Student: Weiyao Li
Research Mentor: Bruce Yang

Introduction

Large language models (LLMs) have shown the potential of revolutionizing natural language processing tasks in diverse domains, sparking great interest in finance. Accessing high-quality financial data is the first challenge for financial LLMs (FinLLMs). While proprietary models like BloombergGPT have taken advantage of their unique data accumulation, such privileged access calls for an open-source alternative to democratize Internet-scale financial data. FinGPT is an open-source large language model for the finance sector. Unlike proprietary models, FinGPT takes a data-centric approach, providing researchers and practitioners with accessible and transparent resources to develop their FinLLMs. Financial sentiment analysis, the task of discerning investor sentiment from financial articles, news, and social media, is an essential instrument for comprehending and forecasting market movements. Conventional models often struggle with several difficulties, including insensitivity to numeric values, difficulties interpreting sentiment without explicit context, and the challenges associated with financial jargon, multilingual data, temporal dependency, insufficient labeled data, and the inherent noise in social media data.

About the training dataset, the FinGPT Sentiment Training dataset is a collection of financial news, each associated with a sentiment classification such as neutral, positive, moderately negative, etc. It is structured to train LLM models, specifically for sentiment analysis tasks within the financial domain. The dataset not only provides the financial statements but also includes instructions that guide the model on how to interpret and classify the sentiment of each piece of news. This structure facilitates the development of models that can accurately determine the sentiment conveyed in financial news, which is critical for financial market analysis, investment strategy, and economic research. Regarding the inference dataset, The Financial PhraseBank (FPB), FIQA (pauri32/fiqa-2018), TFNS (zeroshot/twitter-financial-news-sentiment), and NWGI (oliverwang15/news_with_gpt_instructions) datasets are integral resources in financial sentiment analysis and natural language processing. FPB specializes in sentiment annotation of financial news, making it ideal for market trend analysis. FIQA, part of a

challenge, offers microblogs and headlines with sentiment scores, aiding in financial decision-making algorithms. TFNS focuses on Twitter data, capturing real-time market sentiments, while NWGI provides a unique blend of news articles integrated with GPT-generated instructions, enhancing automated news analysis and generation capabilities in the financial sector.

Objectives

In my research, I explored various tuning strategies to address the volatile characteristics of financial data, ensuring the updated models maintain their precision and applicability. Working as a Machine Learning Engineer, I fine-tuned 3 SOTA Large Language Models (LLMs) — Baichuan2-7B, ChatGLM2-6B, and Qwen-14B-LoRA — employing Supervised Fine-tuning (SFT) and Low-Rank Adaptation (LoRA) techniques with a distributed training process with multiple GPUs. My approach involved instruction-tuning to tailor these broad-spectrum LLMs for the nuanced task of financial sentiment analysis, particularly improving their comprehension of numerical data and sector-specific contexts. The Baichuan2-7B model was the primary focus of this report. Detailed code and findings are thoroughly documented and publicly accessible at FinGPT-AI-Research, with a comprehensive Git commit history provided for transparency.

Results

Following directives from Bruce, I've crafted 2 in-depth technical reports on the process and outcomes of Supervised Fine-Tuning (SFT) applied to Large Language Models (LLMs). These documents, along with the source code for one of them, are now accessible on Medium for public review. In my piece titled "LoRA, Financial Sentimental Analysis with Baichuan2–7B," I detail how I've tailored the Baichuan2–7B model to better grasp financial discourse through Low-Rank Adaptation (LoRA), a technique that refines open-source LLMs for the financial sector.

LoRA significantly improves the models' performance with financial datasets, allowing for a more efficient and nuanced analysis. It also sets a new standard for evaluating model performance across various tasks, whether they're tailored for specific functions, involve multiple tasks simultaneously, or require the model to make informed guesses without prior training (zero-shot instruction tuning). This approach not only sharpens the models' ability to

parse financial sentiment but also serves as a blueprint for future advancements in the intersecting fields of language processing and financial analysis.

In our computational experiment, I trained the Baichuan2-7B model across an array of 8 A800 GPUs, employing a batch size of 4 per device to optimize the training process with DeepSpeed. The model underwent 8 training epochs with a learning rate of 0.0002, utilizing a linear scheduler to adjust this rate over time. Precision during training was maintained at a 16-bit floating point (fp16), balancing computational efficiency with numerical precision. Model performance was periodically assessed, and checkpoints were saved with a frequency factor of 0.1, with a policy in place to retain only the single most recent checkpoint.

