = 100

# Clone the repo.
!git clone https://github.com/vilmibm/lovecraftcorpus

# Find all the files.
```

```
paths = glob.glob("lovecraftcorpus/*.txt")

# Do not use all.
paths = paths[:file_number]
print(sorted(paths))

# each line is a sample in the dataset
# in this case, each line is a paragraph
# Merge.
# TODO: make more sophisticated to deal with short paragraphs.
with open(dataset_file, "w") as output_file:
    for path in paths:
        for line in open(path, "r"):
            for split in line.split("\n"):
                split = split.strip()
                if split != "":
                    print(split, file=output_file)

# Delete repo.
!rm -rf lovecraftcorpus

# Done.
print("Corpus downloaded.")
```

```
Cloning into 'lovecraftcorpus'…
remote: Enumerating objects: 74, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 74 (delta 0), reused 3 (delta 0), pack-reused 70
Receiving objects: 100% (74/74), 1.12 MiB | 5.46 MiB/s, done.
['lovecraftcorpus/alchemist.txt', 'lovecraftcorpus/arthur_jermyn.txt',
'lovecraftcorpus/azathoth.txt', 'lovecraftcorpus/beast.txt',
'lovecraftcorpus/beyond_wall_of_sleep.txt', 'lovecraftcorpus/book.txt',
'lovecraftcorpus/celephais.txt', 'lovecraftcorpus/charles_dexter_ward.txt',
'lovecraftcorpus/clergyman.txt', 'lovecraftcorpus/colour_out_of_space.txt',
'lovecraftcorpus/cool_air.txt', 'lovecraftcorpus/crawling_chaos.txt',
'lovecraftcorpus/cthulhu.txt', 'lovecraftcorpus/dagon.txt',
'lovecraftcorpus/descendent.txt', 'lovecraftcorpus/doorstep.txt',
'lovecraftcorpus/dreams_in_the_witch.txt', 'lovecraftcorpus/dunwich.txt',
'lovecraftcorpus/erich_zann.txt', 'lovecraftcorpus/ex_oblivione.txt',
'lovecraftcorpus/festival.txt', 'lovecraftcorpus/from_beyond.txt',
'lovecraftcorpus/gates_of_silver_key.txt', 'lovecraftcorpus/haunter.txt',
'lovecraftcorpus/he.txt', 'lovecraftcorpus/high_house_mist.txt',
'lovecraftcorpus/hound.txt', 'lovecraftcorpus/hypnos.txt',
'lovecraftcorpus/innsmouth.txt', 'lovecraftcorpus/iranon.txt',
'lovecraftcorpus/juan_romero.txt', 'lovecraftcorpus/kadath.txt',
'lovecraftcorpus/lurking_fear.txt', 'lovecraftcorpus/martins_beach.txt',
'lovecraftcorpus/medusas_coil.txt', 'lovecraftcorpus/memory.txt',
'lovecraftcorpus/moon_bog.txt', 'lovecraftcorpus/mountains_of_madness.txt',
'lovecraftcorpus/nameless.txt', 'lovecraftcorpus/nyarlathotep.txt',
'lovecraftcorpus/old_folk.txt', 'lovecraftcorpus/other_gods.txt',
'lovecraftcorpus/outsider.txt', 'lovecraftcorpus/pharoahs.txt',
'lovecraftcorpus/pickman.txt', 'lovecraftcorpus/picture_house.txt',
'lovecraftcorpus/poetry_of_gods.txt', 'lovecraftcorpus/polaris.txt',
'lovecraftcorpus/randolph_carter.txt', 'lovecraftcorpus/rats_walls.txt',
'lovecraftcorpus/reanimator.txt', 'lovecraftcorpus/redhook.txt',
'lovecraftcorpus/sarnath.txt', 'lovecraftcorpus/shadow_out_of_time.txt',
'lovecraftcorpus/shunned_house.txt', 'lovecraftcorpus/silver_key.txt',
'lovecraftcorpus/street.txt', 'lovecraftcorpus/temple.txt',
'lovecraftcorpus/terrible_old_man.txt', 'lovecraftcorpus/tomb.txt',
'lovecraftcorpus/tree.txt', 'lovecraftcorpus/ulthar.txt',
'lovecraftcorpus/unnamable.txt', 'lovecraftcorpus/vault.txt',
'lovecraftcorpus/what_moon_brings.txt', 'lovecraftcorpus/whisperer.txt',
'lovecraftcorpus/white_ship.txt']
Corpus downloaded.
```

2 Prepare Datasets

[8]:

```
raw_datasets = load_dataset("text", data_files=dataset_file)
raw_datasets
# 4371 lines
```

```
Generating train split: 4371 examples [00:00, 185606.71 examples/s]
```

[8]:

```
DatasetDict({
    train: Dataset({
        features: ['text'],
        num_rows: 4371
    })
})
```

[9]:

```
raw_datasets["train"][666]
# with HuggingFace datasets, every sample is a dictionary
# a sample from the dataset here is a paragraph
# the key is always "text"
```

[9]:

```
{'text': "No word was spoken amidst the distant sound that grew nearer and
nearer, but as I followed the memory-face's mad stare along that cursed shaft of
light to its source, the source whence also the whining came, I, too, saw for an
instant what it saw, and fell with ringing ears in that fit of shrieking
epilepsy which brought the lodgers and the police. Never could I tell, try as I
might, what it actually was that I saw; nor could the still face tell, for
although it must have seen more than I did, it will never speak again. But
always I shall guard against the mocking and insatiate Hypnos, lord of sleep,
against the night sky, and against the mad ambitions of knowledge and
philosophy."}
```

3 Goal

Generate new text

4 Steps

1. Load Dataset
2. Prepare Datasets
3. Encode Text
4. Tokenize Text
5. Build Model

5 Create Tokenizer

[10]:

```python
# Create empty tokenizer and its trainer
tokenizer = Tokenizer(BPE(unk_token="[UNK]")) # subword tokenization and ways to merge them
trainer = BpeTrainer(vocab_size=5_000, special_tokens=["[UNK]", "[PAD]"])
# separates the tokens with a space
tokenizer.pre_tokenizer = Whitespace()

# Batch samples to speed up process
def batch_iterator(batch_size=1000):
    # the batch size is the number of samples that will be processed at once
    # the iterator will yield a batch of samples
    # yield is a keyword in Python that is used like return, except the function will return a generator
    # a generator is an iterator that generates one item at a time
    for i in range(0, len(raw_datasets["train"]), batch_size):
        yield raw_datasets["train"][i : i + batch_size]["text"]

# Train the tokenizer
tokenizer.train_from_iterator(batch_iterator(), trainer=trainer, length=len(raw_datasets["train"]))

# Saves the tokenizer
# when downloading model, we download model, and the tokenizer
tokenizer.save("tokenizer.json")

# Load it fast
# speeds up the process
tokenizer = PreTrainedTokenizerFast(tokenizer_file="tokenizer.json")
tokenizer.add_special_tokens({"pad_token": "[PAD]"})
```

[10]: 0

[11]:
```
# random text that sounds like H.P. Lovcraft
text = "In his house at R'lyeh, dead Cthulhu waits dreaming."

# Tokenize the text
# the tokenizer will split the text into tokens
tokenizer(text)

# input_ids are the token ids
# the input_ids are fed to the model

# token_type_ids are used to distinguish different sequences in the same input

# attention_mask is used to tell the model to ignore the padding tokens
```

[11]:
```
{'input_ids': [368, 169, 470, 100, 44, 6, 118, 4359, 9, 830, 3474, 1012, 282,
3509, 11], 'token_type_ids': [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
'attention_mask': [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]}
```

[12]:
```
# random text that sounds like H.P. Lovcraft
text = "In his house at R'lyeh, dead Cthulhu waits dreaming."

