

2025.11.26

Correlation Recurrent Units: A Novel Neural Architecture for Improving the Predictive Performance of Time-Series Data

Sunghyun Sim, Dohee Kim, Hyerim Bae

202583627
류지현

목차

1. Introduction

2. Related Work

3. Correlation Recurrent Units

4. Experiments

5. Ablation Studies

6. Conclusion

Introduction

- Time-series data: 시간적 사건의 연속적인 현상에 대한 기록
 - 의료, 사회, 금융, 교통 등의 다양한 분야에서 지속적으로 생성되어 저장됨
 - 시계열 예측, 이상치 및 궤적 감지, 인간 패턴 분석을 조사하는 연구에 사용
- Time-series forecasting problem: 특정 시점의 값을 예측하는 문제
 - 통계적 접근 방식 – ARIMA, ETS
 - 딥러닝 – RNN, LSTM, GRU

Introduction

- 시계열 예측에서의 딥러닝

- 1) Seq2Seq 모델을 기반으로 하는 확률적 예측을 위한 새로운 아키텍처 개발

- 확률적 예측을 위한 모델 아키텍처 개발

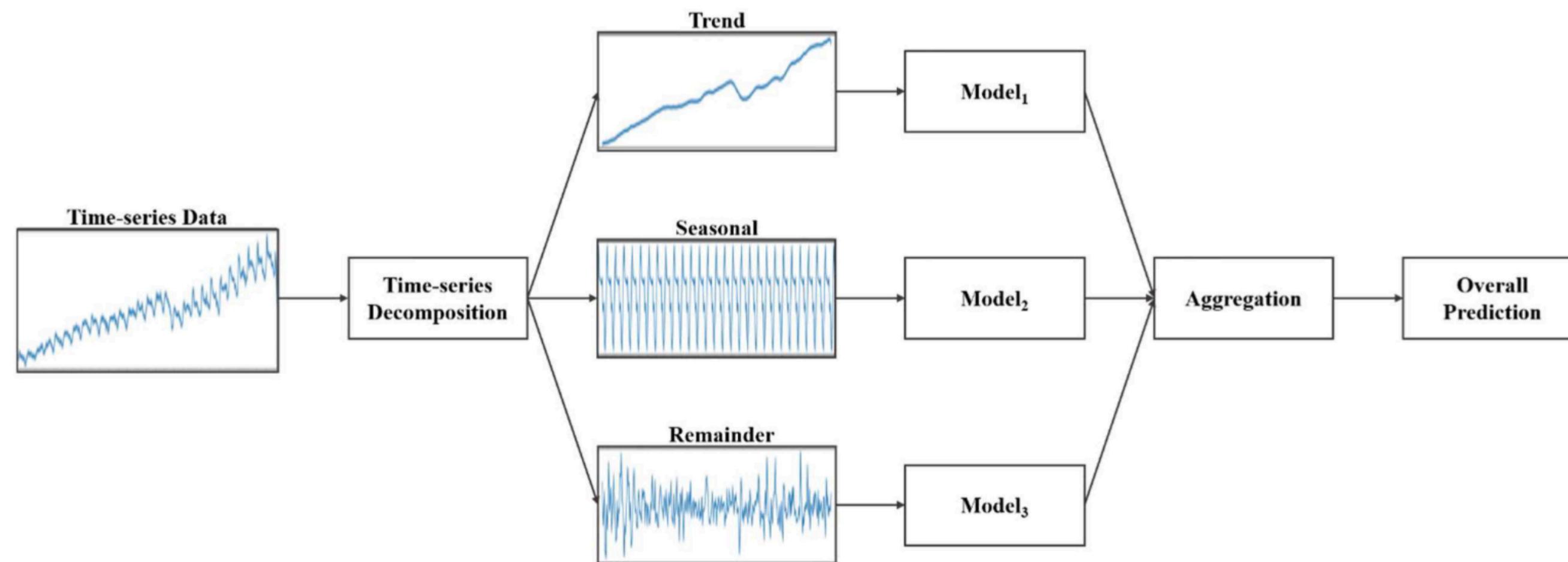
- DeepAR, MQRNN

- 어텐션 메커니즘을 결합한 트랜스포머를 사용하는 모델 아키텍처 개발

- CCT, TFT

Introduction

- 시계열 예측에서의 딥러닝
 - 2) 시계열 분해 기술을 딥러닝 모델과 결합하는 하이브리드 접근 방식
 - LOESS 분해 → 시계열 데이터를 추세, 계절, 잔차로 분해하여 시계열 예측 모델을 학습



Related Work

- 시계열 예측 모델
 - Auto-Regressive Integrated Moving Average(ARIMA)

$$\frac{\phi(B)(1 - B)^d}{\text{AR} \quad |} y_t = \frac{\theta(B)\epsilon_t}{\text{MA}}$$

- 자기회귀(AR): 과거 시점의 데이터가 현 시점의 데이터에 미치는 영향 고려
- 차분(I): 정상성을 만족하지 않을 때 정상성 확보
- 이동평균(MA): 추세 반영

Related Work

- 시계열 예측 모델

- Vector AutoRegression(VAR)

$$\underline{\Phi(B) \mathbf{y}_t = \mathbf{c} + \boldsymbol{\epsilon}_t}$$

다면량 AR

- 자기회귀(AR): 과거 시점의 데이터가 현 시점의 데이터에 미치는 영향 고려
- 다변량 시계열 모델: 2개 이상 변수의 상호 인과관계를 반영
- 각 변수에 대한 계수를 통해 해석 가능

