LTSF-Linear

2022-12-04

증강지능 연구실 황승현

목차

- 시계열 처리 수업 시간에 배운 내용
  - 시계열 통계 모델
  - 시계열 딥러닝 모델
- 트랜스포머란?
- LTSF-Linear

시계열 처리

수업에서 배운 내용

시계열 통계 모델

- 자기회귀(AutoRegressive; AR) 모델
- 이동평균(Mobbing Average; MA) 모델
- 자귀회귀 누적 이동평균(AutoRegressive Integrate Moving Average; ARIMA) 모델
- 벡터자기회귀(Vector Autoregression)

시계열 통계 모델

장점

- 간단함.
  - 파라미터 보고 이해할 수 있음
  - 수학적 표현
- 작은 데이터셋에서 좋은 결과
  - 복잡한 머신러닝 모델과 비교 가능
  - 과적합 위험 없음
- 예측 가능

단점

- 대규모 데이터셋은 머신러닝 모델/신경망 방법이 더 좋음
- 불확실성 방식으로 표현
- 비선형 관계가 많은 데이터를 설명하는데 적합하지 않음

시계열 딥러닝 모델

- 순전파 네트워크 : 시계열에 잘 안 맞음
- 합성곱 신경망(CNN) : 시계열에 잘 안 맞음
- 순환 신경망(RNN) : 시계열에 적합
- 트랜스포머(Transformer)
  - sequence-to-sequence(seq2seq)
  - 어텐션(Attention)
  - 입력되는 순서 파악 가능, 시계열에 적합

