**실무에 바로 쓰는 AI예측/추천 3주차 수업노트**

22.5.19

3주차 배울 내용

**설명력이 중요해 진 인공지능 알고리즘**

* Attention
* Transformer
* DA-RNN
* GAN

학습 내용

02. Seq2Seq모델의 한계

03. Attention

04. 트랜스포머 (Transformer)

05. Dual-Attention RNN

06. Dual-Attention RNN 실습

07. GAN 소개

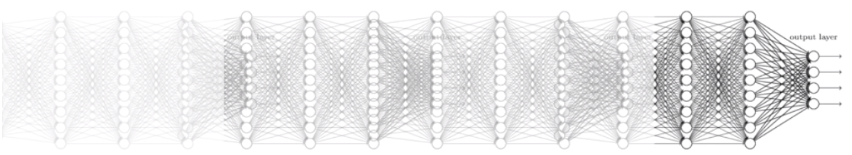
08. DC-GAN

09. GAN+LSTM

10. GAN-LSTM으로 주가예측하기

**02. Seq2Seq모델의 한계**

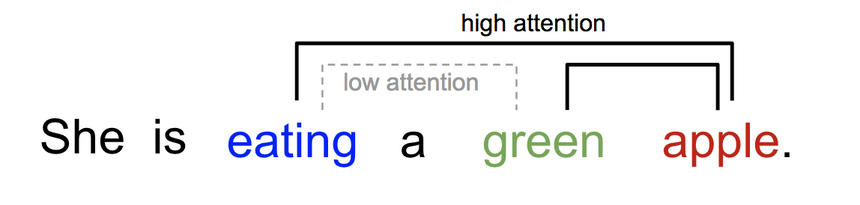
**Vanishing Gradient 문제**



알고리즘의 건망증. 인공지능도 오래된 것을 까먹는다?

**Seq2Seq모형에서의 문제**

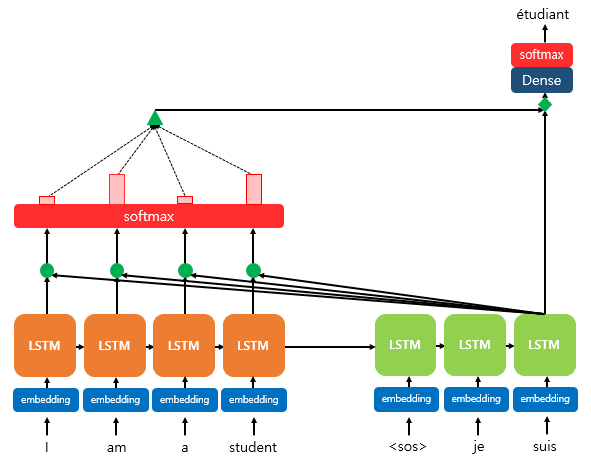
인코더/디코더 반복과정에서 중요한 정보가 손실될 수 있다



**구조적인 문제**

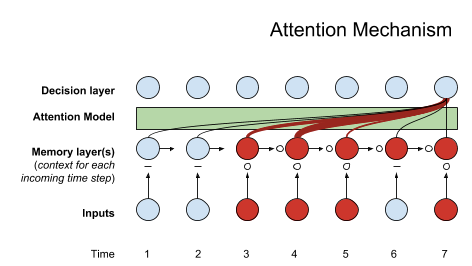
**03. Attention**

Attention 모형은 Seq2Seq의 디코더에서 출력결과를 예측할때마다 중요한 hidden state들을 다시한번 참고하는 모형



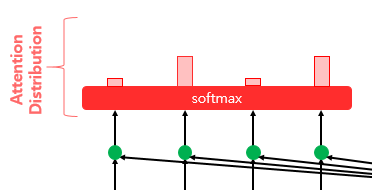
**Attention Score**

현재 디코더 시점 t에서 단어를 예측하기위해 인코더의 모든 은닉상태 각각이 디코더의 현 시점의 은닉 상태 s\_t와 얼마나 유사한지를 판단하는 스코어



