Deep Learning for Time Series Forecasting

☑ Introduction

1) 딥러닝은 multiple input과 output을 지원한다.

데이터 세트에 에너지 소비 값과 함께 온도, 풍속, 구름량 등과 같은 날씨 데이터가 포함되어 있다고 가정한다면 이 경우 에너지 소비 값을 최적으로 예측하기 위해 고려해야 할 여러 변수가 있다. 이와 같은 시리즈를 다변량 시계열이라고 한다.

신경망을 사용하면 임의의 수의 출력 값을 지정할 수 있으므로 다변량 예측 및 다단계 예측 방법이 비교적 간단하다.

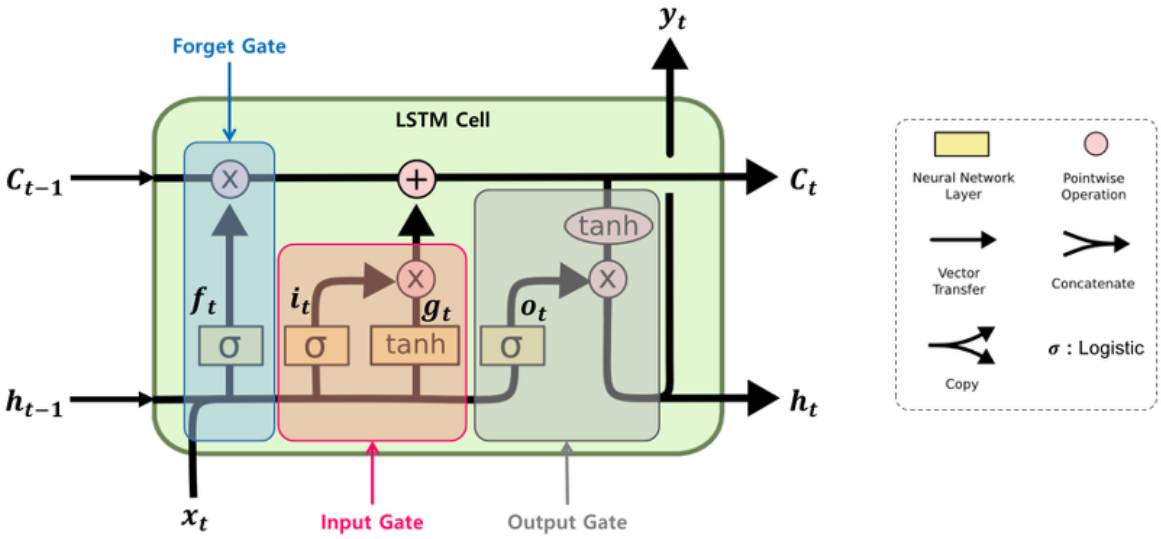
2) 딥러닝 네트워크는 상대적으로 긴 시퀀스에 걸쳐있는 입력 데이터에서 패턴을 추출하는 데 능숙하다.

딥러닝(특히 RNN 계열)은 시계열 데이터에서 가지고 있는 특유의 긴 sequence에 걸쳐있는 패턴을 추출하는데 잘 작동한다.

하지만, 결과물의 해석에 어려움이 있다.

