Your Paper Title

Arlo Steyn	Andre van der Merwe	Stephan Delport
Department of Computer Science	Department of Computer Science	Department of Computer Science
University of Stellenbosch	University of Stellenbosch	University of Stellenbosch
24713848	24923273	242710083
24713848@sun.ac.za	24923273@sun.ac.za	242710083@sun.ac.za

October 23, 2024

Abstract

This is the abstract of the paper. Summarise the main points and contributions here.

Introduction

This is where you introduce the topic of your paper.

Web Scraping Cross Validated

This section of the report provides an overview of the Cross Validated website, details about the scraped data, and the tools used in the scraping process.

Overview of Cross Validated

Data Overview

Tools and Techniques for Data Scraping

Wrangling The Scraped Data

Data Cleaning

Text Normalisation

Structuring And Saving The Data

LLaMA Model Overview and Fine-tuning Process

We used the Large Language Model Meta AI (LLaMA) model, specifically the 3-billion parameter version, which is part of a family of transformer-based models. Llama is an open source LLM provided by Meta and we specifically used the LLaMA-3B-Instruct model for question-and-answering. LLaMA is designed for natural language processing tasks, and its architecture follows a standard decoder-only transformer, making it suitable for autoregressive text generation. This model is ideal for tasks like question answering, summarisation, and dialogue generation, but also has many other abilities and applications.

Model Architecture The LLaMA model is built on the transformer architecture, which uses self attention mechanisms to process input sequences and generate output. It is also autoregressive which means that it generates text one token at a time, using previously generated tokens to predict the next one.

Key components of the LLaMA model architecture include:

- Multi-Head Self-Attention: This allows the model to focus on different parts of the input sequence at the same time, which allows it to capture relationships between words across long sequences.
- Attention Heads: 32 heads per layer
- Attention Head Dimension: 128 dimensions per head
- Feed-Forward Networks (FFN): These are fully connected layers that transform the output of the self-attention mechanism.
- Feed-Forward Network Dimension: 11008 units (intermediate layer)
- Model Size: 3 billion parameters
- Number of Layers: 32 transformer layers
- Hidden Size: 4096 hidden units per layer
- Positional Encoding: Rotary positional encodings (RoPE)
- Layer Normalisation: Applied before self-attention and feed-forward layers

Pre-Trained Weights and Dataset The pre-trained weights for the LLaMA model were obtained from the Unsloth repository, which provided a specialised model version fine-tuned for question-and-answer and conversational tasks. The initial pre-training of LLaMA was done on a large corpus of diverse text, including books, research papers, and web data. This gives the model a strong foundation in understanding complex language structures and domain specific terminology.

We fine-tuned the model using a custom dataset of 39,668 question-and-answer pairs, which were scraped from 'Cross Validated' which focused on data science, artificial intelligence (AI), and machine learning (ML) topics. These questions ranged from simple definitions to more complex conceptual explanations. Fine-tuning allowed the model to better capture the complexity of these technical subjects, which makes it more accurate in responding and understanding to domain specific applications.

Fine-Tuning Process The version of LLaMA we used was fine-tuned for question-and-answering tasks, making it ideal for providing answers to questions related to data science, AI, and ML. Fine-tuning was performed using the Low-Rank Adaptation (LoRA) technique, which allows for efficient fine-tuning of large models by freezing most of the pre-trained model parameters and only updating a small subset of parameters. This drastically reduces the computational resources required, and also still achives high accuracy. We used a combination of gradient accumulation and batch processing to handle the relatively large dataset size during the fine-tuning phase.

The fine-tuning process involved adjusting several hyperparameters, including:

- Learning rate: Set to 2e-4 for optimal convergence without overfitting.
- Batch size: A batch size of 4 per GPU was used to maximise the use of available memory.
- Gradient accumulation: This was set to 4, allowing larger effective batch sizes without exceeding memory limits.

We also used 4-bit quantization during training to optimise memory usage, enabling us to fine-tune the model on hardware with limited GPU resources.

UnSloth Optimisation

What is UnSloth Optimisation

How UnSloth Optimisation Works

Benefits of UnSloth in Our Llama-3.2 Model

Results

Ethics

Conclusion

References

References

[1] Meta AI Research. *The LLaMA-3: Herd of Models*. Available at: `https://ai.meta.com/research/publications/the-llama-3-herd-of-models/`. Accessed: 2024-10-13.

[2] Unsloth Documentation Team. *Unsloth AI Documentation*. Available at: `https://docs.unsloth.ai/`. Accessed: 2024-10-13.

[3] Unsloth AI Tutorials. *How to Fine-Tune Llama 3 and Export to Ollama*. Available at: `https://docs.unsloth.ai/tutorials/how-to-finetune-llama-3-and-export-to-ollama`. Accessed: 2024-10-13.

[4] Stack Exchange Inc. *Cross Validated - Statistical Analysis, Machine Learning, Data Mining, and Data Visualization*. Available at: `https://stats.stackexchange.com/`. Accessed: 2024-10-12.

[5] Google Search. *Search Results: Cross Validated robots.txt*. Available at: `https://www.google.com/search?q=cross+validated+robots.txt&rlz=1C1GCEU_en-gbZA1068ZA1068&oq=cross+val`. Accessed: 2024-10-12.