Related Work

- 시계열 예측 모델

- Error-Trend-Seasonal(ETS)

$$y_t = \ell_{t-1} + b_{t-1} + s_{t-m} + \epsilon_t$$

$$\ell_t = \ell_{t-1} + b_{t-1} + \alpha \epsilon_t$$

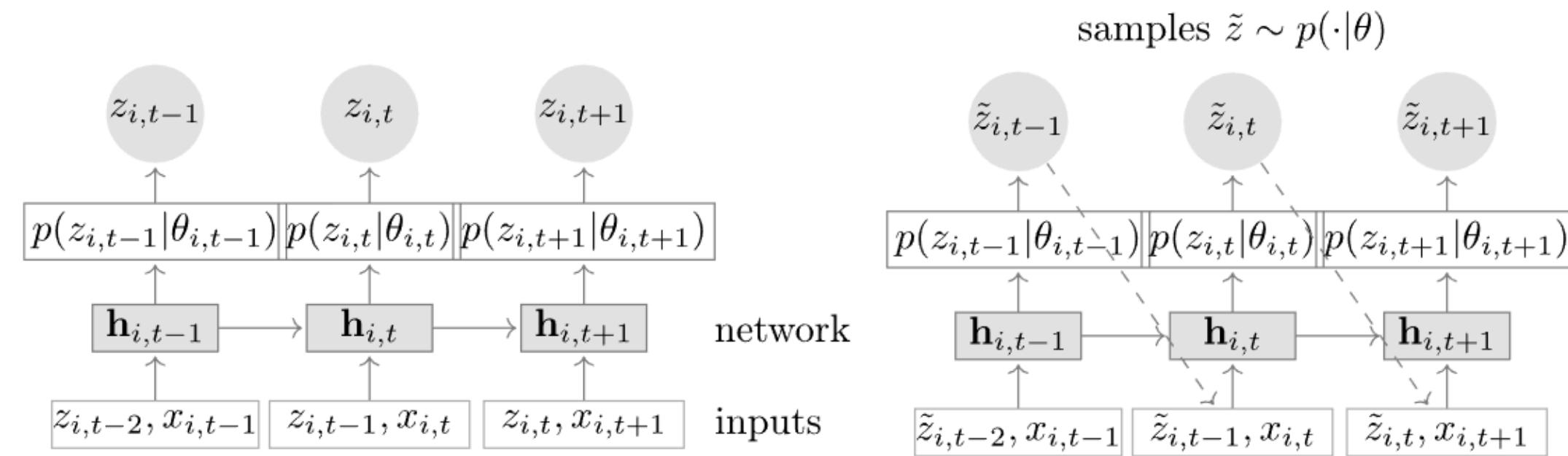
$$b_t = b_{t-1} + \beta \epsilon_t$$

$$s_t = s_{t-m} + \gamma \epsilon_t$$

- 지수평활 시계열 모델
- 오차(E), 추세(T), 계절성(S) 반영

Related Work

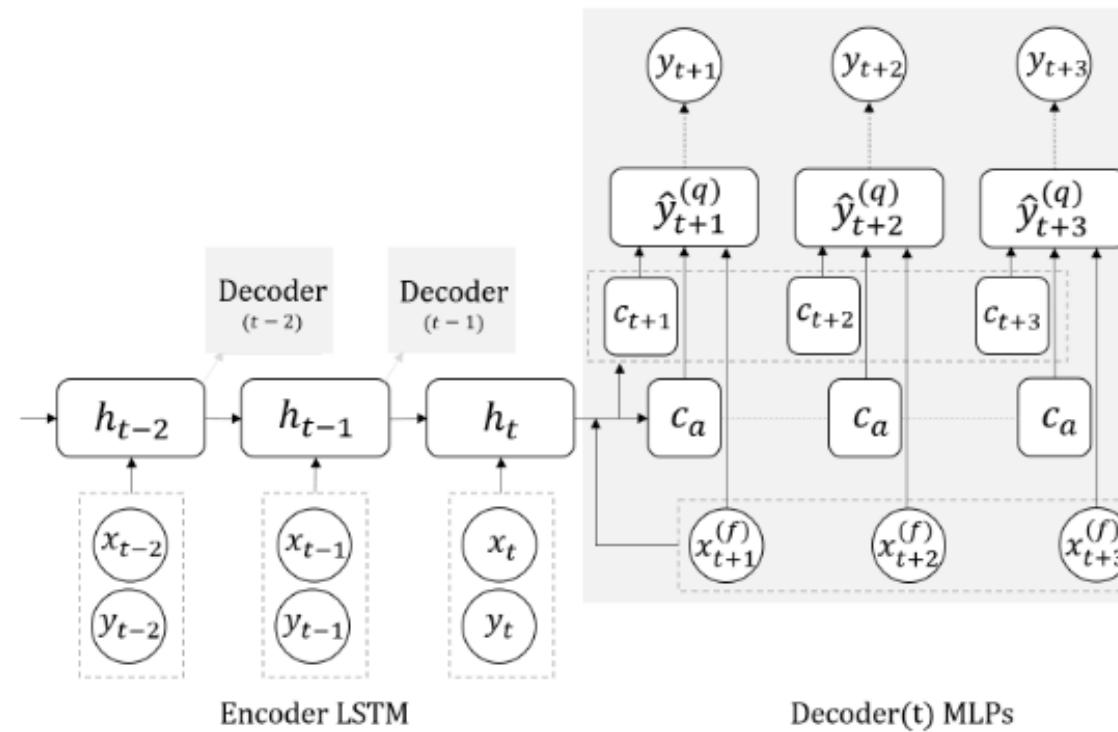
- 시계열 예측 모델
 - DeepAR



- RNN 기반 Autoregressive 구조
- log-likelihood를 최대화하며 학습하여 예측의 불확실성까지 모델링
- 여러 시계열 데이터의 공통 패턴 학습 가능

