2025 09 25
발표 자료

광운대학교 로봇학과
FAIR Lab

김한서

KWANGWOON UNIVERSITY

2D CNN 모델

FAIR Lab

KWANGWOON UNIVERSITY

이번 주 진행사항

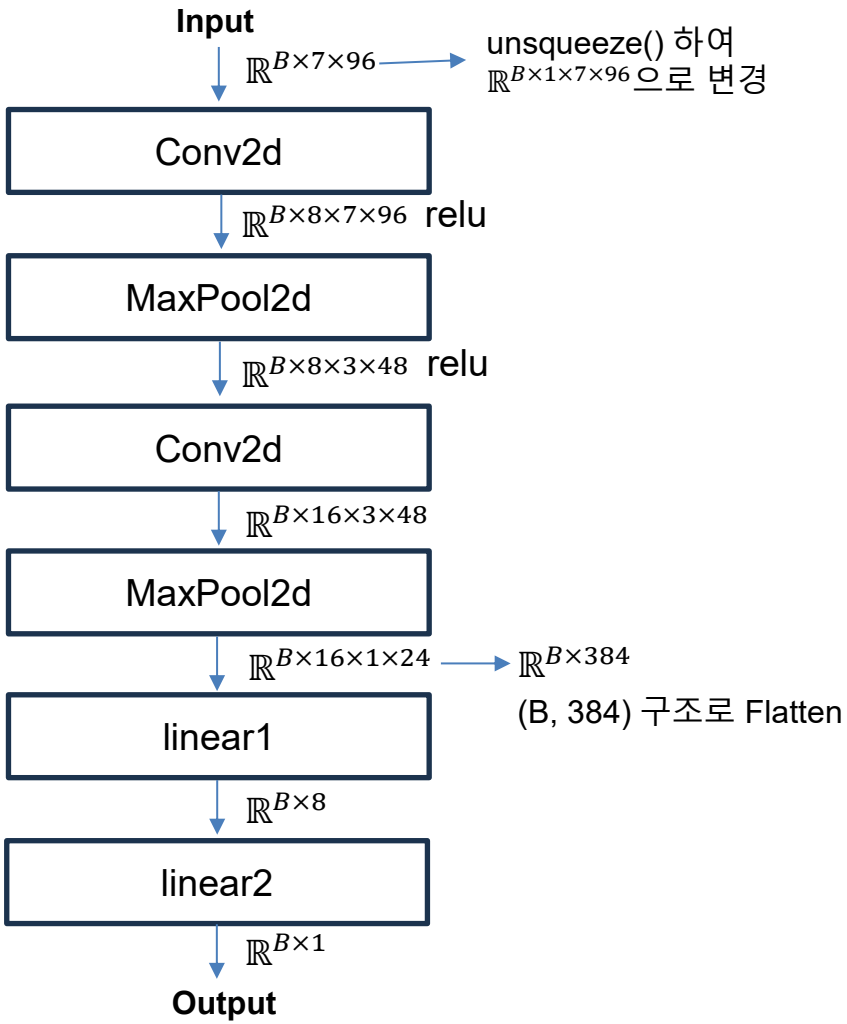
- 2D CNN 모델 학습
  - 채널 8-16 모델 learning rate 상향 전, 후 비교
  - 시각화
- Transformer, Attention 리뷰

모델 구조

FAIR Lab

KWANGWOON UNIVERSITY

*B : Batch Size



데이터셋 및 실험 세팅

- 사용한 종목: 514개 종목
- 데이터 기간 : 2009-12-31 ~ 2023-12-31
  데이터 분할: Train, Valid, Test 6:2:2
- 전처리: 결측치 제거 및 np.inf 삭제
- 정규화: StandardScaler
- Input feature → Open, Close, High, Low, Volume, Vwap, Ticker
  Label feature → 20_day_return_rate

Learning rate	**0.0001**
Epoch	500
Batch size	64
Loss function	MSE Loss
Sequence Length	96
input_feature	7
Output_window	1

