Fine-Tuning Large Language Models with Efficient resources: A Case Study

Sheldon B. Lubar College of Business, University of Wisconsin-Milwaukee

ABSTRACT

This project presents the fine-tuning process of the Llama 2 language model, specifically focusing on the Llama-2-7b-chat-hf model. By utilizing parameter-efficient techniques such as QLoRA, we aim to optimize the model's performance using limited computational resources. By combining these approaches, the system is able to provide output using fewer computational resources. The dataset used in the project is guanaco-llama2-1k which is an already reformatted dataset to follow the standardized Llama 2 prompt template. Fine-tuning is executed with 4-bit precision quantization to reduce VRAM usage. The training progress is monitored using TensorBoard which allows real-time performance tracking. The aim of this project is to showcase that even with resource constraints, effective fine-tuning of large language models is achievable.

Keywords: Fine-Tuning, Llama 2, Parameter-Efficient Techniques, QLoRA, Hugging Face, 4-bit Precision.

1. INTRODUCTION

Large language models (LLMs) are widely used in many industries, fine-tuning these models has become very important step in order to give accurate industry specific answers.

LLMs, such as the GPT-3, LLaMA, BERT language models are widely used and are able to provide nearly accurate results. However, the computational resources required to fine-tune these models often create barriers for researchers and practitioners with limited infrastructure. It is important to address this challenge so that it is easier to adapt powerful AI technologies for many companies and industries ensure their usage in diverse contexts.

Traditional approaches to fine-tuning large language models typically involve resource-intensive processes that demand high-end GPUs or TPUs. While these methods can achieve state-of-the-art results, they are not always feasible in resource-constrained environments. Additionally, existing techniques may lack the efficiency needed to optimize models effectively without sacrificing performance.

To bridge this gap, in this project we are using parameter efficient fine-tuning technique called Quantized Low-Rank Adaptation (QLoRA), and we could observe that these techniques can used to solve the computational constraint. QLoRA is an extended version of LoRA that quantizes or standardizes the weight of each pretrained parameter to just 4 bits from the typical 32-bit weight. QLoRA offers significant memory savings and makes it possible to run an LLM on just one GPU.

In this project, we are fine-tuning the Llama-2-7b-chat-hf model using QLoRA, using a publicly available dataset on hugging face, this dataset is a preformatted dataset designed to align with Llama 2's prompt structure. Our approach combines efficient parameter adaptation with low-precision quantization to enable fine-tuning with limited resources. Throughout the process, TensorBoard is used to monitor training progress in real-time, providing insights into the model's performance.

2. LITERATURE REVIEW

Large Language Models (LLMs) have transformed Natural Language Processing (NLP) by excelling in diverse tasks, with models like GPT (Radford et al.) [1] and BERT (Devlin et al.) [2] marking key advancements. Meta's Llama-2-7b-chat-hf, a 7-billion-parameter model optimized for dialogue applications, represents the next generation of LLMs (Touvron et al.). [3] Fine-tuning, a critical process for adapting pre-trained models to specific tasks, leverages transfer learning to achieve superior results with minimal computational resources (Howard and Ruder et al).[4] This combination of large-scale pre-training and task-specific fine-tuning has driven significant progress in NLP applications such as sentiment analysis and question-answering.

Parameter-efficient fine-tuning (PEFT) techniques like Low-Rank Adaptation (LoRA) (Hu et al.) [5]and Quantized LoRA (QLoRA)** (Dettmers et al.) [6] have revolutionized LLM adaptation. LoRA freezes pretrained weights and introduces trainable low-rank matrices, enabling efficient fine-tuning by significantly reducing the number of updated parameters. QLoRA combines LoRA with 4-bit quantization, allowing fine-tuning of large models (e.g., 65B parameters) on a single 48GB GPU while retaining performance comparable to full 16-bit fine-tuning. These methods drastically lower memory and computational costs, making LLM adaptation accessible for resource-constrained researchers and developers.

Practical implementations of QLoRA highlight its efficiency. The Guanaco project (Dettmers et al.) [5] fine-tuned a 65B parameter model on a single 48GB GPU, achieving 99.3% of ChatGPT's performance in just 24 hours. This was achieved using 4-bit NormalFloat quantization, double quantization, and paged optimizers to optimize memory usage. The Guanaco-LLama2-1k dataset, a subset of the OpenAssistant Guanaco dataset preformatted for Llama 2 models, has proven effective for instruction-following tasks due to its diverse and well-structured samples (Touvron et al.) [6]. These advancements demonstrate that parameter-efficient techniques like LoRA and QLoRA enable fine-tuning large models on consumer-grade hardware without compromising performance (Hu et al., Dettmers et al.) [5]

3. METHODOLOGY

The main components of this project are

Data Preparation:

The dataset used in this project is guanaco-llama2-1k. This dataset is a pre-formatted dataset, it aligns with the Llama 2 prompt template. For this project we are using a pre-formatted dataset, and the volume of the dataset is relatively smaller because of the limited computational resources.