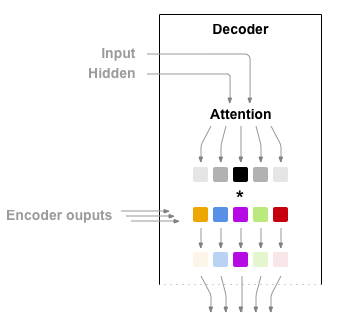
**Attention분포 (a)**

어텐션 스코어는 Scalar값이기 때문에 이를 softmax함수를 통해 Attention분포를 구함



**맥락 벡터 = Attention Value (a\_t)**

어텐션 분포를 인코더 hidden state와 곱해서 맥락벡터를 구함



**04. 트랜스포머 (Transformer)**

**seq2seq VS. Attention VS. Transformer**





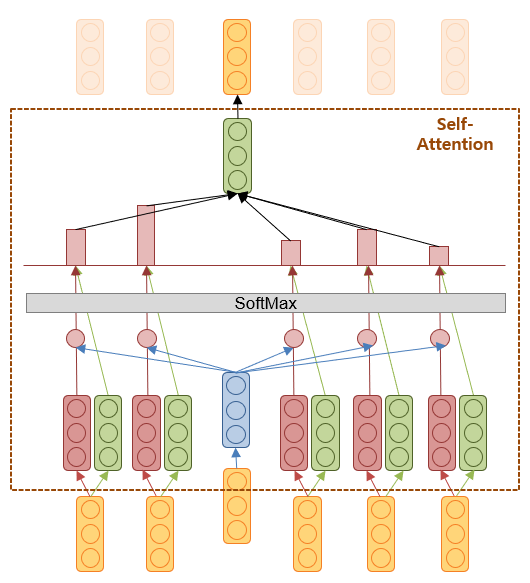
**Transformer의 특징**

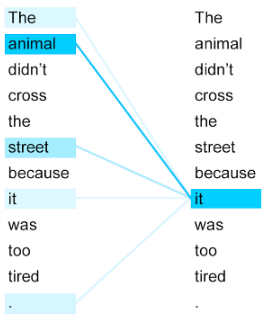
Seq2Seq 구조를 그대로 따르지만 RNN을 사용하지 않고 Attention만을 사용! —성능은 RNN보다 더 우수하고 연산도 빠름

**Key, Query, Value**

* Python Dictionary 와 비슷하게 Key-Value로 구성되어 있음  
   예) dict = {'철수': '3반', '영희': '2반'}
* **Attention(Q,K,V) = Attention Value**  
  - 주어진 쿼리에 대해서 모든 Key와의 유사도를 각각 구함  
  - 구해낸 유사도를 키와 Mapping되어있는 각각의 Value에 반영해줌  
  - 유사도가 반영된 값(Value)를 모두 더해서 Return → Attention Value  
  - Query: t 시점의 decoder의 hidden state (Info를 요청)   
  - Key: 모든 시점의 인코더 셀의 hidden states (Info가 있다고 응답)   
  - Value: 모든시점의 인코더 셀의 hidden state (Info를 제공)

