시계열 예측AI의 기초이론, 최근동향 및 연구사례

2025-09-17

이화여자대학교 인공지능학과
최대진 (djchoi@ewha.ac.kr)

A Short Bio

- Daejin Choi, Ph. D.,
  - 2025.03 – present: Assistant Professor@Dept of AI, Ewha Womans Univ.
  - 2020.09 – 2025.02: Assistant Professor@Dept of CSE, Incheon National Univ.
  - 2019.01 – 2020.08: Research Scientist@Dept of IC, Georgia Tech

- Research Interest
  - Human-centered AI
    - AI in mental health & medical domains
    - Bias & fairness for AI
    - Journalism & fake news
  - Applied ML for real-world problems
    - Networking, security, …

Contents

시계열 이론 기초

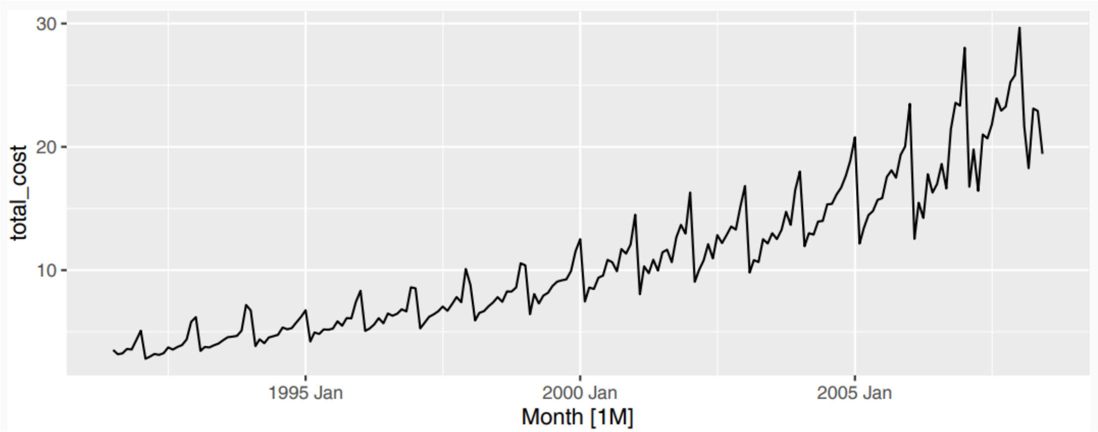
시계열 데이터란?

- 시간의 흐름 순서로 표현된 데이터
- 시계열 데이터 예측: 주어진 시계열 데이터로부터 여러 특징들을 파악해내고 활용하여 미래 시점의 데이터를 예측하는 것

시계열 데이터의 주 특징

- 이전의 값이 현재의 값을 결정하는 데 있어 영향이 있음
- 전체적으로 데이터 흐름에 있어 "패턴"이 존재함

- [주의] 단순한 "순서"데이터는 아님
  - 순서 뿐 아니라 **절대적인 시간 (낮/밤, 봄/여름/가을/겨울 등) 에 따라 특정 규칙이나 패턴이 형성**되는 모습이 있음
  - 주로 장시간 (수십 주, 수십 개월, 수 년) 동안 수집된 경우가 많음



시계열 데이터의 주요 분류

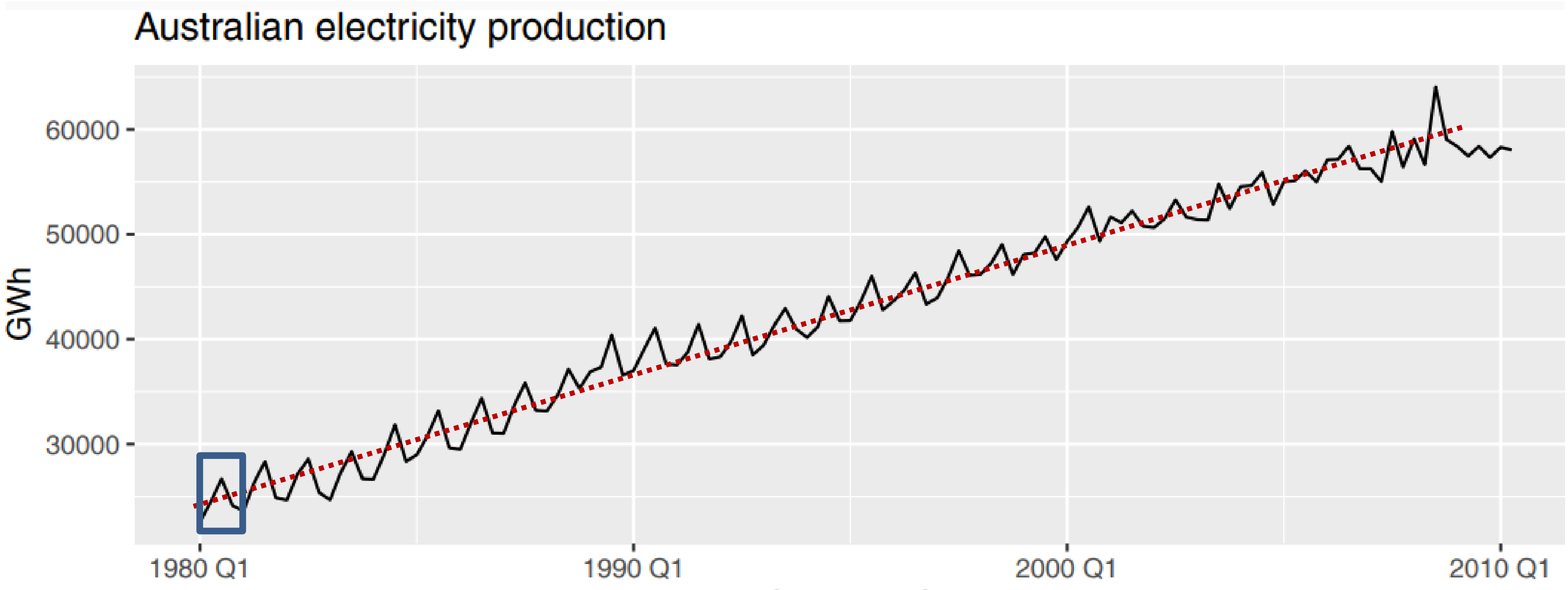
- 단변량 (univariate) vs. 다변량 (multivariate)
  - Single variable vs. multiple variables

- 선형 (linear) vs. 비선형 (non-linear)

- 이산적 (discrete) vs. 연속적 (continuous)
  - 실제로, 산업계에서 대부분 사용되는 시계열 예측은 이산적임
  - E.g., 분당 예측, 시간당 예측, …

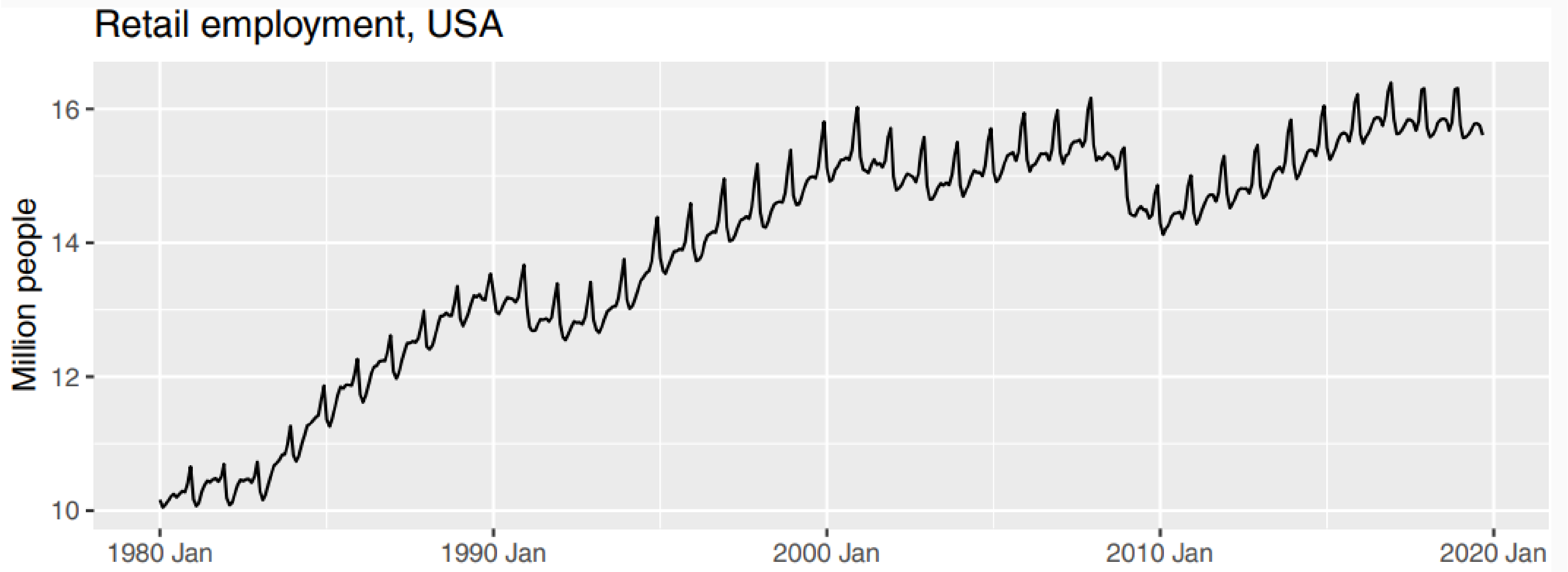
시계열 데이터의 표현: 선 그래프 (line plot)

- Most simple, but most intuitive!