Related Work

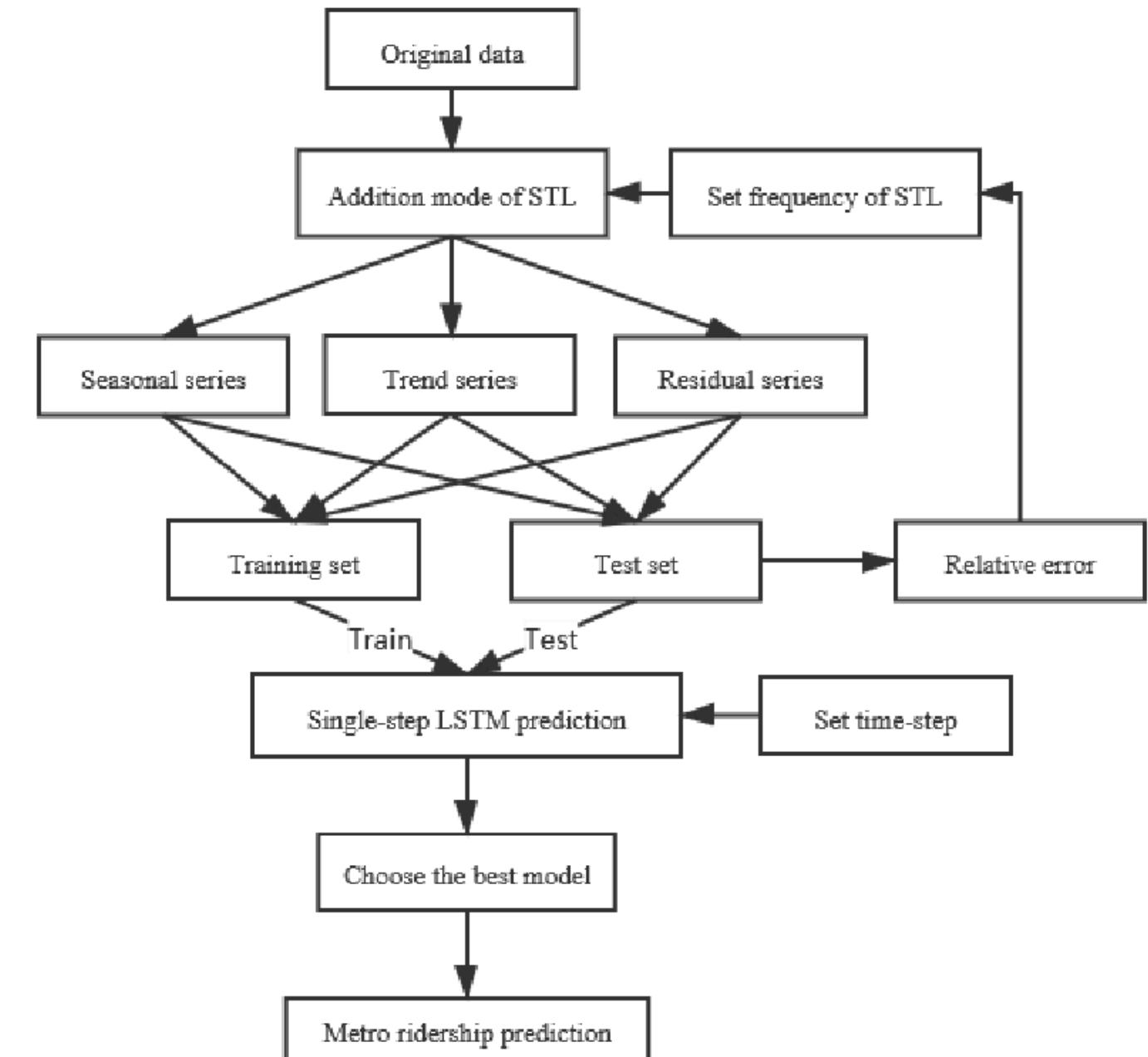
- 시계열 예측 모델
 - Multi-Quantile Recurrent Neural Network(MQ-RNN)



- 분위수를 예측하는 RNN 기반 모델
- RNN으로 구성된 encoder와 두 개의 MLP(global MLP, local MLP)로 구성된 decoder
- global MLP는 encoder 출력과 미래 시점의 입력을 캡슐화
- local MLP는 해당 지점에 대한 분위수를 생성하기 위해 예측 horizon의 각 특정 시점에 작용

Related Work

- 시계열 분해 기반 모델
 - STL-LSTM
 - STL 방법을 통해 계절성, 추세, 잔차로 분해
 - 계절성, 추세, 잔차를 개별 LSTM 모델로 학습
 - 각 LSTM 모델에서 예측된 계절성, 추세, 잔차 값을 합산

