ECE30007 Intro to AI Project

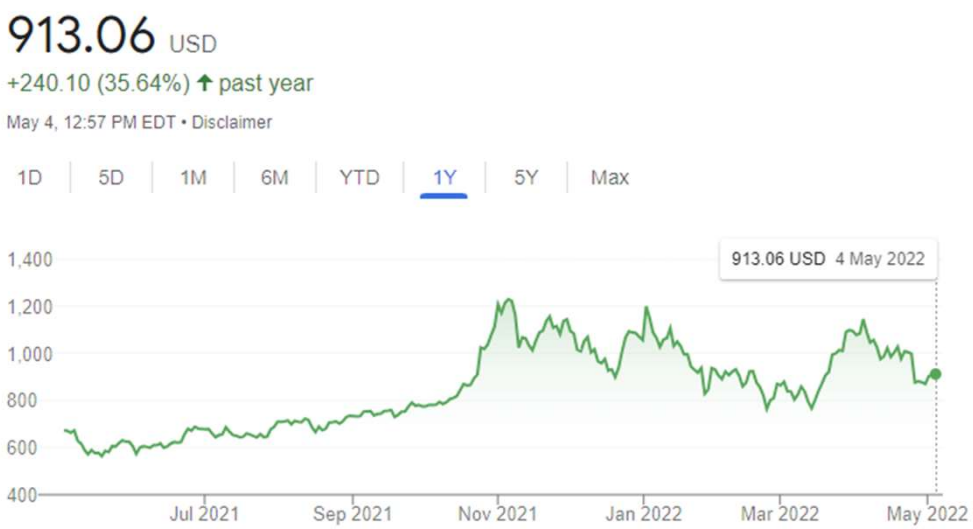
Sequence Learning (RNN, LSTM, and Time Series)

이원형 교수
Dr. Lee, WonHyong
whlee@handong.edu

TIME-SERIES

Time-Series

- Stock Price



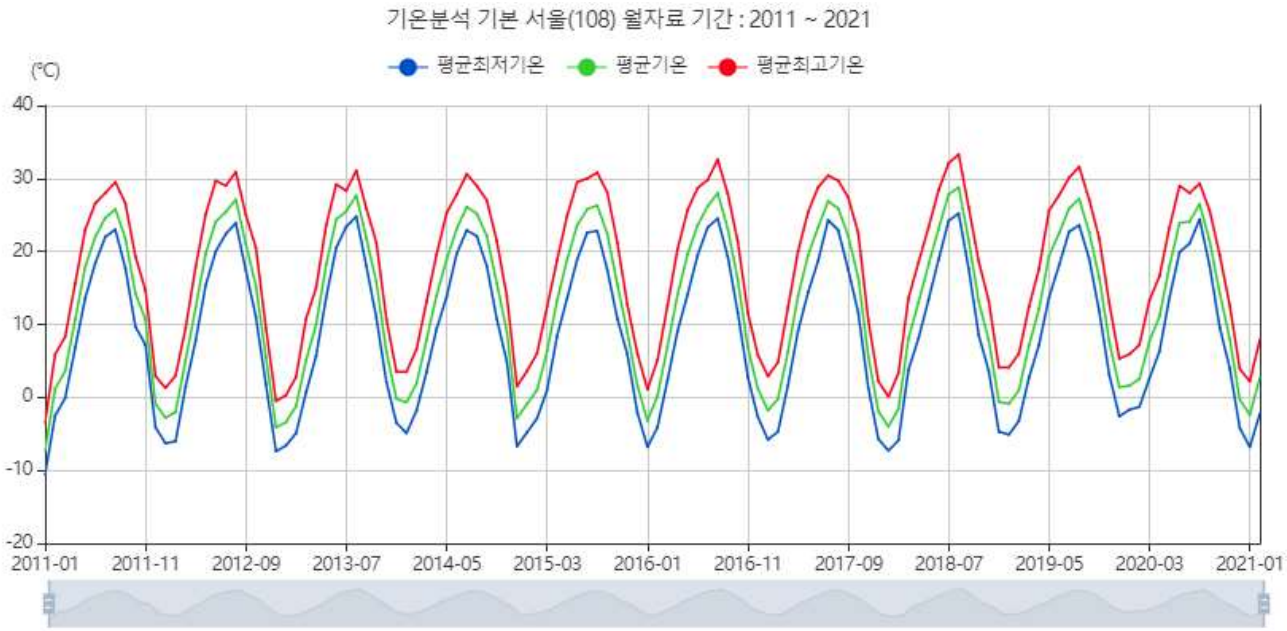
Time-Series

- Cryptocurrency Price

https://joind.io/market/id/1789

Time-Series

- Temperature



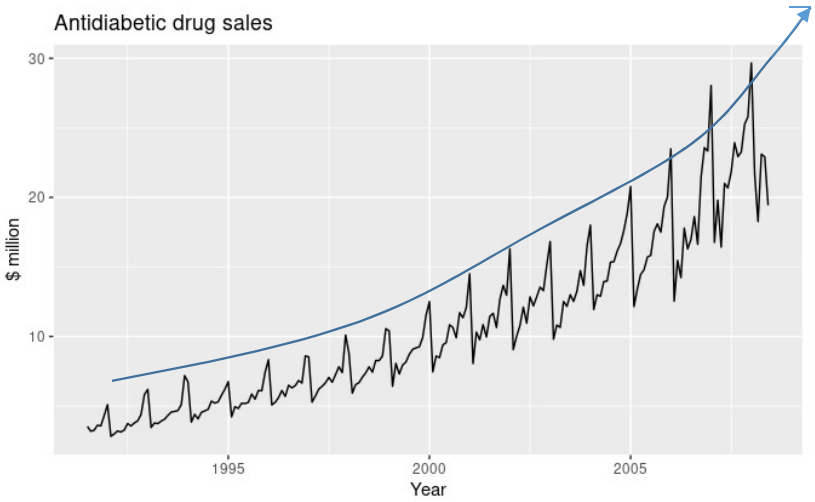
https://data.kma.go.kr/stcs/grnd/grndTaList.do?pgmNo=70

Time-Series

- **Time-Series**
  - Time series is defined as a set of quantitative observation arranged in chronological order(discrete variable).
  - In order words, time series is a series of data points indexed in time order. It is a sequence of discrete-time data with same intervals.
  - A time series graph plots observed values on the y-axis against an increment of time on the x-axis.
- Components
  - **Trend**
  - **Seasonal**
  - **Cyclic**
  - **Irregular**

Time-Series

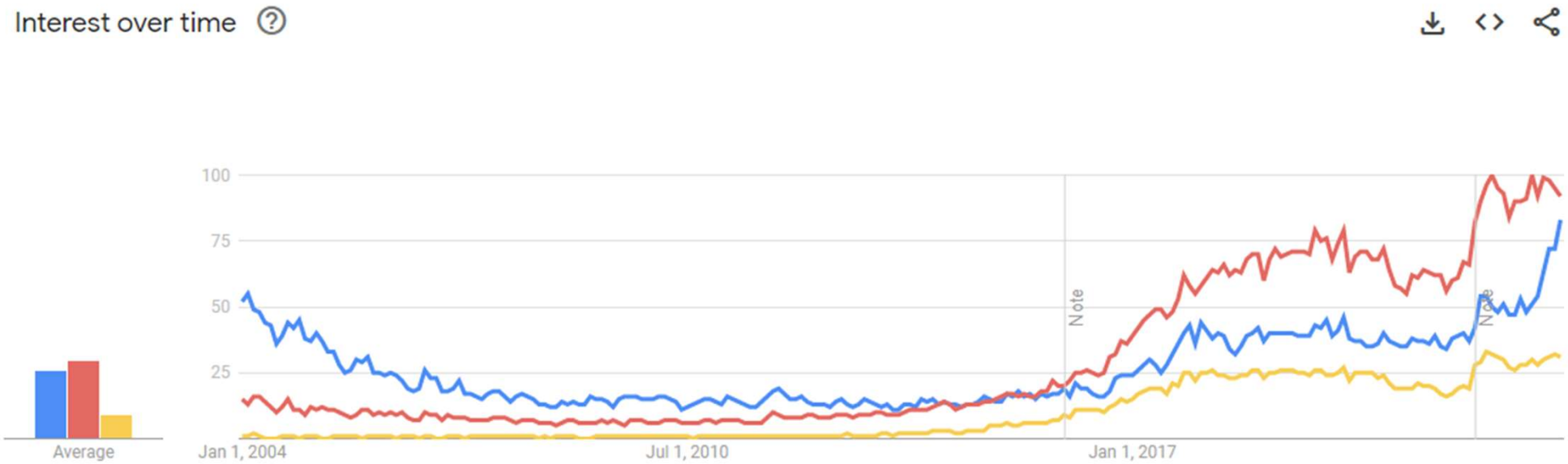
- Components
  - **Trend** : Long-term change in the mean level. General tendency of data to increase or decrease or stagnate over a long period of time.



Here, there is a clear and increasing trend. There is also a strong seasonal pattern that increases in size as the level of the series increases.

Time-Series

- google trends
  - “artificial intelligence”, “machine learning”, “deep learning”



Time-Series

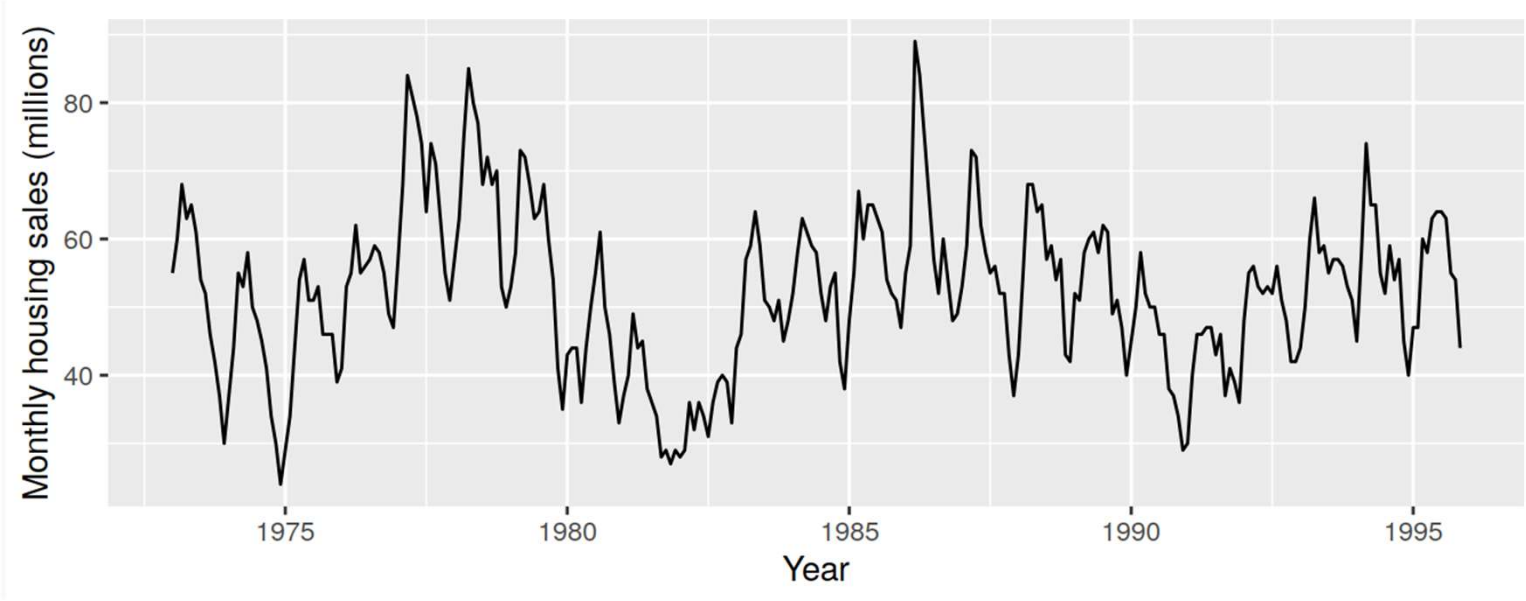
- Components
  - **Seasonal** : A *seasonal* pattern occurs when a time series is affected by seasonal factors such as the time of the year or the day of the week.
    - changes that take place due to the rhythmic forces which operate in a regular and periodic manner.
    - These forces usually have the same or most similar pattern year after year.
    - Seasonality is always of a fixed and known frequency.
    - These variations may be due to seasons, weather conditions, habits, customs or traditions.

