ILLINOIS TECH
College of Computing

Time Series Forecasting with Large Language Models

Anwar Benhnini   Ioan Raicu

Illinois Institute of Technology

Introduction

Traditional time series forecasting models are designed from the ground up for each task, requiring domain-specific architectures, training routines, and tuning. Large Language Models (LLMs) have shown remarkable capabilities in understanding and generating complex sequences, suggesting potential for time series forecasting.

Research Question: Can we leverage the general reasoning capabilities of LLMs to forecast time series data without retraining them?

This project explores adapting frozen pre-trained LLMs for time series forecasting using the Time-LLM framework, which bridges the modality gap between continuous numerical data and discrete textual tokens.

Time-LLM Framework

The Time-LLM framework adapts frozen pre-trained LLMs for time series forecasting without requiring computationally expensive fine-tuning. Key components:

- **Input Embedding:** Normalize time series data and divide into patches
- **Reprogramming:** Transform patch embeddings to align with LLM's word embeddings using learned text prototypes
- **Prompt-as-Prefix (PaP):** Guide the LLM with dataset context, task instructions, and input statistics
- **LLM Processing:** Use frozen pre-trained LLM (Llama-7B) to process the input sequence
- **Output Projection:** Extract and project LLM output representations to produce forecasts

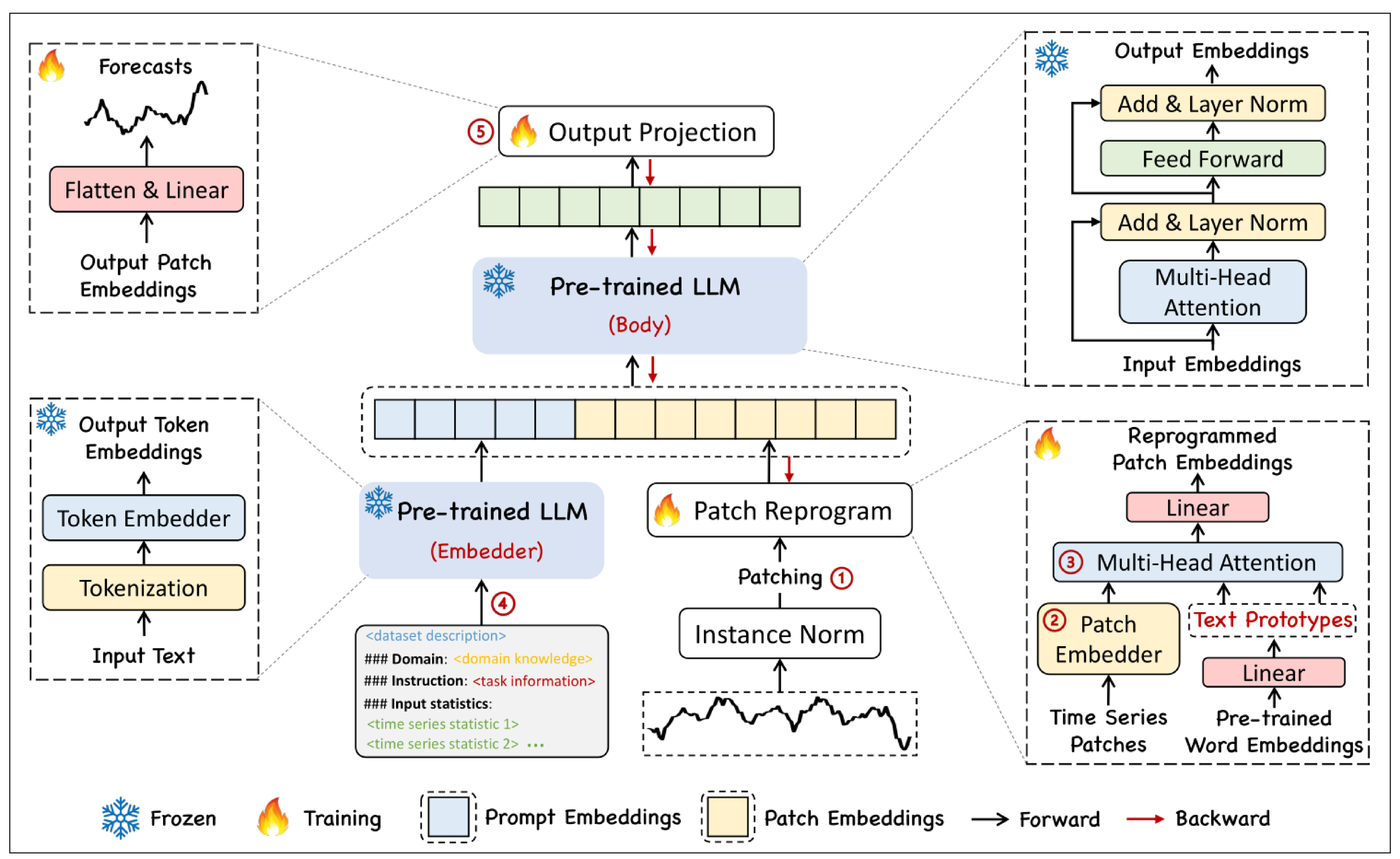

Figure 1. Time-LLM Framework Architecture

References

[1] Ming Jin, Shiyu Wang, Lue Ma, Zhixuan Chu, Jiaming Zhang, Xiang Shi, Pin-Yu Chen, Yuxuan Liang, Ying-Fan Li, Shirui Pan, and Qingsong Wen. Time-llm: Time series forecasting by reprogramming large language models. *International Conference on Learning Representations (ICLR)*, 2024.

Key Findings

- State-of-the-art accuracy on both long-term and short-term forecasting
- Especially strong performance in few-shot and zero-shot settings
- Uses fewer than 6.6 million trainable parameters (only around 0.2% of the total parameters in Llama-7B)
- Removing patch reprogramming or prompt-as-prefix results in 8–17% worse performance
- Most helpful prompt component: input statistics (trend, lag, range)

Hyperparameter Sensitivity Analysis

As the model scales, the performance improves:

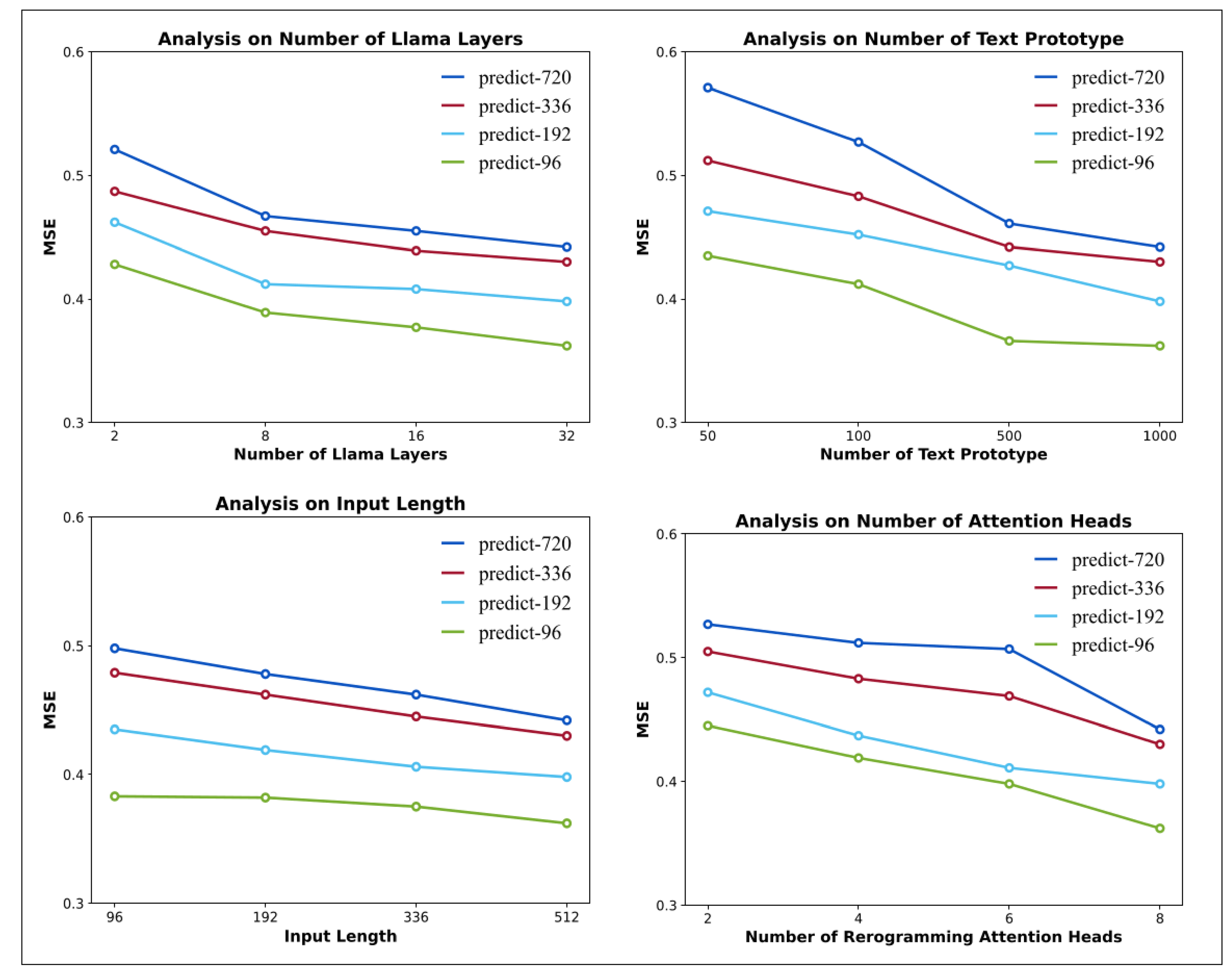

Figure 2. Analysis of hyperparameter sensitivity on ETTh1 dataset

Experimental Setup

- **LLM Backbone:** Llama-7B (8 first layers)
- **Programming Language:** Python 3.11
- **Deep Learning Framework:** PyTorch
- **Libraries:** accelerate, numpy, pandas, transformers, deepspeed
- **Hardware:** 4 NVIDIA V100 GPUs
- **Standard Datasets:** ETTh1, ETTh2, ETTm1, ETTm2, Weather, Electricity, Traffic, ILI, M4, M3-Quarterly

Evaluation Metrics:

- Mean Squared Error (MSE): $MSE = \frac{1}{H}\sum_{h=1}^{H}(Y_h - \hat{Y}_h)^2$
- Mean Absolute Error (MAE): $MAE = \frac{1}{H}\sum_{h=1}^{H}|Y_h - \hat{Y}_h|$

CRYPTEX Dataset Application

The project applied Time-LLM to the historical cryptocurrency price data (Cryptex) from the Data-Sys Lab.

What is CRYPTEX?

- Cryptocurrency time-series data extracted from Binance exchange since September 2019
- OHLC (Open, High, Low, Close) and volume data at fine-grained resolution
- Nearly four years of hourly-level resolution data

Domain Prompt Used: "The Binance Bitcoin Hourly Candlesticks (BTC-H) dataset captures granular financial data from the Binance.us cryptocurrency exchange. It spans nearly four years, from September 2019 to July 2023, with hourly-level resolution. Each record contains second-to-second updates for key trading indicators: Open, High, Low, Close (OHLC) prices and traded volume in USDT..."

Cryptex Experimental Results

Experiment	Input Length	Forecast Horizon	MAE	Notes