제안 모델

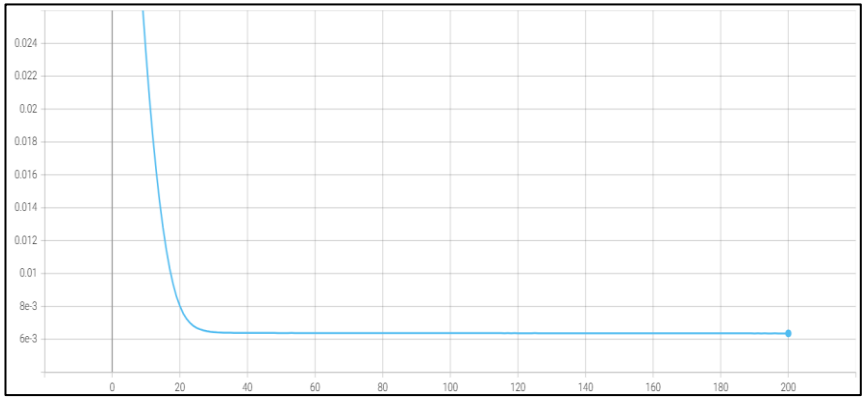
FAIR Lab

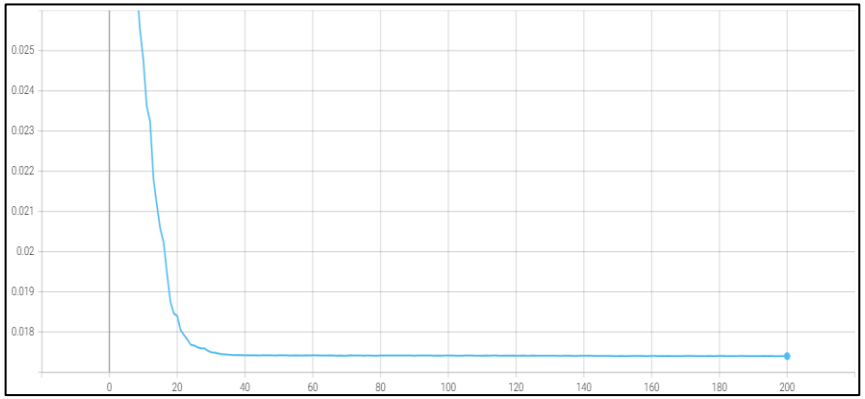
KWANGWOON UNIVERSITY

Learning rate 상향 전, 후 실험 결과

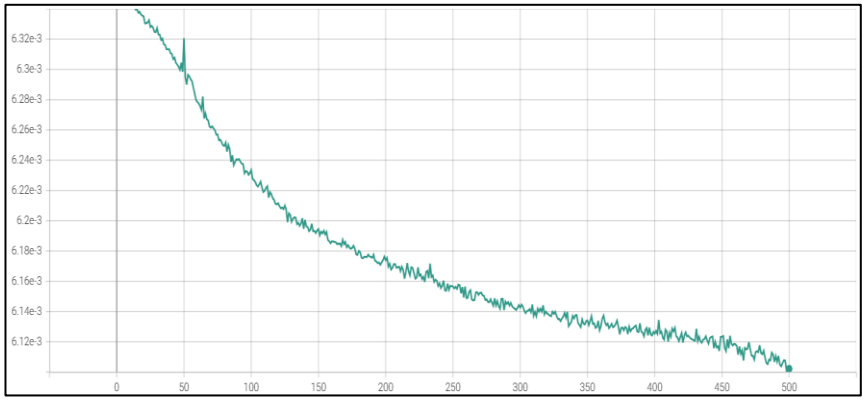
train loss **validation loss**

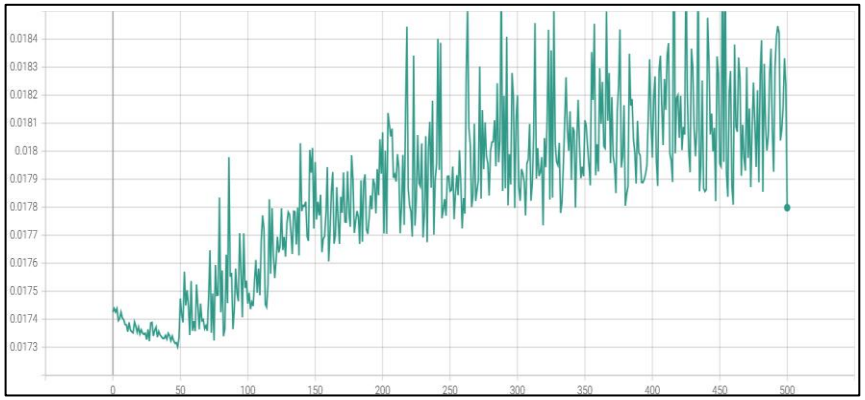
①





②





Test MSE ①	Test MSE ②