Time-Series

- Components
  - **Cyclic** : Apart from seasonal effects, some time-series exhibit variation at not a fixed period due to some other physical cause. Or some time-series exhibit oscillations, which do not have a fixed period but which are predictable to some extent.
    - These fluctuations are usually due to economic conditions, and are often related to the "business cycle."
      - The duration of these fluctuations is usually at least 2 years.
    - In general, the average length of cycles is longer than the length of a seasonal pattern, and the magnitude of cycles tends to be more variable than the magnitude of seasonal patterns.

Time-Series

- seasonal pattern vs. cyclic pattern



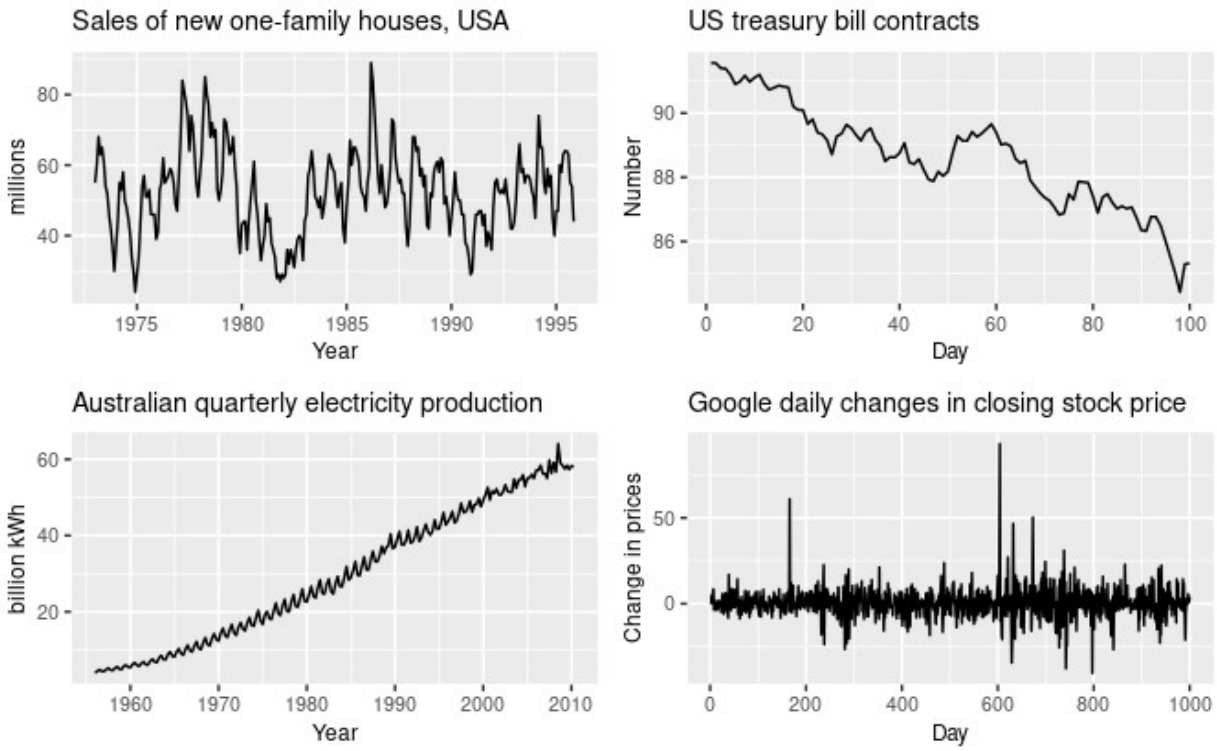
The graph shows strong seasonality within each year,
and some strong cyclic behaviour with period about 6 – 10 years.

Time-Series

- Components
  - **Irregular** : after trend and cyclic variations have been removed from a data, then it's left with residuals that may or may not be random.
    - Or random variations are fluctuations which are a result of unforeseen and unpredictable forces.
    - These forces operate in an absolutely random or erratic manner and do not have any definite pattern.
    - Thus, these variations may be due to floods, famines, earthquakes, strikes, etc.

Time-Series

- Q1) Think what kinds of patterns each plot has.



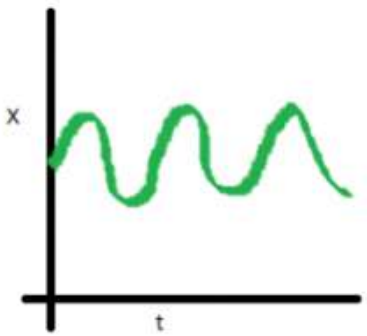
Time-Series

- As you can imagine, it is very difficult to forecast time-series data. We cannot predict with certainty what will occur in the future.
- Therefore, most **statistical forecasting** methods are based on the assumption that the time series can be rendered approximately stationary through the use of mathematical transformations.
- A stationarized series is relatively easy to predict.

Stationary vs Non-Stationary

Stationary Time-Series

- **Stationary Time Series**: data does not have any upward or downward trend or seasonal effects. Mean or variance are consistent over time
- A stationary time series is one whose properties do not depend on the time at which the series is observed. Thus, time series with trends, or with seasonality, are not stationary
  - the trend and seasonality will affect the value of the time series at different times.

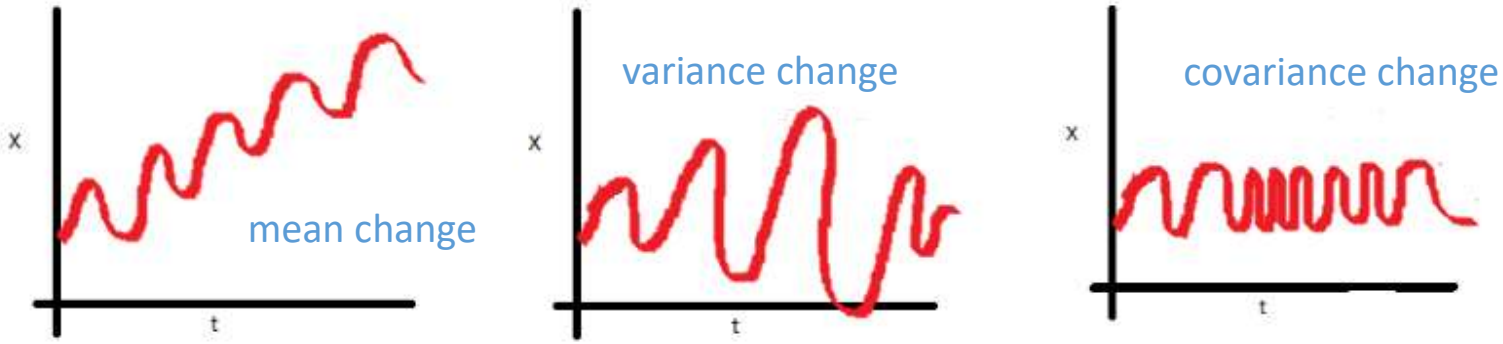


In this plot, variance and covariance are constant with time. This is what a stationary time series looks like.

Stationary Time-Series

- **Non-Stationary Time Series**: data show trends, seasonal effects, and other structures depend on time. Forecasting performance is dependent on the time of observation. Mean and variance change over time and a drift in the model is captured.
- A non-stationary time series can be converted to a stationary time series through a technique called differencing.

Stationary Time-Series



- In the first plot, we can clearly see that the mean varies (increases) with time which results in an upward trend. Thus, this is a non-stationary series. For a series to be classified as stationary, it should not exhibit a trend.
- In the second plot, we certainly do not see a trend in the series, but the variance of the series is a function of time. As mentioned previously, a stationary series must have a constant variance.
- In the third plot, the spread becomes closer as the time increases, which implies that the covariance is a function of time.

Differencing

numpy.diff(*a, n=1, axis=-1, prepend=<no value>, append=<no value>*)[source]¶
Calculate the n-th discrete difference along the given axis.

- Differencing is basically subtracting the previous value from the current value of your time series i.e.

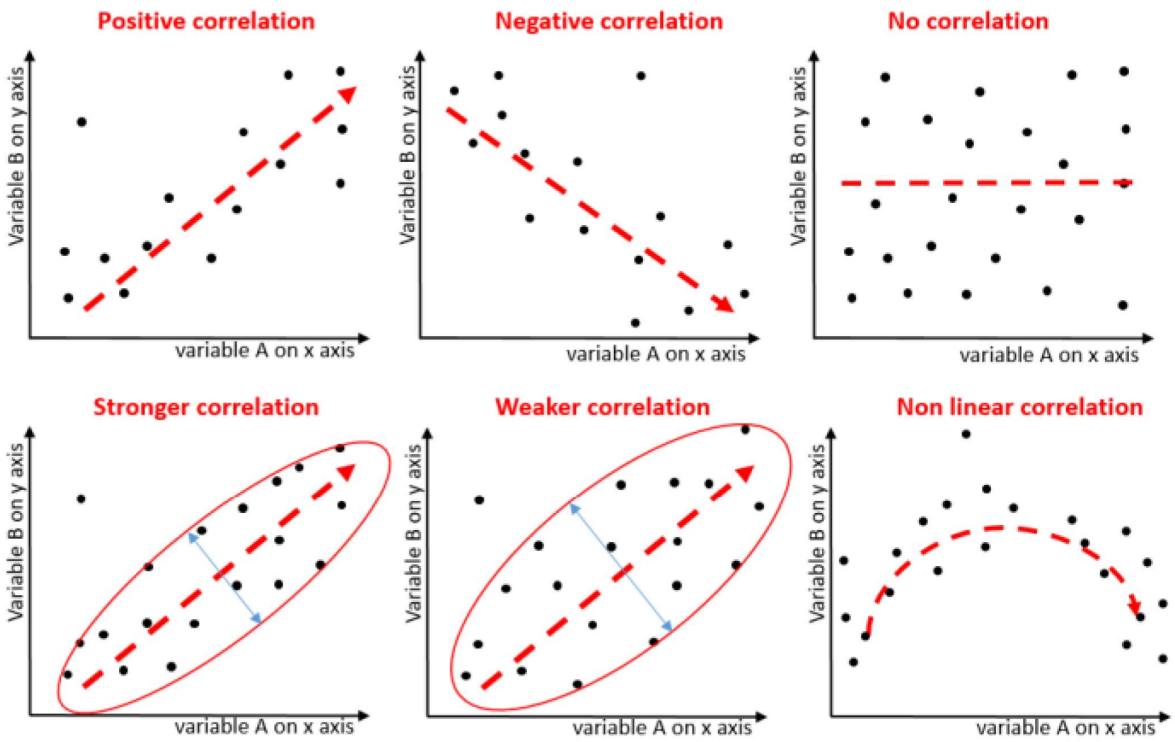
$$Y_t' = Y_t - Y_{t-1}$$

(first order difference)

- For n observations(data), the differenced series will only have n-1 observations as it is not possible to calculate a difference for the first element of a series.
- Differencing <u>stabilizes the mean</u> of the series. (Other methods such as log transform stabilizes the variance.)

