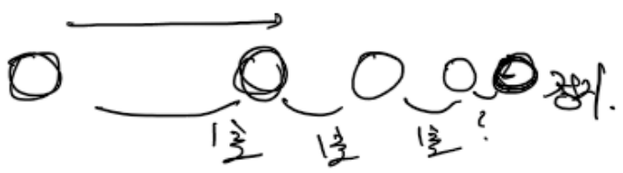
0. 전통적인 TimeSeries Data Prediction

도입

- 지금까지 우리가다루었던 문제들은 정지된 상태에서 정답을 예측하는 문제들이었다.
  - {데이터 : 정답} 형태를 학습시키고 문제 → 해결 을 통해 정답을 도출하는 과정
- 정지된 상태가 아닌 시간에 영향을 받아 흐르고 있는 데이터를 예측하기 위해선 어떻게 해야할까?



  -
  - 조원희의 슈팅을 막아보자.
    - 이영표도 못 뚫은 로봇키퍼에 도전하는 조원희.. 부숴버릴 기세로 풀파워 슈팅함ㅋㅋㅋㅋㅋ I 슛포러브 Shoot for Love - YouTube
    - 공이오는 Sequence를 보고 미리 자리를 잡는다. 현재의 상황 → 미래의 상황

순차 데이터(Sequence Data)

- 자연어 문장
- 심전도
- 동영상
- 주가

구성요소가 **시간**에 의해 순차적으로 발생하는 또는 요소간 순서를 지니고 있는 순차 데이터 이다. 서열데이터는 이전 값들의 결과가 이후에 나타나는 결과값에 영향을 줄수있으므로 이를 동시에 고려해야할 필요가 있다.

Time series Data

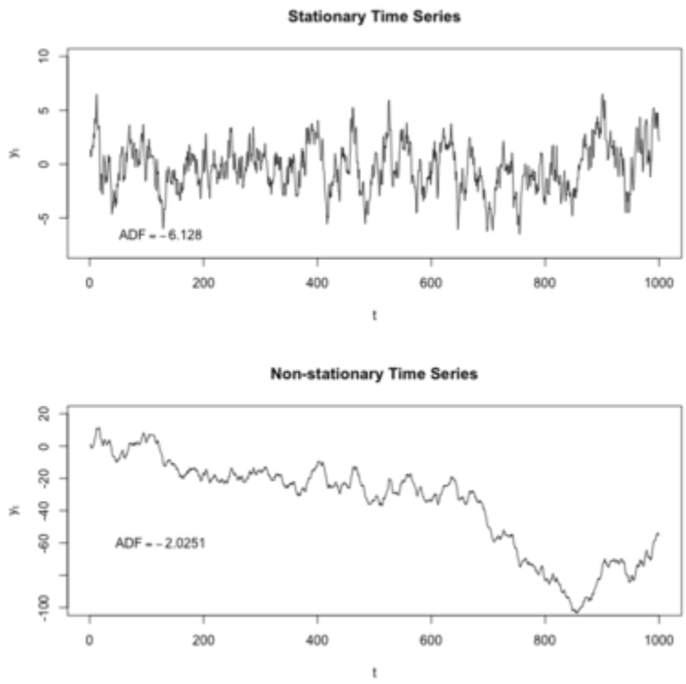
우리가 예측하고 싶어하는 Time Series Data는 크게 2가지로 분류할수 있다.

1. deterministic
   - 어떤 특정한 입력이 들어오면 언제나 똑같은 과정을 거쳐서 언제나 똑같은 결과를 내보내는 데이터
   - 예) 점화식

- $y[0]=1, y[1]=2, y[2]=2^2, \dots$ 을 생각해보자. 이를 일반화 시키면 어떻게 표현할수 있을까?
  - 방법1) $y[n]=2^{n-1}$
  - 방법2) $y_{n+1}=2y_n$ 단, $y[0]=1$
  - 두가지 방식으로 표현 가능하다.

2. stochastic
   - 과거의 데이터가 미래의 데이터에 영향을 미치지만 확률적으로 변화하는 데이터
   - Stationary : 시간이 변해도 평균과 분산이 변하지 않는다. → 특정 분포를 따라감

     Non-Stationary : 매 시간마다 평균과 분산이 바뀜(통계적 수치가 변함) → 분포를 예측하기가 어려움



결론

- 전통적인 시계열 데이터를 예측하는방법은 Markov Chain → Kalman Filter
- RNN은 미리 알고있어야 되는 확률적인 인과관계를 Deep Learning으로 구현한다.