0.435130	0.435151

x: epoch y: loss

① → 2D CNN 학습률 0.000001 실험 결과
② → 2D CNN 학습률 0.0001 실험 결과

Label scale → 1.0
Label → 20_day_return_rate

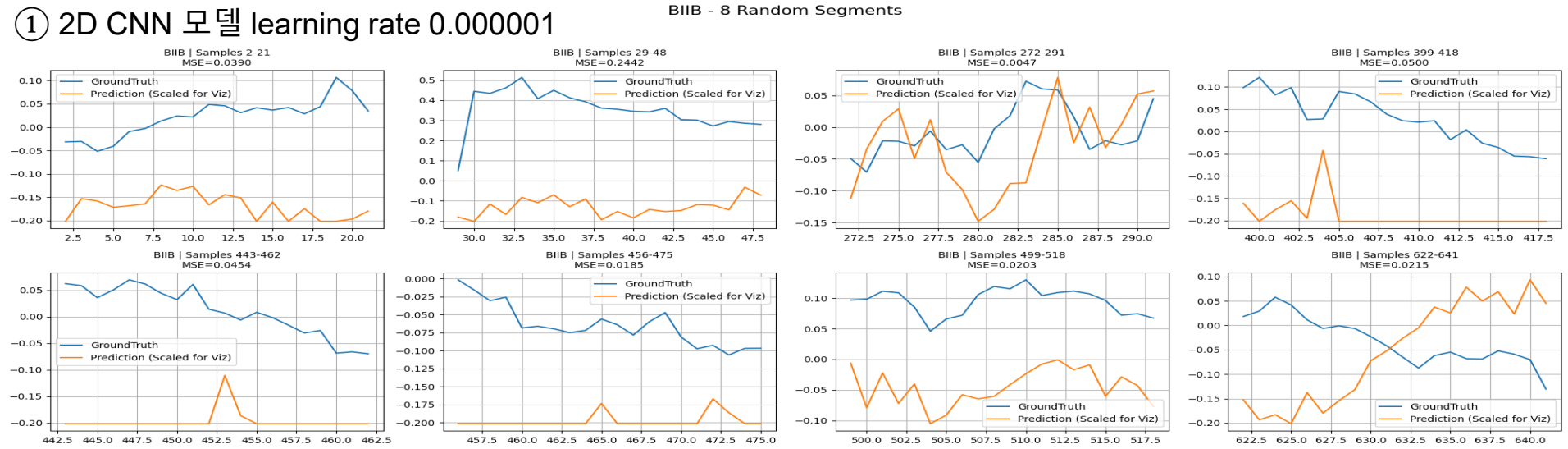
모델 시각화 비교

FAIR Lab

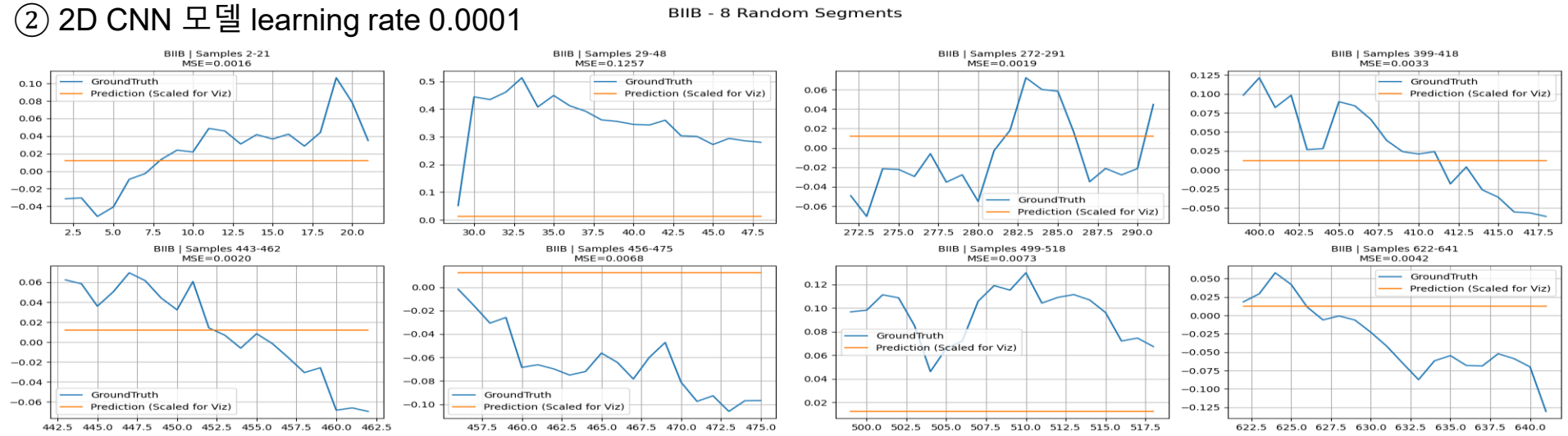
KWANGWOON UNIVERSITY

BIIB.csv 종목

① 2D CNN 모델 learning rate 0.000001



