

Time-LLM: Time Series Forecasting by Reprogramming Large Language Models

Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y. Zhang, Xiaoming Shi,
Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, Qingsong Wen
ICLR 2024

발표자: 안도형

25. 07. 31.

Contents

- Introduction
- Methods
- Conclusion

Introduction

Introduction

- **시계열 예측 모델의 한계**

- 시계열 예측은 수요예측, 재고 최적화, 기후 모델링 등 다양한 산업에서 활용
- But, 이런 시계열 예측은 특정 도메인 전문지식과 맞춤형 모델 개발이 필요하여 일반화가 어려움
- GPT, Llama 같은 LLM은 다양한 NLP 작업을 Few-shot, Zero-shot 방식으로 수행가능

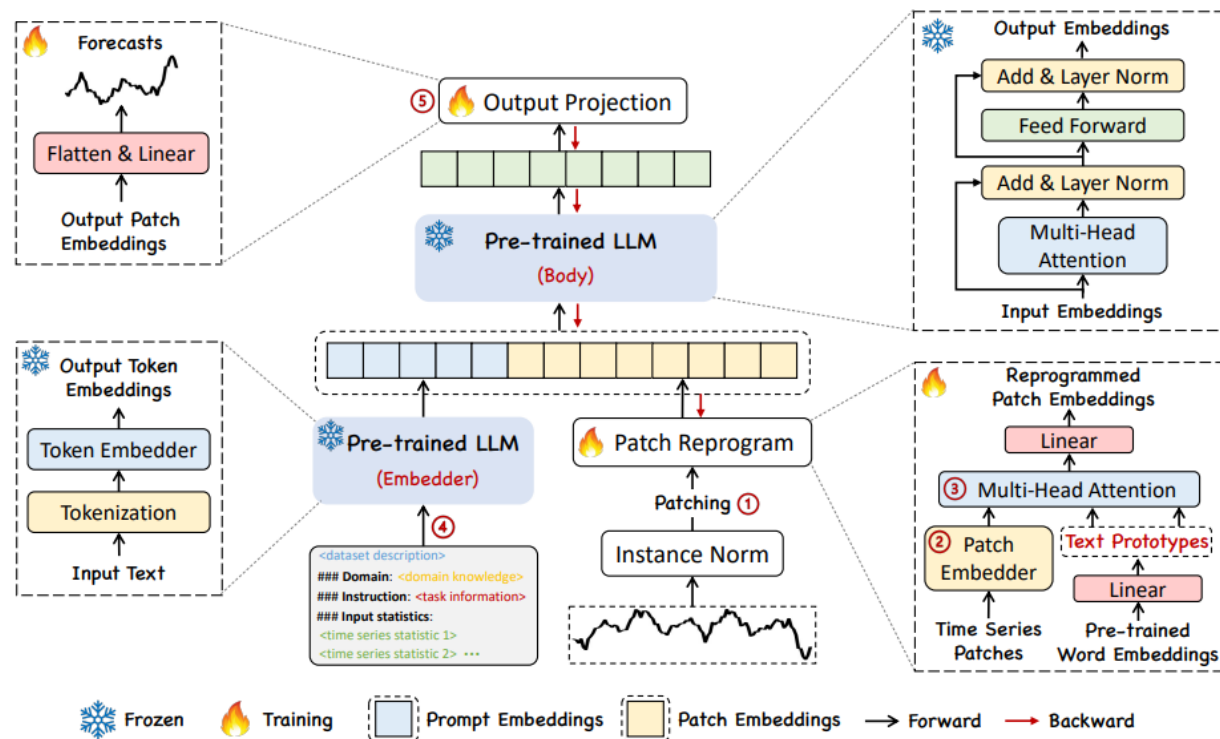
- **LLM을 활용한 시계열 예측의 가능성**

1. 일반화 가능성 – 새로운 도메인에 대해 재학습 없이 활용
2. 데이터 효율성 – 사전학습된 데이터로 적은 데이터로도 학습 가능
3. 추론 능력 – 복잡한 패턴을 인식하고 논리적으로 추론
4. 멀티모달 학습 - LLM은 텍스트/이미지/음성 등 다양한 데이터 활용가능
5. 최적화 용이성 – LLM은 이미 학습되었기에 특정작업에도 바로 적용 가능

> 논문의 연구에서는 Time-LLM: LLM을 시계열 예측에 적용하는 새로운 Reprogramming 프레임워크를 제안

Method

Method : Time LLM



(불꽃 아이콘) → 학습 과정에서 업데이트되는 가중치 (출력 투영 과정).
 (눈꽃 아이콘) → LLM은 동결된 상태(Frozen)로 유지되며 변경되지 않음.
 파란색 박스 → LLM에서 처리하는 주요 과정 (임베딩, 본체, 출력 등).
 노란색 박스 → 패치 변환과 재프로그래밍 과정 (Patch Reprogram).