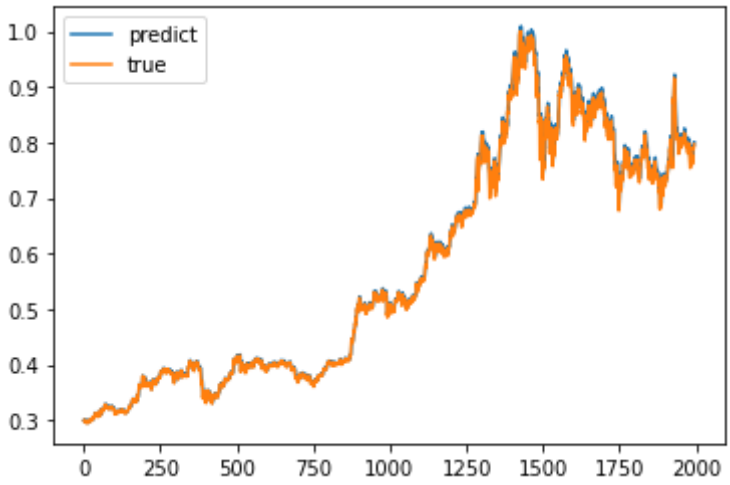
☑ Experiment

1) LSTM

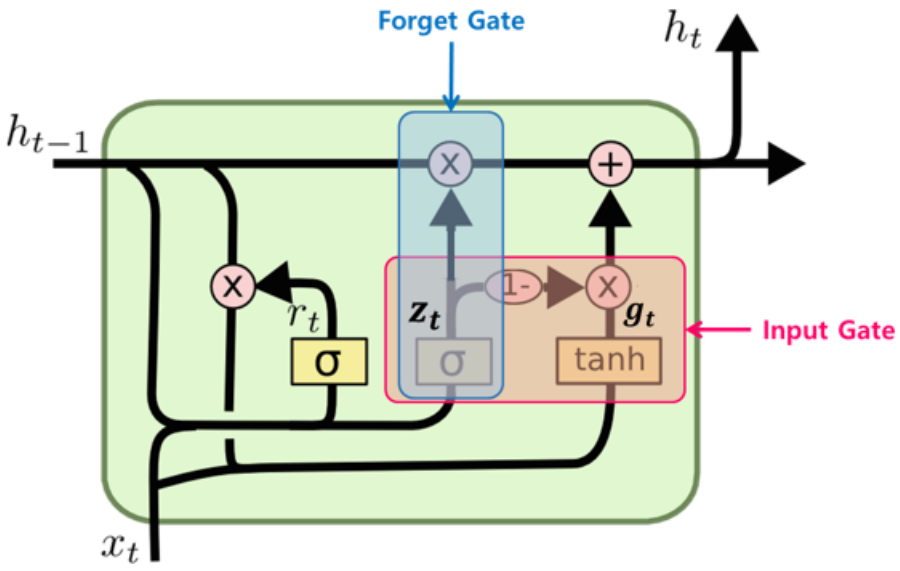


장기기억 c_{t-1}은 셀의 왼쪽에서 오른쪽으로 통과하게 되는데 **forget gate**를 지나면서 일부를 기억(정보)을 잃고, 그 다음 덧셈(+) 연산으로 **input gate**로 부터 새로운 기억 일부를 추가한다. 이렇게 만들어진 c_t는 별도의 추가 연산 없이 바로 출력되며, 이러한 장기 기억 c_t는 타임 스텝마다 일부의 기억을 삭제하고 추가하는 과정을 거치게 된다. 그리고 덧셈 연산 후에 c_t는 복사되어 **output gate**의 $tanh$ 함수로 전달되어 단기 상태 h_t 와 셀의 출력인 y_t를 만든다.

- LSTM의 핵심은 상단의 수평선 셀 스테이트(cell state)
- 하나의 컨베이어 벨트와 같으며 마이너한 선형 연산을 거치고 전체 체인을 관통한다.
- 이로 인해 정보는 큰 변함없이 계속적으로 다음 단계에 전달되게 된다.



2) GRU

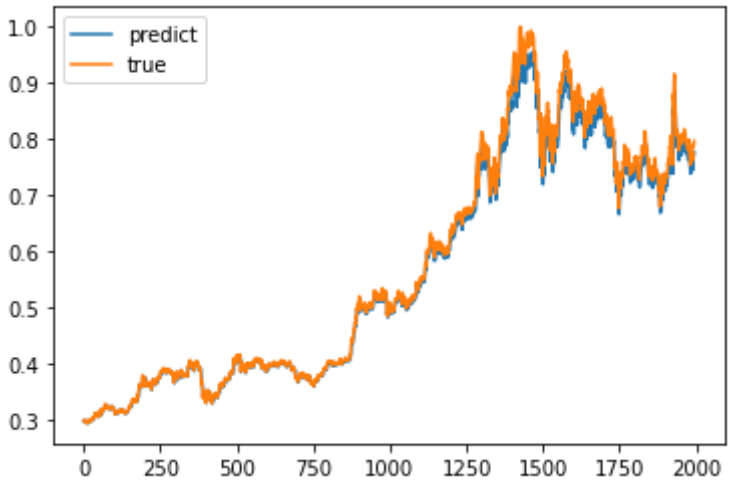


LSTM Cell에서의 두 상태 벡터 c_t와 h_t가 하나의 벡터 로 합쳐졌다.