- 규칙적 패턴, 전반적 경향, ...

다른그래프에서는?



- 전체적인 경향이 존재! → 일반 순서와는 다름
  - 특히, 특정 시간 (예: 월)에 따라 규칙적으로 보이는 패턴이 있음!

  → 이것이 다른 순서데이터와 시계열 데이터와의 차이점

시계열 데이터 구성

시계열 데이터의 주요 구성

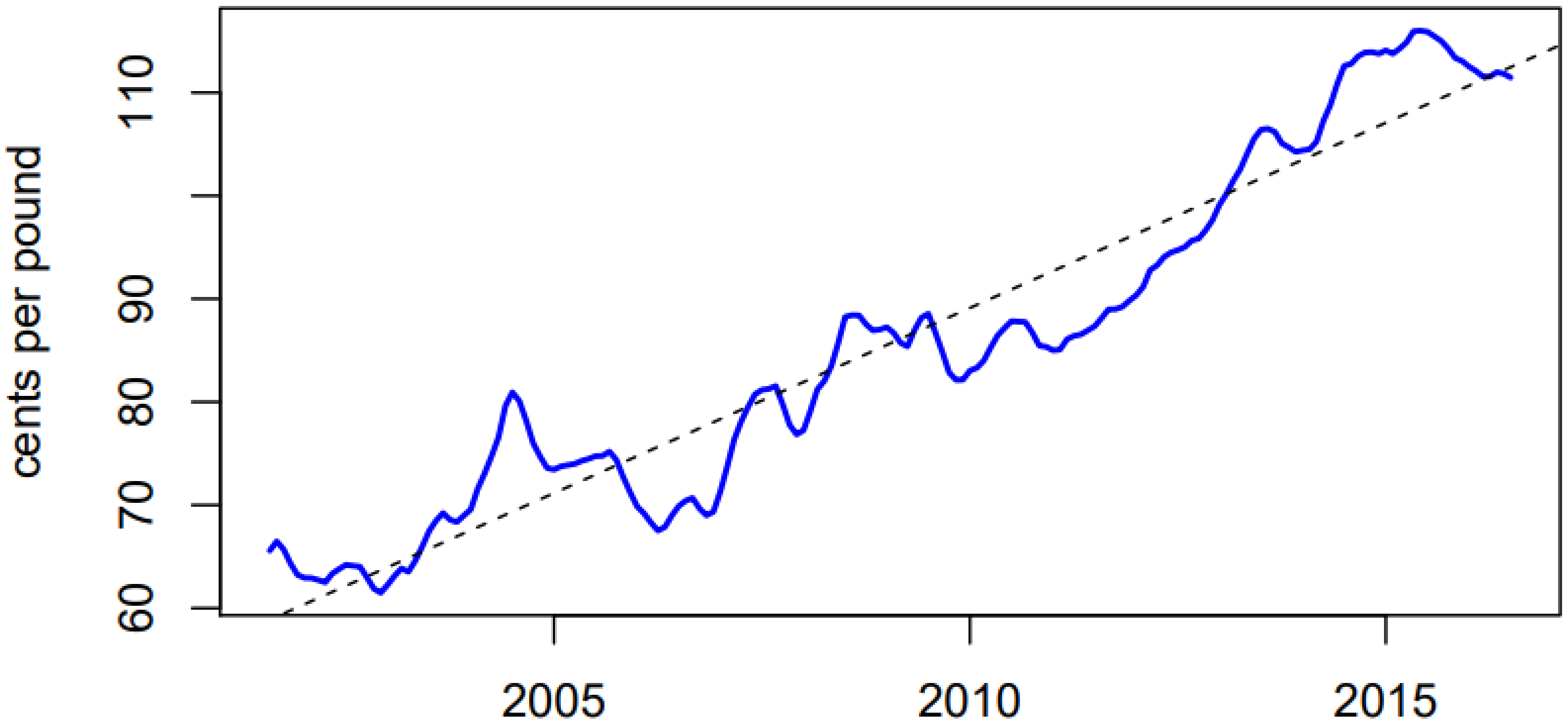
- 시계열 데이터의 경우, 특정 기준에 따라 명백히 보이는 패턴이 존재
  - 계절별로 규칙적인 반복
  - 전체적인 큰 경향성

- 이러한 기준에 따라, 과학자들은 시계열 데이터를 다음과 같은 주요 구성 요소로 분해해 왔음
  - 경향성 (Trend)
  - 계절성 (Seasonal)
  - 반복성 (Cyclic)
  - 기타 규칙적이라고 판단하기 어려운 것들 (Irregular)

- 이러한 구성 요소를 잘 파악하는 것이 중요함!

시계열 구성요소: (1) 경향성

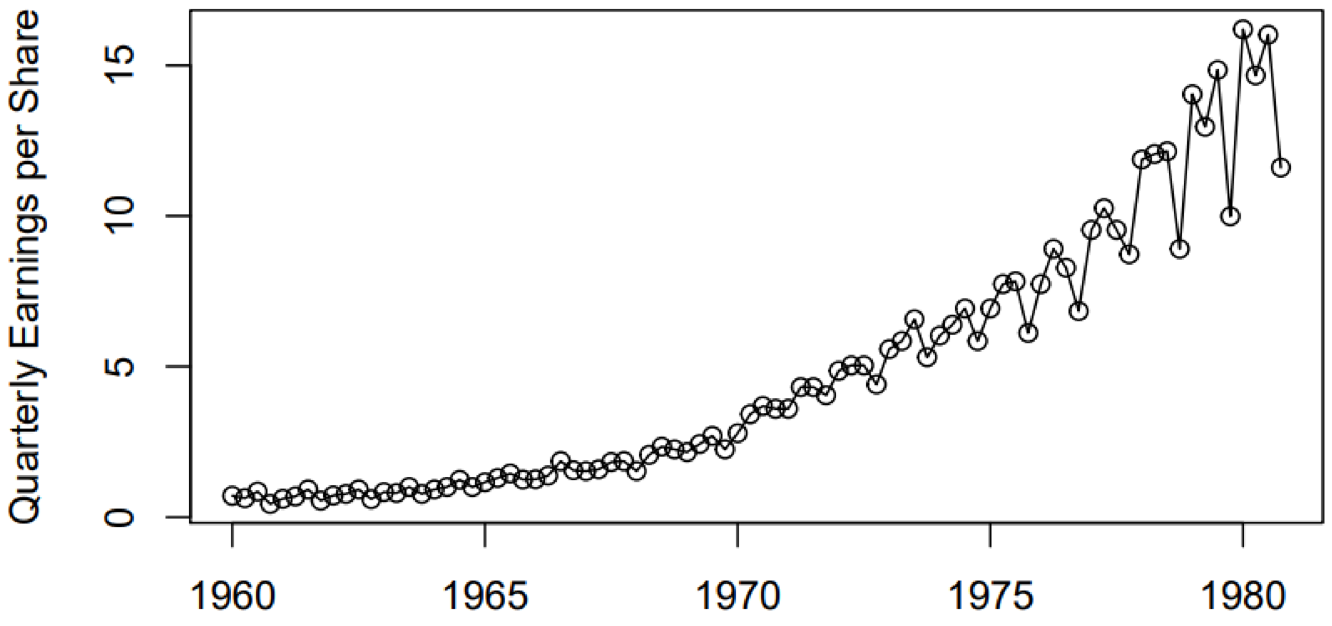
- 시계열 데이터 전체에서 보이는 전반적인 추이 (= 모양)



The price of chicken: monthly whole bird spot price, Georgia docks, US cents per pound, August 2001 to July 2016, with fitted linear trend line.