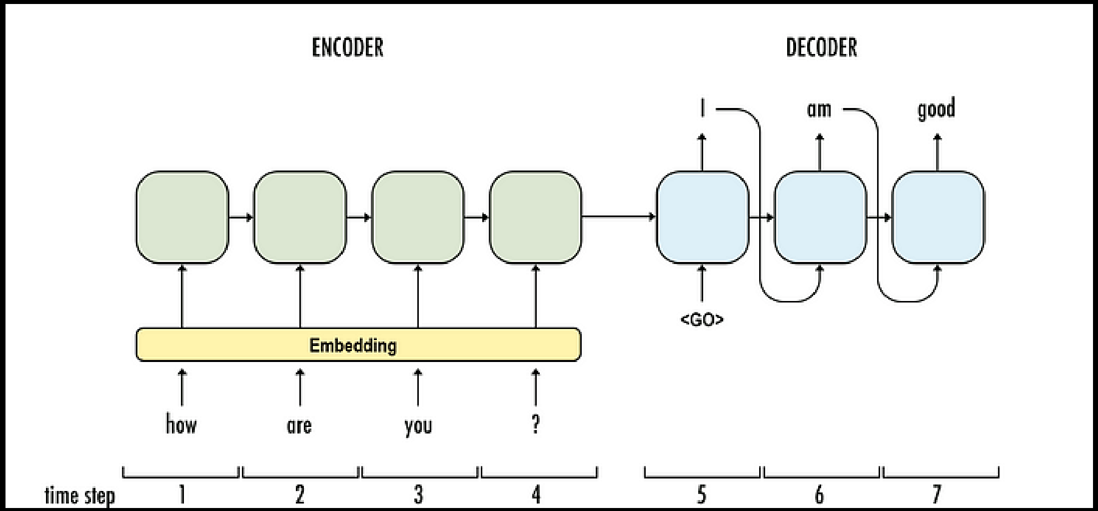
트랜스포머

Transformer

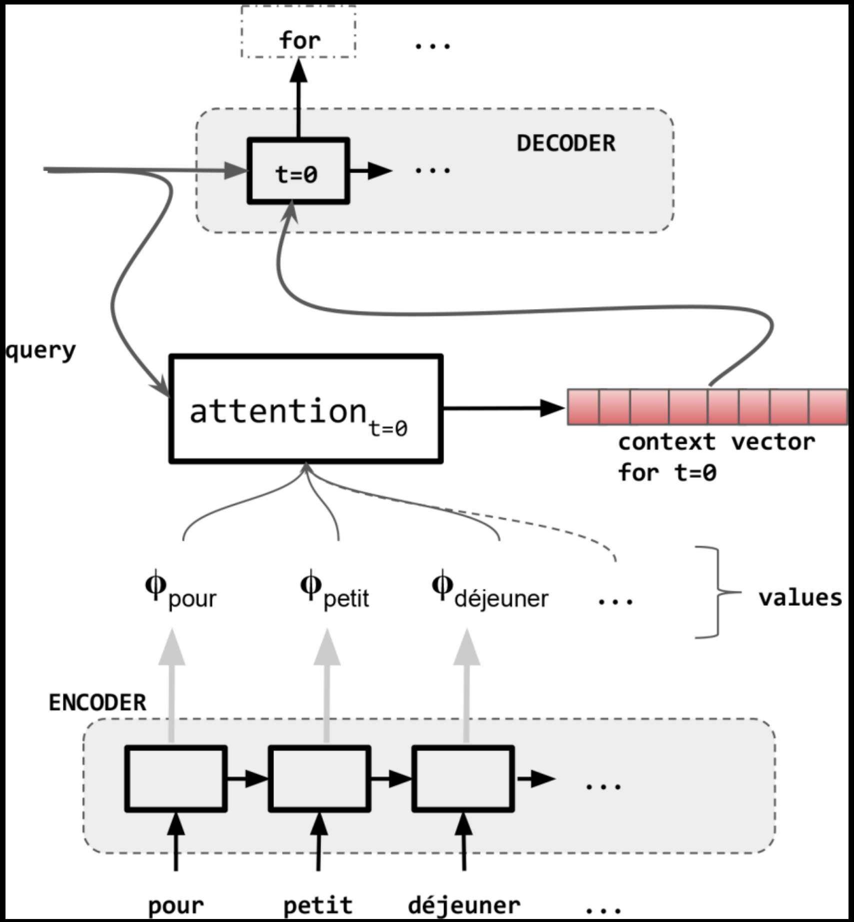
Sequence to Sequence (S2S)

- Input: sequence
- Output: another sequence
- Input possibly different length, as output

- S2S is A type of encoder-decoder model



Attention



- pour petit déjeuner…
- for breakfast…
- Our minds focus on the relevant parts of the input while producing output

Attention Mechanism

- not only the hidden state of the encoder,
  but also the hidden states for each of the intermediate steps

1. compute the ‘similarity' between the current decoder's hidden state and the encoder's hidden state at each point in time.
2. This similarity is the ‘Attention vector’. Find the weighted sum of the encoder hidden states according to the ‘Attention vector’ to get the ‘context vector’.
3. Use the context vector to predict the next word.

트랜스포머(Transformer)

- 순차 데이터 내의 관계 추적.
- 어텐션만으로 구성된 신경망 모델
  - seq2seq의 구조
  - 어텐션(Attention)만으로 구현
  - RNN 계열의 알고리즘 없음

LTSF-Linear

Are Transformers Effective for Time Series Forecasting?

Are Transformers Effective for Time Series Forecasting?

Are Transformers Effective for Time Series Forecasting?

Ailing Zeng[1,2*], Muxi Chen[1*], Lei Zhang[2], Qiang Xu[1]
[1]The Chinese University of Hong Kong
[2]International Digital Economy Academy
{zengailing, leizhang}@idea.edu.cn,{mxchen21, qxu}@cse.cuhk.edu.hk

Abstract

Recently, there has been a surge of Transformer-based solutions for the long-term time series forecasting (LTSF) task. Despite the growing performance over the past few years, *we question the validity of this line of research in this work*. Specifically, Transformers is arguably the most successful solution to extract the semantic correlations among the elements in a long sequence. However, in time series modeling, we are to extract the temporal relations in *an ordered set of continuous points*. While employing positional encoding and using tokens to embed sub-series in Transformers facilitate preserving some ordering information, the nature of the *permutation-invariant* self-attention mechanism inevitably results in temporal information loss.

To validate our claim, we introduce a set of embarrassingly simple one-layer linear models named *LTSF-Linear* for comparison. Experimental results on nine real-life datasets show that *LTSF-Linear* surprisingly outperforms existing sophisticated Transformer-based LTSF models in all cases, and often by a large margin. Moreover, we conduct comprehensive empirical studies to explore the impacts of various design elements of LTSF models on their temporal relation extraction capability. We hope this surprising finding opens up new research directions for the LTSF task. We also advocate revisiting the validity of Transformer-based solutions for other time series analysis tasks (e.g., anomaly detection) in the future.