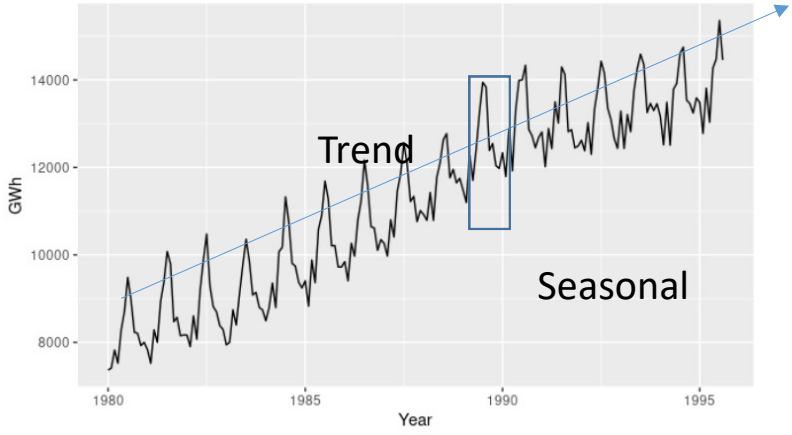
Covariance, Correlation

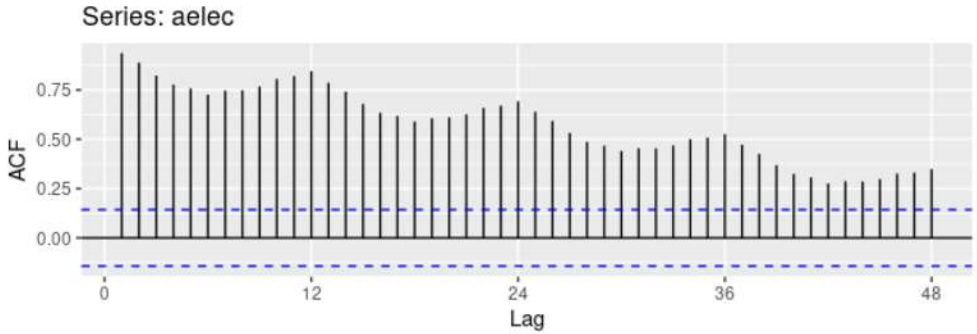
- 상관분석, 상관계수의 개요
  - https://youtu.be/RymrCV3K5J8



Autocorrelation function (ACF)

- 하나의 시계열 데이터에서 현재 값과 하나 이전 값과의 correlation을 계산하면 lag-one correlation
- 현재 값과 n번째 이전 값과 비교하여 correlation을 계산하고, 모든 n에 대해 표시한 것이 Autocorrelation function 의 역할



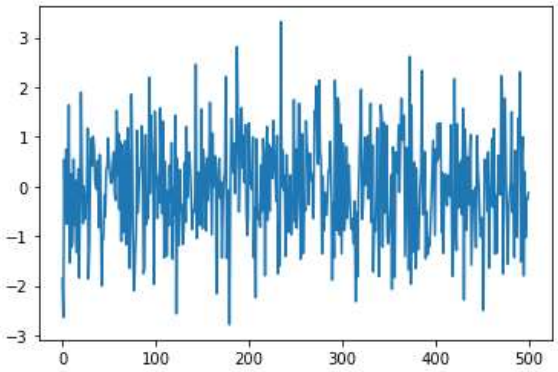



```
from statsmodels.graphics.tsaplots import plot_acf
```

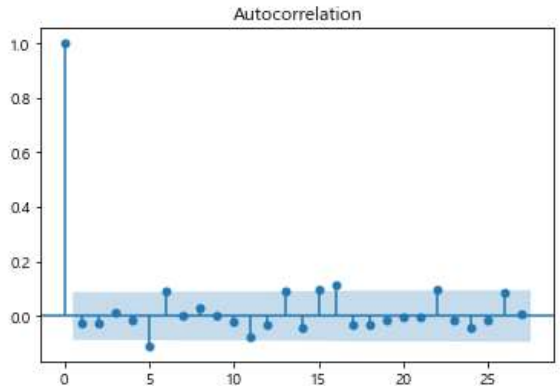
statsmodels.graphics.tsaplots.plot_acf(x, ax=*None*, lags=*None*, *, alpha=*0.05*, use_vlines=*True*, adjusted=*False*, fft=*False*, missing='*none*', title='*Autocorrelation*', zero=*True*, vlines_kwargs=*None*, **kwargs)

Autocorrelation function (ACF)

- Noise




```
noise=np.random.normal(loc= 0, scale = 1, size = 500)
plt.plot(noise)
```




```
plot_acf(noise)
```

Exercise with dataset

- 주택정보

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
df = pd.read_excel('data_set_train.xlsx')
```

df

	aptnm(아파트 이름)	yyyyqrt(거래년도 분기별)	price(가격)	con_year(건축년도)	dong(동)	area(면적)	floor(층수)	Latitude(위도)	Longtitude(경도)	gdp	...	dis_subway(지하철역과의 거리)	brand_r(유명 아파트 브랜드순)	n_home(세대수)
0	강남역우정에쉐르	2006Q1	9000	2004	역삼동	17.23	7	37.494204	127.043545	225613	...	849.353653	0	52
1	강남역우정에쉐르	2006Q1	9000	2004	역삼동	17.23	7	37.494204	127.043545	225613	...	849.353653	0	52
2	개포주공1단지	2006Q1	73000	1982	개포동	50.38	3	37.478407	127.061375	225613	...	1486.178329	0	5040
3	개포주공1단지	2006Q1	70000	1982	개포동	50.64	5	37.484609	127.067275	225613	...	1160.598717	0	5040
4	개포주공1단지	2006Q1	40000	1982	개포동	35.44	4	37.482445	127.051278	225613	...	650.325555	0	5040
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
17395	현대빌라트	2017Q3	179000	1998	청담동	169.63	8	37.526956	127.053126	446835	...	874.719438	0	14
17396	현대이스트빌	2017Q3	122500	1999	역삼동	244.14	4	37.496260	127.046404	446835	...	944.717800	0	12
17397	현대하이츠	2017Q3	64000	2004	역삼동	96.21	1	37.491379	127.034880	446835	...	812.764336	0	12
17398	현대한강	2017Q3	170000	1992	청담동	136.26	8	37.524675	127.056226	446835	...	717.729425	0	18
17399	현대한강	2017Q3	170000	1992	청담동	136.26	8	37.524675	127.056226	446835	...	717.729425	0	18

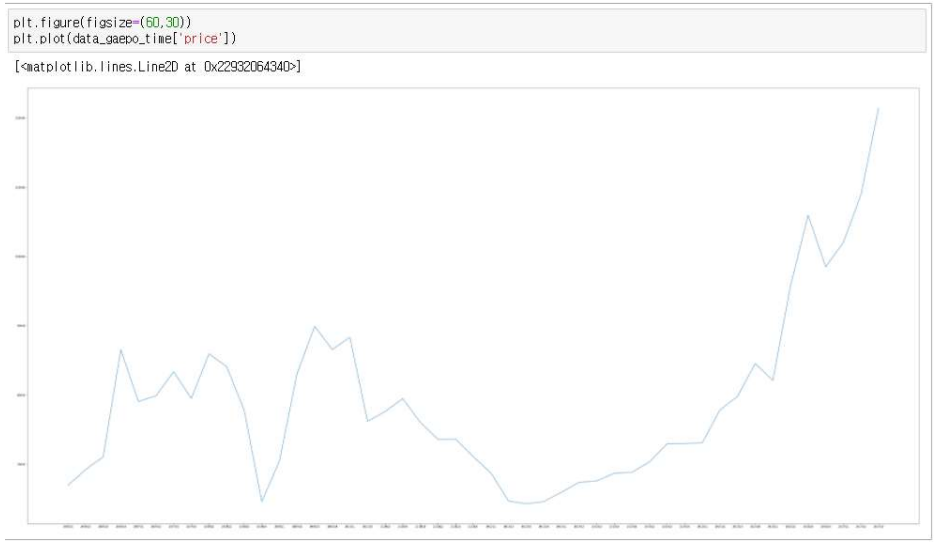
17400 rows × 29 columns

Exercise with dataset

- Try following commands
  - df.shape
  - df.columns
  - df.describe()
  - df.info()
  - df[df['Yongpae(용적률)'].isnull()].head()
    - 용적률 column에서 데이터가 없는 것을 찾아줌
  - df['Yongpae(용적률)']=df['Yongpae(용적률)'].replace(np.nan, df['Yongpae(용적률)'].mean()
    - **Q2) 이 코드가 뜻하는 바는 무엇일까?**
  - df['dong(동)'].value_counts()
  - data1=df[['yyyyqrt(거래년도 분기별)','price(가격)','dong(동)']]
    data1.head()
    - 원하는 column만 가져와 data1에 저장
  - data2=data1.copy()
    data2.columns=['yearqrt','price','dong']
    - Column 이름 변경

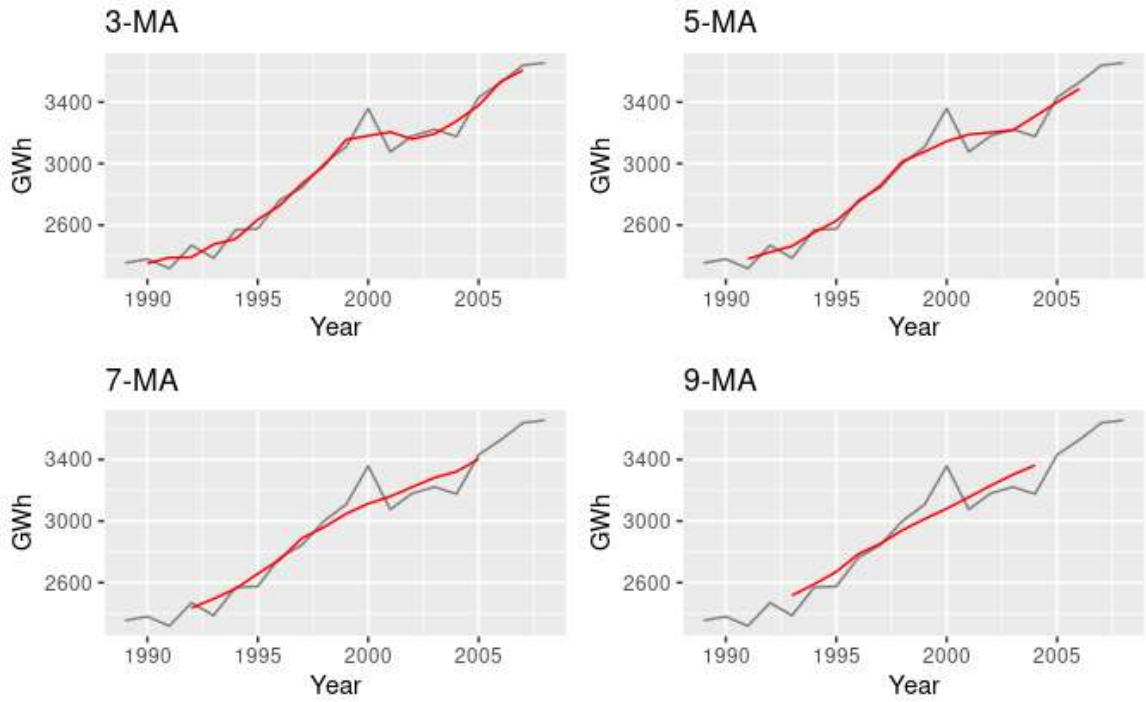
Exercise with dataset

- See how Gaepo-dong price changes over time
  - data_gaepo=data2[data2[‘dong’]==‘개포동’]
  - data_gaepo.head()
- See how Gaepo-dong price changes over time
  - data_gaepo_time=data_gaepo.groupby(‘yearqrt’).mean()
    data_gaepo_time
  - Q3) 위 코드의 결과는?
  - plt.figure(figsize=(60,30))
    plt.plot(data_gaepo_time[‘price’])



Basic Decomposition

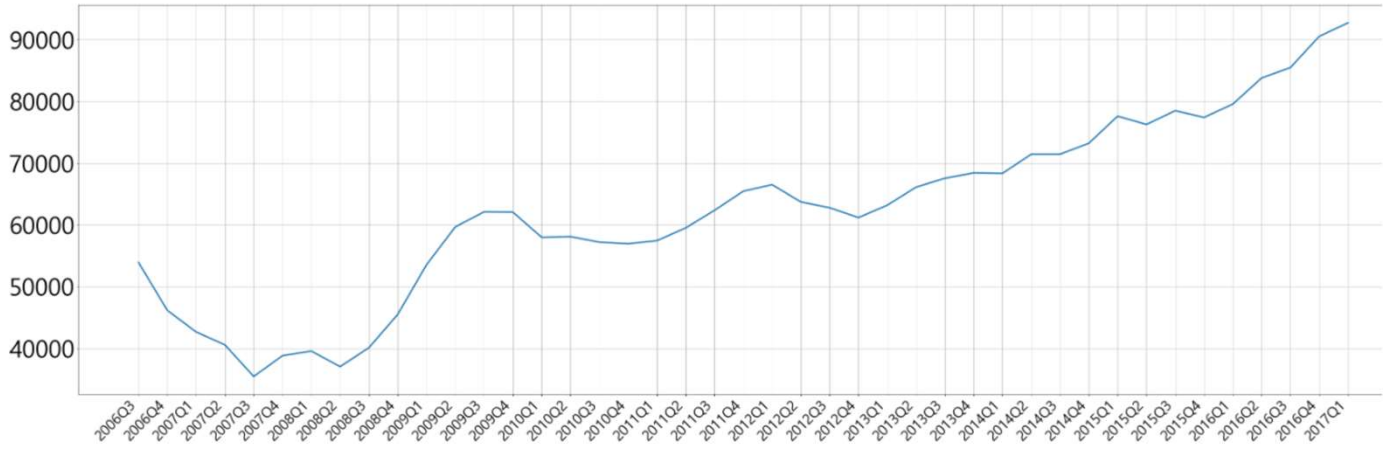
- Moving Average
  - Q4) 빨간 선(Average 결과)이 검정색 선보다 짧은 이유는?



Basic Decomposition

- Moving Average

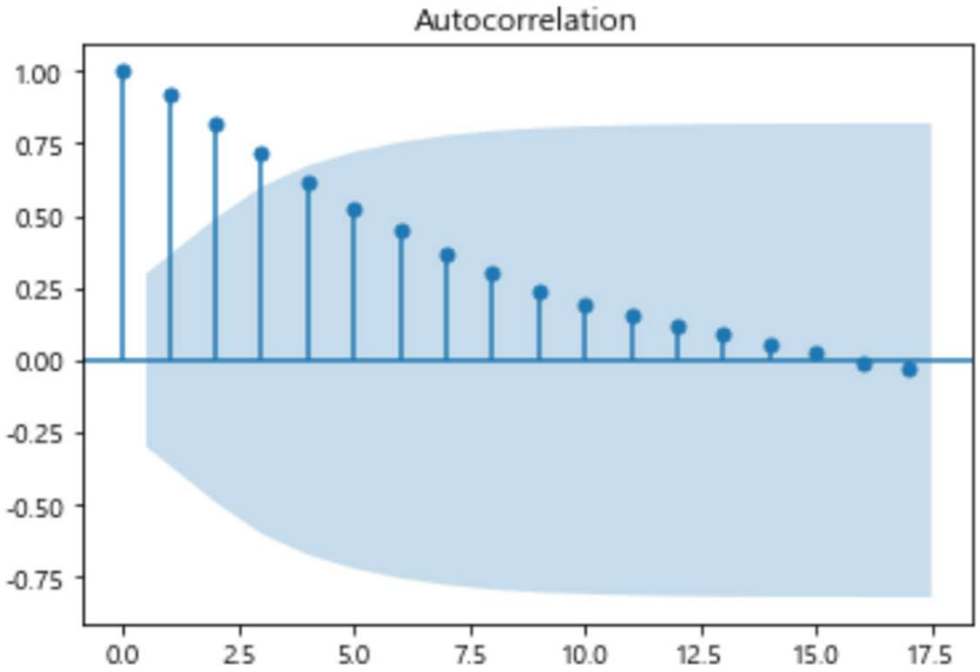
```
nMA = 2 # number of Pre(or post) samples to average