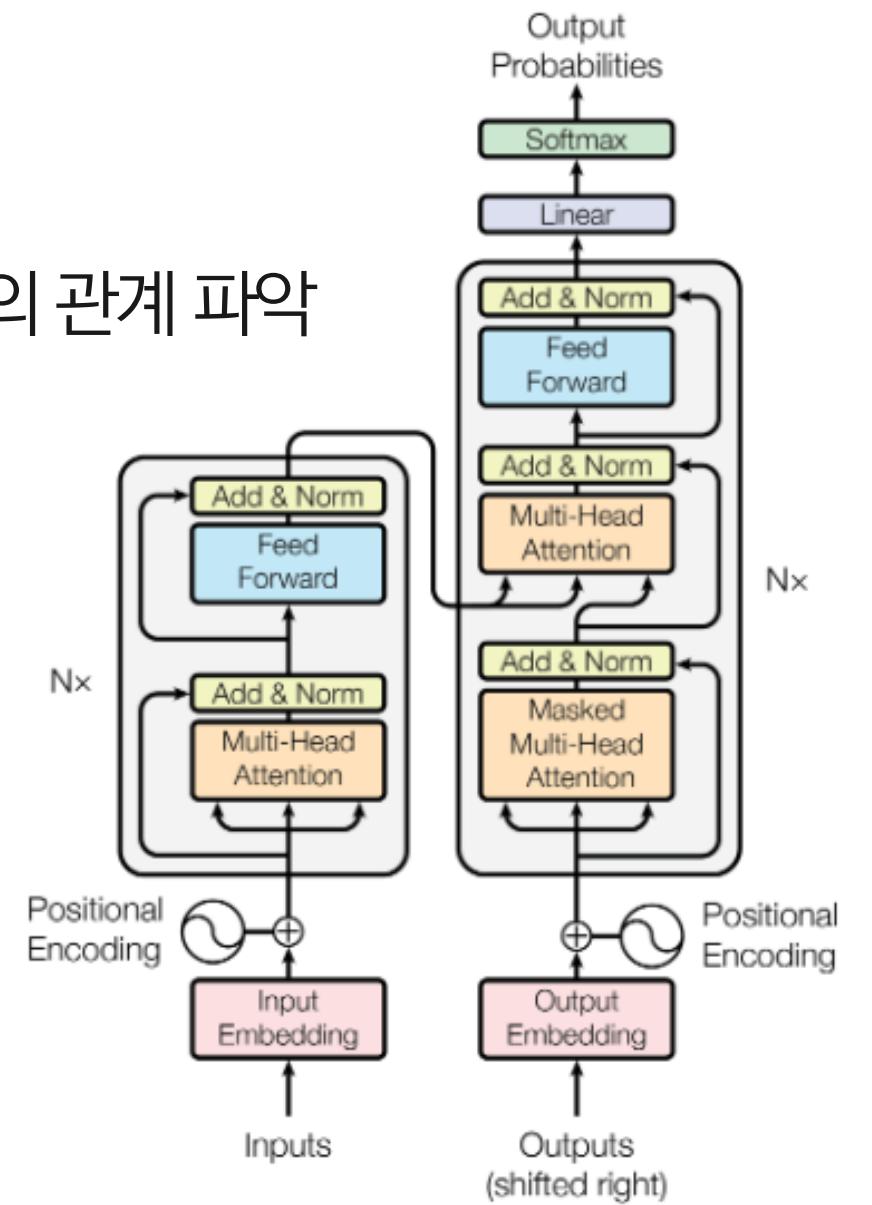


Related Work

- Attention 기반 모델

 - Transformer

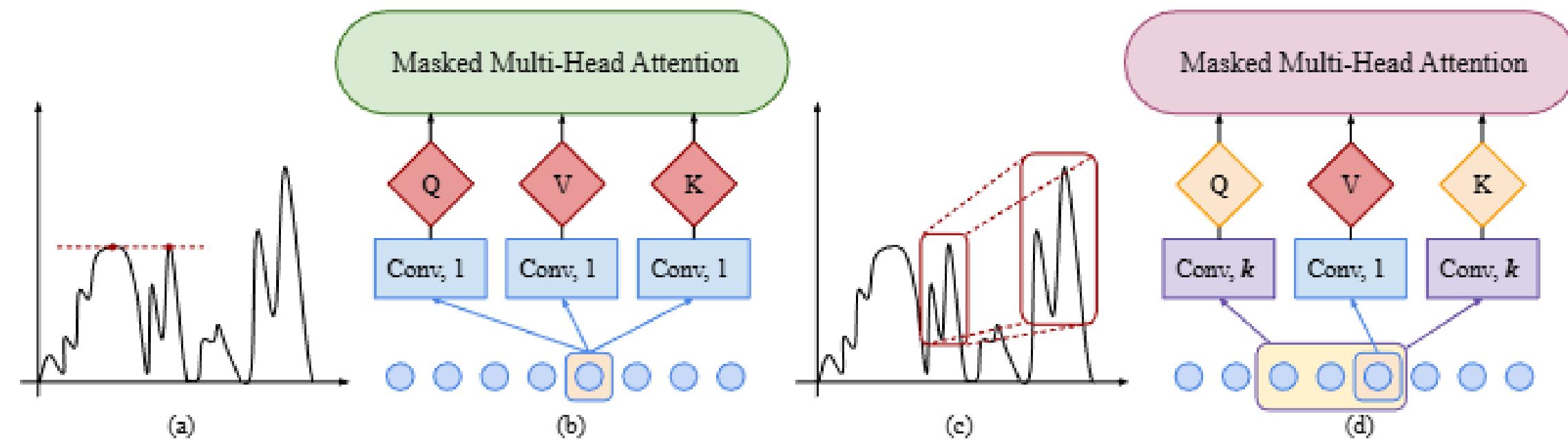
 - attention 메커니즘을 결합한 encoder-decoder 아키텍처 구조를 갖는 것이 일반적
 - encoder의 Multi-Head Attention: 입력값끼리의 관계 파악
 - decoder의 Masked Multi-Head Attention: 지금까지 만들어진 값과 가려진 다음 값의 관계 파악
 - decoder의 Multi-Head Attention: 인코더의 값을 디코더가 만드는 내용과 연결



Related Work

- Attention 기반 모델

 - Causal Convolutional Transformer(CCT)



