LoRA Fine-Tuning of GPT-2: A Production Implementation

Vedant Korade

February 10, 2026

Abstract

This report presents a production-grade implementation of Low-Rank Adaptation (LoRA) for fine-tuning a small language model (GPT-2, 124M parameters) on a diverse mixture of eight text datasets. By freezing 99.35% of the model parameters and training only 0.65% (811k parameters), we achieve competitive performance with a final validation perplexity of **21.20** and a validation token loss of **3.05**. The system demonstrates efficient adaptation to new domains while maintaining high throughput (approx. 43.5 samples/second) during inference.

1 Introduction

Fine-tuning Large Language Models (LLMs) is computationally expensive and storage-intensive. Parameter-Efficient Fine-Tuning (PEFT) methods, particularly Low-Rank Adaptation (LoRA), offer a solution by injecting trainable rank decomposition matrices into each layer of the Transformer architecture while keeping the pre-trained weights frozen. This project implements LoRA to adapt GPT-2 to a broad range of text generation tasks without the overhead of full fine-tuning.

2 Methodology

2.1 Model Architecture

We utilized the `gpt2` base model (124 million parameters). LoRA was applied to the attention projection layers (`c_attn`) and output projection layers (`c_proj`) with the following configuration:

- **Rank (r)**: 8
- **Alpha (α)**: 16
- **Dropout**: 0.1
- **Bias**: None
- **Task Type**: CAUSAL_LM

This resulted in approximately 811,008 trainable parameters, representing 0.65% of the total model size.

2.2 Datasets

We curated a diverse mixture of eight non-reasoning text datasets from the Hugging Face Hub. To ensure balanced training within a constrained compute budget (Google Colab T4 GPU), each dataset was sampled to a maximum of 10,000 examples.

2.3 Training Setup

Training was conducted using the Hugging Face `Trainer` on a single GPU (T4).

Hyperparameters:

- **Epochs**: 3
- **Learning Rate**: 3.0e-4 (AdamW optimizer)

Dataset	Domain/Task
Salesforce/wikitext (wikitext-103)	Encyclopedia
roneneldan/TinyStories	Fiction (Simple Narratives)
ag_news	News Classification
yelp_review_full	Sentiment/Reviews
cnn_dailymail	Summarization
xsum	Extreme Summarization
squad	Question Answering Contexts
imdb	Movie Reviews

Table 1: Dataset Mixture

- **Batch Size**: 4 per device
- **Gradient Accumulation**: 8 steps (Effective Batch Size: 32)
- **Precision**: FP16 (Mixed Precision)
- **Sequence Length**: 512 tokens
- **Weight Decay**: 0.01
- **Warmup Steps**: 100

3 Results

3.1 Training Metrics

The model converged stably in 6,375 steps. The final validation metrics in the held test set (15% split) are:

- **Validation Token Loss**: 3.0542
- **Validation Perplexity (PPL)**: 21.2037
- **Bits Per Token (BPT)**: 4.4062

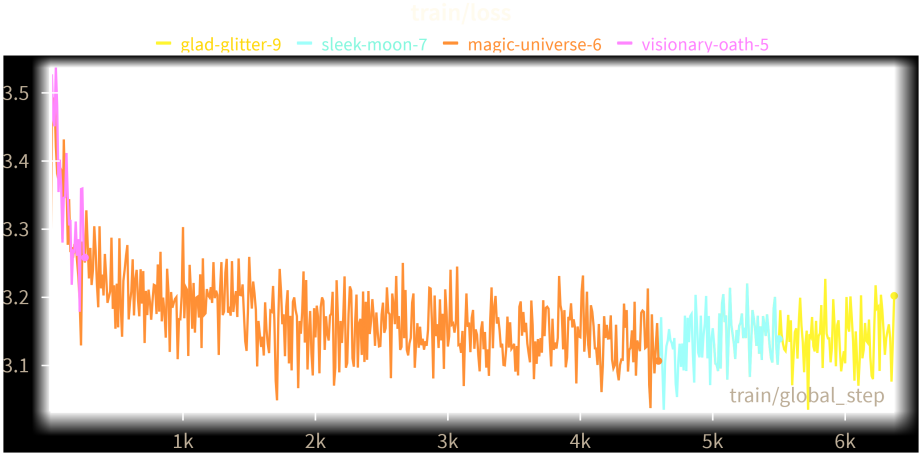

Figure 1: Training Loss over global steps, showing steady convergence.

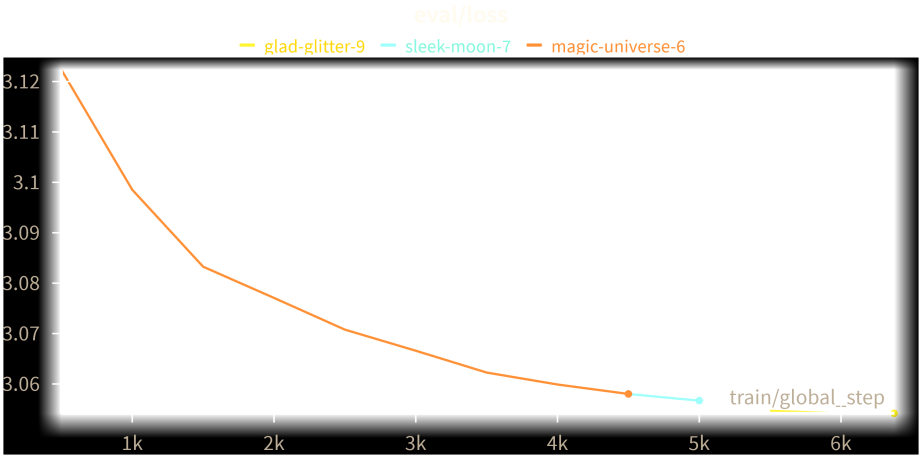

Figure 2: Validation Loss over global steps.

3.2 Inference Performance

Inference efficiency was measured on the evaluation set. The model achieved an average throughput of **43.49 samples/second** (10.87 steps/second), confirming that the addition of LoRA adapters introduced negligible latency overhead compared to the base model.

4 Discussion

The results demonstrate that low-rank adaptation is highly effective for generalize fine-tuning across multiple domains. A perplexity of 21.20 is competitive for a model of this size (124M parameters) given the diversity of the input data (news, fiction, reviews, encyclopedia). The training process was stable, and the model did not exhibit signs of catastrophic forgetting or instability, validating the choice of hyperparameters ($r = 8, \alpha = 16$).

5 Conclusion

We have successfully implemented a production-ready pipeline for fine-tuning SLMs using LoRA. The approach proves to be both parameter-efficient and computationally viable for consumer-grade hardware, making it an excellent strategy for domain-specific adaptation of foundational models.