**셀프어텐션**

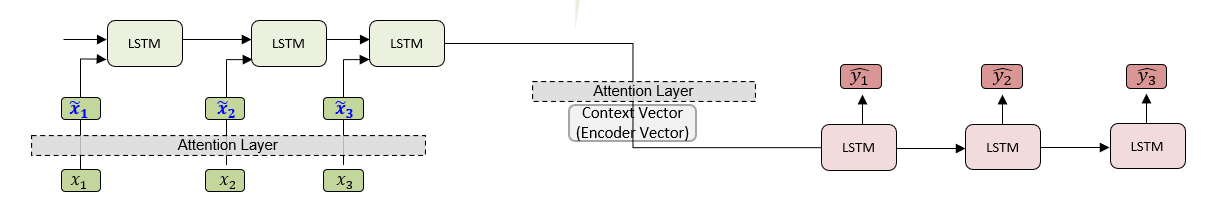




**05. Dual-Attention RNN**

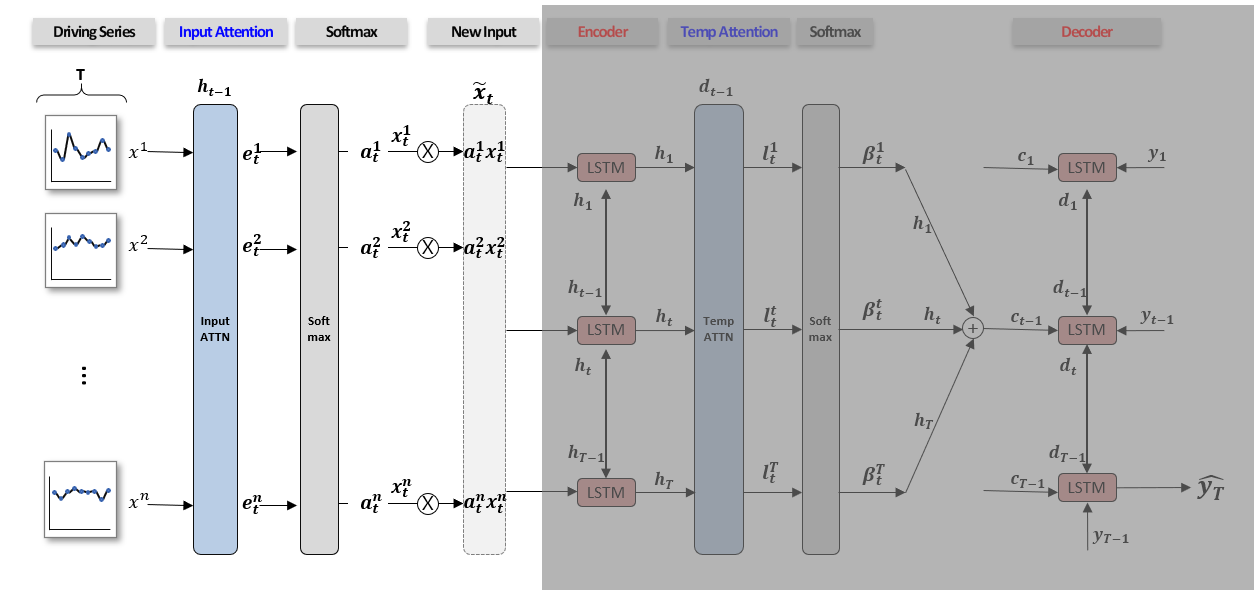
**Dual Attention RNN**

Dual Attention RNN은 두개의 Stage로 나누어진 구조로 가장 기본적인Seq2Seq의 틀을 가지고 있음(Input Attention + Encoder + Decoder )



**Input Attention**

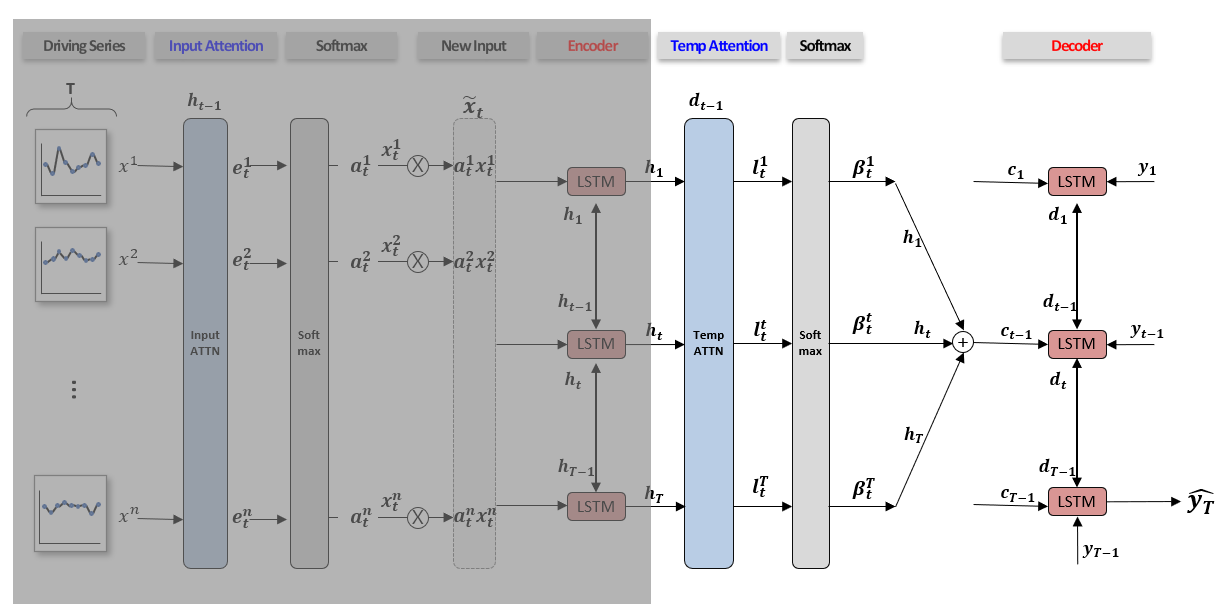
외생변수들 사이에서 Encoder hidden state를 사용하여 중요한 Feature들을 뽑아냄