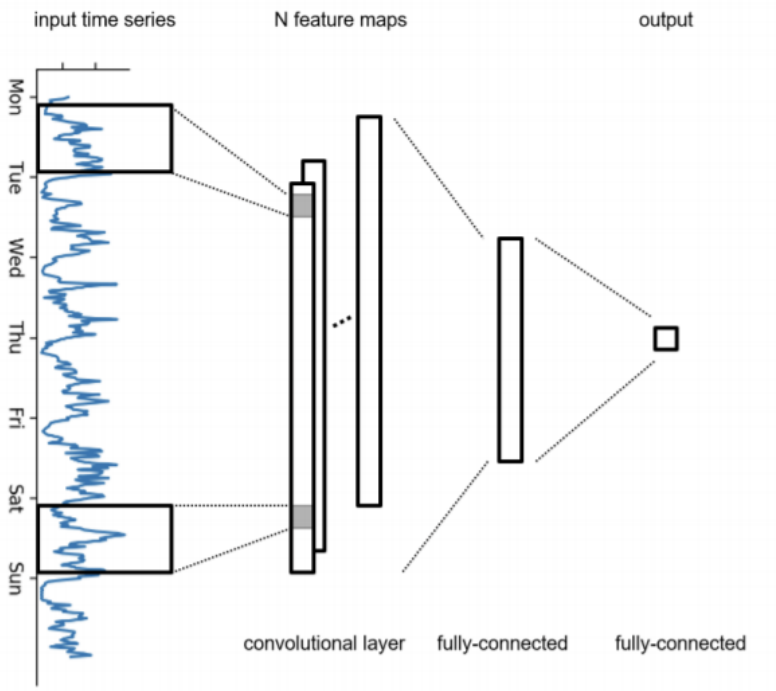
하나의 gate controller인 z_t가 **forget**과 **input** 게이트(gate)를 모두 제어한다. z_t가 `1`을 출력하면 forget 게이트가 열리고 input 게이트가 닫히며, z_t가 `0`일 경우 반대로 forget 게이트가 닫히고 input 게이트가 열린다. 즉, 이전$(t-1)$의 기억이 저장 될때 마다 타임 스텝 t의 입력은 삭제된다.

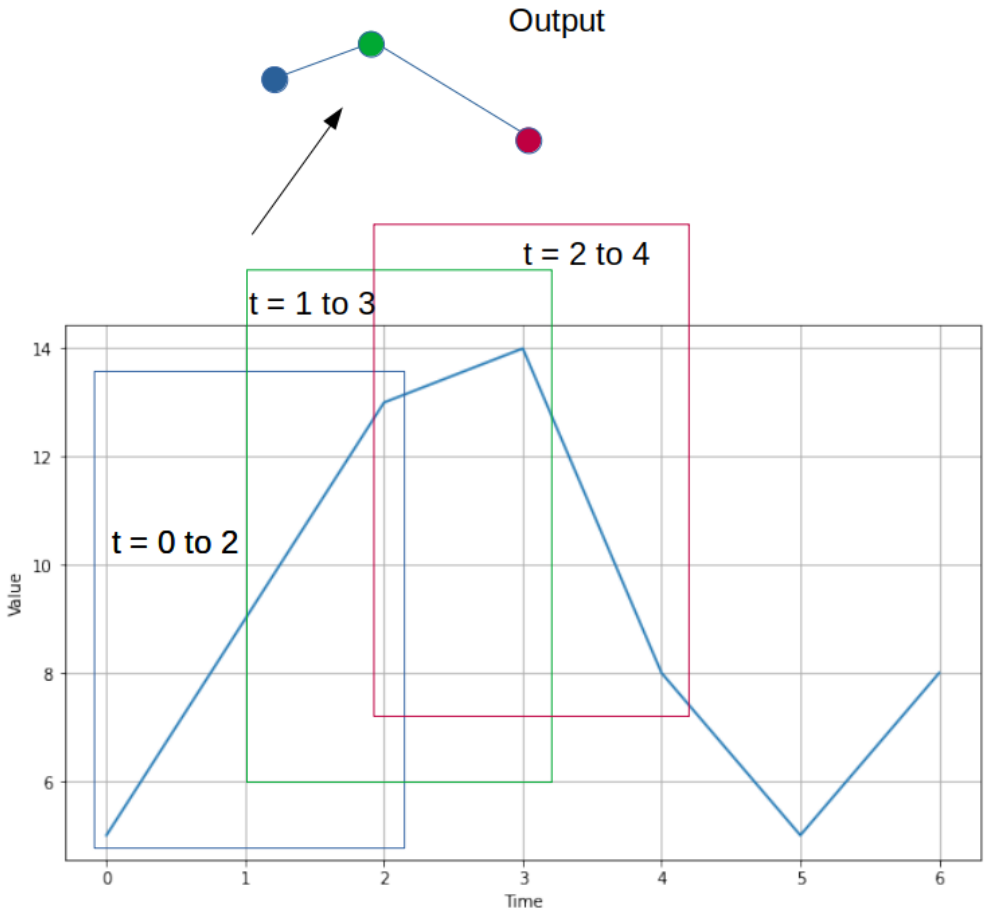
GRU 셀은 output 게이트가 없어 전체 상태 벡터 h_t가 타임 스텝마다 출력되며, 이전 상태 h_{t-1}의 어느 부분이 출력될지 제어하는 새로운 gate controller인 r_t가 있다.

