FinGPT - Transformer Prediction

Joe C, Yash G, Ethan Z, Dash H, Cal R Supervised by Prof. Xiao-Yang Liu

4/23/2024

Introduction

In the dynamic world of financial markets, accurate stock price prediction is paramount. Our project focuses on evaluating and enhancing Meta's Llama 2-7B model for this task. Beginning with a baseline assessment of its predictive ability, we fine-tuned the model using relevant datasets to improve performance. We also explored the intriguing possibility of evolving the enhanced model into a stock trading bot. Through comparative analysis, we aimed to quantify the improvement achieved. By leveraging advanced AI and ML techniques, our goal is to contribute to more informed and strategic decision-making in financial markets, ultimately empowering investors with enhanced predictive capabilities for better outcomes.

Overall Vision:

Step 1 - Prediction:
We tested Llama 2's baseline ability to analyze stock prices and make predictions

Step 2 - Improvement:
We fine tuned the model in order to improve its performance on the current task.

Step 3 - Analysis:
We evaluated prediction abilities of the baseline Llama 2 model vs our retrained model to assess improvement. We are currently working on quantifying the improvement of our model.

Improvement Process

Data Processing:

- We iterated through datasets from Yahoo Finance and cleaned and formatted them correctly for our fine-tuning methods.

Tokenizing and Modifying Input Lengths:

- We used Hugging Face tokenization methods on our dataset. This step breaks down our text input into smaller units called tokens which are fed into the model for training. The tokenizer is also used to prompt the model after training, ensuring consistent interaction with the model.
- We tokenized the dataset and analyzed the distribution of input lengths (shown right) in order to determine a length to truncate all inputs to.

Setting up LoRA (Low-Rank Adaptation):

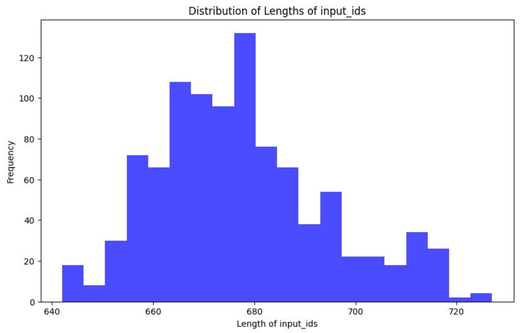
- We reconfigured the model using methods from Hugging Face's PEFT library (Parameter-Efficient Fine-Tuning) in order to optimize training by reducing memory consumption.
- This process works by modifying certain inner layers of the model to reduce the number of parameters while maintaining expressiveness.

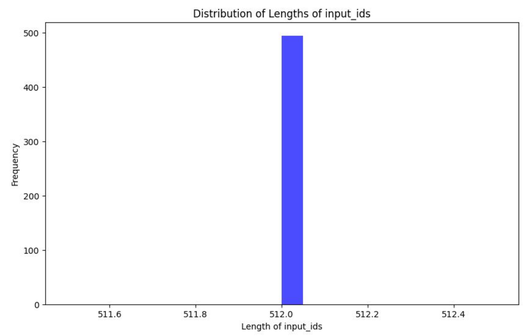
Training:

- We trained the model with a GPU on Google Colab

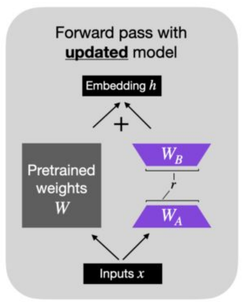
Objectives (Goals)

- Assess model's capabilities
  - Create testing datasets from largely traded stocks
  - Assess performance by calculating predicted data accuracies
- Improve prediction capabilities by 5%
  - Identify highest performing model parameters
  - Use datasets to fine-tune open source large language models
  - Integrate external data sources, such as macroeconomic indicators, industry-specific data, or news/media sentiment
  - Investigate methods for preprocessing and incorporating these additional features into the model.
- Achieve 8% hypothetical ROI
  - Identify different metrics to verify the effectiveness of the return of our model