plt.figure(figsize=(50,15))
tmp=df[df['dong(동)']=="역삼동"]
tp=tmp.groupby('yyyyqrt(거래년도 분기별)').mean().copy()
ls=[]
for j in range(nMA,len(tp)-nMA):
    ls.append(tp['price(가격)'].iloc[j-nMA:j+nMA].mean())
df_mean5=pd.DataFrame({"Time" : tp.index.values[nMA:len(tp)-nMA], "Mean_Price" : ls })
plt.plot(df_mean5["Time"],df_mean5["Mean_Price"],linewidth=3.0)
plt.yticks(fontsize = 50); plt.xticks(fontsize = 30, rotation = 45, ha = 'right')
plt.grid(True)
df_mean5.set_index("Time", inplace=True)
```



Basic Decomposition

- check autocorrelation

```
plot_acf(df_mean5)
plt.show()
```

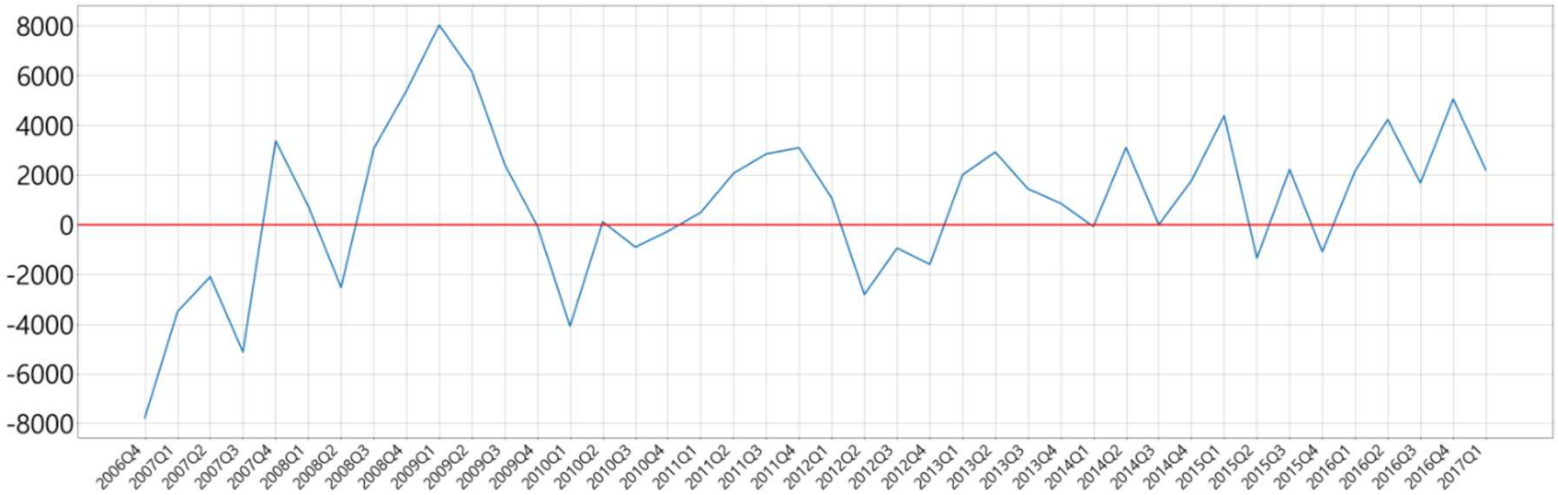


It seems there is visible pattern and non-stationary.

Basic Decomposition

- taking difference and autocorrelation

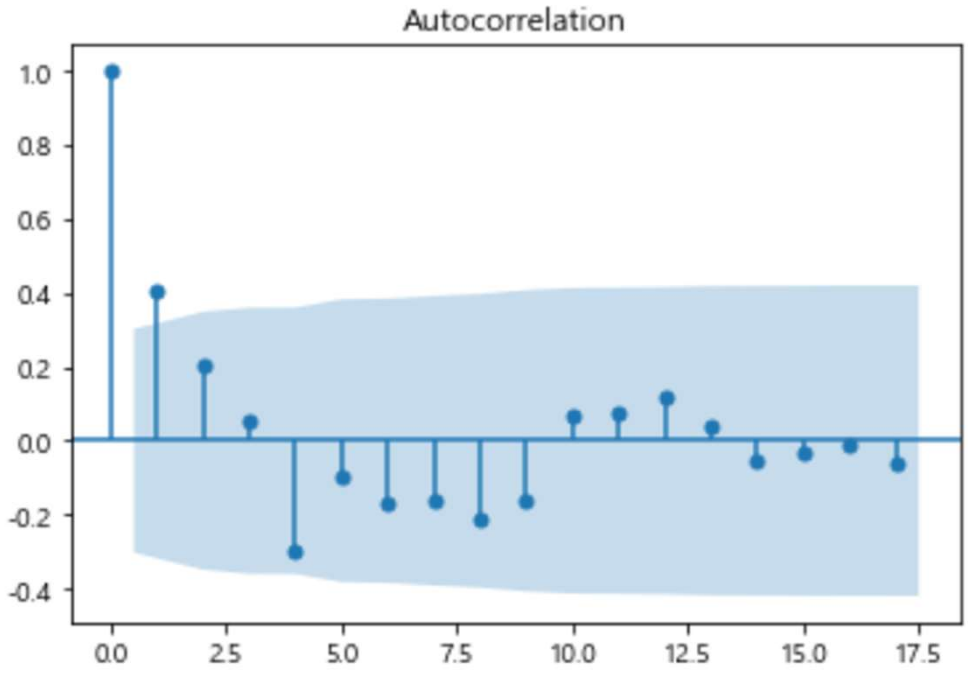
```
# Difference
plt.figure(figsize=(50,15))
diff_1=df_mean5.diff(periods=1).iloc[1:]
plt.yticks(fontsize = 50); plt.xticks(fontsize = 30, rotation = 45, ha = 'right')
plt.plot(diff_1,  linewidth=3.0)
plt.axhline(y=0, color='r', linewidth=3.0)
plt.grid(True)
```



Basic Decomposition

- taking difference and autocorrelation

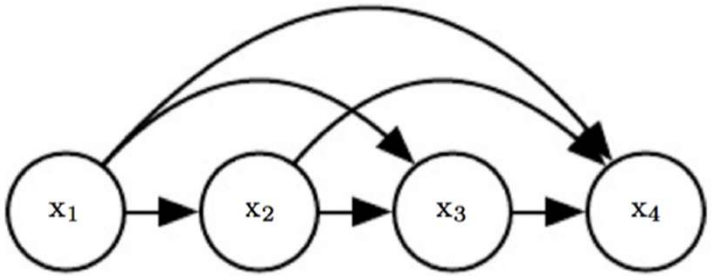
```