- 이런 과정들을 통해 LSTM에 비해 학습할 가중치가 적다.
- 주제별로 LSTM이 좋을때도 있고, GRU가 좋을때도 있는데 비트코인 데이터의 경우 LSTM이 더 좋았음



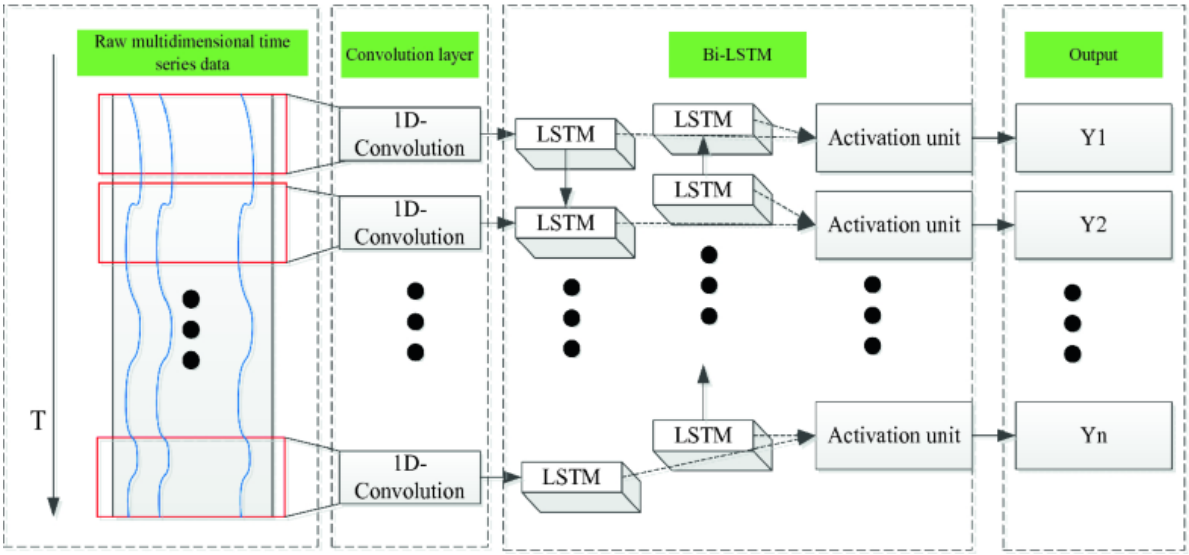
3) 1-D CNN



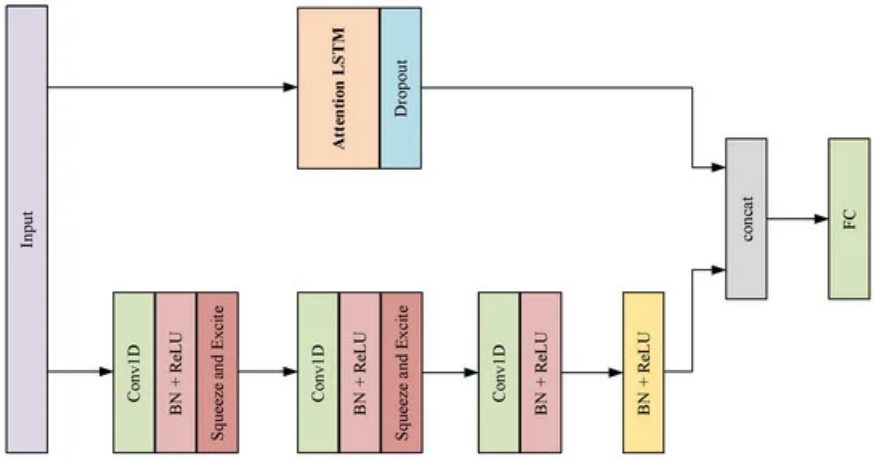


- CNN은 feature extraction에 탁월
- kernel_size=3 이라면 위와 같이 진행됨
- kernel_size가 작으면 성능이 안 좋을 수 있고, 크면 overfitting을 일으킬 수 있다.