시계열 구성요소: (2) 계절성

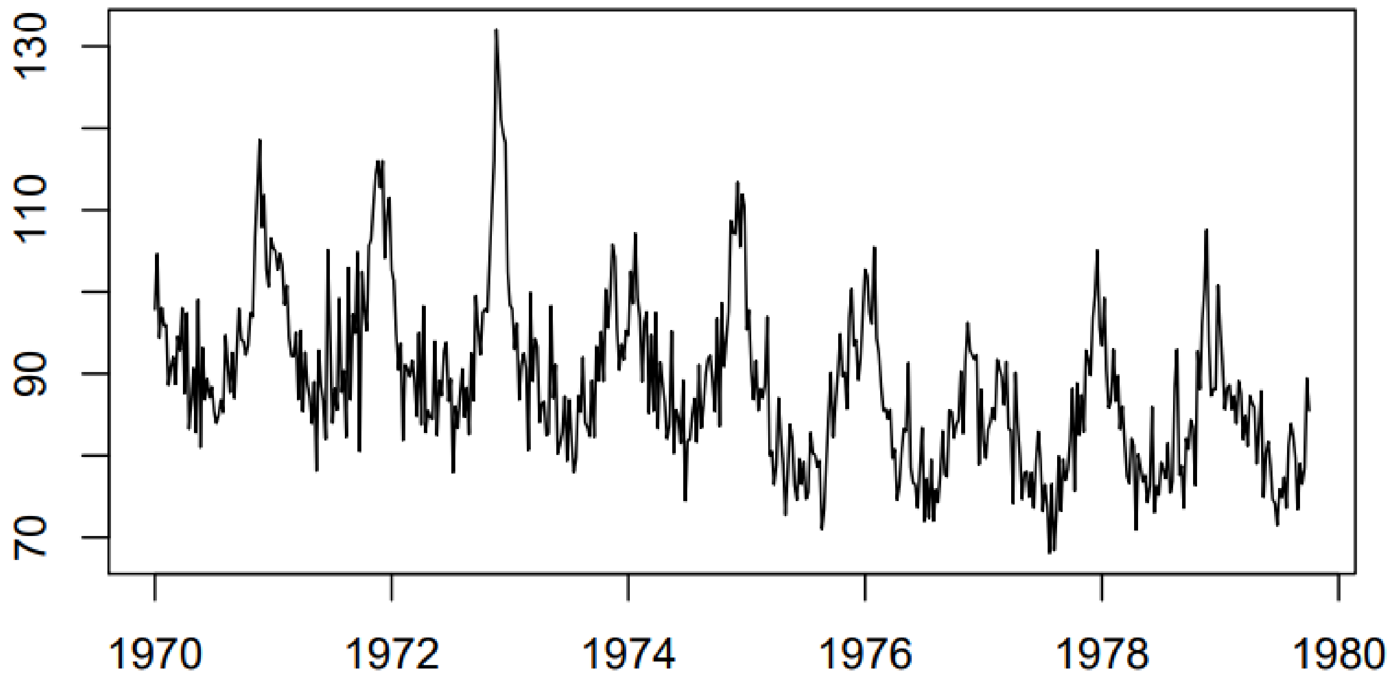
- (여러가지 이유로) 동일 계절마다 반복되는 특징
  - 타 계절과의 차이가 존재
  - 동일 계절에서는 규칙적인 패턴



Johnson & Johnson quarterly earnings per share, 84 quarters, 1960-I to 1980-IV.

시계열 구성요소: (3) 반복성

- 주로 긴 시간 주기에서 반복되는 패턴. 특정 정해진 시간 길이가 아니라, 주기단위로 반복되는 특징이 있음
  - 주변 환경이나 상황에 따라 반복



Average weekly cardiovascular mortality in Los Angeles County. There are 508 six-day smoothed averages obtained by filtering daily values over the 10 year period 1970-1979.

반복성 vs. 계절성

- 직관적으로 “계절“과 관련이 있느냐 없느냐로 나눌 수 있음
  - 하지만, 최근 계절과 직접적인 관련이 없어도, 시간과 관련이 있는 경우 “계절성”으로 판단하기도 함

- 즉, 최근에는 시간대와 관련있는 패턴은 계절성으로, 시간대와 관련없는 반복 패턴은 반복성으로 정의하는 경향이 있음.
  - 추가적으로, 반복성의 경우 일반적으로 긴 기간에 의해 반복되는 경향이 있음
  - 예) 아침시간/저녁시간의 규칙적 패턴 → 계절성
  - 예) 500일 단위의 규칙적 패턴 → 반복성

- 최근에는 경향성 (trend)과 반복성은 합쳐진 요소로 보는 측면도 존재

시계열 구성요소: (4) 비정규적 (irregular) 요소

- 예측 불가능한 모든 오류들을 포함한 요소
  - 예: 갑작스런 사고, 일시적 오류 등
- 통계적인 방법으로 계산 불가
- 큰 영향을 끼치지 않는다는 가정하에 분석되는 경우가 많음
- 즉, 주로 경향성, 계절성, 반복성 요소를 추출해내고, 이러한 요소들을 자세히 분석하는 경우가 많음

시계열 데이터 분해

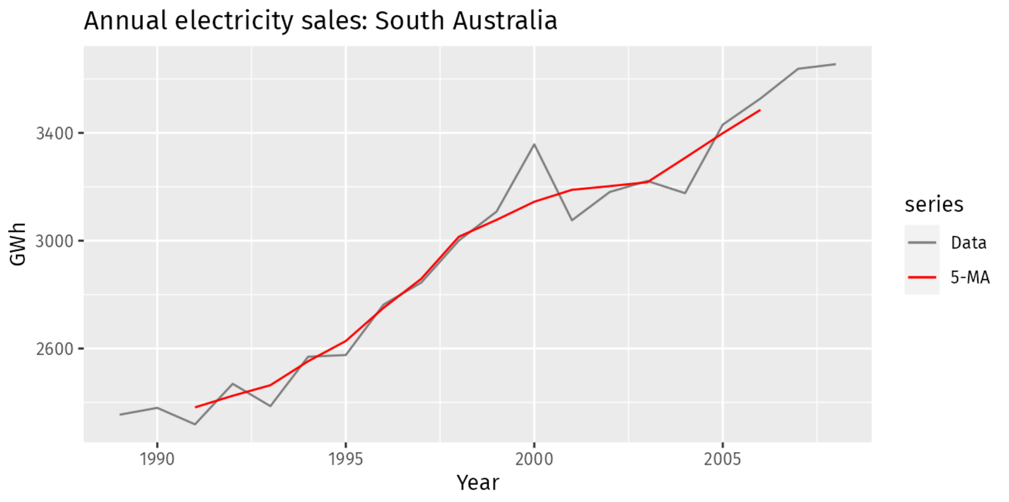
배경: 이동 평균 (Moving Average)

- 여러가지 변동성을 없애고, 전반적인 경향을 보기위해 수행하는 데이터 스무딩 (Smoothing) 방법

- 시계열에서는 경향성+반복성 요소를 추출하기 위한 용도로 사용됨

- m-이동평균 공식:

$$\hat{T}_t = \frac{1}{m}\sum_{j=-k}^{k} y_{t+j}, \qquad m = 2k+1$$

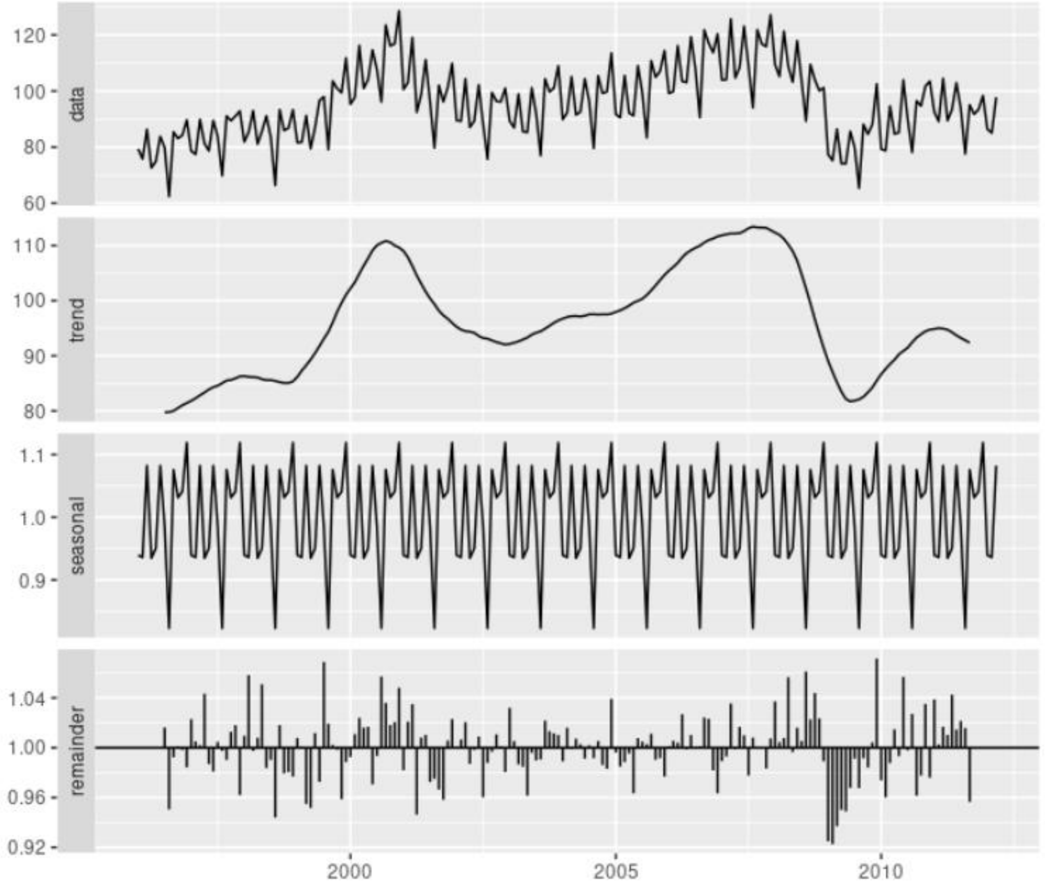


이동평균 (상세)