② 2D CNN 모델 learning rate 0.0001



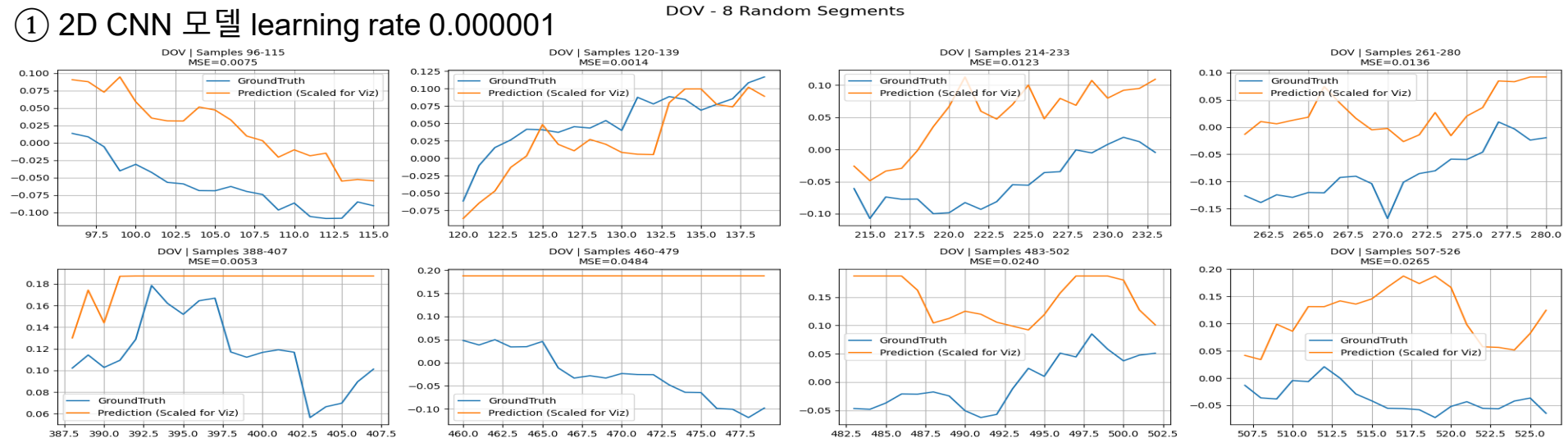
모델 시각화 비교

FAIR Lab

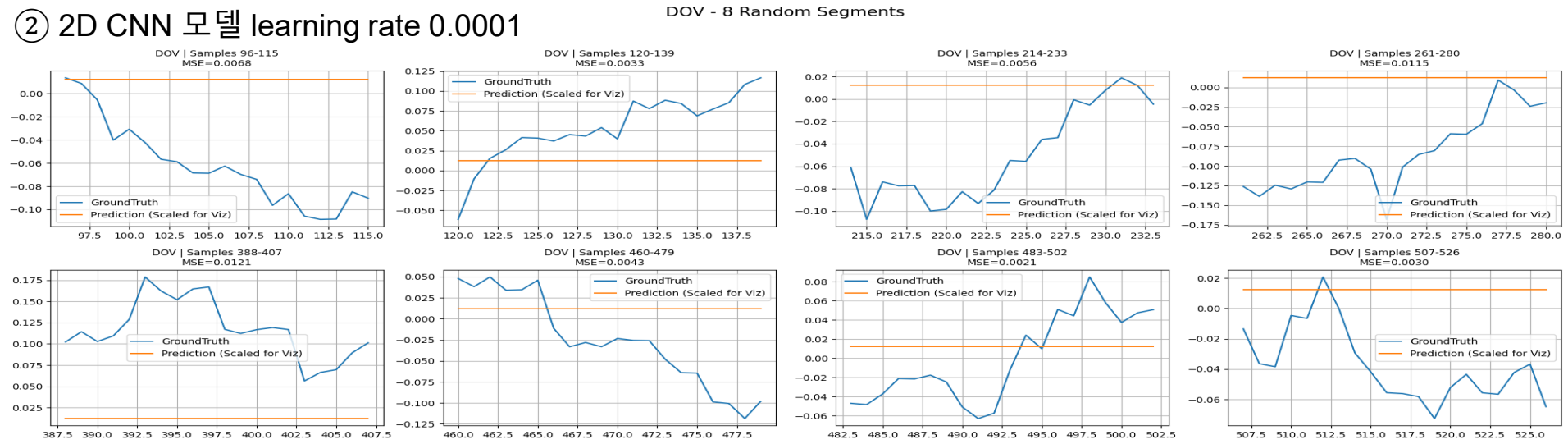
KWANGWOON UNIVERSITY

DOV.csv 종목

① 2D CNN 모델 learning rate 0.000001



② 2D CNN 모델 learning rate 0.0001



정리

실험 결과 정리

- 2D CNN 결과 비교
  - Learning rate 0.000001의 경우, 학습이 부족해 Loss Curve가 초반 이후 거의 감소하지 않고 정체되어 있고, learning rate 0.0001의 경우, 학습이 제대로 진행되지만 Valid Loss가 초반부터 상승하여 과적합 현상을 보이고 있습니다.
- 샘플 시각화 결과
  - Learning rate 0.000001은 학습이 부족했음에도 예측값이 어느정도 정답값을 따라가려는 모습을 보이고 있는 반면, learning rate 0.0001은 과적합으로 인해 모델의 일반화 성능이 낮아져 예측값이 대부분 flat하게 나와 정답값을 따라가지 못하고 있습니다.