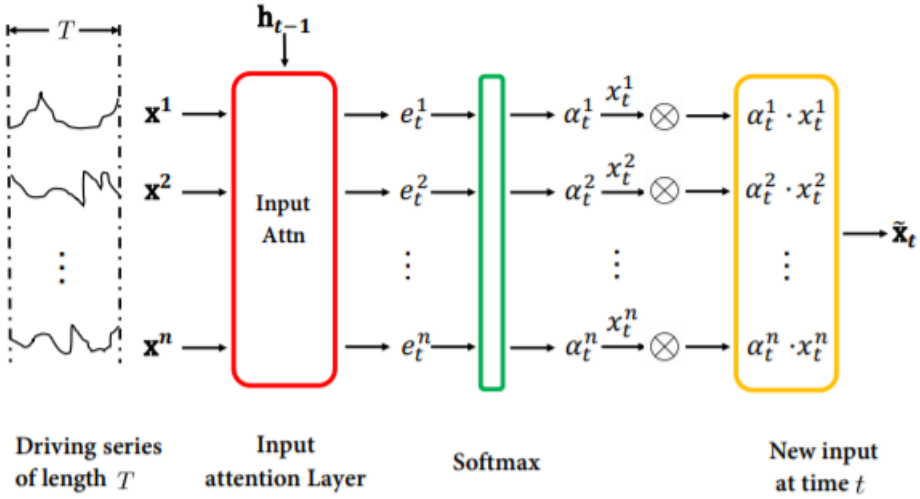
4) CNN-LSTM



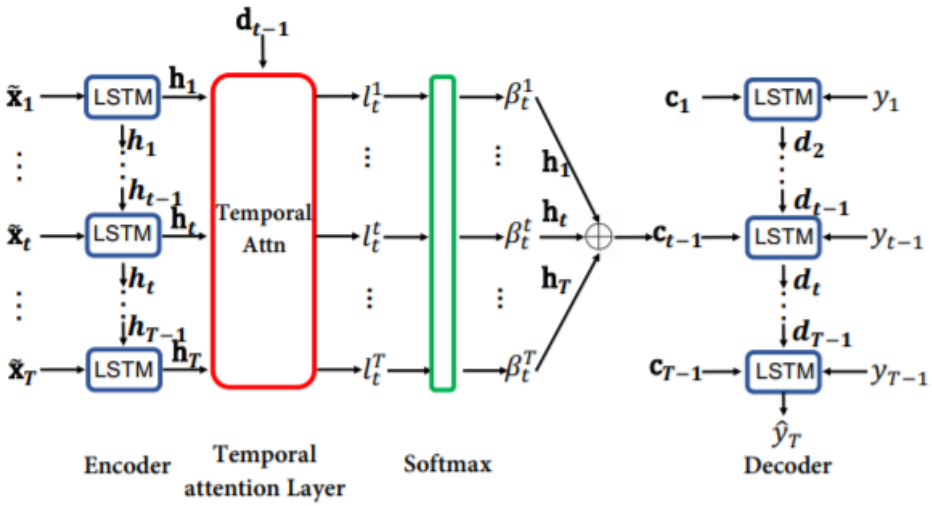
5) Multi-input model



6) DA-RNN (Dual-stage attention-based recurrent neural network)