- m 이 커졌을 때? →더 많이 스무딩 됨 (정보 손실이 높아짐)

- 추가적으로, 양쪽 끝단에서 계산을 할 수 없는 데이터가 많아짐

시계열 분해: 고전적 분해방법



$$\boldsymbol{y_t} = \mathbf{f}(\boldsymbol{S_t}, \boldsymbol{T_t}, \boldsymbol{R_t})$$

- $\boldsymbol{y_t}$ = t에서의 시계열 값
- $\boldsymbol{T_t}$ = t에서의 경향성 요소 값
- $\boldsymbol{S_t}$ = t에서의 계절성 요소 값
- $\boldsymbol{R_t}$ = t에서의 나머지 요소 값 (오차 등)

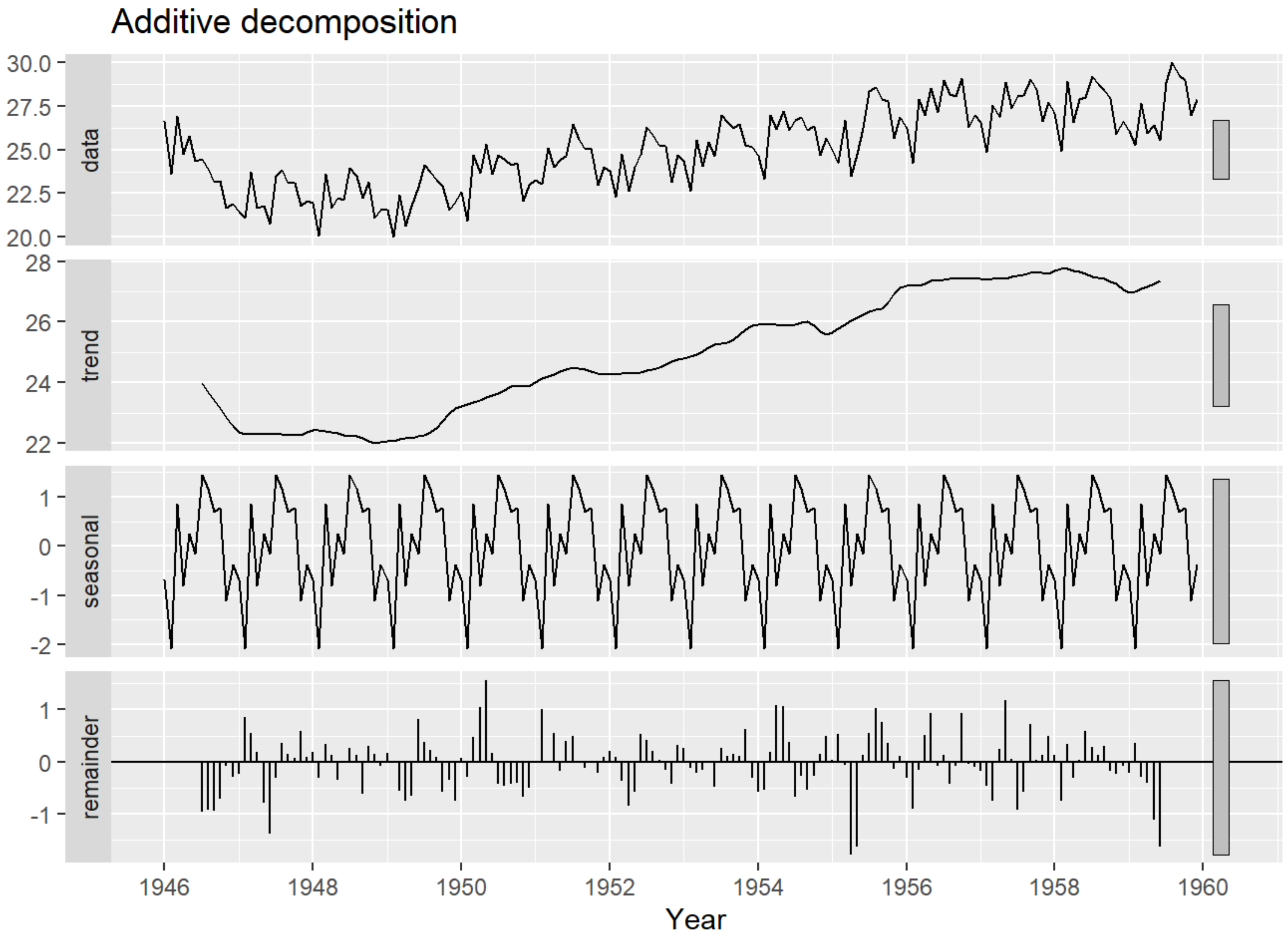
- 덧셈 기반 분해: $\boldsymbol{y_t} = \boldsymbol{S_t} + \boldsymbol{T_t} + \boldsymbol{R_t}$
- 곱셈 기반 분해: $\boldsymbol{y_t} = \boldsymbol{S_t} \times \boldsymbol{T_t} \times \boldsymbol{R_t}$

고전적 시계열 분해 방법

- 이동평균 (= 경향성) 에 기반하여 분해하는 방법

- 분해 방법
  - 해당 시점의 이동평균 값을 구함: $\hat{T}_t$
  - 실제 값에서 이동평균 값을 뺌: $y_t - \hat{T}_t$
  - 미리 지정된 단위의 이전 값들을 모아서 계절 평균을 구함
    - 예: 매월 1일 등
  - 3의 계절평균 값을 0으로 간주하여, 해당 평균값보다 t 시점에서의 값이 얼마나 큰지 (혹은 작은지)에 대한 값을 계산함: 계절성 요소 ($\hat{S}_t$)
  - 계절성, 경향성을 뺀 값을 나머지로 간주 $\hat{R}_t = y_t - \hat{T}_t - \hat{S}_t$

고전적 방식을 통한 분해 결과



고전적 분해방법 (상세)

- 계절적 변화가 별로 없는 경우 → 덧셈기반 분해

- 계절적 변화가 점점 커지는 경우 → 곱셈기반 분해

- 경제 관련된 것 (집 가격, 주식 등) 은 곱셈기반 분해가 좋음

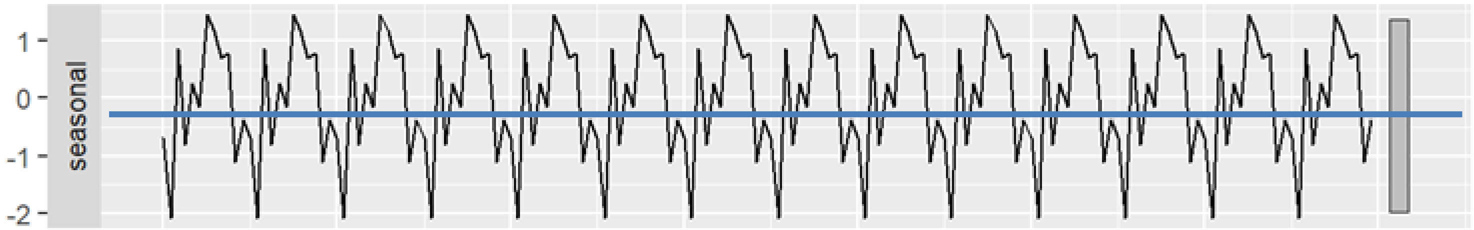
- 곱셈기반 분해는 log의 덧셈 기반 분해 방법으로 쉽게 구현이 가능함

$$\boldsymbol{y_t = S_t \times T_t \times R_t} \quad \Rightarrow \quad \boldsymbol{log y_t = log S_t + log T_t + log R_t}$$

고전적 분해방법 (상세)

- 간단하고 가장 대중적으로 사용됨

- 단점
  - 이동평균에 기반한 분해방법: 이동평균의 크기에 따라 분해가 잘 안되는 경우가 많으며, 앞뒤 분해가 불가능
  - 경향성을 제외한 나머지 값 중 계절성을 우선해서 보기 때문에 무조건 계절성이 존재하는 것처럼 보이는 경향이 있음



다른 분해방법들

- 여러 기관들에서 각 목표/특성에 따라 시계열 분해하는 방법을 조금씩 다르게 가져가고 있고, 현재도 개발 중

- X11, X12, X12-ARIMA, X13-ARIMA, X13-ARIMA-SEATS,…
- Seasonal Extraction in ARIMA Time Series (SEATS)
- Seasonal and Trend decomposition using Loess (STL) and Multiple STL (MSTL)
- These decomposition methods have still been in development!