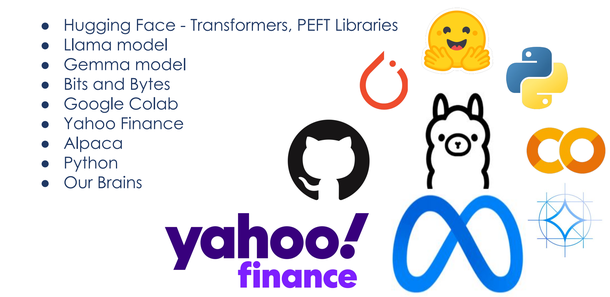


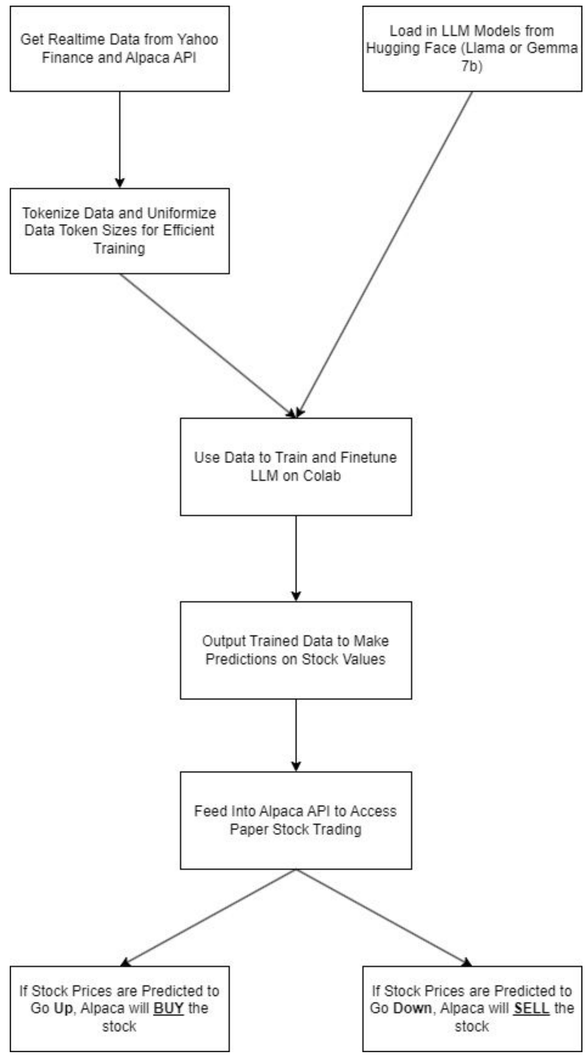
LoRA weights, W_A and W_B, represent ΔW



Materials & Methods

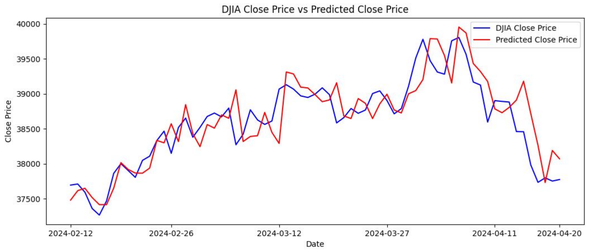
- Hugging Face - Transformers, PEFT Libraries
- Llama model
- Gemma model
- Bits and Bytes
- Google Colab
- Yahoo Finance
- Alpaca
- Python
- Our Brains





Results (Progress)

- Successfully fine tuned and trained Meta's Llama2-7b Model and Google's Gemma-7b Model
- Created a Pipeline to take in Yahoo Finance and Real Time Stock information
- Integration of the Alpaca API to simulate paper trading in real time to verify model decision making
- Formal Testing is still required to evaluate model performance (Using/Finding evaluation metrics)



Conclusions

To start, for the first couple weeks of out process was dedicated to learning and growing our knowledge on the Transformer models themselves. From the knowledge we gained we were able to then test Meta's Llama 2-7B model on how it could evaluate a certain stock. We found that Meta's Llama 2-7B model had a good baseline for evaluating stock prices but we knew we could improve it. From here we were able to retrain the Llama model by using new datasets, manipulating the input queries, and changing the token size. After this we split our focus on evaluating the model and using the model. For using the model we utilized the Alpaca trading API to implement a simple stock analyzer. We were able to connect to Alpaca via the users keys, then the user could pick which stock they wanted out model to analyze and it would output it's analysis based on the last week of data. The evaluation group started to compare the accuracy of the model to real world changes in the stock market. For future outlooks we would like to come up with an algorithm to evaluate our model on a larger scale and improve its training further. Additionally, we can further our use of the model by finding new applications for it's analysis.

References

"Yes, Transformers Are Effective for Time Series Forecasting (+ Autoformer)," n.d. https://huggingface.co/blog/autoformer.

Jiang, Yushan, Zijie Pan, Xikun Zhang, Sahil Garg, Anderson Schneider, Yuriy Nevmyvaka, and Dongjin Song. "Empowering Time Series Analysis With Large Language Models: A Survey." arXiv.org, February 5, 2024. https://arxiv.org/abs/2402.03182.

"Large Language Models Are Zero-Shot Time Series Forecasters." arXiv.org, October 11, 2023. https://arxiv.org/abs/2310.07820.

Liu, Yang, Jiahuan Cao, Chongyu Liu, Kai Ding, and Lianwen Jin. "Datasets for Large Language Models: A Comprehensive Survey." arXiv.org, February 28, 2024. https://arxiv.org/abs/2402.18041.

Song, Xingyou, Oscar Li, Chansoo Lee, Bangding Yang, Daiyi Peng, Sagi Perel, and Yutian Chen. "OmniPred: Language Models as Universal Regressors." arXiv.org, February 22, 2024. https://arxiv.org/abs/2402.14547.

Carroll, Harper. "How to Fine-Tune Llama 2 on Your Own Data." How to Fine-Tune Llama 2 on Your Own Data, brev.dev/blog/fine-tuning-llama-2-your-own-data. Accessed 16 Apr. 2024.

Acknowledgements

Our project was inspired and guided by Professor Xiao-Yang Liu. We thank him for his assistance.

RCOS Helpers

Gnahz Nahte