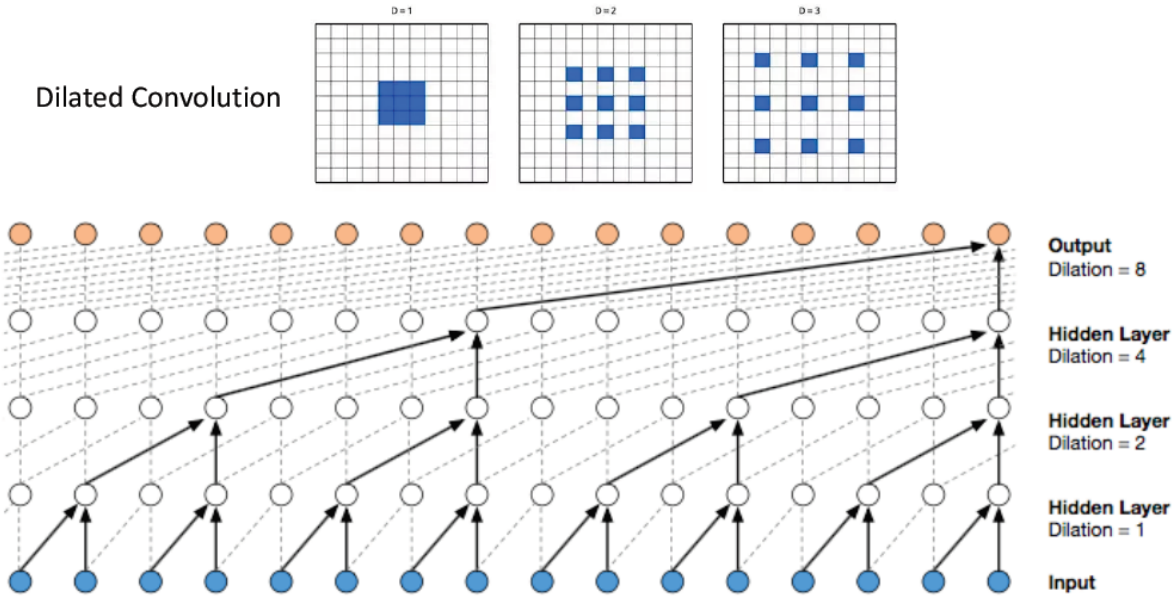
Encoder에서 사용되는 Input Attention은 예측하고자 하는 변수에 영향을 끼치는 외생변수들 중에 의미 있는 변수들에 attention 하여 사용하기 위해 적용된다. 각각 T의 시간 길이를 갖는 n개의 데이터를 사용하고 이를 Encoder에 있는 LSTM에 넣어줘서 hidden state를 뽑아준다.



2번째 Attention인 Temporal attention을 적용해준다. 이 기법은 Encoder에서 얻은 모든 Time step에서의 Hidden state와 각 time step에서의 Decoder LSTM의 hidden state를 비교하여 Attention 한 Context Vector를 추출하기 위한 메커니즘이다. (context vector : 어텐션의 최종결과값)

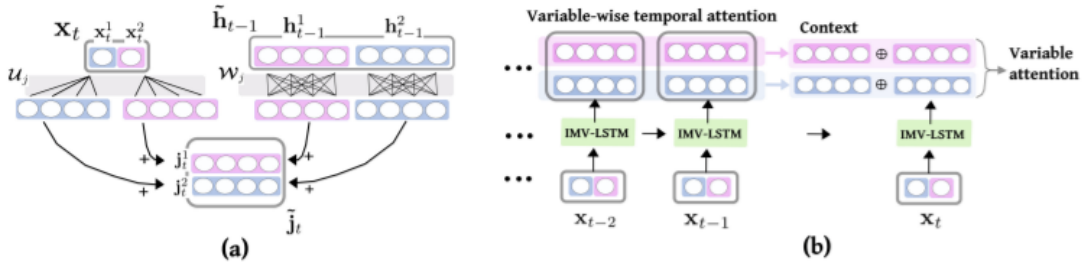
7) WaveNet

- wavenet에서는 아래 그림과 같이 오직 과거의 파형 정보만 접근할 수 있도록 causal convolutional layer를 여러겹 쌓았다.
- 긴 시계열을 다룰 때 RNN을 사용하지 않고 causal convolutional layer를 사용하면 모델을 빠르게 학습할 수 있다.
- DeepMind의 wavenet은 원래 오디오 생성을 위해 만들어졌지만 언어 번역과 시계열 예측으로 좋은 결과를 보여주었다.



wavenet은 일정 스탭(dilation)을 건너뛰면서 filter를 적용하는 dilated convolution을 사용한다.
Dilated Convolution은 필터 내부에 zero padding을 추가해 강제로 receptive field를 늘리는 방법이다.
결국 필터를 통해 어떤 사진의 전체적인 특징을 잡아내기 위해서는 receptive field는 높으면 높을수록 좋다.

8) IMV-LSTM (Interpretable Multi-Variable LSTM)



x_t : 각 변수들 ex) 매매량, 종가 등등

y_t : 타겟 변수, 예측하고자 하는 변수

h_t : hidden state matrix (t: timesteps)