TIME-SERIES관련 예측 모델: STOCASTIC

ARIMA Overview

- Auto-Regressive Integrated Moving Average
  - **AR** (Autoregression, 자기회귀): 이전에 관찰된 패턴이 다시 보이는 현상
    - i.e., 과거의 (lagged) 패턴으로부터 현재 패턴이 모델링 가능하다는 의미
  - **I** (Integrated): 차분 (differential)을 통해 시계열의 정상성 (stationary)을 확보하는 것
  - **MA** (Moving Average): 잔차 오류 (residual error)와 실제 값과의 차이간 모델링을 통해 대상 값을 모델링 하는 방법

- Putting all together → ARIMA
  - AR & MA → "stationary" 시계열을 예측하기 위한 선형적 모델
  - I → 주어진 시계열 데이터의 "정상성"을 확보하기 위한 방법

ARIMA Overview (Cont’d)

- ARIMA모델의 parameter 는 AR(p), I(d), MA(q)의 p, d, q로 구성됨.
  - p (lag order)
  - d (degree of differencing)
  - q (order of moving average)

- 즉, p, d, q의 값에 따라 AR, I, MA 각 모델을 포함할 수 있음
  - 예: p = 0, d = 0 → MA모델

AutoRegressive (AR) 모델

- Motivation
  - Current value of the series, y_t, can be explained as a linear combination of p past values (i.e., y_{t-1}, $y_{t-2}, \ldots, y_{t-p}$), together with a random error in the same series
- Definition
  - An autoregressive model of order p, abbreviated AR(p), is of the form

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + \varepsilon_t$$

    where ε_t is white noise and y_t is **stationary** time series
  - NOTE: $\phi_1, \phi_2, \ldots, \phi_p$ ($\phi_p \neq 0$) are model parameters
- $AR(0)$ = white noise
- $AR(1)$ → $y_t = \phi_1 y_{t-1} + \epsilon_t$
  - $|\phi_1|$ is close to 0 → almost white noise
  - $\phi_1 < 0$ → the series tends to oscillate between positive and negative values
  - $\phi_1 = 1$ → Random walk (not stationary)
  - $\phi_1 > 1$ → Explosive AR (still stationary, but not useful)

Moving Average (MA)

- moving average models here ≠ moving average smoothing
- Motivation
  - (Recall) in AR models, current observation (y_t) is regressed using the previous observations ($y_{t-1}, y_{t-2}, \dots, y_{t-p}$), plus an error term ($\epsilon_t$) at current time point
  - One problem of AR model is the **ignorance of correlated noise** structures (which is unobservable) in the time series
  - In other words, the imperfectly predictable terms in current time, ϵ_t, and previous steps, ϵ_{t-1}, ϵ_{t-2},…, ϵ_{t-q}, are also **informative** for predicting observations

- Equation (q-th order)

$$y_t = c + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \cdots + \theta_q \varepsilon_{t-q}$$

where ε_t is white noise.

ARMA: AR + MA

- ARMA is a concatenation of AR and MA
- AR → causality from past values
- MA → (autocorrelated) errors (that have not been considered by AR)

- Definition
  - A time series $\{\boldsymbol{y_t}\}$ is $ARMA(p, q)$ if it is **stationary** and

$$\boldsymbol{y_t} = \boldsymbol{\epsilon_t} + \sum_{\boldsymbol{i=1}}^{\boldsymbol{p}} \boldsymbol{\phi_i y_{x-i}} + \sum_{\boldsymbol{j=1}}^{\boldsymbol{q}} \boldsymbol{\theta_j \epsilon_{t-j}}$$

where $\boldsymbol{\phi_i} \neq 0$ and $\boldsymbol{\theta_j} \neq 0$ and $\boldsymbol{\epsilon_t}$ is white noise

"Stationarize" Non-stationary Time Series

- One limitation of ARMA models is the stationarity condition
- In many situations, time series can be divided into two components
  - A non-stationary trend series
  - A zero-mean stationary series
- How to divide?
  - **Detrending**: subtracting with an estimate for trend and deal with residuals
  - **Differencing:** modeling the trend as "a random walk with drift" stochastic process

$$X_t = \delta + X_{t-1} + w_t = \delta t + \sum_{i=1}^{t} w_i$$

  - **Smoothing**
  - **Decomposition**
  - ...

Differencing

- Differencing helps to stabilize the mean

- (First-order) differencing
  - The "differenced series" is the *change* between each observation in the original series: $y'_t = y_t - y_{t-1}$
  - The differenced series will have only $T - 1$ values since it is not possible to calculate a difference y'_1 for the first observation

AR + MA + Differencing = ARIMA

- Equation: ARIMA(p, d, q)

$$y'_t = c + \underbrace{\phi_1 y'_{t-1} + \cdots + \phi_p y'_{t-p}}_{\textbf{AR}} + \underbrace{\theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q}}_{\textbf{MA}} + \varepsilon_t$$

Differencing (here, first-order diff.)

- White noise: ARIMA(0, 0, 0)
- AR(p): ARIMA(p, 0, 0)
- MA(q): ARIMA(0, q, 0)
- How to choose parameters? → maximum likelihood estimation or non-linear least-squares estimation

Seasonal ARIMA

- A problem of ARIMA is the lack of seasonality

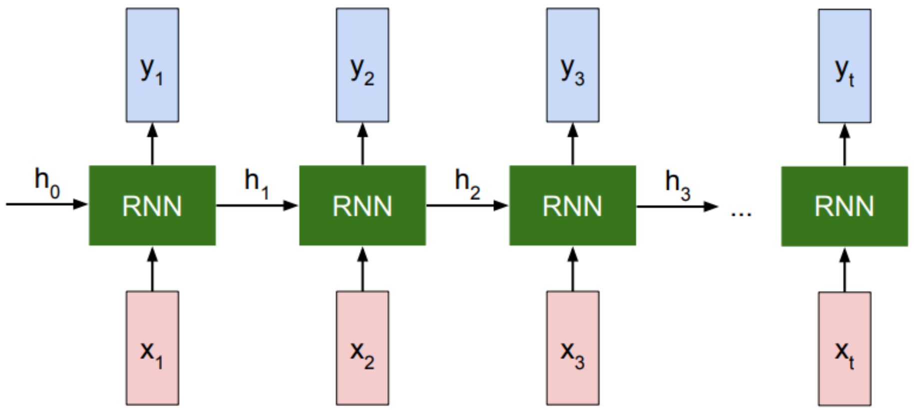
  → SARIMA can address

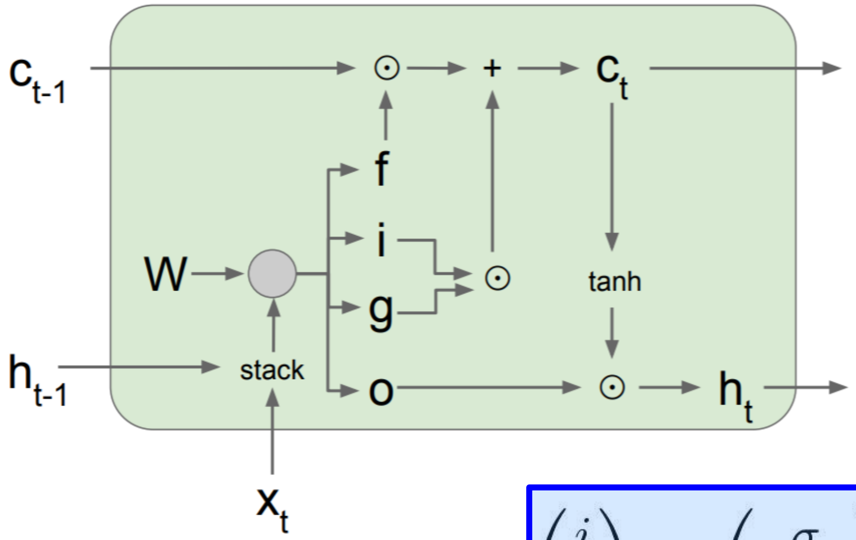
- Key Idea
  - Add seasonal terms in ARIMA model

ARIMA	$\underbrace{(p, d, q)}$	$\underbrace{(P, D, Q)_m}$
	↑ Non-seasonal part of the model	↑ Seasonal part of of the model

DEEP LEARNING MODELS

RNN/LSTM is an intuitive for encoding sequences





i: Input gate, whether to write to cell
f: Forget gate, Whether to erase cell
o: Output gate, How much to reveal cell
g: Gate gate (?), How much to write to cell