****

**Encoder**

모든 timestep에 대해 attention weights를 구해준 것을 받으면 Original Input Data와 곱해서 새로운 x\_t를 구현

**Decoder**

****

새로운 input을 LSTM을 거친뒤에 hidden state를 뽑아 두번째 Attention인 "Temporal Attention"을 적용

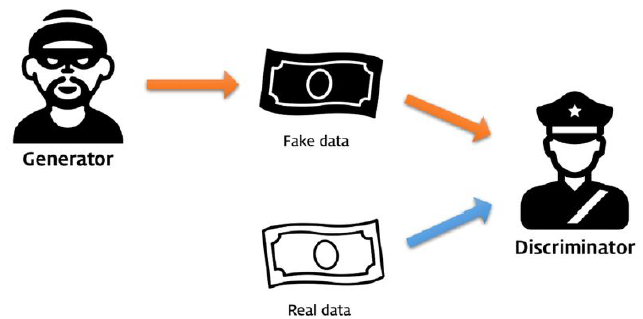
**07. GAN 소개**

GAN이란? (Generative Adversarial Network)

**Generative : 생성하는**

GAN은 한마디로 어떤 분포 혹은 분산 자체를 만들어내는 모델(그럴듯한 가짜를 만들어냄!)

**Adversarial(적대적)**

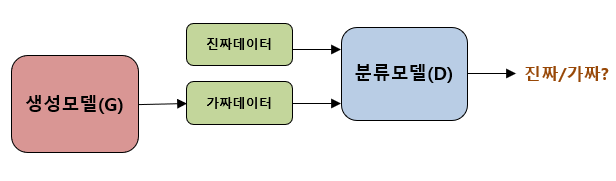


* 위조범: Generator (생성자 G) —단속을 피하기위해 더욱 정교한 가짜를 만듬
* 경찰: Discriminator (구분자 D) —점점더 정교한 방법으로 판별해내는 방법을 개발해 냄
* 생성자와 구분자가 적대적으로 경쟁하면서 서로가 발전함
* 경쟁하듯 실제로는 함께 학습의 방향성을 지시해주는 Unsupervised Learning