Markov Procss

기본형 : 이전의 사건들이 영향을 미쳐 현재 사건이 일어난다면?

도입

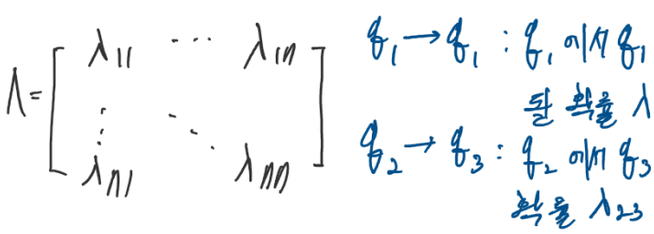
- 사건 q_t가 발생할수 있는 경우의 수가 다음과 같다고 하자.

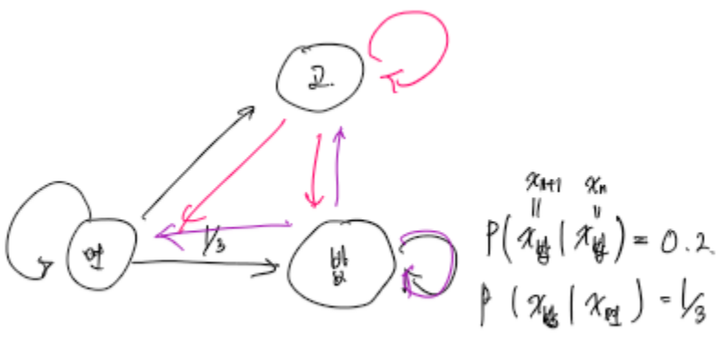
- $q_t \in \{S_1, S_2, \dots, S_N\}$

- 이때 다음과 같은 Time Series Data가 생성될수 있는 확률을 다음과 같이 정의할수 있다.
  - $p(q_0, q_1, \dots, q_t) = p(q_0)p(q_1|q_0)p(q_2|q_1, q_0)\dots p(q_t|q_{t-1}, \dots, q_0)$

Markovian Property

- Markovian Property는 지금의 상태가 미래를 측정하는 기준이 되기 때문에, 이미 현재의 상태에 과거의 확률이 축적되어있다고 가정한다.
  - $p(q_{t+1}|q_0, q_1, \dots, q_t) = p(q_{t+1}|q_t)$
- 따라서 위의 수식을 다음과 같이 정리 할수 있다.
  - $p(q_0, q_1, \dots, q_t) = p(q_0)p(q_1|q_0)p(q_2|q_1)\dots p(q_t|q_{t-1})$
  - 결국 너무 먼 과거에 영향받아 미래가 결정되는것이 아니라, 현재의 상황을 통해 미래가 결정된다는 것을 의미한다.(수식이 간단해지고 고려할 사항이 적어짐)
  - 예
    - 오민엽의 저녁 메뉴
    - 칼국수(밀) → 순대국밥(쌀) → 밀면(밀) → 대구탕(쌀) → 우육탕(밀) → 떡볶이(쌀) →?
      - 전날 떡볶이(쌀)를 먹었으니 오늘은 밀 음식을 먹을 예정
  - a에서 b로 상태가 전환될 확률을 다음과 같이 표기 하자.
    - $P_{ab} = p(q_{t+1} = a|q_t = b)$
    - $\lambda_{ab} = p(q_{t+1} = a|q_t = b)$
  - 그러면 다음과 같은 행렬식으로 상태 전환 확률을 표기할수 있다.