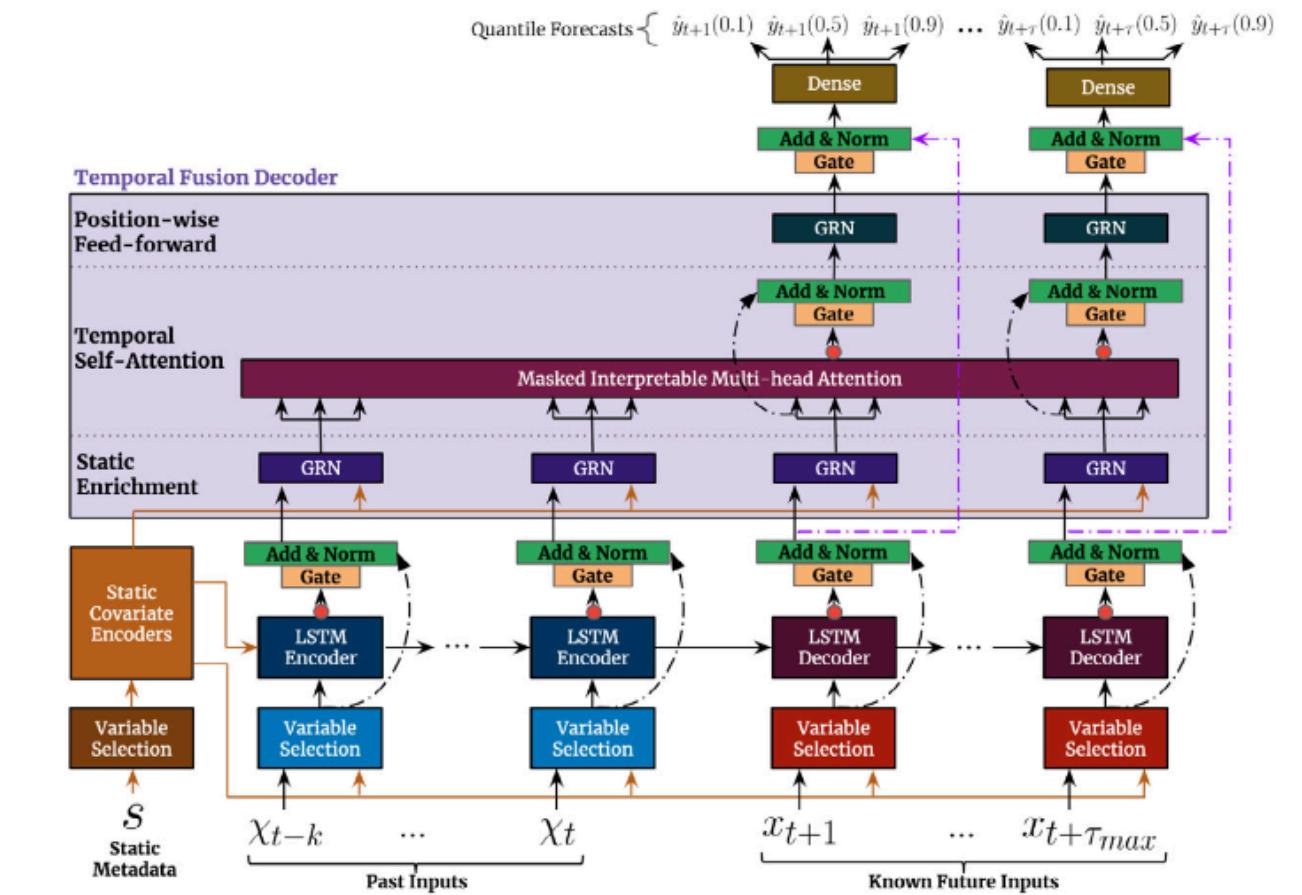
 - causal convolution: 지역적 맥락을 활용 → 계산량 증가
 - sparse attention: 전체 과거 시점에 대해 attention을 계산하는 대신 특정 부분집합만 계산 → 계산량 감소

Related Work

- Attention 기반 모델

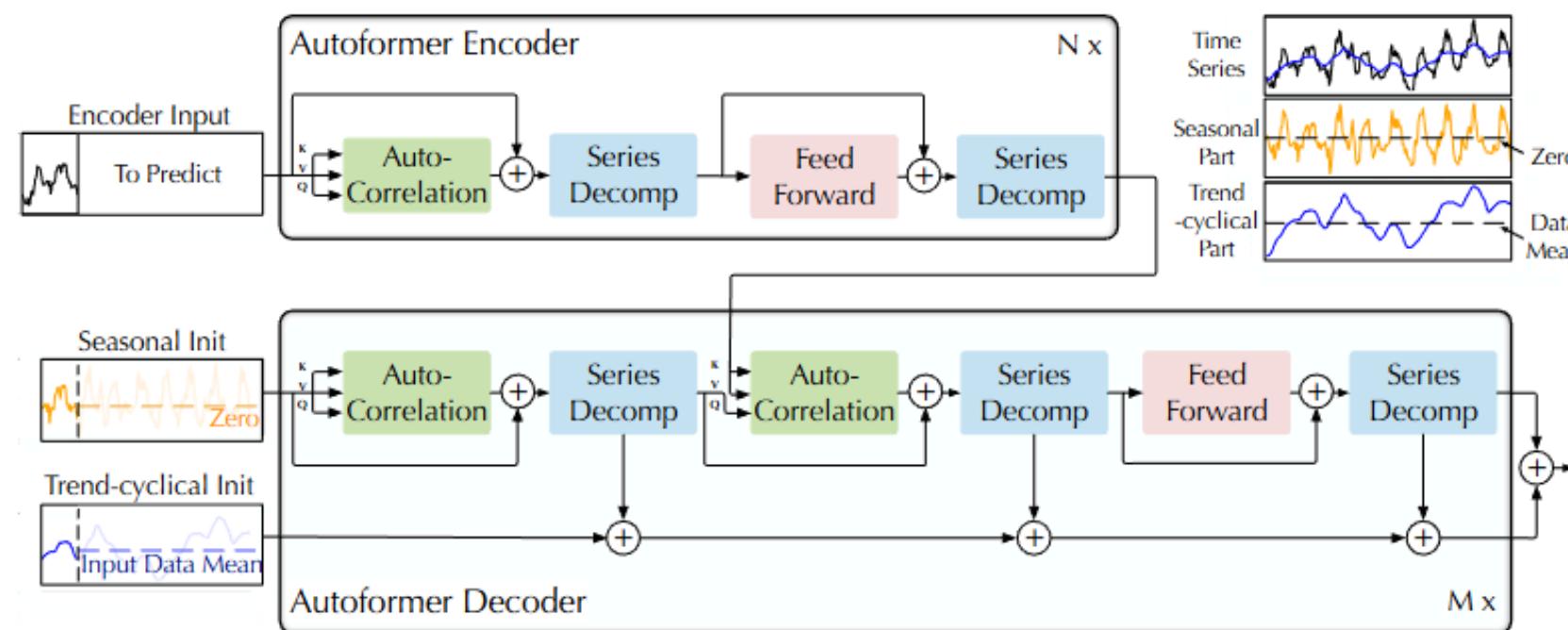
 - Temporal Fusion Transformer(TFT)

 - 정적 공변량 임베딩: 시간에 따라 변하지 않는 정보 반영
 - 변수 선택 수행: 각 변수가 예측에 얼마나 중요한지 결정
 - 관련 없는 부분을 건너뛰는 게이팅 구성
: 추가적인 LSTM, attention이 필요하지 않다고 판단할 경우 건너뜀



Related Work

- Attention 기반 모델
 - Autoformer



- 인코더 내부에서 시계열 데이터의 추세, 계절성 분해
- Autocorrelation: 시계열의 주기성을 활용하여 부분 시계열 단위로 계산

Related Work

- 기존 연구의 한계
 - 레이어 수준이 아닌 모델 수준에서 예측 정확도를 높이려고 시도
 - 시계열 추세 구성 요소 간의 상호 상관 관계에 대한 고려가 부족
 - Transformer 기반 모델은 성능이 좋지만 무거움

Correlation Recurrent Units

A. STL state

- 모델 내부에 STL 분해 방법을 사용하여 입력값을 추세, 계절성, 잔차로 분해하는 STL state 도입

Step1. 추세 분해

- 원본 시계열에서 추정된 추세 분해

Step2.