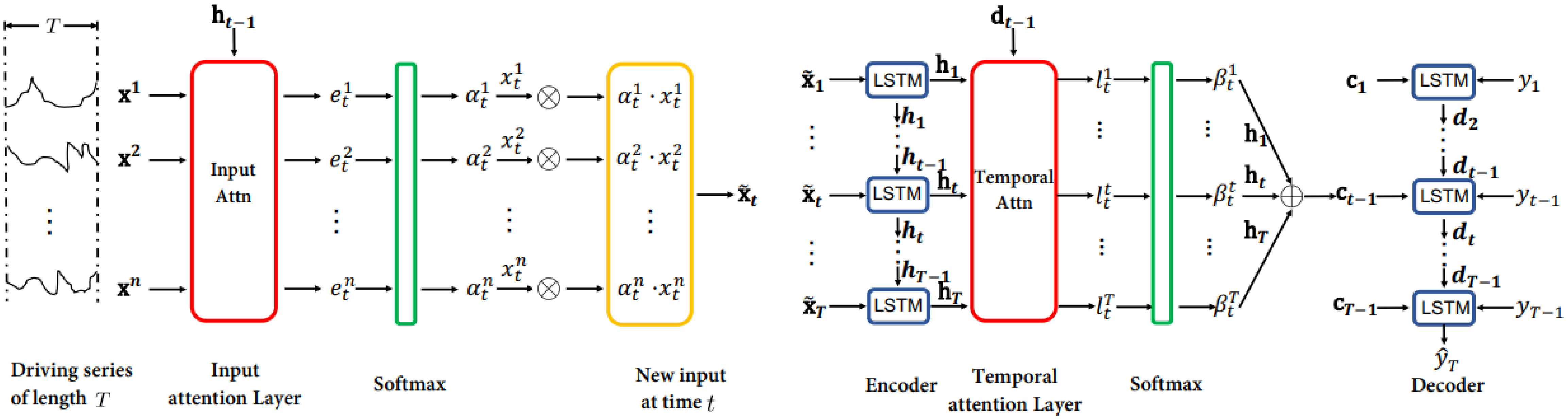
$$\begin{pmatrix} i \\ f \\ o \\ g \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{pmatrix} W \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix}$$

$$c_t = f \odot c_{t-1} + i \odot g$$

$$h_t = o \odot \tanh(c_t)$$

A Dual-Stage Attention-Based RNN [Yao, 2017]

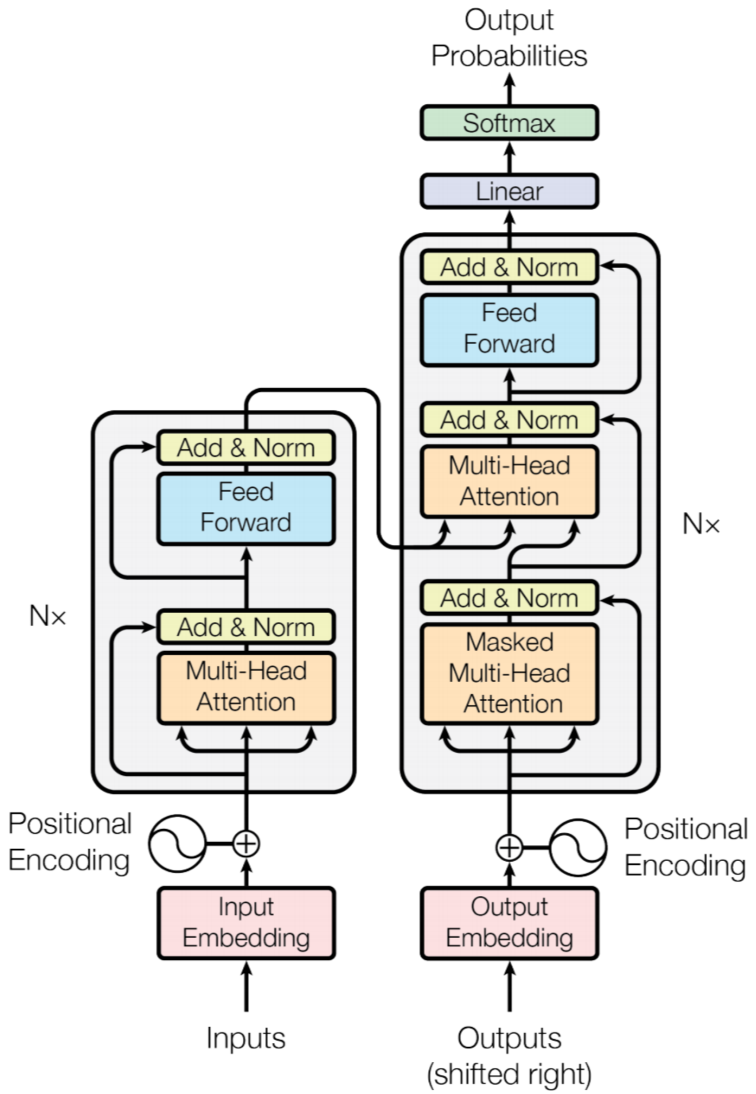
- Dual Attention
  - Among features
  - Among time-series



(a) Input Attention Mechanism (b) Temporal Attention Mechanism

Appearance of Transformer

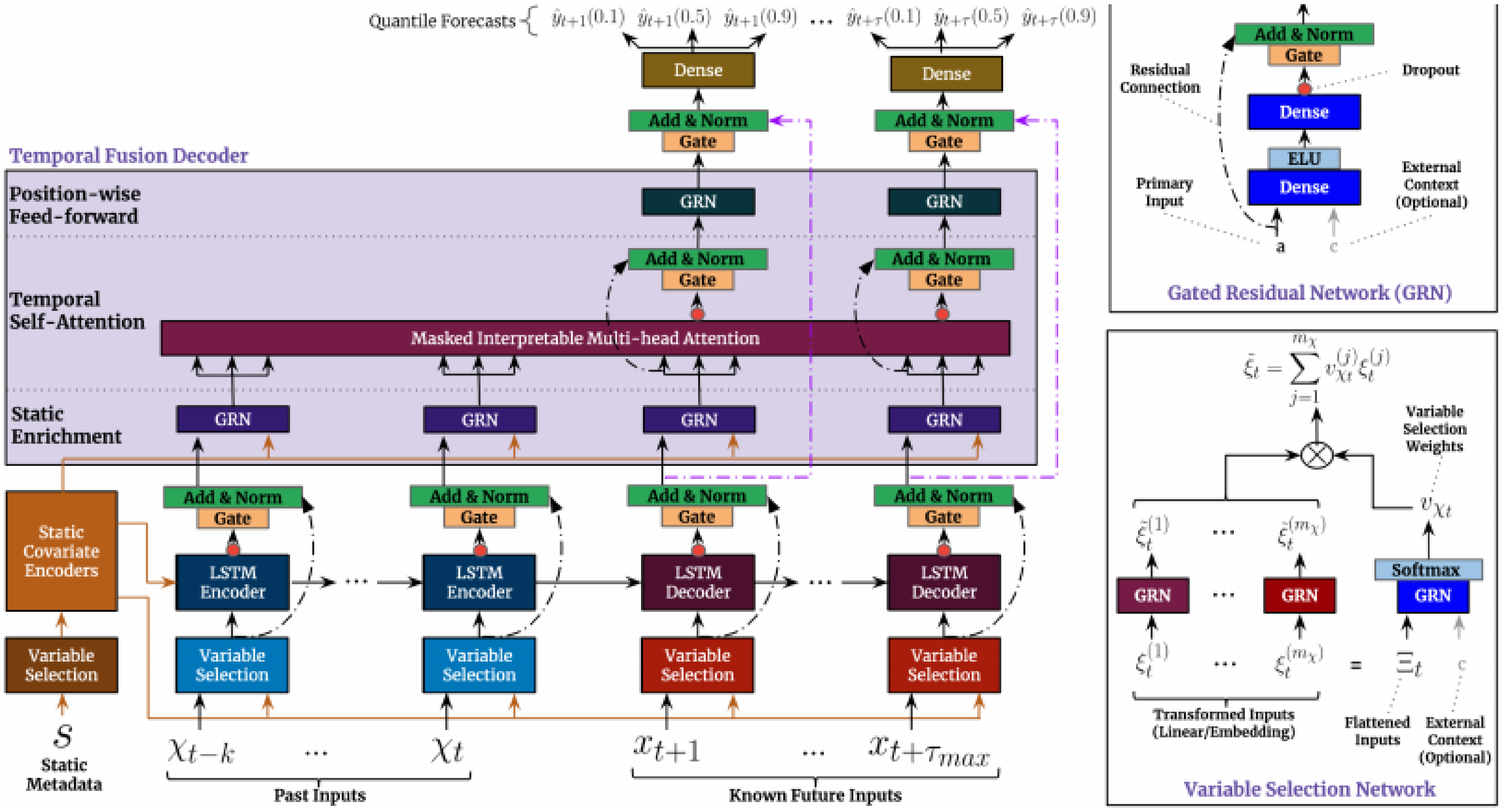
- Model: advanced version of AE
  - Final cost/error function is standard cross-entropy error on top of a softmax classifier
- Sequence-to-sequence prediction
  - Task: machine translation with parallel corpus (predict each translated word)
- A lot of variants for time-series forecasting
  - Temporal Fusion Transformer (TFT), 2020
  - Informer, 2021
  - Autoformer, 2021
  - FEDformer, 2022
  - PatchTST, 2023