Introduction

Time series are ubiquitous in today's data-driven world. Given historical data, time series forecasting (TSF) is a long-standing task that has a wide range of applications, including but not limited to traffic flow estimation, energy management, and financial investment. Over the past several decades, TSF solutions have undergone a progression from traditional statistical methods (e.g., ARIMA (Ariyo, Adewumi, and Ayo 2014)) and machine learning techniques (e.g., GBRT (Friedman 2001)) to deep learning-based solutions, e.g., (Bai, Kolter, and Koltun 2018; Liu et al. 2022).

Transformer (Vaswani et al. 2017) is arguably the most successful sequence modeling architecture, demonstrating unparalleled performances in various applications, such as natural language processing (NLP) (Devlin et al. 2018), speech recognition (Dong, Xu, and Xu 2018), and computer vision (Liu et al. 2021b). Recently, there has also been a surge of Transformer-based solutions for time series analysis, as surveyed in (Wen et al. 2022). Most notable models, which focus on the less explored and challenging long-term time series forecasting (LTSF) problem, include LogTrans (Li et al. 2019) (NeurIPS 2019), Informer (Zhou et al. 2021) (AAAI 2021 Best paper), Autoformer (Xu et al. 2021) (NeurIPS 2021), Pyraformer (Liu et al. 2021a) (ICLR 2022 Oral), Triformer (Cirstea et al. 2022) (IJCAI 2022) and the recent FEDformer (Zhou et al. 2022) (ICML 2022).

The main working power of Transformers is from its multi-head self-attention mechanism, which has a remarkable capability of extracting semantic correlations among elements in a long sequence (e.g., words in texts or 2D patches in images). However, self-attention is *permutation-invariant* and "anti-order" to some extent. While using various types of positional encoding techniques can preserve some ordering information, it is still inevitable to have temporal information loss after applying self-attention on top of them. This is usually not a serious concern for semantic-rich applications such as NLP, e.g., the semantic meaning of a sentence is largely preserved even if we reorder some words in it. However, when analyzing time series data, there is usually a lack of semantics in the numerical data itself, and we are mainly interested in modeling the temporal changes among *a continuous set of points*. That is, the order itself plays the most crucial role. Consequently, we pose the following intriguing question: ***Are Transformers really effective for long-term time series forecasting?***

Moreover, while existing Transformer-based LTSF solutions have demonstrated considerable prediction accuracy improvements over traditional methods, in their experiments, all the compared (non-Transformer) baselines perform autoregressive or iterated multi-step (IMS) forecasting (Ariyo, Adewumi, and Ayo 2014; Salinas, Flunkert, and Gasthaus 2017; Bahdanau, Cho, and Bengio 2014; Taylor and Letham 2017), which are known to suffer from significant error accumulation effects for the LTSF problem. Therefore, in this work, we challenge Transformer-based LTSF solutions with direct multi-step (DMS) forecasting strategies to validate their real performance.

Not all time series are predictable, let alone long-term forecasting (e.g., for chaotic systems). We hypothesize that

*These authors contributed equally.