- 추세가 제거된 시계열을 주기에 따라 분할한 후 LOESS를 적용하여 임시 계절성 패턴을 추출

Step3.

- 임시 계절성에서 이동 평균과 LOESS를 순차적으로 적용

Correlation Recurrent Units

A. STL state

- 모델 내부에 STL 분해 방법을 사용하여 입력값을 추세, 계절성, 잔차로 분해하는 STL state 도입

Step4.

- 임시 계절성에서 저역 통과 필터링 결과를 빼내어 순수한 계절성 성분을 얻고 평균이 0이 되도록 정규화

Step5.

- 원본 시계열에서 현재 추정된 계절성을 제거

Step6.

- 계절성이 제거된 시계열에 LOESS를 적용하여 최종 추세 성분을 추출

Correlation Recurrent Units

A. STL state

- 잔차를 활용하여 이상치에 대한 강건성 확보
 - 최종 추세와 계절성을 활용하여 잔차 계산

$$R_t = Y_t - T_t - S_t.$$

- robustness weight 계산

$$\rho_t = \frac{B(|R_t|)}{6\text{median}(|R_t|)}.$$

- robustness weight를 활용하여 LOESS 가중치 조정

Correlation Recurrent Units

B. 자기 상관 및 상관 게이트

- 자기 상관 게이트(AG) → 현 시점의 입력이 이전 시점의 은닉 상태와 얼마나 강한 자기상관을 가지는지 계산

$$\hat{a}_t^s = \sigma(W_{x^s h^s} x_t^s + W_{h^s h^s} h_{t-1}^s + b_{\hat{a}^s})$$

$$\hat{a}_t^t = \sigma(W_{x^t h^t} x_t^t + W_{h^t h^t} h_{t-1}^t + b_{\hat{a}^t})$$

- 상관 게이트(CG) → 현 시점의 입력이 이전 시점의 은닉 상태와 얼마나 강한 상관을 가지는지 계산

$$\hat{c}_t^s = \sigma(W_{x^s h^s} x_t^s + W_{h^s h^s} h_{t-1}^t + b_{\hat{c}^s})$$

$$\hat{c}_t^t = \sigma(W_{x^t h^t} x_t^t + W_{h^t h^t} h_{t-1}^s + b_{\hat{c}^t})$$

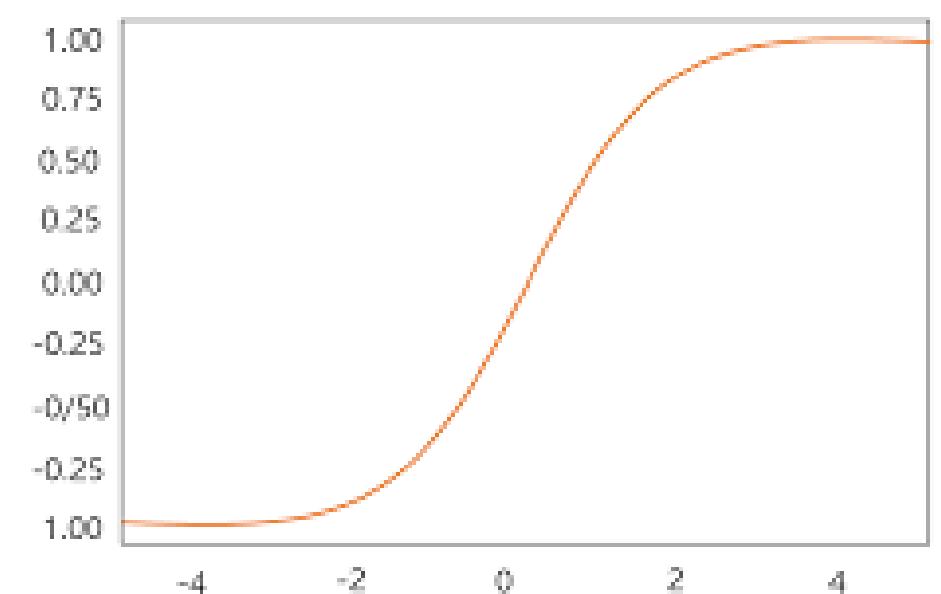
Correlation Recurrent Units

C. Hidden state 업데이트

$$h_t^s = \lambda \times \tanh(W_{x^s h^s} x_t^s + W_{h^s h^s} (\hat{a}_t^s \odot h_{t-1}^s) + b_{h^s}^s) \\ + (1 - \lambda) \times \tanh(W_{x^s h^t} x_t^s + W_{h^t h^t} (\hat{c}_t^s \odot h_{t-1}^t) + b_{h^t}^s)$$