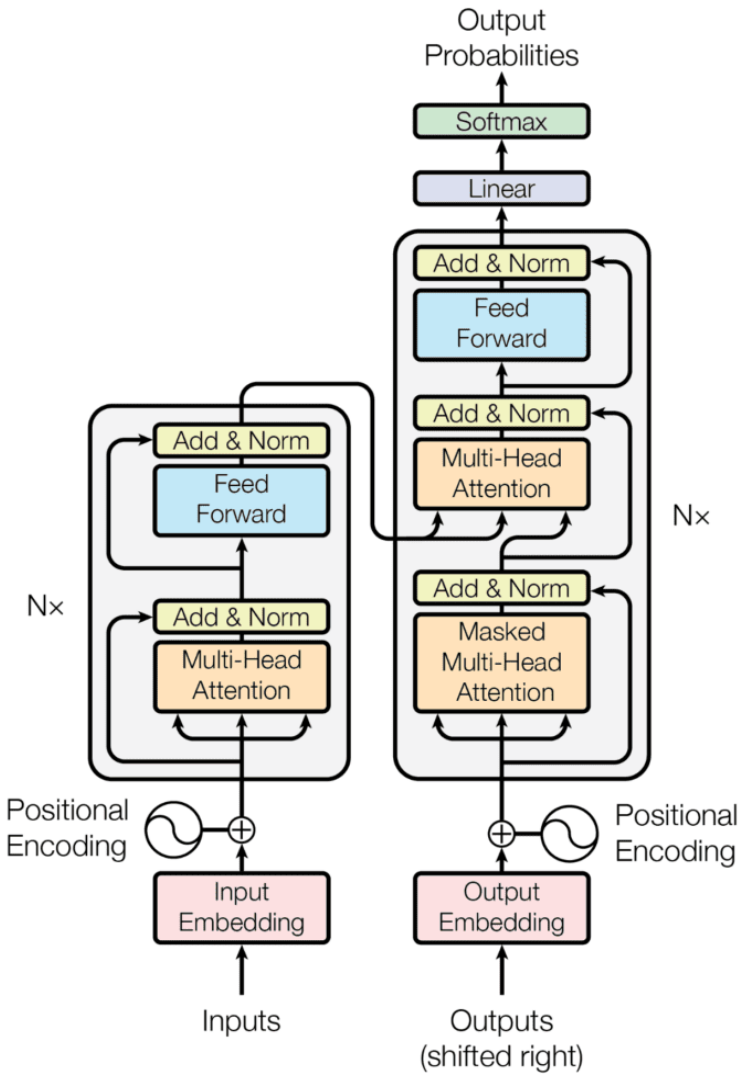
모델 채널 수	Learning rate	Label scale	Test MSE	학습 소요 시간
8 → 16	0.000001	1.0	0.435130	4시간 6분 44초
8 → 16	0.0001	1.0	0.435151	7시간 43분 16초

이후 계획

- 하이퍼파라미터 변경 후 실험 진행

  → Output_window를 1에서 48로 올린 뒤 진행

Transformer 리뷰



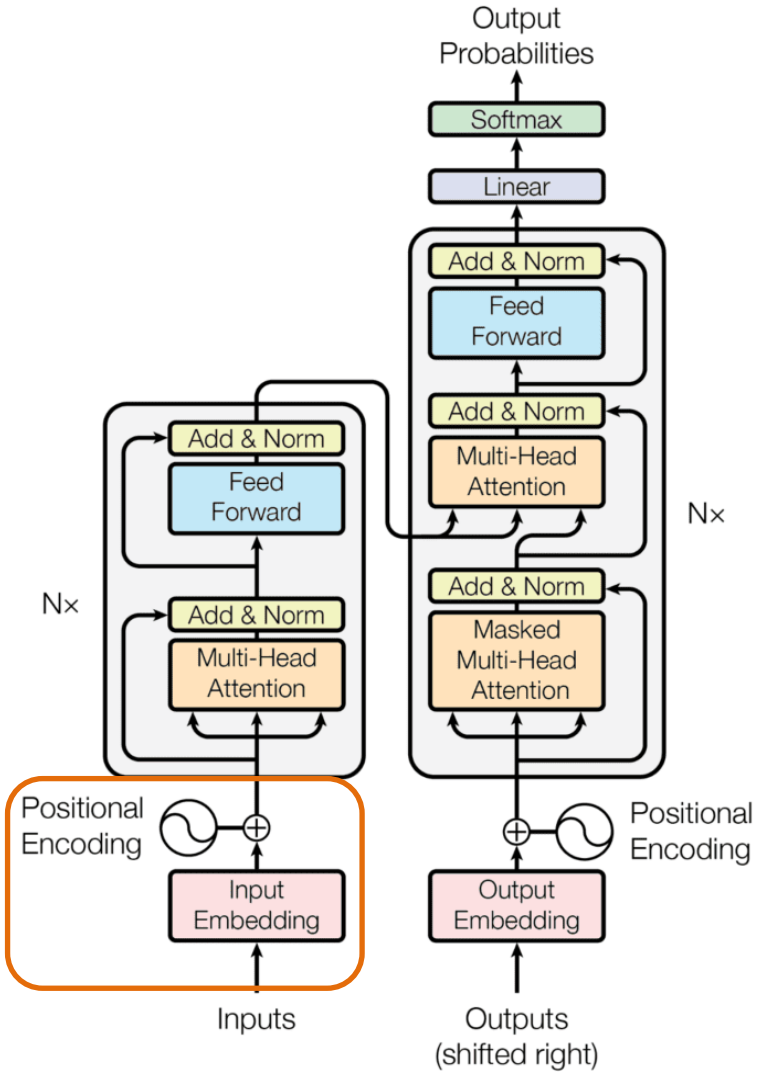
- 인코더는 입력 문장의 의미와 문맥을 파악하여 핵심 정보로 압축, 디코더는 인코더의 핵심 정보를 참고, 이후에 한 토큰씩 예측하여 완성된 문장 출력