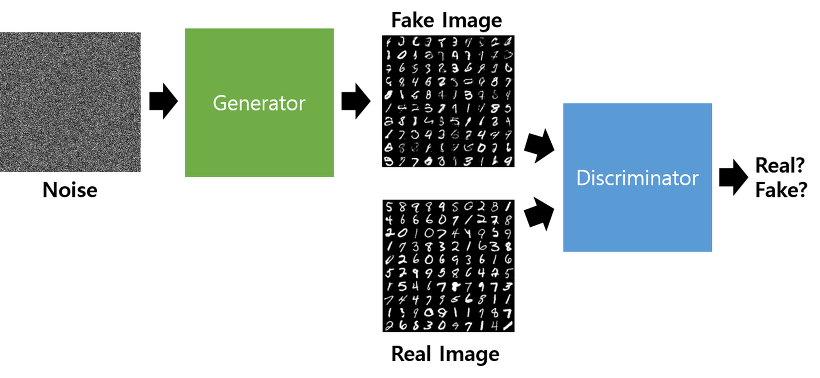
**Network**

인공신경망/딥러닝으로 만들어졌으며, 생성이라는 문제를 풀기위해 딥러닝으로 만들어진 모델을 적대적 학습이라는 독특한 방식으로 학습시키는 알고리즘

**GAN의 학습 과정**

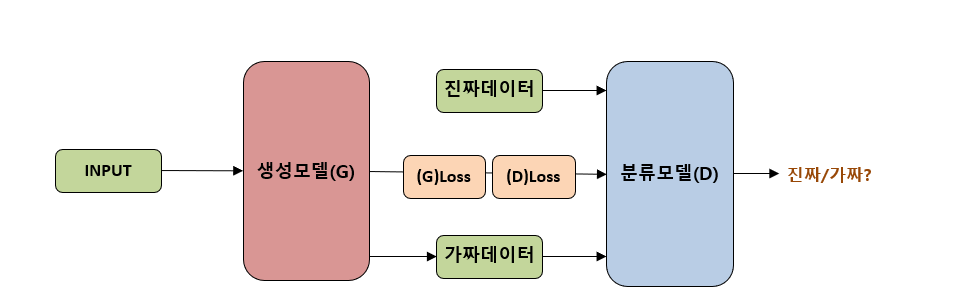


G한번 D한번 이 아닌 진짜데이터와 가짜데이터를 D의 인풋으로 합쳐서 학습 시킴



**Gan 의 Loss Function들 (학습의 목적함수)**

GAN은 **MinMax Problem**! 목적함수(V)는 G는 최소한으로 D는 최대한으로 끌어내는 방향으로 학습해야함

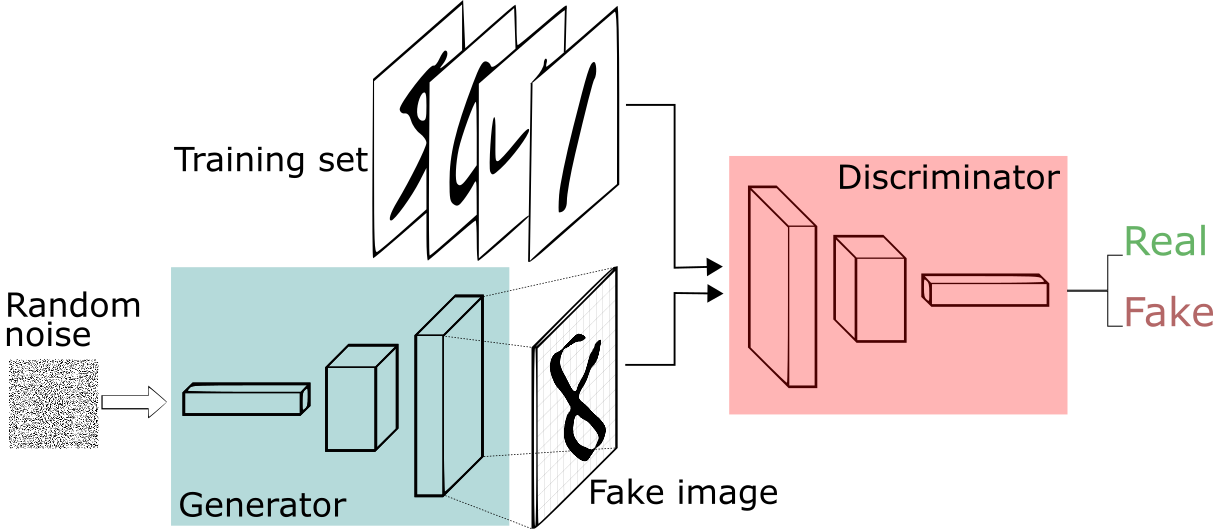


**GAN의 제한**

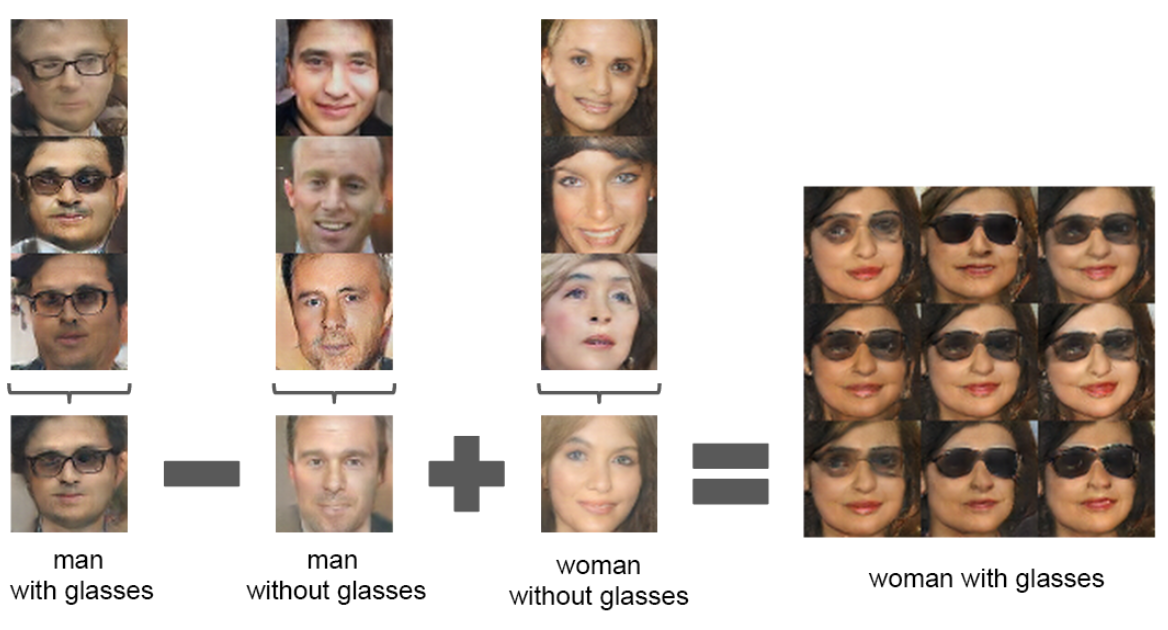
* GAN은 학습이 굉장히 불안정함(D와 G가 고루 학습이 되어야 하는데 편향적으로 학습될 수 있음)
* 성능평가의 어려움(사람의 눈으로 파악할 수 밖에 없음. Overfitting을 판별하기도 어려움 )