Figure 2: The model framework of TIME-LLM. Given an input time series, we first tokenize and embed it via ① patching along with a ② customized embedding layer. ③ These patch embeddings are then reprogrammed with condensed text prototypes to align two modalities. To augment the LLM's reasoning ability, ④ additional prompt prefixes are added to the input to direct the transformation of input patches. ⑤ The output patches from the LLM are projected to generate the forecasts.

• Time LLM 의 전체적인 모델 프레임 워크

- 기존 LLM을 수정하지 않고, 시계열 데이터를 자연어 형태로 변화하여 예측

Method : 흐름 및 주요 단계

- Patching (패치변환)
 - 입력 시계열 데이터는 **Patch** 단위로 분할.
 - **Instance Normalization**(인스턴스 정규화)를 통해 데이터의 분포를 조정
 - 이후 **Patch Embedder**를 사용하여 패치를 특정 차원의 벡터로 변환
- Patch Embedding(임베딩 변환)
 - 패치 데이터를 **사용자지정(customized)** 임베딩 레이어를 통해 벡터화
 - 이 과정에서 **Text Prototypes**을 활용하여 시계열 데이터를 자연어 표현과 정렬(Alignment)
- Patch Reprogram(패치 재프로그래밍)
 - 기존 LLM이 자연어를 처리하는 방식에 맞춰 패치 데이터를 재구성.
 - 다중 헤드 어텐션(Multi-Head Attention)과 선형 변환(Linear)을 적용하여, 시계열 데이터를 LLM이 처리할 수 있는 입력 데이터로 변환
- Prompt-as-Prefix(PaP, 프롬프트 추가)
 - LLM이 시계열 데이터를 더 잘 이해할 수 있도록 prompt 추가
 - 프롬프트 구성 :
 1. 도메인 지식(domain knowledge) - 해당 데이터의 의미 및 특성 설명
 2. 작업 지시(Task Introduction) - 예측해야할 타임 스텝과 작업 방향을 지정
 3. 입력통계(Input Statistics) - 최소값, 최대값, 중간값, 추제 정보 제공
- Output Projection(출력 투영)
 - LLM이 생성한 출력을 다시 시계열 데이터 형태로 변환
 - 예측된 결과를 선형변환(Linear Projection) 및 Flatten 과정을 통해 최종 예측값을 도출

Method : 모델의 구조

Input Transformation : 입력 변환

시계열 데이터 정규화

1. 각 입력 채널을 개별적으로 평균 0, 표준편차 1로 정규화하여 시계열 분포 변화를 완화
2. Reversible Instance Normalization (RevIN)을 사용하여 시계열 데이터의 변화에도 적응하도록 함
-> RevIN은 시계열의 절대적 크기는 무시하고 상대적 패턴만 순수하게 보존

L_p : 패치 길이

s : 슬라이딩 윈도우 크기

2. 패치변환(Patching)

1. 시계열 데이터를 고정 길이의 패치단위로 분할
2. 연속적인 패치를 구성하여 모델이 로컬 시계열 패턴을 효과적으로 학습하도록 유도
3. 패치수는 다음과 같음

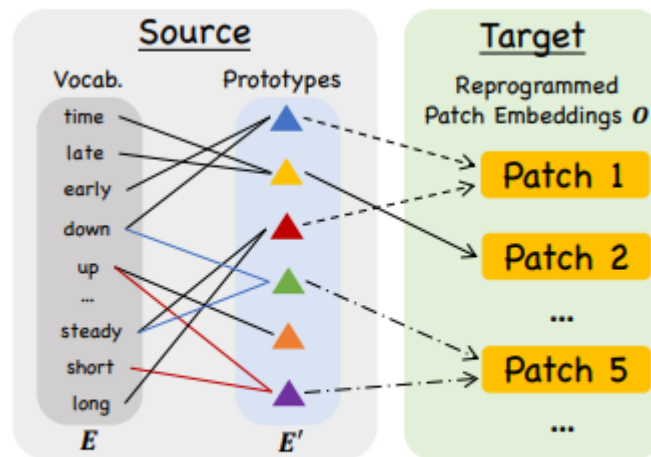
$$P = \lfloor \frac{(T-L_p)}{s} \rfloor + 2,$$

3. 패치 임베딩(Patch Embedding)