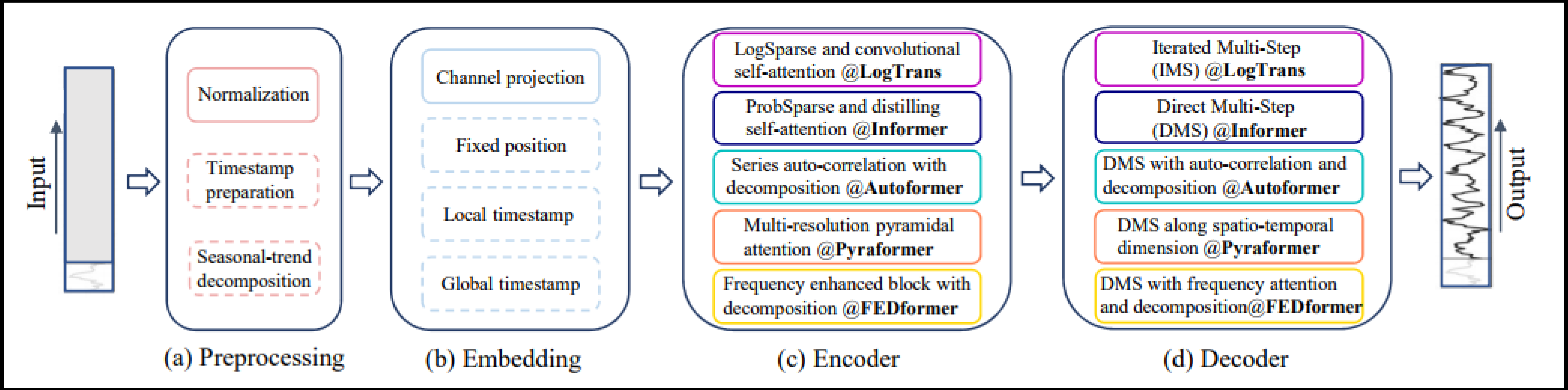
- AAAI-23

- Ailing Zeng(1,2)
- Muxi Chen(1)
- Lei Zhang(2)
- Qiang Xu(1)
- 1 The Chinese University of Hong Kong
- 2 International Digital Economy Academy

LTSF

- long-term time series forecasting
- 시계열 처리 모델
  - non-Transformer
    - Autoregressive
    - Iterated multi-step (IMS)
  - Transformer
- Are Transformers really effective for long-term time series forecasting?

트랜스포머 기반 시계열 분석



선형 모델 소개

- Dlinear
  - 디컴포짓 방식 + 선형 레이어
- Nlinear
  - 입력에서 시퀀스의 마지막 값 뺌
  - 선형 레이어 통과 후 마지막 값 더함

실험- 9가지 데이터셋

Table 1: The statistics of the nine popular datasets for the LTSF problem.