* Mode Collapsing (훈련된 Multi-model 데이터의 일부가 누락)

**08. DC-GAN**

DC-GAN은 Deep Convolutional GAN: 기존 GAN에서 존재하던 Fully Connected Layer구조를 CNN로 대체하면서 GAN의 기존 단점을 극복!



* 기본적으로 GAN과 전체적인 구조는 같음! (Noise→ Generator → Fake Data → Discriminator )
* FC대신 CNN을 사용 → Conv & Pooling 교차  
  - 전체 데이터의 특정 부분에 대한 "특징"을 추출하는 Conv Layer  
  - 추출한 특징중 중요한것만 Filter해주는 Pooling Layer  
  - 부분의 특징을 읽어내는 성능이 탁월→ 학습이 잘됨
* Batch Norm을 G와 D모두에 사용



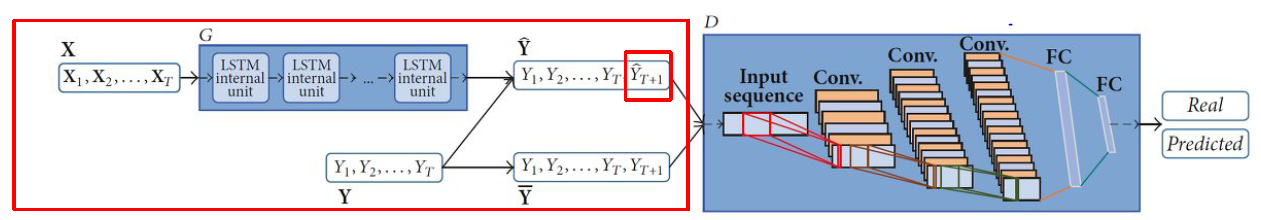
**AI Fake Image 처리기술로 DC-GANN이 많이 사용된다**