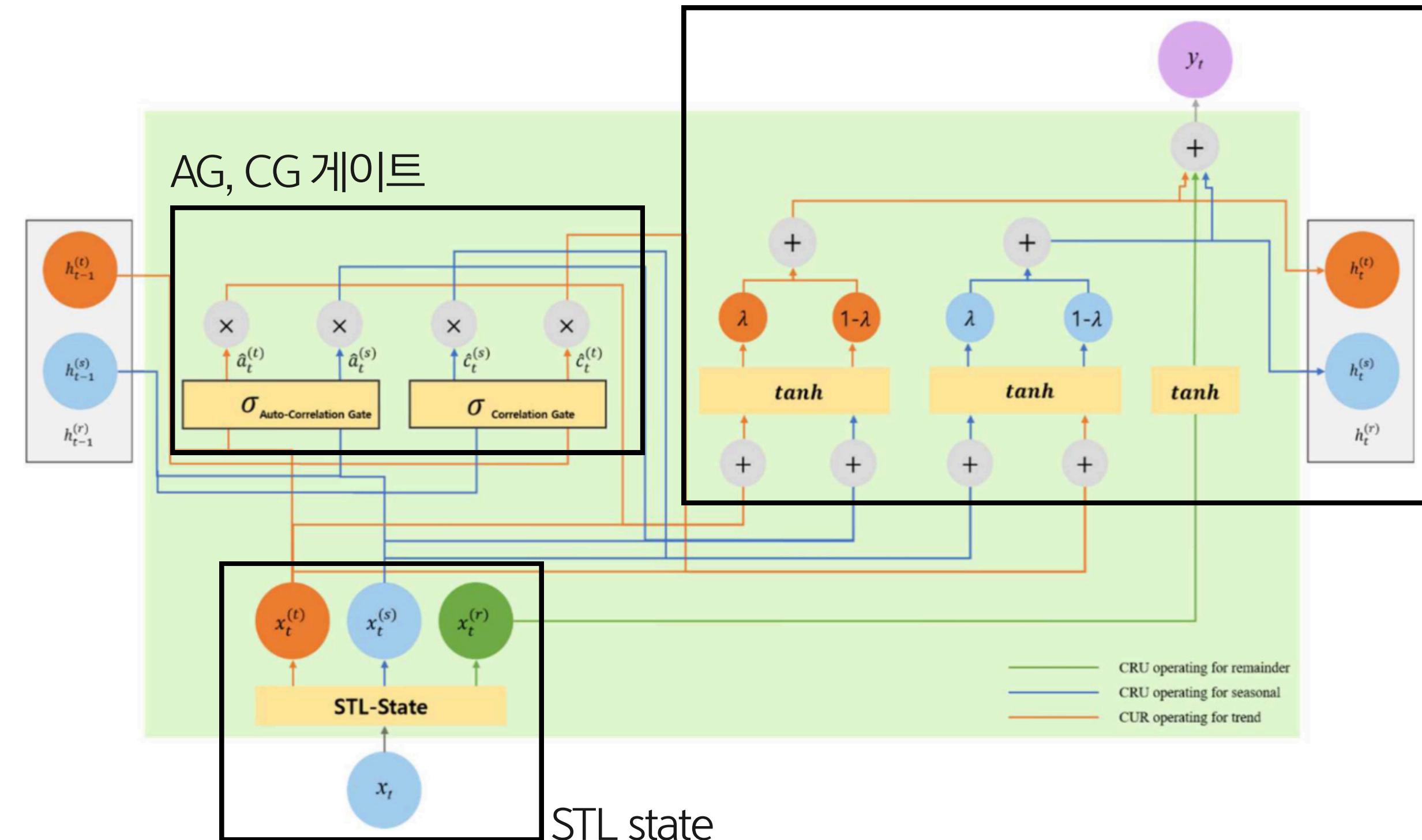
$$y = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

$$h_t^t = \lambda \times \tanh(W_{x^t h^t} x_t^t + W_{h^t h^t} (\hat{a}_t^t \odot h_{t-1}^t) + b_{h^t}^t) \\ + (1 - \lambda) \times \tanh(W_{x^s h^t} x_t^t + W_{h^s h^s} (\hat{c}_t^t \odot h_{t-1}^s) + b_{h^s}^t)$$



Correlation Recurrent Units

Hidden state 업데이트



Experiments

- 단변량, 다변량 시계열 데이터에 대한 실험
 - RMSE, MAPE, DILATE

TABLE I
DESCRIPTIONS OF DATASETS FOR THE EXPERIMENTS

	Data type	Notation	Num. rows	Num. columns	Collected duration	Unit	Data description
EXP-I	Univariate	UTD1	4567	1	1999.11.01- 2018.02.09	Daily	Baltic dry index (BDI) as the major freight indices for the dry cargo provided by the Baltic Exchange
		UTD2	3298	1	2008.07.04- 2021.10.20	Daily	EU emissions trading system (EU-ETS) price provided by the EU transaction log
		UTD3	2233	1	2019-07-26- 2019-10-26	Hours	Particulate matter (PM10) observation data from the Busan port area in Korea
		UTD4	50401	1	2015-01-01- 2020-09-30	Hours	Data collected from 1 randomly selected customer per hour of electricity consumption
EXP-II	Multivariate	MTD1	3151	4	2005.07.01- 2018.02.09	Daily	BDI, BCI, BPI, and BSI are included as freight indices for the dry cargo provided by the Baltic Exchange.
		MTD2	2140	11	2011.11.04- 2021.12.02	Daily	The eleven-market global stock index (Stock) from Yahoo! finance data
		MTD3	2233	5	2019-07-26- 2019-10-26	Hours	Five air pollution observation datasets from Busan port in Korea
		MTD4	50401	370	2015-01-01~ 2020-09-30	Hours	Data of per hour of electricity consumption from 370 customers

Experiments

- EXP-I
 - ARIMA, ETS, LSTM, STL-LSTM, DeepAR, MQ-RNN, Transformer, CCT, TFT, Autoformer와 비교
 - 모든 단변량 데이터셋에서 단기 및 중장기 예측 모두 가장 우수한 성능
 - 예측 기간이 길어질수록 기존 모델과의 성능 격차는 줄어드는 경향