This and related figures from paper: https:arxiv.org/pdf/1706.03762.pdf

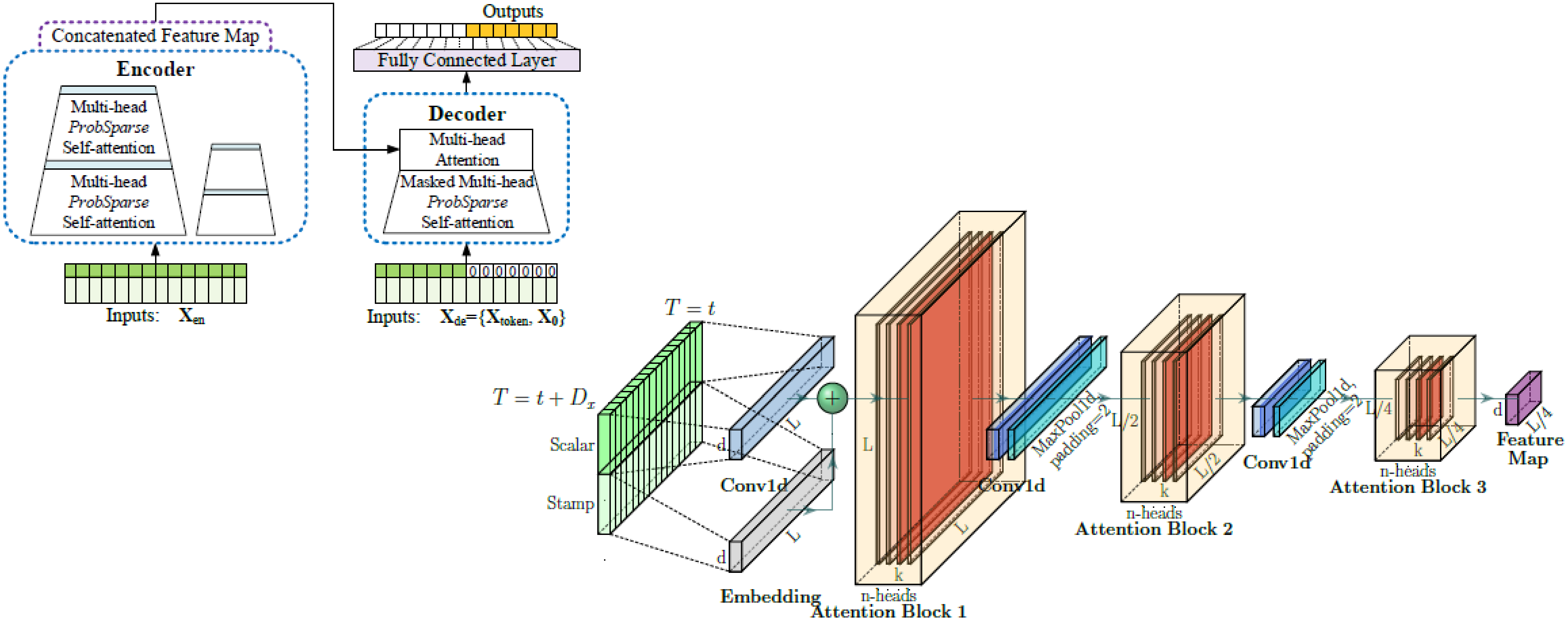
Temporal Fusion Transformer (2020)

- Good for multivariate time series with providing feature importance



Informer (2021)

- Targeting efficient "long-sequential" time-series forecasting



Autoformer (2022)

- Reflecting “trend decomposition” & “autocorrelation” in model architecture

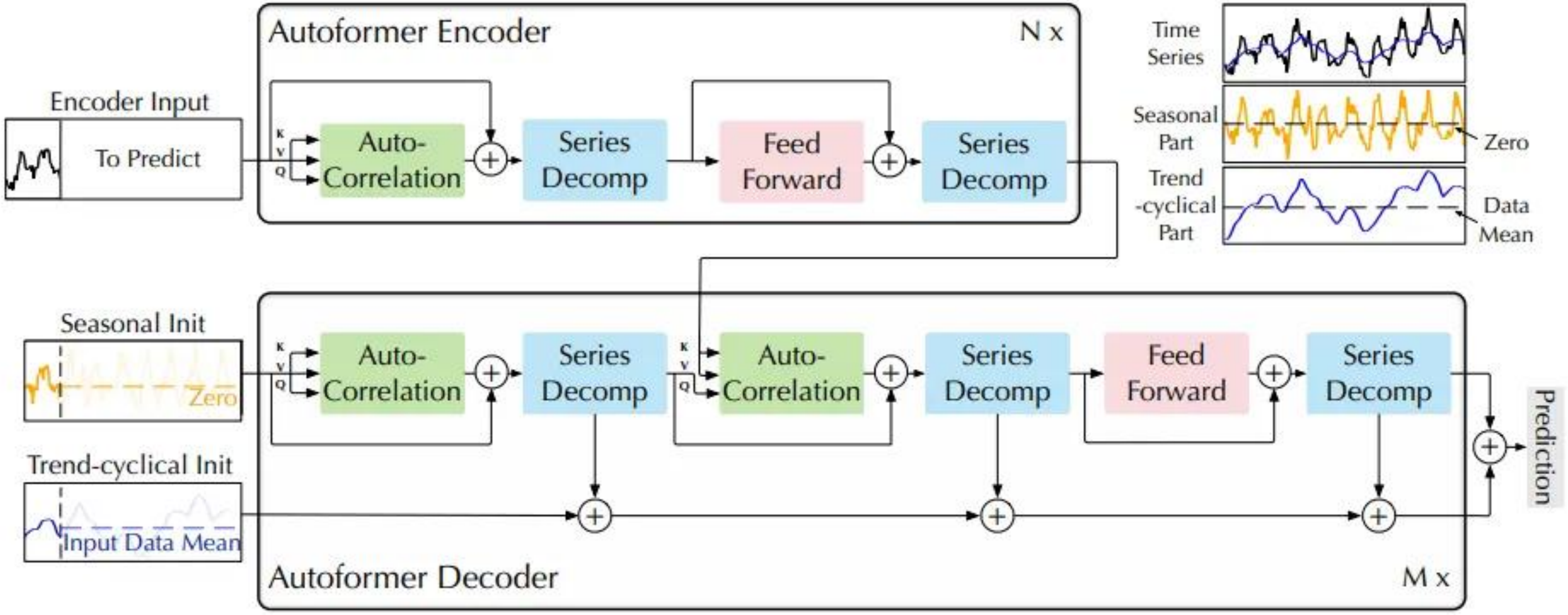


Figure 1: Autoformer architecture. The encoder eliminates the long-term trend-cyclical part by series decomposition blocks (blue blocks) and focuses on seasonal patterns modeling. The decoder accumulates the trend part extracted from hidden variables progressively. The past seasonal information from encoder is utilized by the encoder-decoder Auto-Correlation (center green block in decoder).

FEDformer (2024)

