Name: Sai Ram Gunturu
ID: C00313478

Note: This is a completely new notebook created by me

GPT Model Fine-Tuning Notebook Project Log

Dataset: Wikitext-2 (raw version from Hugging Face)

Phases:

Phase	Techniques Used	Reason for the Technique Used	Duration	Difficulty Level (1-10)
Dataset Selection	Used the Wikitext-2 dataset from Hugging Face	To fine-tune a model on high-quality, natural language data from Wikipedia articles	20 min	6
Data Preparation	Tokenization with GPT2TokenizerFast, filtering empty texts, and setting up a Data Collator for causal LM	To convert raw text into token IDs and prepare padded batches for causal language modeling	30 min	7
Baseline Testing	Generated text using the pre-trained GPT-2 model without fine-tuning	To establish a reference performance and assess the need for domain-specific fine-tuning	3 hours	8
Model Fine-Tuning	Fine-tuned GPT-2 using the Trainer API with mixed precision (fp16), proper batch sizes, and disabled 'wandb' logging	To adapt the pre-trained model to capture the style and language patterns of Wikitext-2, reducing training loss and improving predictions	5 hours	9

Evaluation & Saving the model	Evaluated training loss (decreased from ~3.64 to ~3.10, averaging ~3.25) and saved model using save_pretrained()	To verify robust learning and persist the model and tokenizer for future inference or deployment	10 min	6
Inference	Created a text-generation pipeline with explicit parameters (max_length, truncation, temperature, top_k, top_p)	To generate consistent, coherent, and contextually relevant text from user prompts, matching the style observed during training	15 min	6

1. Business Understanding

My objective in this project is to predict high-quality text output by fine-tuning a pre-trained GPT-2 model on the Wikitext-2 dataset. While the original GPT-2 model generates general language, fine-tuning adapts it to the specific style and structure of Wikipedia articles. This fine-tuned model should therefore produce outputs that are coherent, contextually rich, and reflective of the dataset's formal tone.

2. Data Understanding & Preparation

I sourced the Wikitext-2 dataset from Hugging Face, which contains raw Wikipedia text. The dataset is divided into training (~36,718 examples), validation (~3,760 examples), and test (~4,358 examples) sets.

The raw text was tokenized using GPT2TokenizerFast, setting the EOS token as the pad token. Empty texts were filtered out to prevent errors. A data collator for causal language modeling (with mlm=False) was used to create consistent batches for training.

3. Modeling & Training

3.1 Baseline Model

I first evaluated the pre-trained GPT-2 model by generating text from a sample prompt. Although the generated output was fluent, it did not fully capture the style and nuance of Wikipedia text, which confirmed the need for fine-tuning.

3.2 Enhanced Model with Technical Indicators

To fine-tune the model, I used the Trainer API with the following setup:

- **Epochs:** 1 (for quick experimentation)
- **Batch Size:** 2 per GPU
- **Mixed Precision:** Enabled (fp16=True) to speed up training and lower memory usage
- **Checkpointing:** Saving every 500 steps, with a maximum of 2 checkpoints
- **WandB Logging:** Disabled (report_to=[]) to simplify tracking

The training loss steadily decreased from about 3.64 (step 500) to around 3.10 (step 11,500) with an average of ~3.25. The entire training process (11,884 steps) took approximately 31.6 minutes, demonstrating that the model effectively learned the language modeling task.

4. Evaluation

The fine-tuned model was evaluated both quantitatively and qualitatively.

- **Quantitative Evaluation:**

  The average training loss of ~3.25 and the steady loss reduction indicate robust learning.

- **Qualitative Evaluation:**

  When tested with prompts like "An Excellent movie" and "good morning," the model generated coherent, contextually relevant text that reflects the language style of Wikipedia. A truncation warning was noted, so explicit truncation (truncation=True) is recommended for consistent inference.

5. Conclusions

Transitioning from the baseline GPT-2 model to a fine-tuned version has significantly improved output quality. The fine-tuning process has resulted in:

- An average training loss of ~3.25, demonstrating effective learning.
- Generated outputs that are coherent and reflect the formal, detailed style of Wikipedia.
- A robust model that, despite some sensitivity to sampling parameters, is well-prepared for further evaluation and deployment.

Key learnings include the importance of fine-tuning with domain-specific data, the critical role of sampling parameters in inference, and the benefits of mixed precision training. These insights

guide future steps, such as further fine-tuning (if necessary), integrating the model into a deployment pipeline, and continuously monitoring output quality.

This notebook demonstrates that constant fine-tuning and parameter adjustments are essential for building a robust text-generation model that can meet practical application requirements.