plot_acf(diff_1)
plt.show()
```



closer to stationary

Basic Decomposition

- 예측?
  - Autoregressive Model
    - AR(p) : An autoregressive process of order p
    - Present value of the series, X_t, by a function of p past values, X_{t-1}, X_{t-2}, …, X_{t-p}.
    - $X_t = c + \varphi_1 X_{t-1} + \varphi_2 X_{t-2} + \varphi_3 X_{t-3} + \cdots + \varphi_p X_{t-p} + Z_t$
      - *Where $\{Z_t\}$ is white noise, i.e., $\{Z_t\}$ ~ W N(0, σ^2),*
        and Z_t, is uncorrelated with X_s $for\ each\ s < t$.

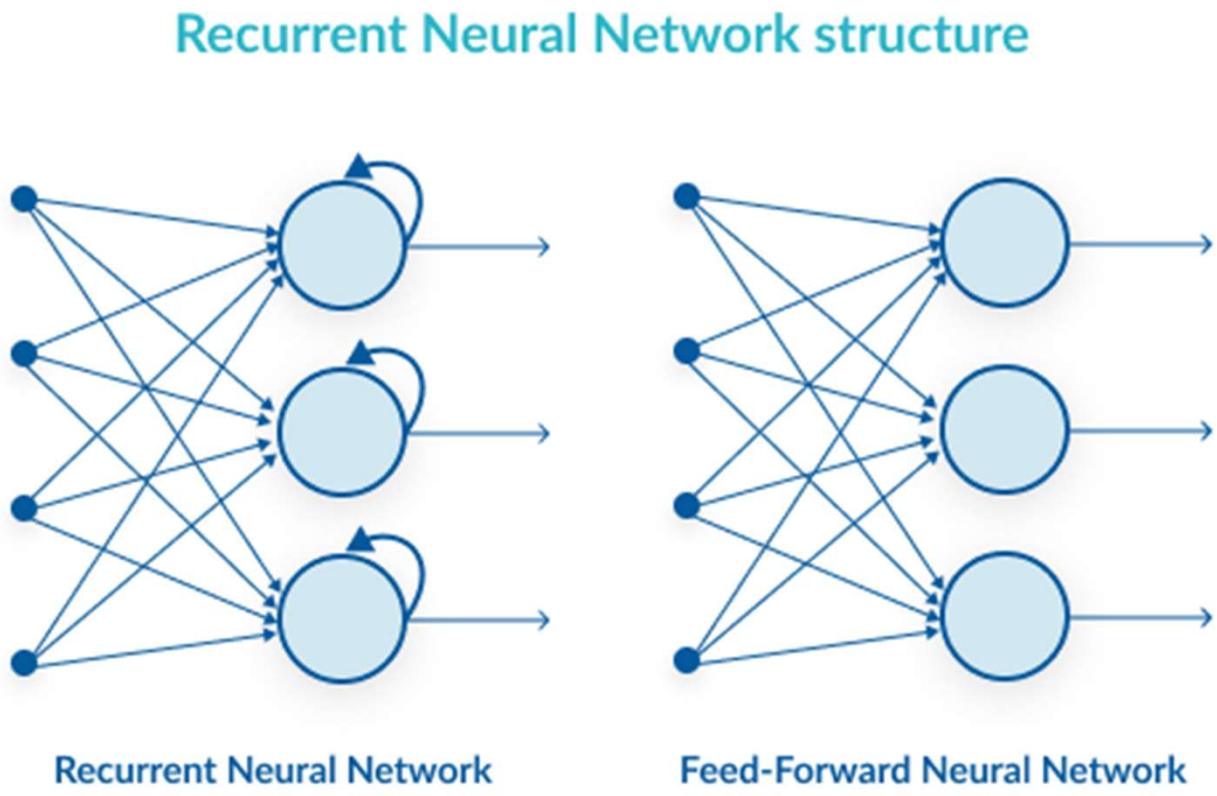


Basic Decomposition

- 예측?
  - Autoregressive Model
    - *AR(1) :* $X_t = c + \varphi_1 X_{t-1} + Z_t$
      - *For an AR(1) model:*
      - *when* φ_1 *=0,* X_t *is equivalent to white noise;*
      - *when* φ_1 *=1 and c=0,* X_t *is equivalent to a random walk;*
      - *when* φ_1 *=1 and c≠0,* X_t *is equivalent to a random walk with drift;*
      - *when* φ_1 *<0,* X_t *tends to oscillate around the mean.*

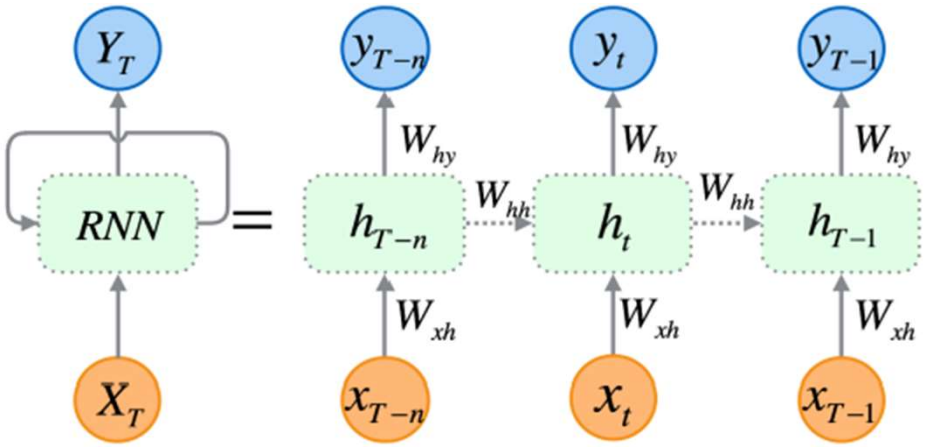
RNN

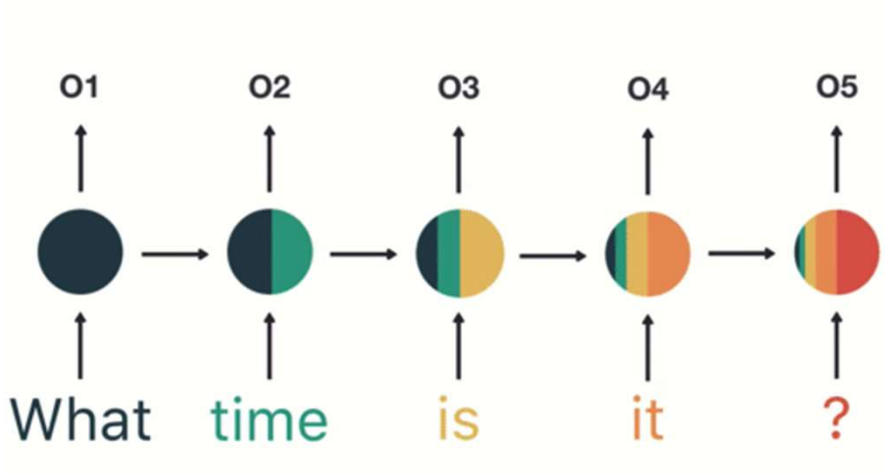
RNN (Recurrent Neural Network)



RNN (Recurrent Neural Network)

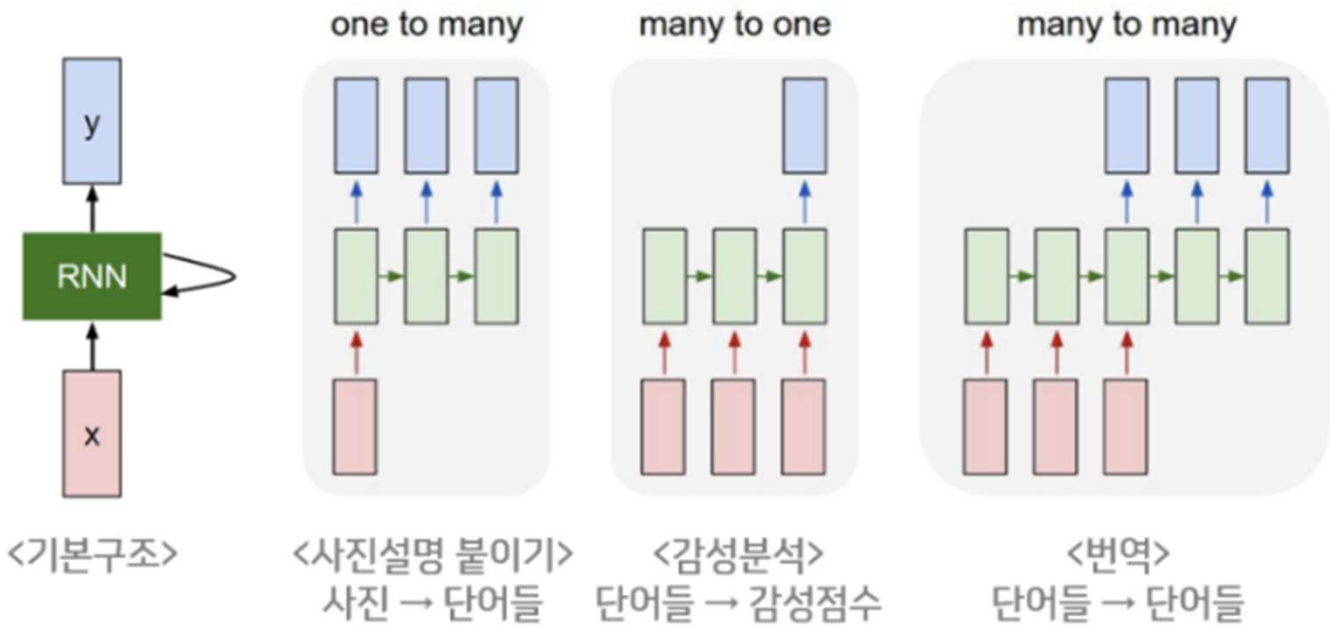