**09. GAN+LSTM**

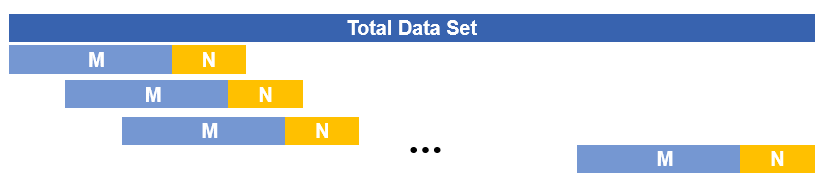
**GAN for Time Series Data1: 주가예측**

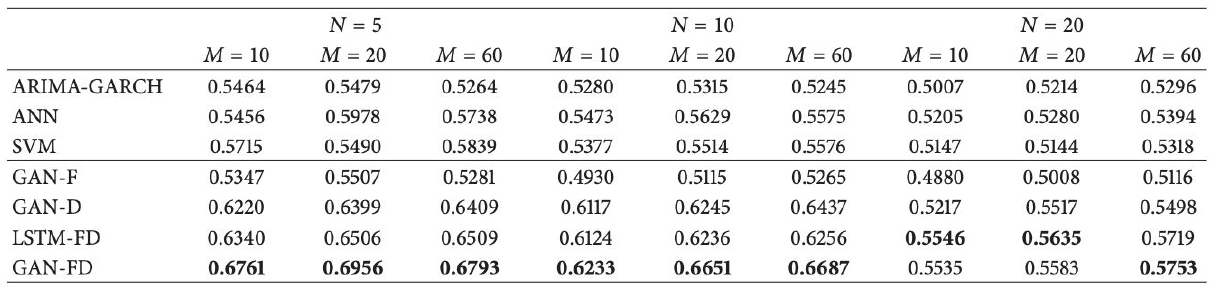
* GAN 을 LSTM와 결합시킨 시도
* 주가예측은 굉장히 어려운문제! (Non-Decidable, Non-Stationary, Stochastic Variable )
* 일반적 Index를 Input으로 사용 하였음
* Rolling Segmentation을 Train/Test에 적용 → 모델 파라메터 Update Cycle을 구현

**구현 방법**

* Y: Market Index를 t+1만큼 예측 (1분)   
  - T=242 (거래일 당 242분)
* **G(생성자)**  
    
  - 1~t 시점의 데이터를 LSTM에 통과   
  - Layer Output과 1~t시점의 주가를 합쳐서 t+1시점의 주가를 예측  
  - "지표"들로 주가를 예측
* **D(구별자)**  
  - 1~t 시점의 주가 + (t+1)시점의 주가를 받았을때 이 실제가 진짜인지 가짜인지를 훈련  
  - D와 G를 번갈아가면서 훈련시킴
* **Evaluation Metrics**  
  - G: RMSRE→ RMSE대신 사용한 이유는 42개 주식에 대한 결과를 동일한 비중으로 가져가기 위해  
  - D: Direction Prediction Accuracy (DPA) → 예측값 자체보단 방향의 정확도를 측정 → 높은 DPA: 이득이되는 거래 확률 Up

**데이터**

* 244거래일 \* 242분 = 59048 개 거래 데이터
* CSI300(Y)종가 : 42개 주가로 모델링 진행
* Input: 13 Indicators (X)  
  - 시가, 최고가, 최저가, 거래량, Turnover, Bias, 볼린저벤드값, MA값, Stochastic Index, MAXD, Relative Strength Index
* Rolling Segmentation 방식   
    
  - M: 학습기간/ N: Test기간  
  - 훈련 한번 완료 후 N기간동안 이동해서 학습  
  - 트레이더의 거래 스타일이 바뀌는것을 감안  
  - M={10, 20, 60}, N={5, 10, 20} → 1주, 2주, 1달, 1분기  
   - 전체 데이터를 (M+N)의 길이만큼 잘라서 학습 → M=20, N=5일때 가장 성능이 좋았음 → 모든경우 N이 5일때 가장 성능이 좋음



10. GAN-LSTM으로 주가예측하기