Model Configuration:

The base model, Llama-2-7b-chat-hf, is loaded from Hugging Face. For this project, we configure it for parameter-efficient fine-tuning. The model is adapted using Low-Rank Adaptation (LoRA), which reduces the number of parameters that need to be trained. This step includes setting key LoRA hyperparameters, such as the attention dimension (lora_r), scaling factor (lora_alpha), and a dropout rate (lora_dropout). These parameters help improve training efficiency while preserving performance. We are also using 4-Bit Quantization (QLoRA) to optimize memory usage; the model is loaded with 4-bit quantization enabled using the BitsAndBytes configuration. This reduces the memory footprint during training by lowering the precision of the model weights, allowing the training process to run on a single GPU while maintaining a high level of performance. The quantization type used is "nf4", and nested quantization is not applied for simplicity. The 4-bit quantization parameters are selected carefully to maintain a balance between memory efficiency and model accuracy.

Training and Monitoring:

The training process is done with the model processing 4 samples at once, and gradient checkpointing is used to manage memory more efficiently. The model is trained for a single epoch, with regular logging steps to keep track of progress. We are using TensorBoard to closely monitor the model's performance, this will help visualize key metrics like loss, accuracy, and gradient norms in real time.

Model Evaluation:

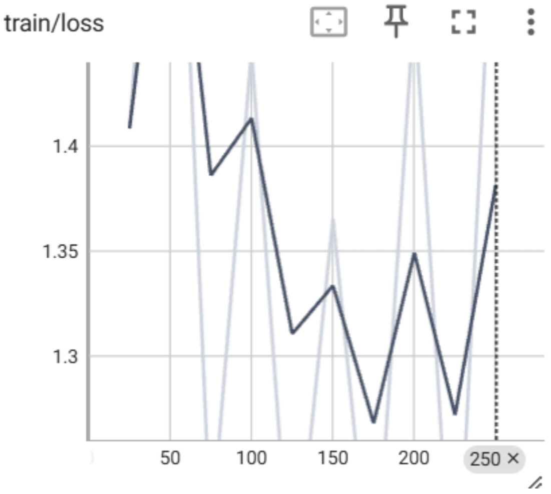
After training, the model is evaluated based on the performance of the fine-tuning task. The fine-tuned model is then saved. The trained model is stored as "Llama-2-7b-chat-finetune" for future inference or evaluation tasks.

Performance Visualization:

TensorBoard is used to visualize training logs in real-time. This provides an outlook on the model's performance during training and helps identify patterns and understand the model further.

4. RESULTS AND ANALYSIS

The Llama-2-7b-chat-hf model is fine-tuned using a QLoRA-based approach. This approach showed promising results in terms of training loss and processing efficiency.



The training loss fluctuated during the training process, between 1.66 and decreasing to 1.36 by the end of the epoch. Although this range is acceptable there were a few loss variations, this could be due to the complexity of the task. Despite these fluctuations, the model showed reasonable convergence with a training runtime of 1,454 seconds (approximately 24 minutes) and a throughput of 0.688 samples per second, which shows efficient usage of the hardware.

Although the model displayed promising results during training, additional tests on a test set will be helpful to evaluate the overall performance and compare it with other pre-trained models like GPT-3, BERT, and other versions of LLaMA.

5. DISCUSSION

During the training process the model was able to showcase efficient results by using parameter-efficient technique - QLoRA and 4-bit precision quantization. While the model's training loss decreased steadily, it is crucial to further evaluate its performance with a test set. The primary advantage of using this model is the memory efficiency and the reduced computational resources. Despite some training loss fluctuations, the overall training procedure was completed in a reasonable time frame, and the model showcases potential for deployment in resource-constrained environments.

The fluctuation in training loss suggests that further parameter tuning could be beneficial to improve the model's performance. Future research could be evaluation on a different dataset and comparisons with other fine-tuned models.

6. CONCLUSION

This project aimed to fine-tune the Llama-2-7b-chat-hf model for improved performance with limited resources in using parameter efficient fine-tuning method which is QLoRA and 4-bit quantization. While the model demonstrated successful fine-tuning with a reasonable decrease in training loss, the results cannot be justified without trying it for different datasets. We have also used memory-efficient technique to scale model performance with limited computational resources. To proceed further with this work the evaluation and optimization are important to ensure that the model achieves higher accuracy and relevance in different applications where the resources are limited.

FUTURE WORK

To explore further, we could fine-tune the Llama 2 model on a huge dataset and by using domain-specific datasets limited resources. This will help evaluate how well the model adapts and performs in other industries such as manufacturing, healthcare, finance, or law, where understanding the unique language and terminology is crucial.

REFERENCES

1. Radford, A. (2018). Improving language understanding by generative pre-training.
2. Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Pre-training of deep bidirectional transformers for language understanding. arXiv. *arXiv preprint arXiv:1810.04805*.
3. Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., ... & Scialom, T. (2023). Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.
4. Howard, J., & Ruder, S. (2018). Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*.
5. Dettmers, T., Pagnoni, A., Holtzman, A., & Zettlemoyer, L. (2024). Qlora: Efficient finetuning of quantized llms. *Advances in Neural Information Processing Systems*, *36*.
6. Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., ... & Scialom, T. (2023). Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.