Report on LoRA Tuning using PEFT from Hugging Face

Introduction to LoRA Tuning

Low-Rank Adaptation (LoRA) is a technique for fine-tuning pre-trained models more efficiently by modifying only a small, low-rank portion of the model's parameters. Instead of modifying all parameters, LoRA decomposes the weight matrices into smaller matrices, significantly reducing computational costs and memory usage while retaining similar or even better performance than full fine-tuning.

Key Concepts of LoRA

LoRA works by decomposing large weight matrices into smaller low-rank matrices. This allows fine-tuning large models with much fewer trainable parameters, reducing computational requirements while still achieving effective performance. In large language models like GPT-3 or Bloom, LoRA can reduce the number of trainable parameters to as little as 0.02% of the total parameters.

PEFT Library and Hugging Face Integration

The PEFT library offers implementations of efficient fine-tuning techniques like LoRA. Hugging Face supports pre-trained models compatible with PEFT, making it easy to experiment with LoRA on models like GPT-2, GPT-3, and Bloom.

Experiment Overview

We used a small Bloom model and fine-tuned it using LoRA on the "fka/awesome-chatgpt-prompts" dataset from Hugging Face, which contains prompts for motivational coaching. The goal was to observe how the pre-trained model performs before and after LoRA fine-tuning.

Process and Results

1. Pre-trained Model Inference: The pre-trained Bloom model generated generic responses, not tailored to the motivational coaching task.
2. Dataset Preparation: We tokenized the dataset and selected a small subset for fine-tuning.
3. LoRA Configuration: We set the rank parameter (`r=4`) and applied dropout to avoid overfitting.
4. Fine-tuning: We fine-tuned the model for 2 epochs with a learning rate of 0.03.
5. Post-fine-tuning Inference: After fine-tuning, the model generated much more relevant and context-specific responses related to motivational coaching.

Key Learnings

1. LoRA Efficiency: LoRA allows fine-tuning large models with a small fraction of the total parameters, leading to significant resource savings while maintaining good performance.
2. Cost-Effectiveness: LoRA enables fine-tuning on limited hardware resources (e.g., Google Colab), reducing training time and computational costs.
3. Hyperparameter Sensitivity: The choice of parameters like rank, dropout, and learning rate greatly affects the fine-tuning process and results.
4. Task-Specific Improvement: Fine-tuning with LoRA improved the model's ability to generate task-specific responses, like motivational coaching.