TABLE III
EXP-I RESULTS OF ONE-STEP-AHEAD PREDICTION OF UNIVARIATE TIME-SERIES DATA

Model	UTD1			UTD2			UTD3			UTD4		
	RMSE	MAPE	DILATE	RMSE	MAPE	DILATE	RMSE	MAPE	DILATE	RMSE	MAPE	DILATE
ARIMA	35.496	3.806	797.75	0.987	3.645	0.665	4.866	17.400	17.217	191.311	2.022	19183
ETS	33.458	3.748	746.41	0.975	3.571	0.580	4.257	16.815	14.942	190.581	1.956	17874.
LSTM	33.379	3.676	621.32	0.899	3.462	0.544	3.811	14.775	11.930	188.454	1.943	17606
STL-LSTM	27.844	3.177	458.11	0.890	3.122	0.464	3.584	13.613	9.587	188.396	1.873	16493
MQ-RNN	24.976	2.929	384.99	0.855	2.915	0.412	3.494	13.389	9.148	185.632	1.760	14546
DeepAR	24.612	2.288	371.87	0.834	2.623	0.359	3.225	12.606	8.993	184.700	1.701	14342
Transformer	23.826	2.201	347.60	0.757	2.481	0.308	2.963	11.183	8.494	175.772	1.553	11501
CCT	22.838	2.123	308.19	0.734	2.182	0.256	2.475	9.346	7.610	172.279	1.461	10242
TFT	22.964	1.855	311.44	0.718	2.080	0.237	2.622	10.650	7.933	170.787	1.360	8891
Autoformer	21.321	1.318	271.69	0.699	1.956	0.218	2.356	7.775	7.414	168.110	1.337	8597
CRU*	20.551	1.140	252.23	0.650	1.907	0.193	2.212	6.524	7.239	165.840	1.329	8128

Experiments

- EXP-II
 - LSTM, STL-LSTM, DeepAR, Transformer, CCT, TFT, VAR, LSTNet, DSANet, Autoformer와 비교

- 단기 및 중기 예측에서 다른 모델 대비 최소 10% 이상의 뛰어난 성능
- 장기 예측에서는 일부 데이터셋에서 DSANet 및 어텐션 기반 모델들이 CRU보다 더 나은 성능을 보이는 경우 존재

→ 사용 가능한 변수의 수 증가에 따른 어텐션 기법의 효과로 인해 정확도가 크게 향상

TABLE VII
EXP-II RESULTS OF ONE-STEP-AHEAD PREDICTION OF MULTIVARIATE TIME-SERIES DATA

Model	MTD1			MTD2			MTD3			MTD4		
	RMSE	MAPE	DILATE	RMSE	MAPE	DILATE	RMSE	MAPE	DILATE	RMSE	MAPE	DILATE
VAR	21.089	2.393	2011	30.047	2.132	17968	0.00237	15.510	12.32	415.749	8.283	4.330
LSTM	16.416	1.872	1823	27.961	2.001	15571	0.00219	13.402	9.87	400.267	7.936	3.978
STL-LSTM	15.706	1.785	1726	24.472	1.676	11975	0.00216	13.464	10.03	384.857	7.685	3.687
DeepAR	14.341	1.657	1657	22.167	1.522	10676	0.00207	13.412	9.90	357.639	7.058	3.039
Transformer	12.248	1.417	1554	21.285	1.473	8449	0.00197	12.369	8.82	304.929	6.059	2.063
CCT	10.465	1.277	1524	20.644	1.440	7758	0.00196	12.124	8.40	291.281	5.806	1.807
TFT	11.873	1.393	1595	20.628	1.449	8196	0.00166	11.054	7.41	291.281	5.806	1.793
LSTNet-A	12.248	1.417	1618	17.800	1.288	6541	0.00165	10.386	6.54	287.109	5.767	1.758
DSANet	10.156	1.267	1551	17.536	1.235	6328	0.00156	10.155	6.30	276.753	5.615	1.613
Autoformer	9.845	1.201	1507	17.320	1.206	5755	0.00150	9.916	6.06	274.444	5.533	1.569
CRU*	8.385	1.042	1390	17.171	1.172	5494	0.00137	8.863	4.96	271.753	5.405	1.360

Ablation Studies

- CRU 구성 요소의 효과 분석
 - STL State 효과
 - 기본 RNN에 STL State를 추가했을 때 RNN 대비 약 25%의 성능 향상
 - AG, CG 효과
 - STL State가 적용된 RNN에 CG 또는 AG 중 하나만 추가했을 때 약 16%의 추가 성능 향상이 있었으며, 두 게이트를 모두 사용한 CRU가 단일 게이트 사용 시보다 평균 13% 더 우수한 성능

Ablation Studies

- 시계열 데이터의 특성에 따른 CRU 가중치 분석
 - 추세 또는 계절성 구성 요소만 포함된 데이터로 CRU를 학습시킨 후, CRU 내부의 추세, 계절성, 잔차에 대한 가중치를 분석
 - 추세 데이터로 학습했을 때는 추세 관련 가중치가 가장 높고, 계절성 데이터로 학습했을 때는 계절성 관련 가중치가 가장 높게 나타남
- CRU가 입력 시계열 데이터의 특성에 따라 내부적으로 중요도를 성공적으로 학습하고 반영함

Conclusion

- CRU

- 1) STL state: 새로운 시계열 입력 데이터를 세 가지 시계열 구성 요소(추세, 계절성 및 잔차)로 분해하는 셀
- 2) AG 및 CG: 시계열 구성 요소와 은닉 상태 간에 존재하는 상관 관계 및 자기 상관 관계를 추정하는 게이트

- 성능 검증

- 단변량 및 다변량 시계열 데이터를 통해 단기 및 중기 예측에서 최고 성능 확인
- 다변량 장기 예측에서 일부 어텐션 기반 모델이 우수한 경우 존재

- 학습 메커니즘 분석

- 입력 시계열의 특성(추세/계절성)에 따라 CRU 내부 파라미터가 적절하게 학습됨을 분석

Q&A



Discussion



Discussion

- 1** 다변량 시계열에서의 다른 시계열의 구성요소간의 상관관계(변수 A의 추세와 변수 B의 추세 등)를 반영하는 방법은 없을까?

- 2** 추세와 계절성을 분해하고 남은 잔차는 현 시점의 값만 반영해도 정보의 손실이 없을까?

- 3** CRU를 점 예측이 아닌 확률분포를 출력하도록 확장할 수 있을까?