Methods		IMP.	Linear*		NLinear*		DLinear*		FEDformer		Autoformer		Informer		Pyraformer*		*Repeat**	
Metric		MSE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Electricity	96	27%	**0.140**	**0.237**	0.141	**0.237**	**0.140**	**0.237**	0.193	0.308	0.201	0.317	0.274	0.368	0.386	0.449	1.588	0.946
	192	24%	**0.153**	0.250	0.154	**0.248**	**0.153**	0.249	0.201	0.315	0.222	0.334	0.296	0.386	0.386	0.443	1.595	0.950
	336	21%	**0.169**	0.268	0.171	**0.265**	**0.169**	0.267	0.214	0.329	0.231	0.338	0.300	0.394	0.378	0.443	1.617	0.961
	720	17%	**0.203**	0.301	0.210	**0.297**	**0.203**	0.301	0.246	0.355	0.254	0.361	0.373	0.439	0.376	0.445	1.647	0.975
Exchange	96	45%	0.082	0.207	0.089	0.208	**0.081**	0.203	0.148	0.278	0.197	0.323	0.847	0.752	0.376	1.105	**0.081**	**0.196**
	192	42%	0.167	0.304	0.180	0.300	**0.157**	0.293	0.271	0.380	0.300	0.369	1.204	0.895	1.748	1.151	0.167	**0.289**
	336	34%	0.328	0.432	0.331	0.415	**0.305**	0.414	0.460	0.500	0.509	0.524	1.672	1.036	1.874	1.172	**0.305**	**0.396**
	720	46%	0.964	0.750	1.033	0.780	**0.643**	**0.601**	1.195	0.841	1.447	0.941	2.478	1.310	1.943	1.206	0.823	0.681
Traffic	96	30%	**0.410**	0.282	**0.410**	**0.279**	**0.410**	0.282	0.587	0.366	0.613	0.388	0.719	0.391	2.085	0.468	2.723	1.079
	192	30%	**0.423**	0.287	**0.423**	**0.284**	**0.423**	0.287	0.604	0.373	0.616	0.382	0.696	0.379	0.867	0.467	2.756	1.087
	336	30%	0.436	0.295	**0.435**	**0.290**	0.436	0.296	0.621	0.383	0.622	0.337	0.777	0.420	0.869	0.469	2.791	1.095
	720	26%	0.466	0.315	**0.464**	**0.307**	0.466	0.315	0.626	0.382	0.660	0.408	0.864	0.472	0.881	0.473	2.811	1.097
Weather	96	19%	**0.176**	0.236	0.182	**0.232**	**0.176**	0.237	0.217	0.296	0.266	0.336	0.300	0.384	0.896	0.556	0.259	0.254
	192	21%	**0.218**	0.276	0.225	**0.269**	0.220	0.282	0.276	0.336	0.307	0.367	0.598	0.544	0.622	0.624	0.309	0.292
	336	23%	**0.262**	0.312	0.271	**0.301**	0.265	0.319	0.339	0.380	0.359	0.395	0.578	0.523	0.739	0.753	0.377	0.338
	720	20%	0.326	0.365	0.338	**0.348**	**0.323**	0.362	0.403	0.428	0.419	0.428	1.059	0.741	1.004	0.934	0.465	0.394
ILI	24	48%	1.947	0.985	**1.683**	**0.858**	2.215	1.081	3.228	1.260	3.483	1.287	5.764	1.677	1.420	2.012	6.587	1.701
	36	36%	2.182	1.036	**1.703**	**0.859**	1.963	0.963	2.679	1.080	3.103	1.148	4.755	1.467	7.394	2.031	7.130	1.884
	48	34%	2.256	1.060	**1.719**	**0.884**	2.130	1.024	2.622	1.078	2.669	1.085	4.763	1.469	7.551	2.057	6.575	1.798
	60	34%	2.390	1.104	**1.819**	**0.917**	2.368	1.096	2.857	1.157	2.770	1.125	5.264	1.564	7.662	2.100	5.893	1.677
ETTh1	96	1%	0.375	0.397	**0.374**	**0.394**	0.375	0.399	0.376	0.419	0.449	0.459	0.865	0.713	0.664	0.612	1.295	0.713
	192	4%	0.418	0.429	0.408	**0.415**	**0.405**	0.416	0.420	0.448	0.500	0.482	1.008	0.792	0.790	0.681	1.325	0.733
	336	7%	0.479	0.476	**0.429**	**0.427**	0.439	0.443	0.459	0.465	0.521	0.496	1.107	0.809	0.891	0.738	1.323	0.744
	720	13%	0.624	0.592	**0.440**	**0.453**	0.472	0.490	0.506	0.507	0.514	0.512	1.181	0.865	0.963	0.782	1.339	0.756
ETTh2	96	20%	0.288	0.352	**0.277**	**0.338**	0.289	0.353	0.346	0.388	0.358	0.397	3.755	1.525	0.645	0.597	0.432	0.422
	192	20%	0.377	0.413	**0.344**	**0.381**	0.383	0.418	0.429	0.439	0.456	0.452	5.602	1.931	0.788	0.683	0.534	0.473
	336	26%	0.452	0.461	**0.357**	**0.400**	0.448	0.465	0.496	0.487	0.482	0.486	4.721	1.835	0.907	0.747	0.591	0.508
	720	14%	0.698	0.595	**0.394**	**0.436**	0.605	0.551	0.463	0.474	0.515	0.511	3.647	1.625	0.963	0.783	0.588	0.517
ETTm1	96	21%	0.308	0.352	0.306	0.348	**0.299**	**0.343**	0.379	0.419	0.505	0.475	0.672	0.571	0.543	0.510	1.214	0.665
	192	21%	0.340	0.369	0.349	0.375	**0.335**	**0.365**	0.426	0.441	0.553	0.496	0.795	0.669	0.557	0.537	1.261	0.690
	336	17%	0.376	0.393	0.375	0.388	**0.369**	**0.386**	0.445	0.459	0.621	0.537	1.212	0.871	0.754	0.655	1.283	0.707
	720	22%	0.440	0.435	0.433	0.422	**0.425**	**0.421**	0.543	0.490	0.671	0.561	1.166	0.823	0.908	0.724	1.319	0.729
ETTm2	96	18%	0.168	0.262	**0.167**	**0.255**	**0.167**	0.260	0.203	0.287	0.255	0.339	0.365	0.453	0.435	0.507	0.266	0.328
	192	18%	0.232	0.308	**0.221**	**0.293**	0.224	0.303	0.269	0.328	0.281	0.340	0.533	0.563	0.730	0.673	0.340	0.371
	336	16%	0.320	0.373	**0.274**	**0.327**	0.281	0.342	0.325	0.366	0.339	0.372	1.363	0.887	1.201	0.845	0.412	0.410
	720	13%	0.413	0.435	**0.368**	**0.384**	0.397	0.421	0.421	0.415	0.433	0.432	3.379	1.338	3.625	1.451	0.521	0.465

- Methods* are implemented by us; Other results are from FEDformer (Zhou et al. 2022).