1. 분할된 패치를 선형 레이어를 통해 임베딩 벡터로 변환
2. 임베딩 된 패치 데이터를 LLM이 처리할 수 있는 형식으로 변환

Method : 모델의 구조

Patch Reprogramming 패치 재프로그래밍



(a) Patch Reprogramming

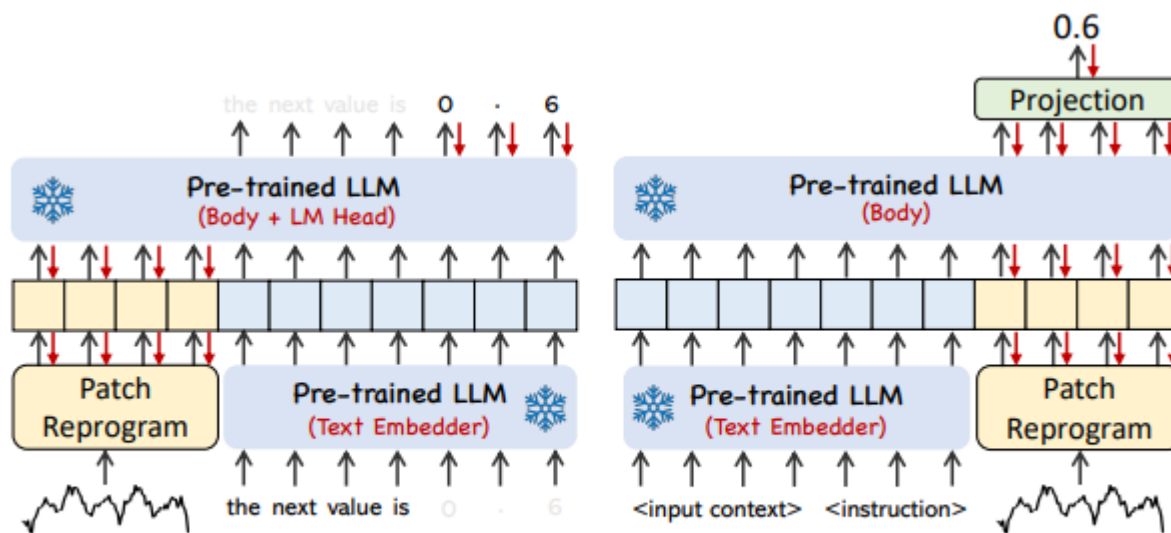
- LLM이 이해할 수 있도록 시계열 데이터를 자연어 형태로 변환
- Text Prototypes을 사용하여 패치 정보 요약
- Multi-head Cross-Attention을 적용하여 시계열 데이터를 자연어 표현

Method : 모델의 구조

Prompting : 프롬프트 기법

L_p : 패치 길이

s : 슬라이딩 윈도우 크기



(b) Patch-as-Prefix and Prompt-as-Prefix

- Patch-as-Prefix VS Prompt-as-Prefix 두가지 방법 비교
- LLM의 예측 성능을 향상시키기 위해 Prompt-as-Prefix(PaP) 기법 도입

Method : 모델의 구조

Prompting : 프롬프트 기법

1. Patch-as-Prefix

시계열 데이터를 패치 단위로 변환 후, 이를 LLM에 직접 입력하여 다음 값을 예측하도록 학습

예측값은 숫자(0.6)과 같은 형태로 직접 생성

문제점

LLM은 숫자 처리 성능이 낮아 정밀도가 떨어짐

토큰화 방식이 다르기 때문에 숫자 출력을 일관되게 변환하기 어려움. 해서, 장기 예측이 어려움

2. Prompt-as-Prefix(PaP)

LLM이 시계열 데이터를 더 효과적으로 학습할 수 있도록 프롬프트 추가

패치 변환 후, 추가적인 프롬프트를 입력하여 LLM이 시계열 데이터를 더 효과적으로 학습할 수 있도록 유도

출력은 자연어 기반으로 생성된 후, 최종적으로 Projection Layer(출력변환 레이어)를 통해 시계열 예측값으로 변환됨.

3. 프롬프트 구성요소

Dataset Context : 데이터의 의미와 특성 제공

예) "이 데이터는 전력 소비량을 나타내며, 특정 시점에서 급증할 가능성이 있음."

Task Instruction : LLM이 수행해야할 예측 작업 정의 - 예측할 타임 스템과 예측 방식 설명

예) "다음 24시간 동안의 전력 소비량을 예측하십시오"

Input Statistics : 입력 데이터의 통계 정보 제공

예) 최소, 최대, 중간값, 변화 추세(upward/downward) 등.

Method : 모델의 구조

Output Projection : 출력 반환

- LLM에서 생성된 출력을 다시 시계열 데이터 형태로 변환
- 출력투영 레이어(Projection Layer) 를 사용하여 예측값을 생성.
- LLM의 자연어 출력을 정량적 예측값으로 변환하는 역할.

Conclusion

Conclusion

- Time-LLM은 기존의 LLM을 수정하지 않고도 시계열 예측에 적용할 수 있는 강력한 프레임워크임을 입증

- Patch Reprogramming을 통해 시계열 데이터를 LLM이 이해할 수 있는 텍스트 프로토타입으로 변환
- Prompt-as-Prefix 기법을 사용하여 LLM의 논리적 추론 능력을 강화, 기존의 전문 시계열 모델보다 더 높은 성능 기록
- 이를 통해 시계열 예측을 자연어처리(NLP)문제처럼 다룰 수 있으며, LLM을 활용한 새로운 접근 방식이 가능성을 시사

- 향후 연구 방향

- 최적의 Reprogramming 기법 연구
- LLM의 지속적인 사전학습 연구
- 멀티모달 모델로 확장
- LLM을 활용하여 예측 뿐만 아니라 다양한 시계열 분석에도 사용