- Unfolding diagram





RNN (Recurrent Neural Network)

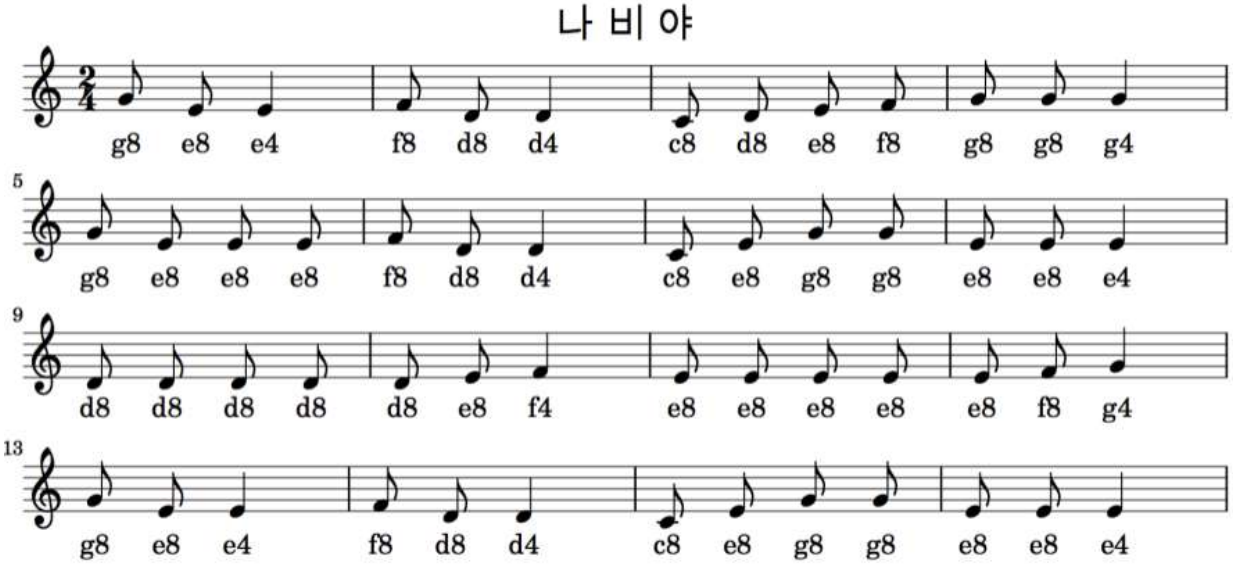
- 기본 구조를 변형한 다양한 유형



https://tykimos.github.io/2017/04/09/RNN_Layer_Talk/

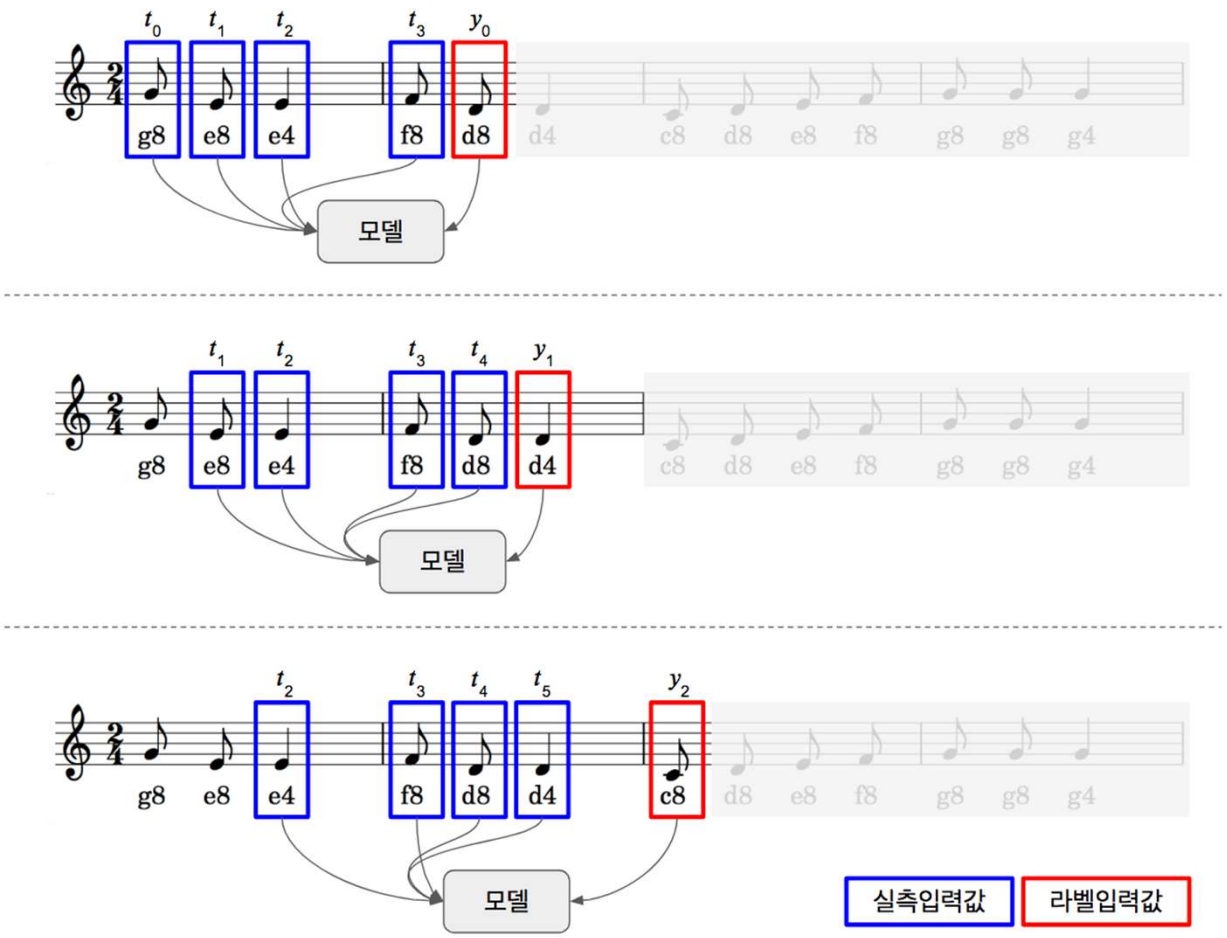
Sequence Data

- 나비야
  - c(도), d(레), e(미), f(파), g(솔), a(라), b(시)
  - 4(4분음표), 8(8분음표)



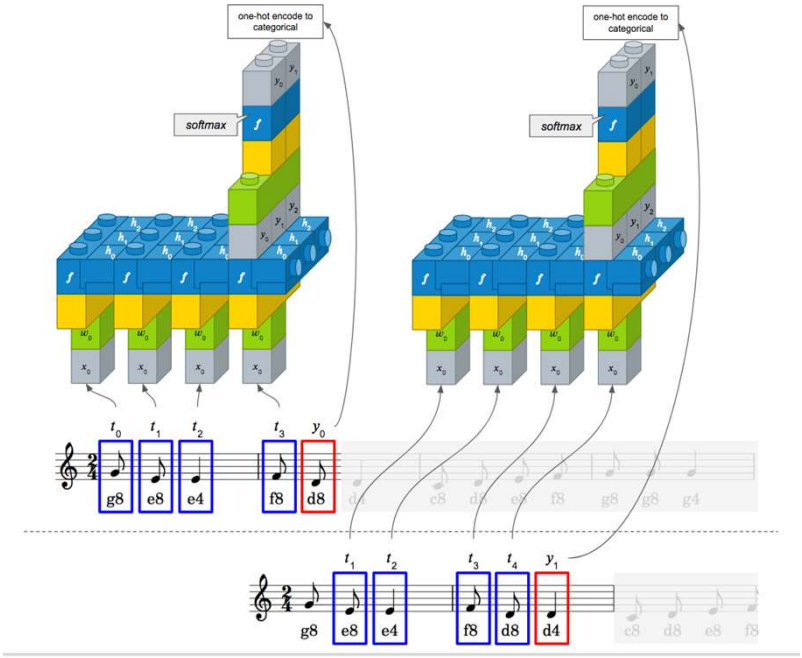
Sequence Data

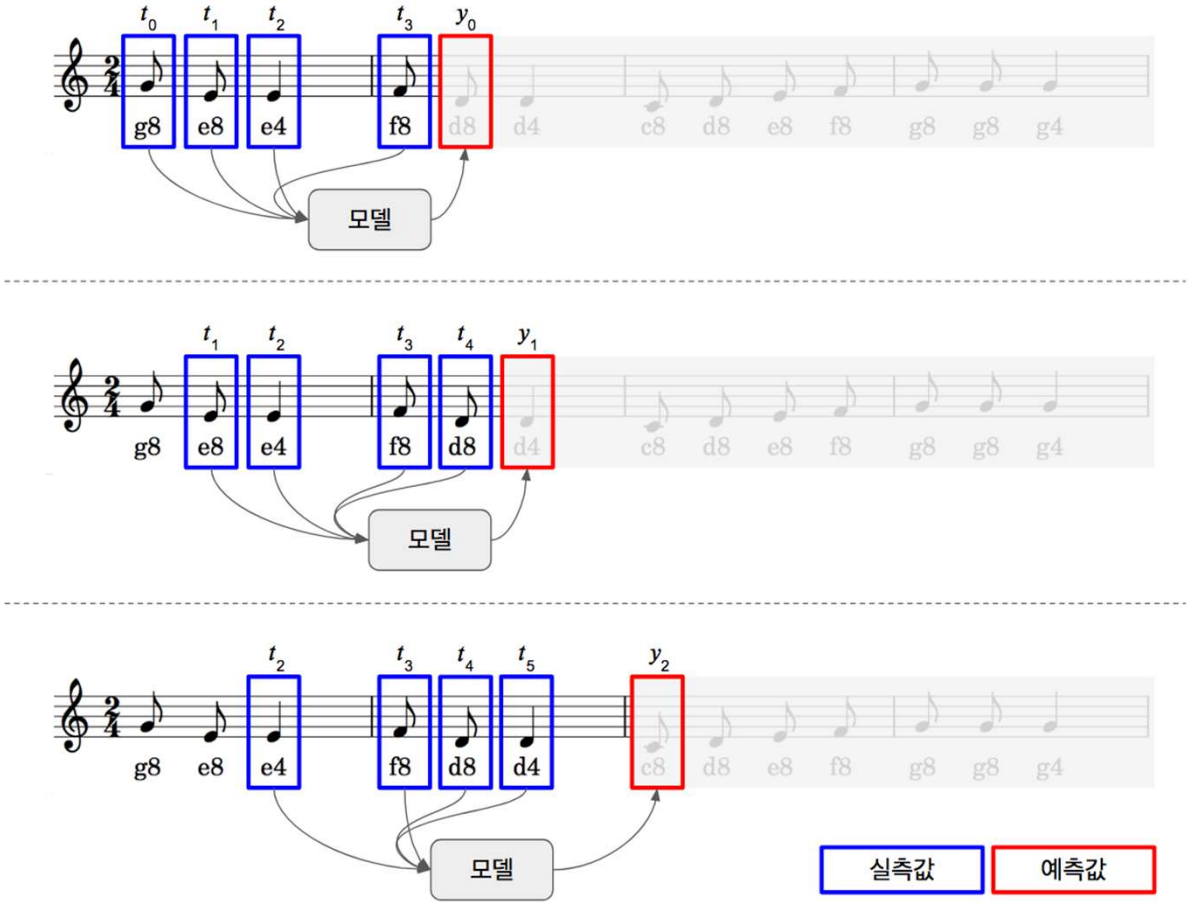
- 학습 데이터 준비



Sequence Data

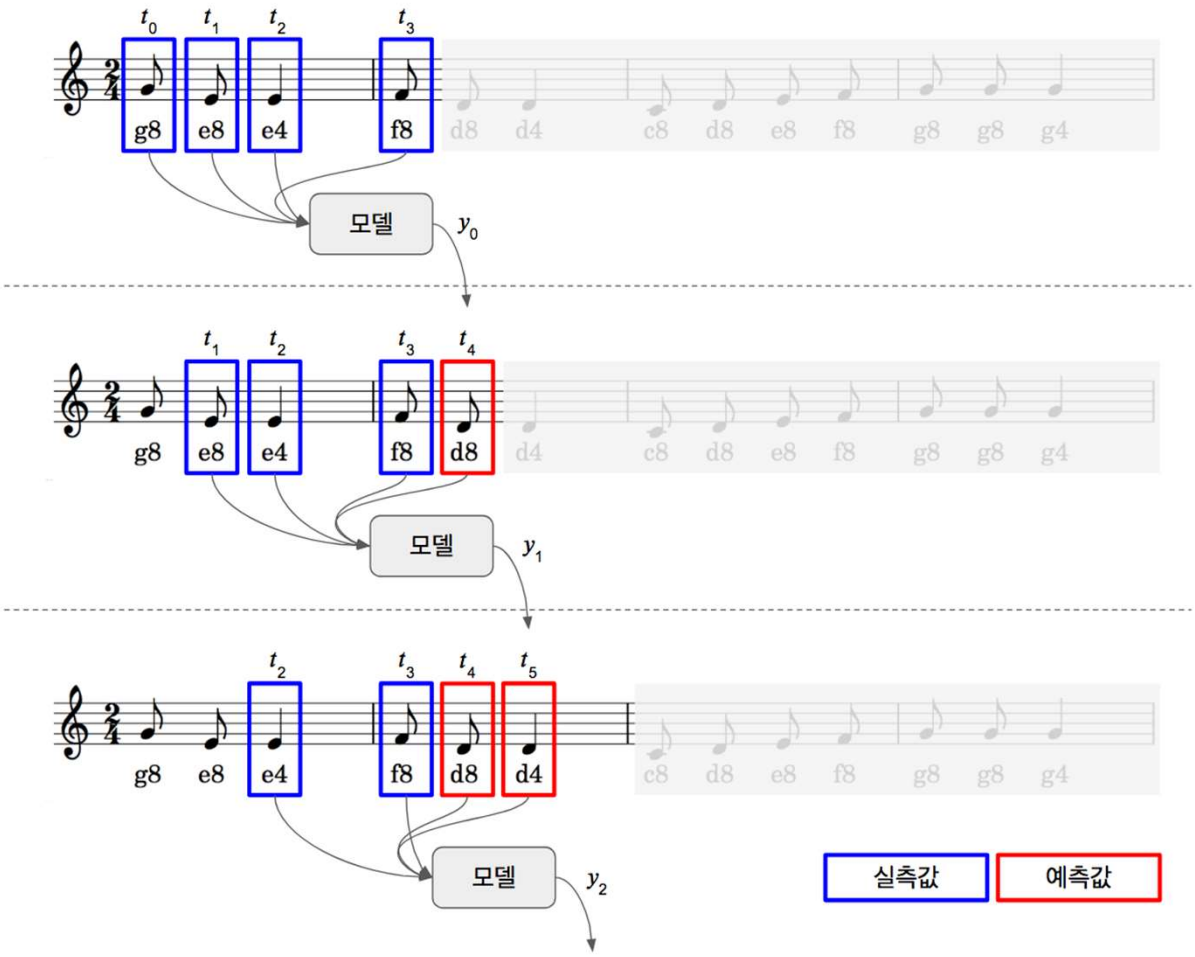
- 예측하는 과정
  - 각 스텝만 예측하는 경우





Sequence Data

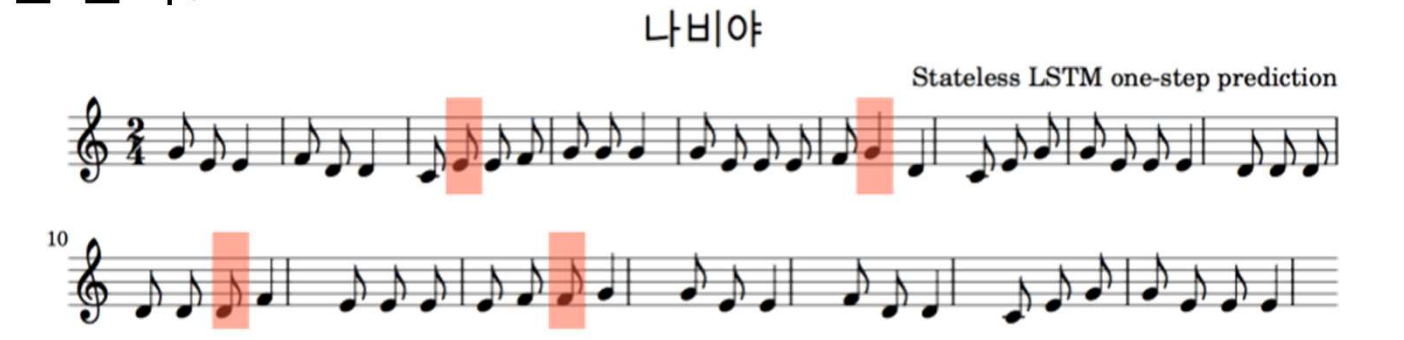
- 예측하는 과정
  - 첫 값만 주고
    곡 전체를 예측하는 경우

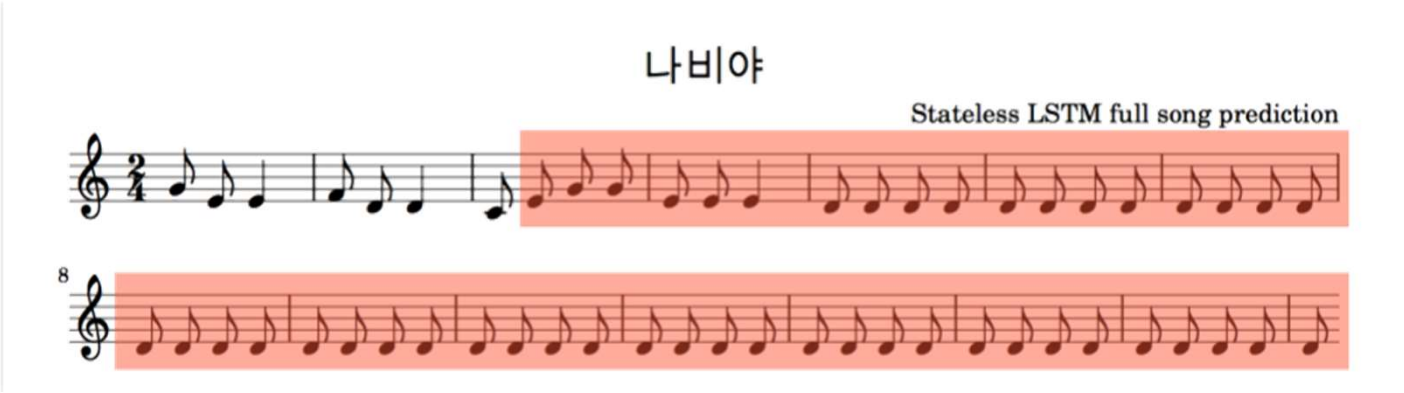


Sequence Data

코드는 뒤에서 소개

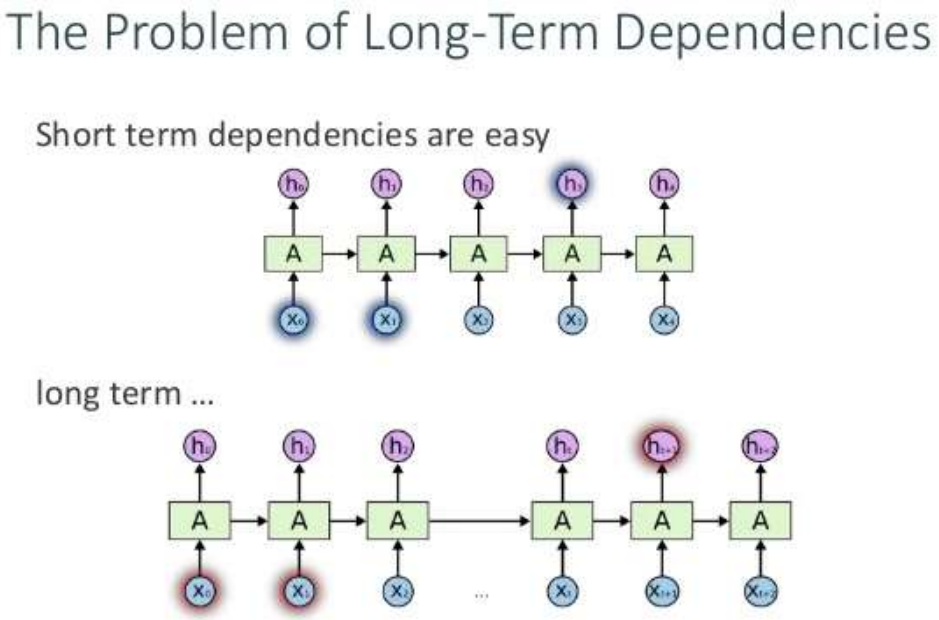
- 결과 (한 스텝 예측 결과와 곡 전체 예측 결과)
  - 왜 이런 결과?





RNN의 한계

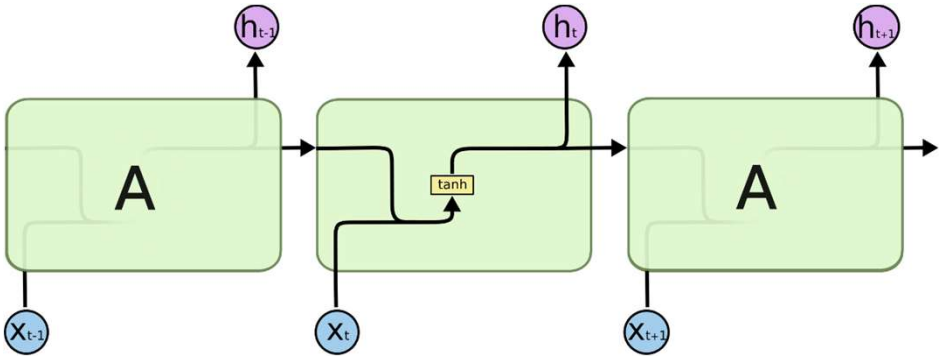
- 관련 정보와 그 정보를 사용하는 지점 사이 거리가 멀어지는 경우 학습 능력이 현저하게 저하됨

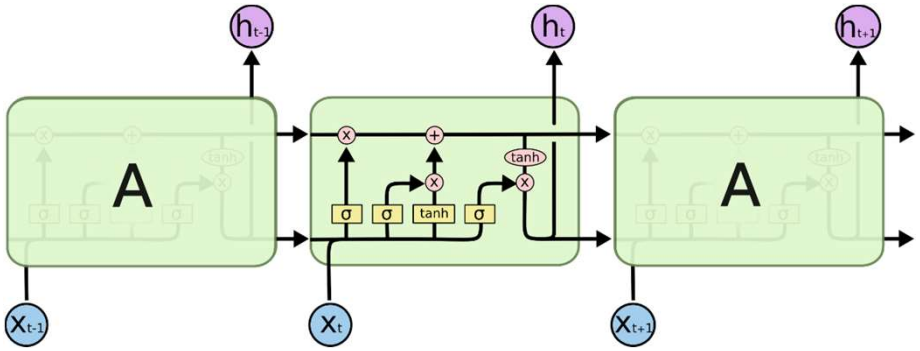


LSTM

LSTM(Long Short-Term Memory)

- 기본구조
  - RNN(좌), LSTM(우)

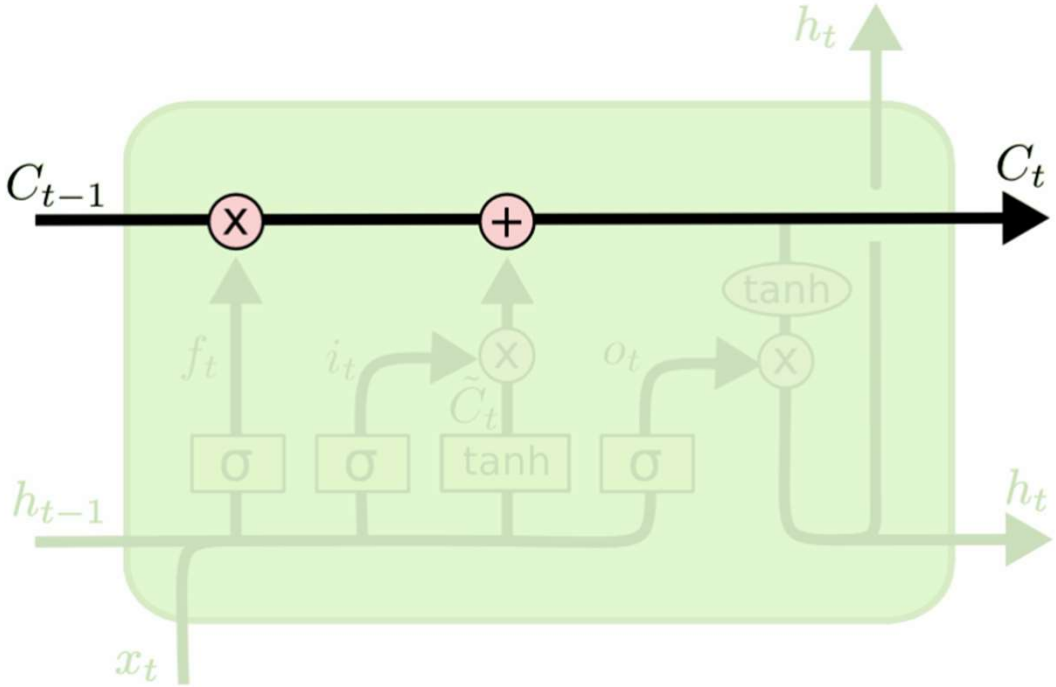


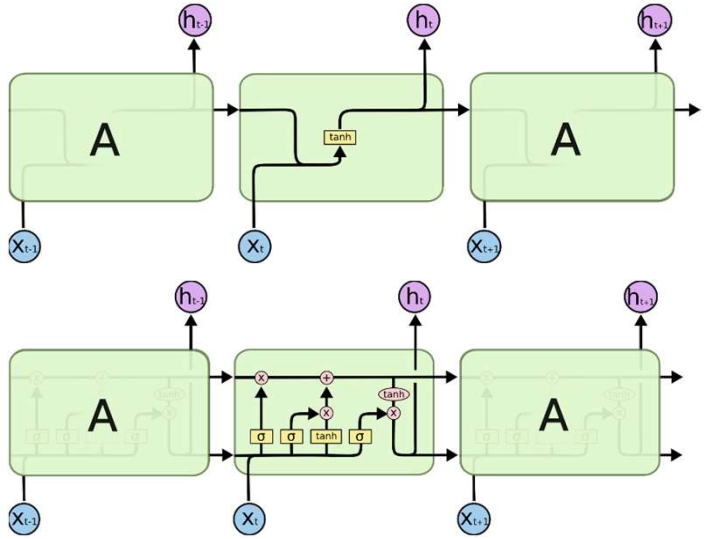


https://colah.github.io/posts/2015-08-Understanding-LSTMs/