Table 2: Multivariate long-term forecasting errors in terms of MSE and MAE, the lower the better. Among them, ILI dataset is with forecasting horizon $T \in \{24, 36, 48, 60\}$. For the others, $T \in \{96, 192, 336, 720\}$. The best results are highlighted in bold and the best results of Transformers are highlighted with an underline. IMP. is the best result of linear models compared to the results of Transformer-based solutions.

실험- look-back window 크기

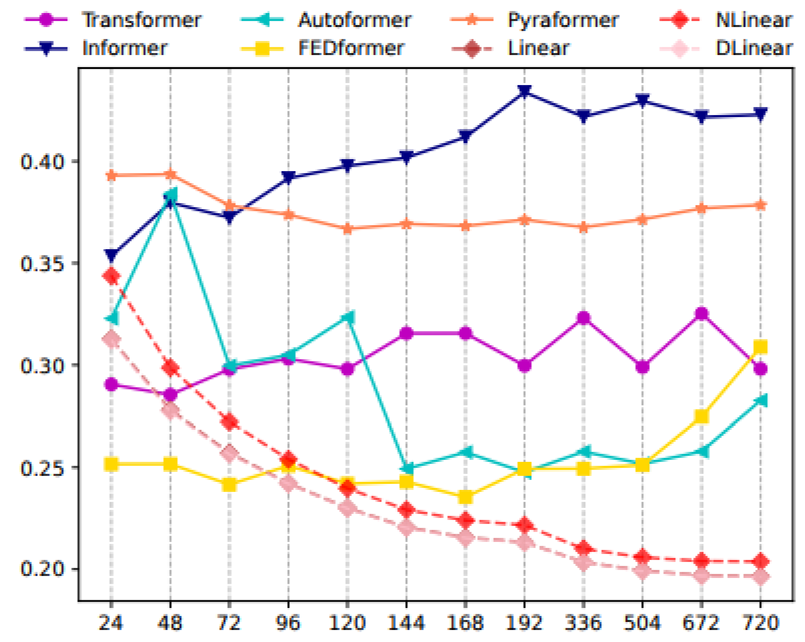


Figure 4: The MSE results (Y-axis) of models with different look-back window sizes (X-axis) of long-term forecasting (T=720) on Electricity.

실험 - 데이터셋의 크기

Methods	FEDformer		Autoformer	
Dataset	*Ori.*	*Short*	*Ori.*	*Short*
96	0.587	**0.568**	0.613	**0.594**
192	0.604	**0.584**	**0.616**	0.621
336	0.621	**0.601**	0.622	**0.621**
720	0.626	**0.608**	0.660	**0.650**

Table 7: The MSE comparisons of two training data sizes.

실험 - 효율성

Method	MACs	Parameter	Time	Memory
DLinear	**0.04G**	**139.7K**	**0.4ms**	**687MiB**
Transformer×	4.03G	13.61M	26.8ms	6091MiB
Informer	3.93G	14.39M	49.3ms	3869MiB
Autoformer	4.41G	14.91M	164.1ms	7607MiB
Pyraformer	0.80G	241.4M	3.4ms	7017MiB
FEDformer	4.41G	20.68M	40.5ms	4143MiB

× the same one-step decoder.

Table 8: Comparison of practical efficiency of LTSF-Transformers under L=96 and T=720 on the Electricity. MACs are the number of multiply-accumulate operations. The inference time averages 5 runs.

기존 트랜스포머 기반 모델의 한계

- 트레이닝 데이터 크기 작을 때 성능 안 좋음
  - 실제로, 데이터 크기 늘리면 성능 향상
  - 현실 데이터는 크기 늘릴 수 없음
- 일부 순서 정보 보존할 수 있으나 전체 시간 정보 보존 불가
  - self-attention mechanism의 한계
  - permutation-invariant
  - 순서에 관계 없음
  - 시계열: 순서 중요

결론

- 트랜스포머보다 선형 모델이 더 우월하다 X
- LTSF-Transformer가 효과적이지 않다 O
- Transformer의 특성상 시계열 처리에 적합하지 않을 수도

향후 계획

- 반박 논문 리뷰
  - A Time Series is Worth 64 Words: Long-term Forecasting with Transformers.
  - Transformer 기반 시계열 분석 모델 SOTA
  - ICLR 2023

Q&A

Thank you!