Tue, 06/04/2024 - 16:22

Forecasting Financial Asset Returns with Large Language Models, GPT-TS Case Study

Ruslan Melnikov

Forecasting financial time series data can indeed be challenging due to various factors such as volatility, non-linearity, seasonality, and external market influences. These complexities make it crucial to utilize sophisticated techniques and models to generate accurate predictions.

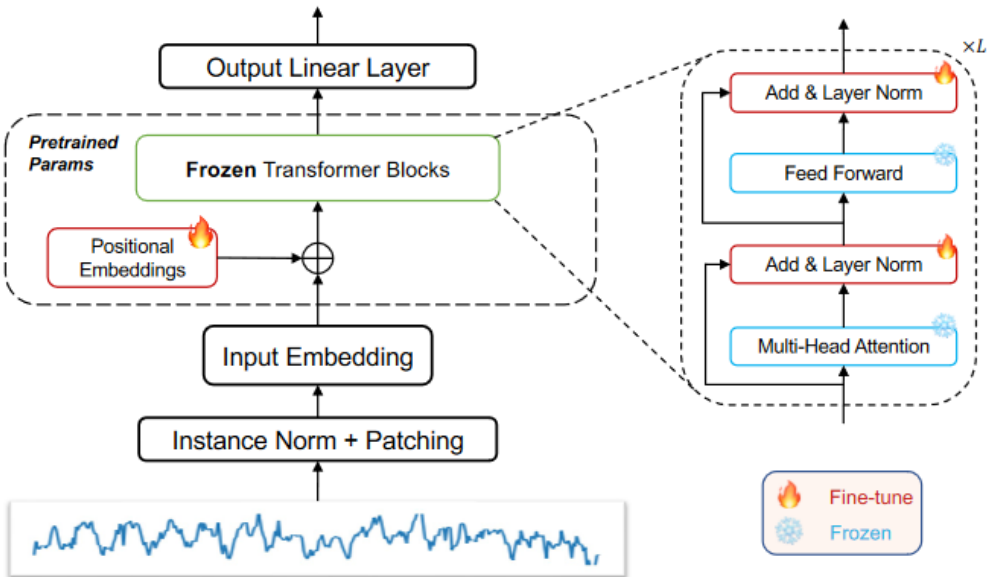
Common approaches to forecasting financial time series include using statistical models, such as ARIMA and its variations, to analyze linear dependencies and trends, or employing machine learning algorithms like RNN or LSTM to uncover complex patterns in the data. Alternative neural network architectures for time series forecasting, such as N-HITS, N-BEATS, or Transformers, offer various approaches and techniques for modeling time series data, each of which has its own strengths and capabilities. Transformers were originally developed for processing sequential data such as text; however, they can also be successfully applied to time series analysis. Their ability to capture long-term dependencies and process context at different levels can be useful for accurate time series forecasting.

In this article, we explore how large language models, specifically GPT-2 (Generative Pre-trained Transformer), can be adapted and applied to forecasting the returns of financial assets. Large language models based on the transformer architecture have number of the advantages. We provide a description of the process of fine-tuning the model and approaches to improving forecast quality. Finally, we assess the potential of using GPT-2 with fine-tuning for forecasting returns for two asset classes (ETF-SPY) and construct an ETF portfolio based on the model's forecasts.

Fine-tuning GPT-2 for time series forecasting

The GPT-2 model, with its transformer architecture consisting of millions of parameters, exhibits remarkable capabilities in understanding and generating

human-like text based on input data. A pre-trained model from the language domain can be adapted for time series analysis with minimal modifications through precise parameter tuning.



The architecture retains the positional embedding layers and self-attention blocks from pre-trained models. Since the self-attention blocks and FFN (feedforward neural networks) hold the majority of the learned knowledge from pre-trained language models, these blocks are typically frozen during fine-tuning. However, positional embeddings and normalization layers are left unfrozen and are retrained during the fine-tuning process. As a result, pretrained parameters are transferred to time series forecasting tasks.

To understand the context in sequential data, a patching method is used, where adjacent time steps are aggregated to form a single token based on a patch. Patching allows for a significant increase in the historical input time horizon while maintaining the same token length and reducing information redundancy for transformer models.