The model's performance was evaluated across 4 distinct sentiment analysis datasets—fpb, fiqa, tfns, and nwgi—yielding a comprehensive accuracy profile. Accuracy (Acc.) was tracked through several lenses, including the overall accuracy, macro, and micro F1 scores, which consider both the precision and recall of the model. Additionally, weighted scores were calculated to account for class imbalance, and a proprietary Bloomberg Performance Test (Bloomberg PT) was applied to gauge the model's real-world applicability. For the Financial PhraseBank (fpb) dataset, the model achieved an accuracy (Acc.) of 85.31% with a macro F1 score of 84.28% and a micro F1 score mirroring the accuracy. In the Financial Question Answering (fiqa) dataset, the accuracy was slightly lower at 84.72%, with a macro F1 score of 75.04% and a micro F1 score of 84.72%. The Twitter Financial News Sentiment (tfns) dataset saw the model perform with an accuracy of 88.98%, a macro F1 score of 86.39%, and a micro F1 score equal to the accuracy. Lastly, for the News With GPT Instructions (nwgi) dataset, the accuracy was 65.65%, the macro F1 score stood at 65.93%, and the micro F1 score was also 65.65%.

Furthermore, weighted Bloomberg Performance Tests (BloombergG PT) yielded the following scores: 85.28% for fpb, 86.12% for fiqa, 89.04% for tfns, and 64.86% for nwgi, indicating a strong alignment with professional financial analytics benchmarks. These metrics collectively demonstrate the model's adeptness at financial sentiment analysis, with robust performance across different datasets, showcasing its utility in various financial analytics applications.

In the report "Beginner's Guide to FinGPT: Training with LoRA and ChatGLM2–6B," I submitted the coding part, meticulously running through the complete process of training a FinGPT model with detailed code examples. The guide begins with Part 1, focusing on data preparation through code, including initializing directories, loading and preparing the dataset, and concatenating and shuffling it for a diverse training sample. Part 2 details the dataset formatting and tokenization, crucial for converting raw data into a machine-learning-friendly format, along with code to save the prepared dataset. Part 3 involves setting up the FinGPT training parameters with LoRA on ChatGlm2–6b, including training arguments, quantization configurations, model loading and preparation, and LoRA configuration, all provided in code format. Part 4 demonstrates the coding process for loading data and executing the FinGPT training, covering training configuration setup, the actual training process, and model saving and downloading. Finally, Part 5 focuses on inference and benchmarking stages through code, where the model is loaded, benchmarks are run to assess its performance, and its results are compared with FinGPT V3.1. This guide serves as a step-by-step, code-centric roadmap for beginners to effectively train their FinGPT models.

Main Things Learned

Data-Centric Approach in FinLLMs: A key takeaway from my research was the significance of a data-centric approach when training Financial Large Language Models (FinLLMs), particularly FinGPT. This approach is critical in ensuring that the training data accurately reflects the intricacies and nuances of financial contexts and sentiments. It highlighted the need for meticulously curated datasets that could provide a realistic and comprehensive basis for training models to understand and analyze financial language effectively.

Advanced Model Tuning Techniques: Another important aspect of my learning was the application of advanced tuning techniques like Supervised Fine-tuning (SFT) and Low-Rank Adaptation (LoRA). These techniques were instrumental in refining Large Language Models for specialized tasks, such as financial sentiment analysis. By employing these methods, I was able to enhance the models' ability to process numerical data and understand sector-specific contexts, which are crucial in the financial domain.

Challenges in Financial Sentiment Analysis: The research also provided me with deep insights into the unique challenges associated with financial sentiment analysis. I learned about the difficulties in processing financial jargon, handling multi-lingual data, and dealing with the inherently noisy nature of social media data. These challenges underscored the need for specialized models that are tailored to the specific requirements of financial sentiment analysis.

Evaluating Model Performance: A significant part of my learning involved understanding and applying various performance evaluation metrics. I learned to use metrics like overall accuracy, macro and micro F1 scores, and the proprietary Bloomberg Performance Test to assess the effectiveness of the models. These metrics provided a multi-dimensional view of the models' performance, helping to gauge their ability to accurately interpret and analyze financial language.

Practical Application of LLMs: Working with the Baichuan2-7B model, I gained practical experience in applying Large Language Models to real-world scenarios. A key aspect of this was learning to balance computational efficiency with numerical precision, especially the use of 16-bit floating point precision in training. This experience was crucial in understanding the technical aspects of deploying LLMs in a practical, efficient manner.

Model Training Process: The hands-on process of distributed training process across multiple GPUs, optimizing batch sizes, and adjusting learning rates using a linear scheduler was a vital part of my learning. This experience shed light on the complexities and technicalities involved in efficiently training large-scale models, emphasizing the importance of resource management and optimization strategies in machine learning.

Importance of Code Documentation and Transparency: Through this research, I also learned the importance of thorough documentation and transparency in AI projects. By making my code and findings publicly accessible, I realized the value of contributing to the open-source community. This transparency not only fosters collaboration but also ensures the reproducibility and reliability of AI research.

Overall, this research journey in the field of financial sentiment analysis using Large Language Models provided me with a comprehensive understanding of the various facets involved in AI and machine learning, specifically in the context of the financial sector. The experience was enriching and provided a strong foundation for my future endeavors in this field.