https://dgkim5360.tistory.com/entry/understanding-long-short-term-memory-lstm-kr

LSTM(Long Short-Term Memory)

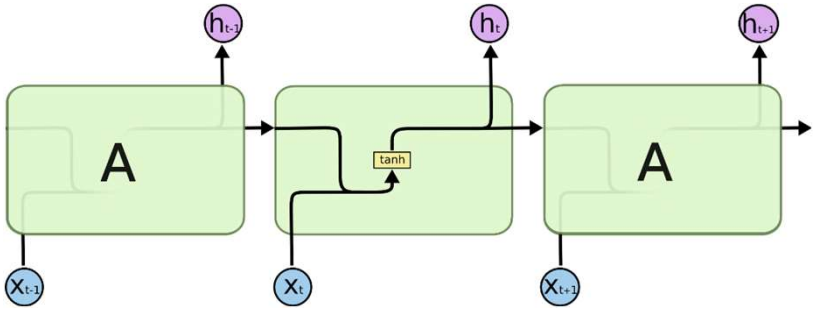
- 핵심 아이디어
  - cell state, 장기 기억을 유지

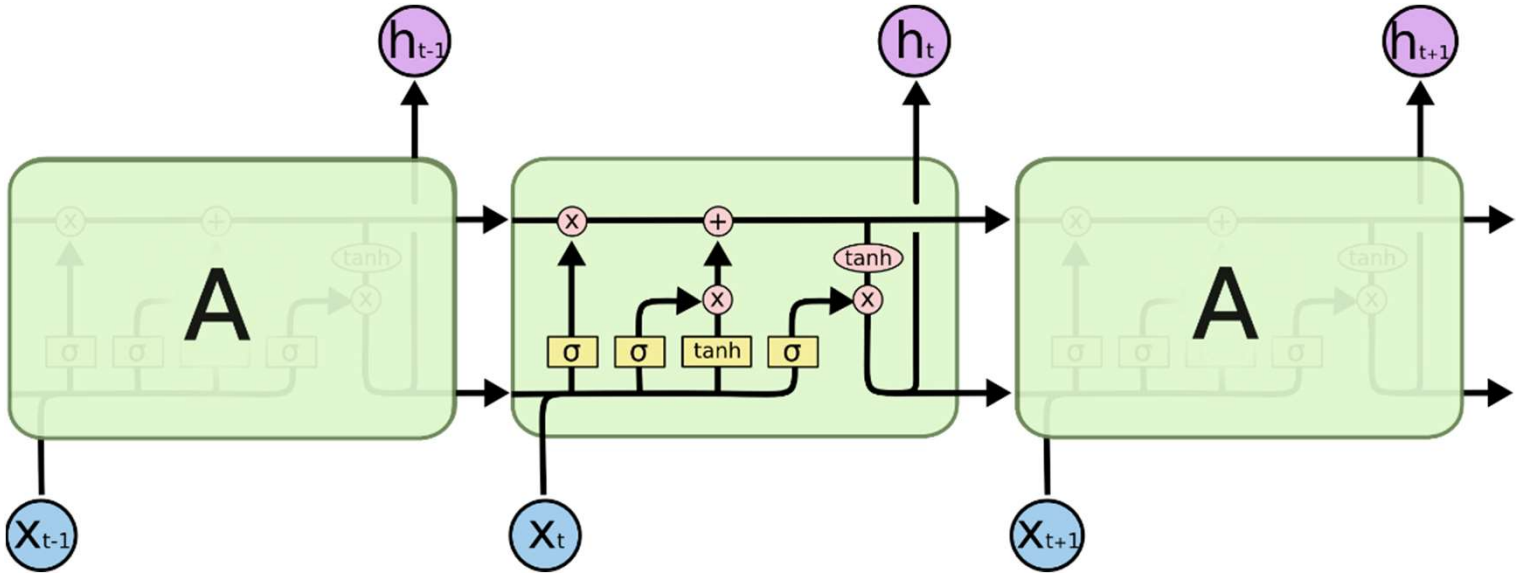




LSTM(Long Short-Term Memory)

- 핵심 아이디어
  - cell state, 장기 기억을 유지

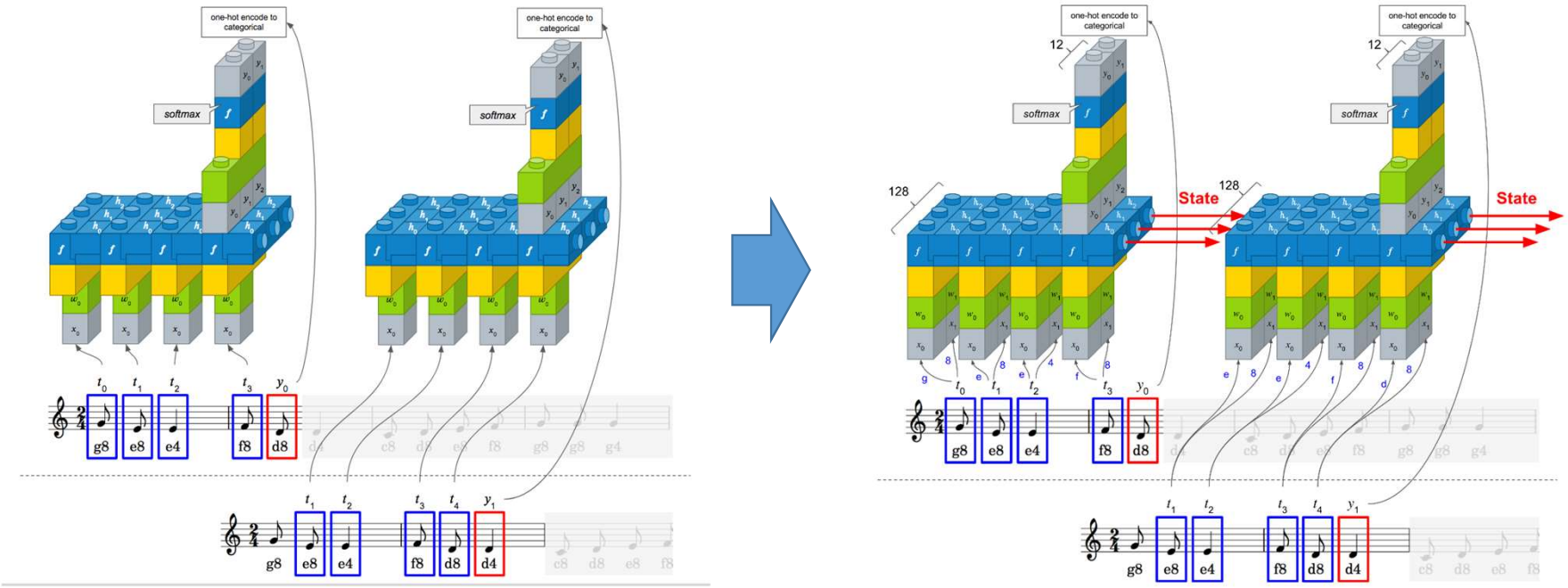




https://colah.github.io/posts/2015-08-Understanding-LSTMs/
https://dgkim5360.tistory.com/entry/understanding-long-short-term-memory-lstm-kr

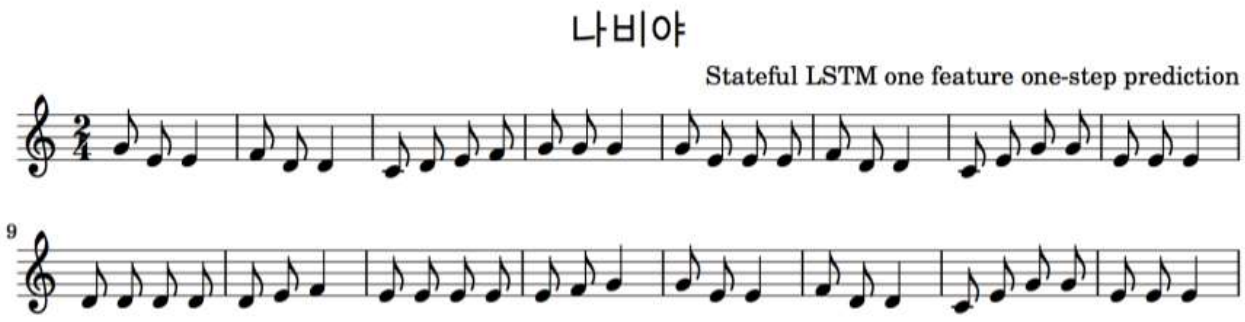
LSTM(Long Short-Term Memory)

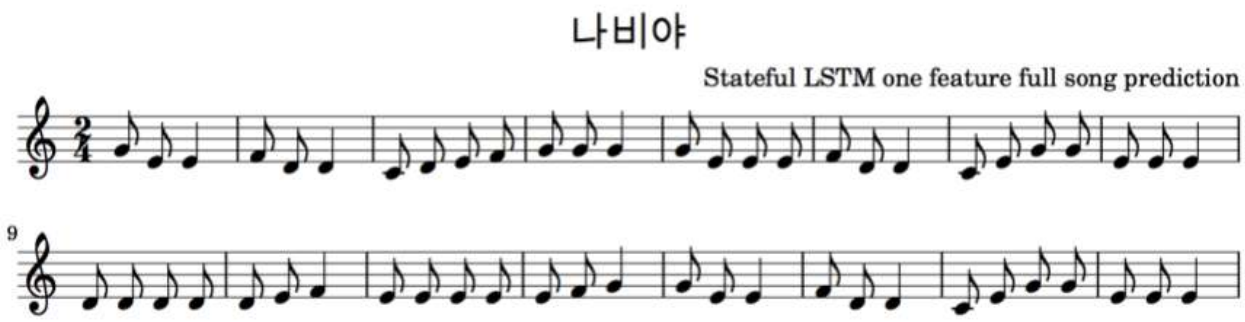
- RNN(좌), LSTM(우)
  - 상태유지를 하느냐 안하느냐의 차이 (코드 상에서도 "상태유지 모드" 차이)



LSTM(Long Short-Term Memory)

- 결과 (한 스텝 예측 결과와 곡 전체 예측 결과)





LSTM(Long Short-Term Memory)

- 코드
  - RNN.ipynb
  - LSTM.ipynb
  - 자세한 설명: https://tykimos.github.io/2017/04/09/RNN_Layer_Talk/

LSTM(Long Short-Term Memory)

ASSIGNMENT

Assignment (HW7)

- 보고서 작성
  - Q1) 각 그래프에 대한 답변
  - Q2)에 대한 답변
  - Q3) 결과 이미지 캡처
  - Q4) 답변
  - Q5) 분기별 역삼동 아파트 가격을 LSTM을 이용하여 예측해보라.
    - 한 스텝씩 예측 vs 기간 전체 예측 비교
    - 입력 범위는 다양하게 테스트 해보고 가장 좋은 결과 사용

  - Deadline: 5/4(목) 밤 11시 59분

예고편..(?)

- Text도 Sequence 정보
  - 간단한 챗봇 구현(좌), 성경말투 번역(우)