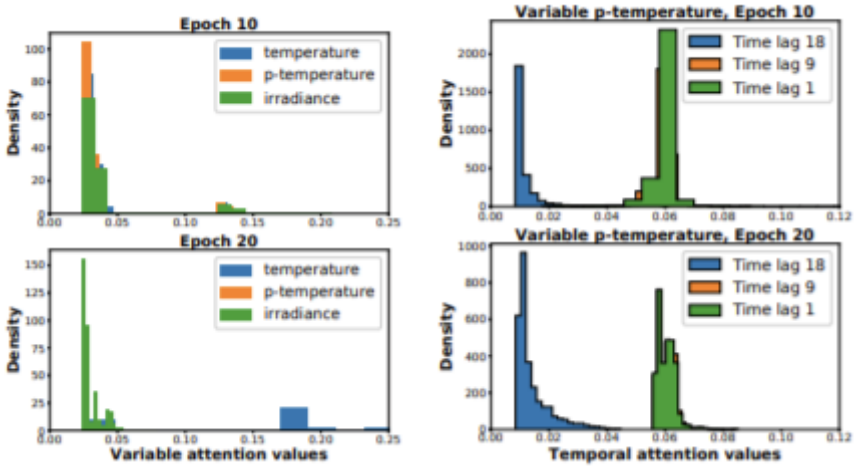
$context$: attention-value

	time	high	low	open	volumefrom	volumeto	close	conversionType	conversionSymbol	timeUTC
0	1576144800	7219.93	7149.15	7170.18	1536.79	11041994.81	7166.14	direct	NaN	2019-12-12 19:00:00
1	1576148400	7208.36	7166.03	7166.14	785.11	5654607.43	7199.79	direct	NaN	2019-12-12 20:00:00
2	1576152000	7222.43	7194.87	7199.79	672.46	4855213.22	7199.97	direct	NaN	2019-12-12 21:00:00
3	1576155600	7211.80	7187.29	7199.97	807.61	5821087.81	7201.38	direct	NaN	2019-12-12 22:00:00
4	1576159200	7229.22	7152.84	7201.38	1606.06	11537799.57	7173.50	direct	NaN	2019-12-12 23:00:00

$$\tilde{\mathbf{j}}_t = \tanh\left(\mathcal{W}_j \circledast \tilde{\mathbf{h}}_{t-1} + \mathcal{U}_j \circledast \mathbf{x}_t + \mathbf{b}_j\right),$$

해석 가능한 RNN은 중요 데이터 부분의 통찰력을 얻어 우수한 예측 성능을 달성할 수 있다 .

변수 중요도와 변수별 시간 중요도의 두 가지 중요도 해석에 중점을 둔다.



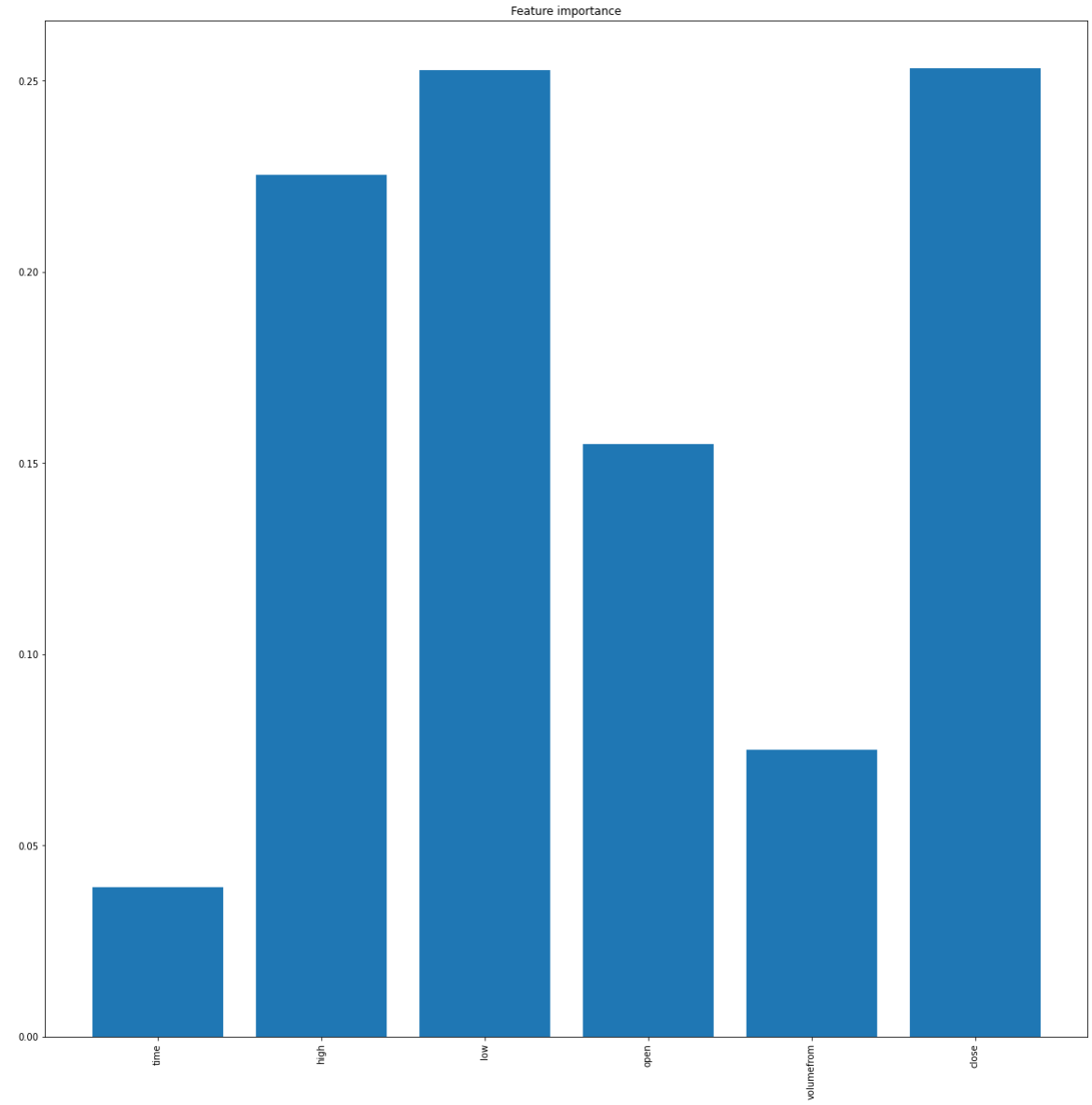
왼쪽 그림은 서로 다른 에포크에서 3가지 변수에 대한 어텐션 히스토그램, 오른쪽은 변수 "p-온도"의 시간 어텐션에 대한 히스토 그램. 이러한 히스토그램에서 변수중요도를 완전히 식별하기는 어렵다.