Exp 1	512	96	0.0138	Long-term forecast
Exp 2	192	24	0.0126	Medium-term forecast
Exp 3	96	6	0.0112	Short-term forecast

Table 1. Summary of Cryptex experiments

Experiments with Cryptex data yielded MAE losses in the order of 0.01, which represents promising results using only historical price data.

Sentiment Integration Experiment

We explored integrating external sentiment analysis into the Time-LLM framework by injecting sentiment prompts into the Domain section of the Prompt-as-Prefix:

Positive Sentiment: "Bitcoin has surged following increased institutional adoption and favorable regulatory developments in major markets. Investor sentiment is strongly bullish..."

Negative Sentiment: "Recent headlines report a crackdown on cryptocurrency exchanges, combined with sharp sell-offs and declining on-chain activity. Investor confidence is falling..."

Neutral Sentiment: "Bitcoin is trading within a narrow range as market participants await further economic signals. News coverage remains balanced..."

Results showed limited influence of artificial sentiment cues on model predictions, possibly because the LLM already possesses predefined knowledge of Bitcoin price movements during the dataset's historical window.

Future Work

- Develop methods for Time-LLM to provide probabilistic forecasts and natural language explanations
- Further investigate integrating external information (like news sentiment) into the framework
- Identify optimal data granularity for cryptocurrency forecasting