```
output = predict('빨리 퇴근하고 싶다.')
```

```
Input: 빨리 퇴근하고 싶다.
Output: 몸과 마음이 좀 쉬어야 할 것 같아요 .
```

```
output = predict('저녁에는 닭고기 요리를 먹을 거야.')
```

```
Input: 저녁에는 닭고기 요리를 먹을 거야.
Output: 피할 수 있으면 피하세요 .
```

```
output = predict('너 여자친구 있어?')
```

```
Input: 너 여자친구 있어?
Output: 그 사람으로 잊혀지기도 하죠 .
```

```
output = predict('오늘 날씨는 어때?')
```

```
Input: 오늘 날씨는 어때?
Output: 친구들과 같이 놀러가세요 .
```

```
prediction('오늘은 현장에서 예배를 드렸다.')
```

```
Bible: 오늘 이것 은 온 날 의 예배 하 는 절기 의 삼 분 의 일 이 니라
```

```
prediction('성경에 있을 법한 말을 해야 괜찮은 결과가 나온다.')
```

```
Bible: 성경 을 따르 아 쓰 ㄴ 것 을 너희 가 정하 ㄴ 것 으로 말미암 아 너희 가 평생 에
기록 하 ㄹ지니라
```

```
prediction('그래도 연습용으로는 괜찮다.')
```

```
Bible: 그 는 남종 으로 는 흠 이 없 고
```

```
prediction('이게 맞나.')
```

```
Bible: 이것 들 은 맞 았 으나
```

```
prediction('재미로 돌려보는 정도.')
```

```
Bible: 그 가 부르짖 는 것 을 보 고 돌 아 다니 는 것 을 바라보 았 으니 그 는 정의 로
말미암 아 도움 이 되 는 남 의 인자 하 ㅁ 이 니라
```

IFU

Thank you!