# Tokenize the text
# the tokenizer will split the text into tokens
print(tokenizer(text)["input_ids"]) # token indices

tokens = [tokenizer.decode([index]) for index in tokenizer(text)["input_ids"]]
print(tokens)
```

```
[368, 169, 470, 100, 44, 6, 118, 4359, 9, 830, 3474, 1012, 282, 3509, 11]
['In', 'his', 'house', 'at', 'R', "'", 'ly', 'eh', ',', 'dead', 'Cthulhu', 'wa',
'its', 'dreaming', '.']
```

6 Tokenize Dataset

[13]:
```
# start with sequence length of 256
# pads the sequences to the same length
sequence_length = 256

# takes a dictionary as input
def tokenize_function(example):
    # tokenize the text
    tokenized_example = tokenizer(
        example["text"],
        truncation=True,
        padding=True,
        max_length=sequence_length,
    )
    return {"input_ids": tokenized_example["input_ids"]}

# tokenize entire dataset
tokenized_datasets = raw_datasets.map(tokenize_function, batched=True,␣
 ↪remove_columns=raw_datasets["train"].column_names)
```

```
Map: 100%|      | 4371/4371 [00:00<00:00, 12201.74 examples/s]
```

[14]:
```
# print sample number 666
# returns input_ids
print(tokenized_datasets["train"][666])
```

```
{'input_ids': [1004, 2376, 127, 3391, 1394, 93, 1254, 584, 128, 1134, 3237, 102,
3237, 9, 195, 109, 35, 1480, 93, 1328, 10, 607, 6, 71, 357, 104, 240, 1022, 128,
1487, 4493, 103, 297, 111, 282, 2108, 9, 93, 2108, 2932, 1492, 93, 121, 1569,
361, 9, 35, 9, 540, 9, 382, 148, 94, 2222, 291, 113, 382, 9, 102, 1701, 152,
1188, 107, 2237, 92, 128, 1862, 103, 4436, 304, 442, 2500, 182, 942, 93, 3322,
59, 261, 102, 93, 1974, 11, 4244, 234, 35, 628, 9, 742, 109, 35, 413, 9, 291,
113, 2378, 127, 128, 35, 382, 24, 420, 234, 93, 514, 607, 628, 9, 148, 2316,
113, 394, 233, 519, 305, 365, 35, 330, 9, 113, 586, 483, 1338, 437, 11, 528,
879, 35, 1186, 2153, 894, 93, 4683, 102, 560, 100, 61, 227, 34, 77, 68, 124, 71,
9, 64, 1322, 103, 1072, 9, 894, 93, 340, 986, 9, 102, 894, 93, 357, 4167, 1688,
103, 1720, 102, 4074, 77, 11, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]}
```

7 Collate the Data

[15]:
```
# data collator is used to batch the samples together
# data pump for training
# collate means to collect and combine
data_collator = DataCollatorForLanguageModeling(tokenizer=tokenizer, mlm=False)
```

8 Create the Model

[16]:
```
model_config = GPT2Config(
    vocab_size=tokenizer.vocab_size, # the size of the vocabulary
    pad_token_id=tokenizer.pad_token_id, # the token id for padding
    n_ctx=sequence_length, # context length
    n_positions=sequence_length, # positions in context, the order of the␣
  ↪tokens in sequence
    n_embd=512, # embedding dimension
    n_head=8, # number of heads in the multi-head attention models
    n_layer=6, # number of layers
)

model = GPT2LMHeadModel(model_config)
model

# wte: word token embeddings; which means the embeddings of the tokens
# wpe: word position embeddings; which means the embeddings of the positions of␣
  ↪the tokens
# drop: dropout
# sequence length: 256
# dropout layer: 0.1
# dropout is a regularization technique; it prevents overfitting
# normalisation is done first, which differs to transformers
# normalisation means to scale the input to have a mean of 0 and a standard␣
  ↪deviation of 1
# Conv1D means 1D convolution; convolutions are used to extract features from␣
  ↪the input
# LayerNorm means layer normalisation; normalisation is used to improve the␣
  ↪training of the model
```

[16]:

```
GPT2LMHeadModel(
  (transformer): GPT2Model(
    (wte): Embedding(5000, 512)
    (wpe): Embedding(256, 512)
    (drop): Dropout(p=0.1, inplace=False)
    (h): ModuleList(
      (0-5): 6 x GPT2Block(
        (ln_1): LayerNorm((512,), eps=1e-05, elementwise_affine=True)
        (attn): GPT2Attention(
          (c_attn): Conv1D()
          (c_proj): Conv1D()
          (attn_dropout): Dropout(p=0.1, inplace=False)
          (resid_dropout): Dropout(p=0.1, inplace=False)
        )
        (ln_2): LayerNorm((512,), eps=1e-05, elementwise_affine=True)
        (mlp): GPT2MLP(
          (c_fc): Conv1D()
          (c_proj): Conv1D()
          (act): NewGELUActivation()
          (dropout): Dropout(p=0.1, inplace=False)
        )
      )
    )
    (ln_f): LayerNorm((512,), eps=1e-05, elementwise_affine=True)
  )
  (lm_head): Linear(in_features=512, out_features=5000, bias=False)
)
```

9 Create Trainer

10 Save Trainer

[21]:

```
output_path = "output"

# Create the Trainer

training_args = TrainingArguments(
    output_dir=output_path, # output directory
    overwrite_output_dir=True, # overwrite the content of the output directory
    num_train_epochs=10, # number of training epochs
    #per_device_train_batch_size=16, # batch size for training per device (e.g.␣
 ↪multiple GPUs), which took 8 minutes
    #per_device_train_batch_size=32 # double, because i was only using around␣
 ↪3GB of VRAM, which took 7.5 minutes
    per_device_train_batch_size=46 # increase, because i was only using around␣
 ↪6GB of VRAM with 32, which also took 7.5 minutes
)

trainer = Trainer(
    model=model, # the model
    args=training_args, # training arguments
    data_collator=data_collator, # data collator
    train_dataset=tokenized_datasets["train"] # training dataset
)

# Train
trainer.train()

# Save
tokenizer.save_pretrained(output_path)
model.save_pretrained(output_path)
```

```
/home/solaris/miniconda3/envs/deep_learning_gpt2/lib/python3.10/site-
packages/transformers/optimization.py:391: FutureWarning: This implementation of
AdamW is deprecated and will be removed in a future version. Use the PyTorch
implementation torch.optim.AdamW instead, or set `no_deprecation_warning=True`
to disable this warning
  warnings.warn(
 52%|       | 500/960 [03:55<03:44,  2.05it/s]
```

```
{'loss': 4.8177, 'learning_rate': 2.3958333333333334e-05, 'epoch': 5.21}
```

```
100%|          | 960/960 [07:35<00:00,  2.11it/s]
```

```
{'train_runtime': 455.0859, 'train_samples_per_second': 96.048,
'train_steps_per_second': 2.109, 'train_loss': 4.731611124674479, 'epoch': 10.0}
```

[22]:

```
# Encode the conditioning tokens.
input_ids = tokenizer.encode("The most merciful thing in the world, I think, is the inability of the human mind to correlate all its contents.", return_tensors="pt").cuda()
print(input_ids)

# Generate more tokens.
generated_ids = model.generate(
    input_ids,
    max_length=100,
    do_sample=True,
    temperature=0.5
)
generated_sequence = tokenizer.decode(generated_ids[0], clean_up_tokenization_spaces=True)
print(generated_sequence)
```

```
tensor([[ 184,  325, 3454,  205,   92,   93,  552,    9,   35,  678,    9,  114,
           93,   92, 3974,  103,   93,  577,  609,  111,  421,  695,  227,  156,
          282, 4911,   11]], device='cuda:0')
The most merciful thing in the world, I think, is the in ability of the human
mind to cor rel ate all its contents. I was a moment, and made a half - place
which an old man's only because of the most of the world was a certain con stell
ations. I was not that I was in the house, and not known to the ancient, but I
was not even if the first time before. The thing was a strange, and I saw that I
was no
```