Fine-tuning Phi2 Model using QLoRA for Natural Language Inference (NLI)

November 4, 2024

1 Introduction

This report outlines the fine-tuning of the Phi2 Large Language Model (LLM) using QLoRA for the task of Natural Language Inference (NLI). The task was performed on a subset of the SNLI dataset and evaluated on its effectiveness in improving model accuracy on NLI.

2 Dataset and Model Selection

The dataset used is the SNLI dataset from Hugging Face, with specific splits as follows:

- **Training Set**: 1000 samples, selected by choosing every 550th sample from a total of 550,000 samples.
- **Validation Set**: 100 samples, selected by choosing every 100th sample from a total of 10,000 samples.
- **Test Set**: 100 samples, selected by choosing every 100th sample from a total of 10,000 samples.

The model chosen for fine-tuning was **Phi2** from Hugging Face.

3 Fine-Tuning Process and Hyperparameters

The Phi2 model was fine-tuned using QLoRA for 5 epochs. During each epoch, the model was saved to ensure continuity. The following hyperparameters were used for fine-tuning:

- **Batch size**: 8
- **Learning rate**: 2.5×10^{-5}
- **Number of epochs**: 5

The model and dataset were initially processed on a CPU environment, switching to GPU only when ready to run the final fine-tuning for optimal resource usage.

4 Results and Analysis

4.1 1. Accuracy Comparison

The balanced accuracy of the pretrained model on the test set was observed to be 38%. After fine-tuning, the model achieved an improved balanced accuracy of 72%, indicating a significant enhancement in the model's understanding of NLI.

4.2 2. Time Taken for Fine-Tuning

The total time taken to fine-tune the model over 5 epochs was approximately [Enter time in HH:MM:SS format].

4.3 3. Model Parameters

The fine-tuning process involved adjusting only a small percentage of the model's parameters:

- **Trainable Parameters**: 17,448,960
- **Total Parameters**: 1,538,841,600
- **Percentage of Parameters Fine-Tuned**: 1.13%

5 LoRA Configuration

The following table provides details on the configuration parameters used for fine-tuning the Phi2 model using QLoRA.

Parameter	Value
r	32
lora_alpha	64
target_modules	{ "q_proj", "k_proj", "v_proj", "o_proj", "gate_proj", "up_proj", "down_proj", "lm_head" }
bias	none
lora_dropout	0.05
task_type	CAUSAL_LM

Table 1: LoRA Configuration Parameters for Phi2 Model Fine-Tuning

5.1 4. Resources Used

The fine-tuning was conducted on an **A40 GPU** with 48 GB of VRAM, which was essential in managing the large model size and batch processing requirements. Additional system memory usage was observed to be approximately [Enter system memory usage if available].

5.2 5. Failure Cases and Analysis

5.2.1 Corrected Failure Cases

Several failure cases from the pretrained model were successfully corrected after fine-tuning. For example:

- Pretrained Model Misclassification: Premise and hypothesis pairs that were previously misclassified as "neutral" were correctly identified as "contradiction" or "entailment" after fine-tuning, indicating improved understanding of entailment relationships.

5.2.2 Uncorrected Failure Cases

Some cases remained challenging for the fine-tuned model, such as:

- Complex syntactic structures or nuanced meanings that involve implicit or indirect relationships. These cases may require further model capacity or fine-grained linguistic understanding beyond current capabilities.

5.2.3 Possible Explanations

The improvement in corrected cases likely stems from the QLoRA fine-tuning's ability to better capture patterns in NLI-specific contexts, which were not fully represented in the pretrained model. However, the uncorrected cases suggest limitations in the dataset diversity or model's representation of nuanced contexts, which could be addressed with additional training data or more specialized model architectures.

6 Conclusion

Fine-tuning the Phi2 model with QLoRA significantly improved the model's balanced accuracy on the NLI task from 38% to 72%. This improvement demonstrates the efficacy of targeted fine-tuning for domain-specific tasks such as NLI, where pretrained LLMs may initially underperform.