By freezing certain components and unfreezing others, pretrained parameters are effectively transferred for the task of time series forecasting. This helps leverage the knowledge acquired by the pretrained language model on large text corpora to improve forecasting performance. Such an approach allows for the effective adaptation of pretrained language models, such as GPT-2, to time series forecasting tasks, utilizing their knowledge while precisely fine-tuning specific components to better match the data characteristics.

Approaches to improving forecast quality

In this section, we will describe two approaches that can improve the accuracy of time series forecasting. Both are based on providing additional information to the model, which can further enhance forecasting accuracy.

First and foremost is the decomposition of the time series. Decomposition helps in understanding the underlying structure of the time series. Knowing how trend and seasonality influence the data can provide insights that are useful for building more accurate forecasting models.

For decomposition we use STL method (Seasonal-Trend decomposition using LOESS). STL is employed to decompose a time series into three components: trend, seasonality, and residual. This method utilizes LOESS (locally estimated scatterplot smoothing) to extract smooth estimates of the three components. The obtained decomposition of the time series is tested in two ways: first, by reassembling the overall time series as the sum of the decomposed components, which triples the length of the series; second, by using each component as a separate channel, thereby forming a tensor of increased dimensionality.

The second approach for improving forecast quality is optimizing the number of training steps or the length of the training window. The goal of time-series forecasting is to predict the values for the upcoming H timestamps based on the observed values from the preceding K timestamps. When optimizing the window length, we take into account that the architecture of transformers, including models like GPT, allows them to effectively capture long-range dependencies in data. This capability is particularly useful when dealing with long time series. The self-attention mechanism in transformers enables each token in the input sequence to attend to all other tokens, regardless of their position, allowing dependencies to be captured over long distances. This means that longer time series can provide the model with more context for learning, enabling it to make more informed predictions.

Assessing the accuracy of forecast returns model

In this section we will be forecasting future data points for the following assets: SPY and GLD ETFs, and AAPL and MSFT stocks. We use minute-level time series of returns for the period from 2013 to 2018.

We perform fine-tuning annually with the data divided into subsequences of two days (K=780 minutes) for each asset over the last two years. We then forecast the return for the next hour (H=60 minutes) and make comparisons relative to a

naive approach using MSE and MAE metrics. Naive approach assumes that the return of the next hour is equal to the previous hour.

Below are the accuracy estimates of the forecasts for each year, along with the differences relative to the naive approach:

ETFs

SPY	MSE	MAE	MSE naïve	MAE naïve	Percent change MSE
2013	0.07	0.18	0.13	0.25	48.91%
2014	0.07	0.18	0.13	0.25	46.95%
2015	0.14	0.24	0.27	0.33	47.44%
2016	0.11	0.22	0.20	0.30	44.99%
2017	0.03	0.12	0.06	0.16	47.71%
2018	0.16	0.26	0.29	0.36	45.41%

GLD	MSE	MAE	MSE naïve	MAE naïve	Percent change MSE
2013	0.26	0.28	0.46	0.42	43.62%
2014	0.12	0.21	0.24	0.31	48.96%
2015	0.12	0.21	0.24	0.32	47.76%
2016	0.15	0.23	0.29	0.35	47.44%
2017	0.07	0.16	0.13	0.24	48.38%
2018	0.06	0.16	0.12	0.23	47.84%

Stocks

AAPL	MSE	MAE	MSE naïve	MAE naïve	Percent change MSE
2013	0.45	0.41	0.84	0.58	46.87%
2014	0.29	0.33	0.58	0.47	48.76%
2015	0.43	0.42	0.82	0.58	46.95%
2016	0.36	0.37	0.68	0.52	47.31%
2017	0.20	0.27	0.39	0.39	48.03%
2018	0.43	0.43	0.79	0.60	45.43%

MSFT	MSE	MAE	MSE naïve	MAE naïve	Percent change MSE
2013	0.37	0.36	0.70	0.50	46.52%
2014	0.22	0.32	0.43	0.45	48.54%
2015	0.47	0.42	0.87	0.57	45.73%
2016	0.36	0.37	0.68	0.52	46.89%
2017	0.16	0.25	0.30	0.35	47.72%
2018	0.45	0.45	0.87	0.64	48.02%

We observe that for each year, the forecast accuracy exceeds the naive estimate by almost two-fold.