mixture attention: 시간 attention과 변수 attention이 변수별 상태를 병합하기 위해 도출된다. 이 두 단계는 확률론적 혼합 모델로 결합되어 후속 학습, 예측 및 해석을 용이하게 한다.

Table 1: RMSE and MAE with std. errors

Dataset	PM2.5	PLANT	SML
STRX	$52.51 \pm 0.82, 47.35 \pm 0.92$	$231.43 \pm 0.19, 193.23 \pm 0.43$	$0.039 \pm 0.001, 0.033 \pm 0.001$
ARIMAX	$42.51 \pm 1.13, 40.23 \pm 0.83$	$225.54 \pm 0.23, 193.42 \pm 0.41$	$0.060 \pm 0.002, 0.053 \pm 0.002$
RF	$38.84 \pm 1.12, 22.27 \pm 0.63$	$164.23 \pm 0.65, 130.90 \pm 0.15$	$0.045 \pm 0.001, 0.032 \pm 0.001$
XGT	$25.28 \pm 1.01, 15.93 \pm 0.72$	$164.10 \pm 0.54, 131.47 \pm 0.21$	$0.017 \pm 0.001, 0.013 \pm 0.001$
ENET	$26.31 \pm 1.33, 15.91 \pm 0.51$	$168.22 \pm 0.49, 137.04 \pm 0.38$	$0.018 \pm 0.001, 0.015 \pm 0.001$
DUAL	$25.31 \pm 0.91, 16.21 \pm 0.42$	$163.29 \pm 0.54, 130.87 \pm 0.12$	$0.019 \pm 0.001, 0.015 \pm 0.001$
RETAIN	$31.12 \pm 0.97, 20.11 \pm 0.76$	$250.69 \pm 0.36, 190.11 \pm 0.15$	$0.048 \pm 0.001, 0.037 \pm 0.001$
IMV-Full	$24.47 \pm 0.34, 15.23 \pm 0.61$	$157.32 \pm 0.21, 128.42 \pm 0.15$	$0.015 \pm 0.002, 0.012 \pm 0.001$
IMV-Tensor	$\mathbf{24.29 \pm 0.45, 14.87 \pm 0.44}$	$\mathbf{156.32 \pm 0.31, 127.42 \pm 0.21}$	$\mathbf{0.009 \pm 0.0009, 0.006 \pm 0.0005}$

밝은색 일수록 예측하는데 기여를 많이한 변수이다.



Data

더 큰 데이터로 실험해볼 예정..

1) bitcoin data

2) 자전거 대여

3) ?

논문방향

- 시계열에 관한 딥러닝 기술 소개 및 비교
- IMV-LSTM 이용해서 중요한 변수들과 timestep을 찾은 뒤, 주어진 데이터 내에서 가장 좋은 예측모델 찾기
- 혹은, 기술들을 hybrid 시켜 새로운 방법 제시