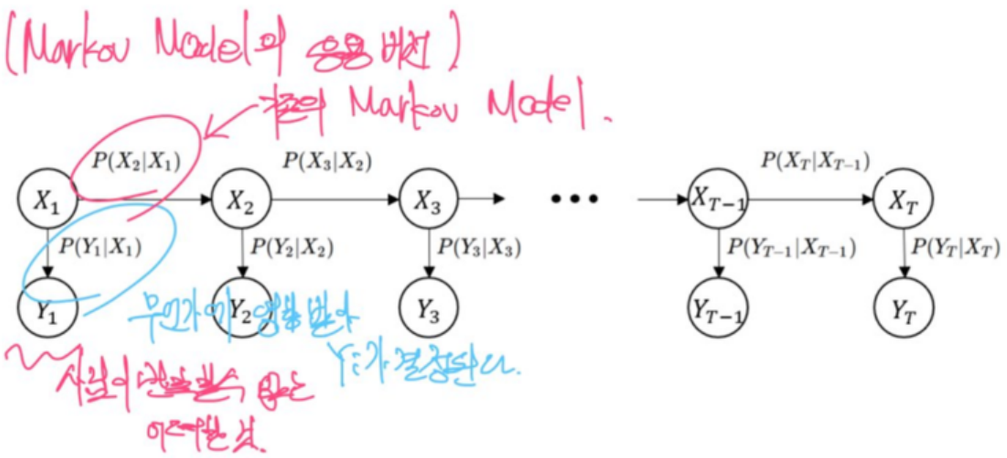




Hidden Markov Models

Markov Models은 너무 복잡하다. 언제쩍 과거까지 영향을 받는거지?

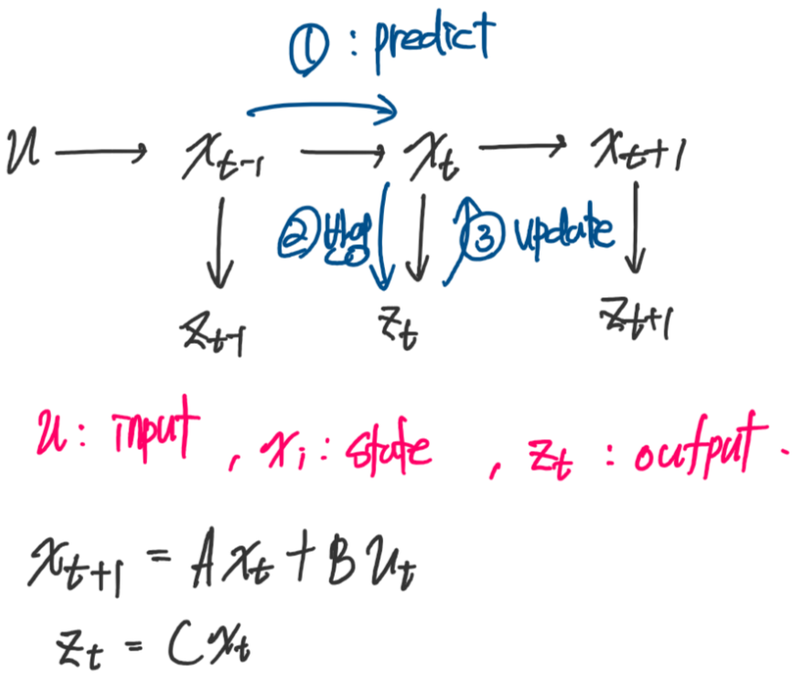
- Markov Chain 과 비슷하지만 과거 상태에서 영향받은 무언가에 영향받아 확률이 결정된다.



Kalman Filter

역학적인 결과를 측정해 시계열 데이터를 미리 예측

- 칼만 필터는 상태 예측과 측정값 업데이트를 반복적으로 수행하여 상태를 예측한다.
- 예를들어 로봇의 위치 계산
  - 사람은 지도를 보면서 자신의 위치를 대략적으로 판단할수 있지만, 로봇의 경우 다양한 센서를 이용해 확률기반으로 위치를 예측한다.
  - https://medium.com/@celinachild/kalman-filter-소개-395c2016b4d6
- 시계열 데이터를 예측하는 가장 일반적인 방법
- 모션 모델과 측정 모델이 linear할 경우 Kalman Filter를 사용한다.
- 모션 모델과 측정 모델이 Gaussian 분포(Normal distributions)를 따를 경우 Kalman Filter를 사용한다.
- A, B, C에 대한 역학적인 결과를 미리 알고 있어야 한다



u : input , x_t : state , z_t : output .

$$x_{t+1} = Ax_t + Bu_t$$

$$z_t = Cx_t$$

차이점

- 전통적인 시계열 데이터를 예측하는방법은 Markov Chain → Kalman Filter
- RNN은 미리 알고있어야 되는 확률적인 인과관계를 Deep Learning으로 구현한다.