In the next step, we analyze the approaches to improving estimation quality discussed earlier, including optimizing the number of training steps and decomposing time series using STL. For these methods, we will continue to evaluate accuracy using MSE. Additionally, we will assess the profitability of simple strategies implemented based on the model's forecasts: a 'Long only' strategy, which involves buying when the forecasted return for the next hour is positive; a 'Long-short' strategy, which adds a short position when the forecasted return is negative; and a 'Naive long only' strategy, which involves buying when the return of the previous hour is positive.

In the approach that involves optimizing the number of training steps, we increase the number of training steps and present the results for 2-6-20 days. We observe a tendency where an increase in steps leads to a decrease in prediction accuracy. At the same time, this increase in steps improves the profitability metrics. This is valid for both the ‘Long only’ and ‘Long-Short’ strategies.

Seqlen SPY	MSE next hour return
2 Days	0.0572
6 Days	0.0589
20 Days	0.0609

2 Days - seq len

Strategy	Long Only	SPY	Long-Short	Naïve Long Only
Sharp	0.648	1.332	-0.493	1.039
CAGR	4.39%	12.15%	-3.37%	7.38%
Volatility	8.54%	11.79%	11.79%	8.27%

6 Days - seq len

Strategy	Long Only	SPY	Long-Short	Naïve Long Only
Sharp	1.317	1.333	0.610	1.040
CAGR	9.52%	12.16%	6.88%	7.39%
Volatility	7.54%	11.79%	11.79%	8.27%

The results of time series decomposition testing using STL for the SPY ETF over the interval 2013-2018 for two methods: the combined time series (sum of decomposed components) and each component as a separate channel. The test reveals that while these approaches improve accuracy, they also deteriorate the profitability metrics of the base strategies used in the analysis of approach quality.

Method	MSE next hour return
Without decomposition	0.0589
The combined time series	0.0586
Separate channel	0.0572

Without decomposition:

Strategy	LongOnly	SPY	Long-Short	Naïve LongOnly
Sharp	1.317	1.333	0.610	1.040
CAGR	9.52%	12.16%	6.88%	7.39%
Volatility	7.54%	11.79%	11.79%	8.27%

The combined time series:

Strategy	LongOnly	SPY	Long-Short	Naïve LongOnly
Sharp	0.676	1.332	-0.411	1.039
CAGR	5.18%	12.15%	-1.78%	7.38%
Volatility	8.00%	11.79%	11.79%	8.27%

Separate channel:

Strategy	LongOnly	SPY	Long-Short	Naïve LongOnly
Sharp	0.648	1.332	-0.493	1.039
CAGR	4.39%	12.15%	-3.37%	7.38%
Volatility	8.54%	11.79%	11.79%	8.27%

In the end, to evaluate the quality of using the GPT-2 model, we construct an equal-weighted portfolio for a universe of 6 ETFs (SPY, GLD, AGG, VOO, VWO, EFA) for the ‘Long only’ strategy, which entails buying when the forecasted return for the next hour is positive. Forecasts for each ETF were updated every hour, based on a training window of 20 days.

For comparison, we use the returns from an equally weighted portfolio of all ETFs as a baseline portfolio (benchmark) for the same period.

ETF Portfolio	Sharp	CAGR	Volatility
Portfolio Baseline	0.54	4.48%	8.28%
Long Only Strategy	0.67	3.26%	4.87%

As a conclusion, our experiment demonstrates the potential of using advanced natural language processing models, such as GPT-2, for analyzing financial time series. After fine-tuning, the GPT-2 model outperforms the naive forecast by more than 50% across all classes of models. The portfolio, formed based on the model's forecasts for a universe of 6 ETFs, achieves a Sharpe ratio of 0.67, compared to the equally weighted portfolio's Sharpe ratio of 0.54. While the model's accuracy has significantly surpassed naive estimates, the return of the constructed portfolio remains low.

Although our test resulted in an increased Sharpe ratio accompanied by a decrease in overall portfolio profitability, the significant improvement in forecast accuracy highlights the substantial potential for creative approaches in using LLM models. For example, in our test, we limited ourselves to the GPT-2 model, but it is possible to use a wide range of alternative large language models (LLMs) that may better capture the dynamics of specific time series. Additionally, increasing the number of model parameters, combined with extending the forecast window from an hour to several days, can further improve the quality of the predictions.