- Learning "short-term" frequency (high-frequency) with Fourier Transform

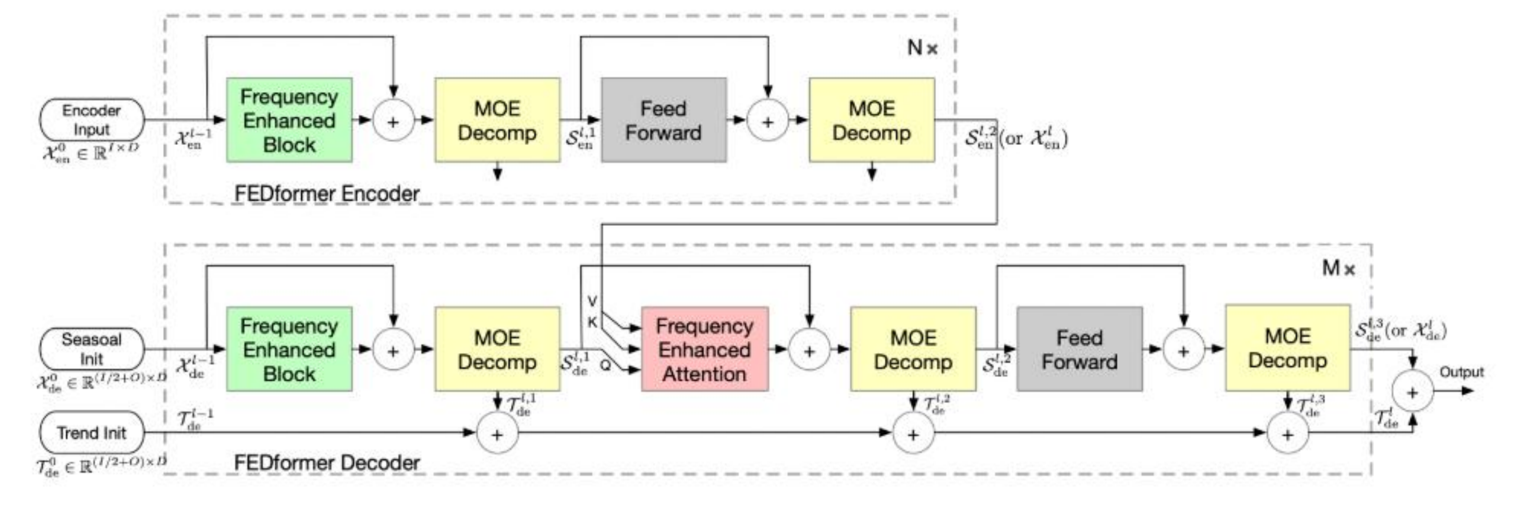


Figure 1. Overall structure of FEDformer

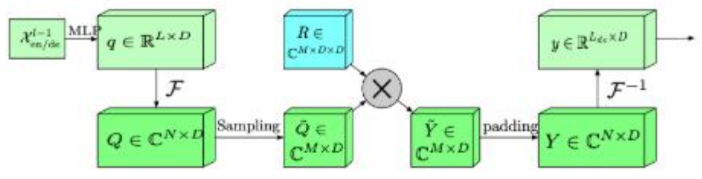


Figure 2. Frequency Enhanced Block (FEB)

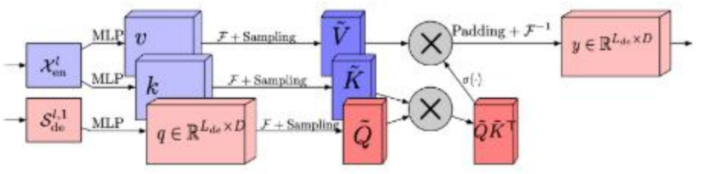
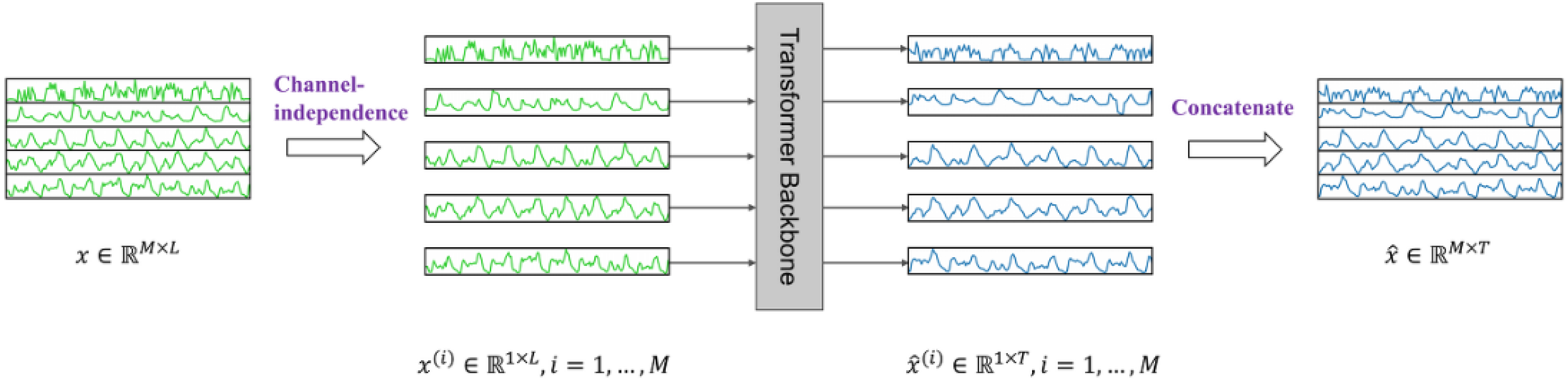


Figure 3. Frequency Enhanced Attention (FEA)

PatchTST (2023)

- SOTA model over other transformer-based models for diverse benchmark datasets
- Segmenting time-series values and put the values into ViT (vision transformer)



Summary of Sequence-based Deep learning model

- 시계열 특성 (long-term, short-term 특징, 주기성, autocorrelation 등)을 반영하는 형태로 발전

- 전통적인 시계열 모델 → 신경망 기반 모델 → 트랜스포머 모델 → 시계열 특성 반영 트랜스포머 모델

- 산업계에서 popular하게 사용되지는 않음! Why?
  - Overparameterized despite similar performance
  - Why? → 시계열이 명확하게 보이는 곳에는 통계적인 모델도 충분히 좋은 성능을 보이기 때문

Are Transformers Effective for Time Series Forecasting? (2022)

- Simple linear model outperforms transformer-based model

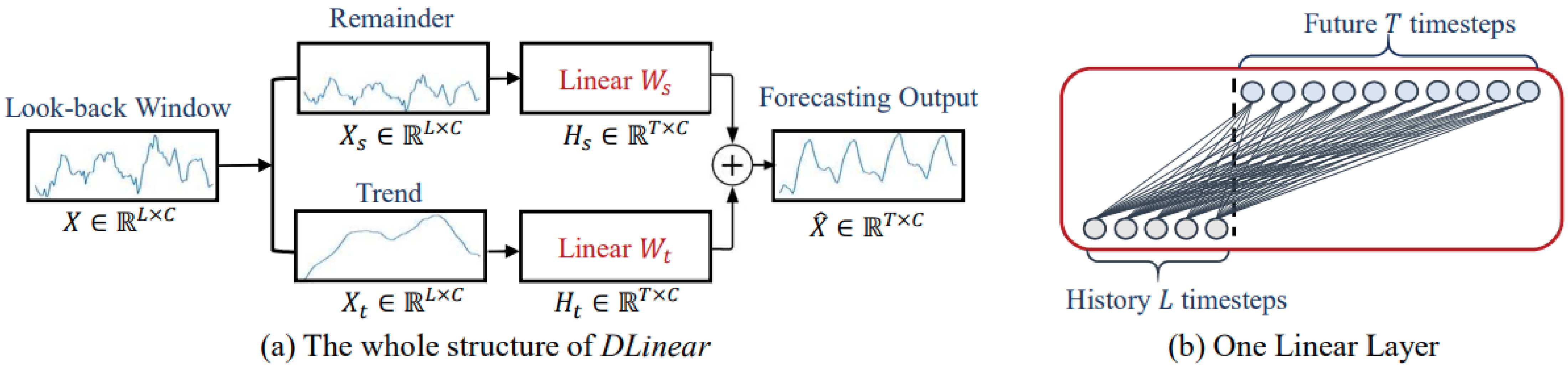


Figure 2: Illustration of the Decomposition Linear Model.

- Avoid using transformer without consideration!

Summary

- Time-series의 이해 및 기초 이론
  - 정의 및 특징
  - 시계열 데이터 구성 및 분해

- Time-series 예측 모델
  - 전통적 모델 (ARIMA)
  - 딥러닝 기반 모델

- ~~Time-series 관련 연구 소개~~

References

- https://otexts.com/fpp2/decomposition.html
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. In Advances in Neural Information Processing Systems (NeurIPS 2017) (pp. 5998–6008).
- Lim, B., Arık, S. Ö., Loeff, N., & Pfister, T. (2021). Temporal fusion transformers for interpretable multi-horizon time series forecasting. International Journal of Forecasting, 37(4), 1748–1764. https://doi.org/10.1016/j.ijforecast.2021.03.012
- Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., & Zhang, W. (2021). Informer: Beyond efficient transformer for long sequence time-series forecasting. In Proceedings of the AAAI Conference on Artificial Intelligence (AAAI 2021) (Vol. 35, No. 12, pp. 11106–11115).
- Wu, H., Xu, J., Wang, J., & Long, M. (2021). Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. In Advances in Neural Information Processing Systems (NeurIPS 2021), 34, 22419–22430.
- Zhou, T., Ma, Z., Wen, Q., Wang, X., Sun, L., & Jin, R. (2022). FEDformer: Frequency enhanced decomposed transformer for long-term series forecasting. In Proceedings of the International Conference on Machine Learning (ICML 2022) (pp. 27268–27286).
- Nie, Y., Han, Z., Cai, L., Song, X., & Sun, L. (2023). A time series is worth 64 words: Long-term forecasting with transformers. In International Conference on Learning Representations (ICLR 2023).
- Zeng, A., Chen, M., Zhang, L., Xu, Q., Wu, Q., Zhang, Z., Xu, Y., & Tian, Y. (2023). Are transformers effective for time series forecasting? In Proceedings of the AAAI Conference on Artificial Intelligence (AAAI 2023) (Vol. 37, No. 9, pp. 11121–11128).