Transformer 리뷰

FAIR Lab

KWANGWOON UNIVERSITY

Embedding, Positional Encoding



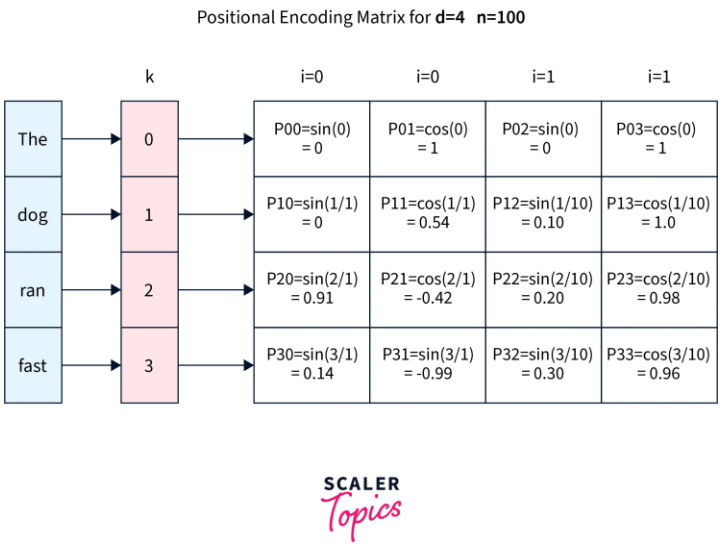
- 입력값이 Input Embedding에 들어가면서 embedding vector로 변환됨
- Positional Encoding을 통해 문장의 순서를 알 수 있도록 Embedding vector에 토큰의 위치 정보를 더해줌

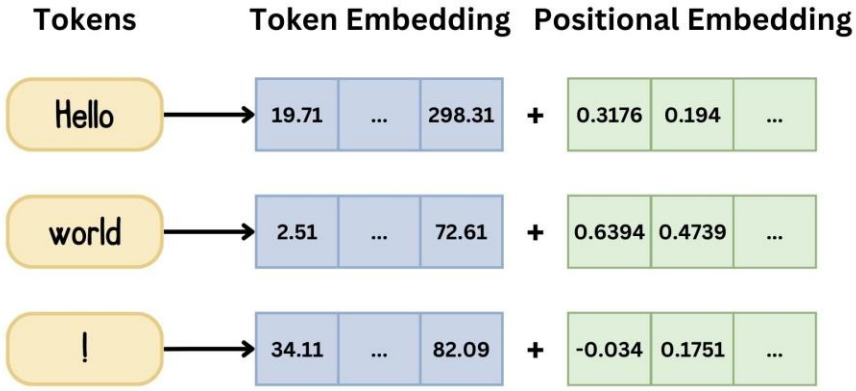
Transformer 리뷰

FAIR Lab

KWANGWOON UNIVERSITY

Positional Encoding, Embedding 차이점





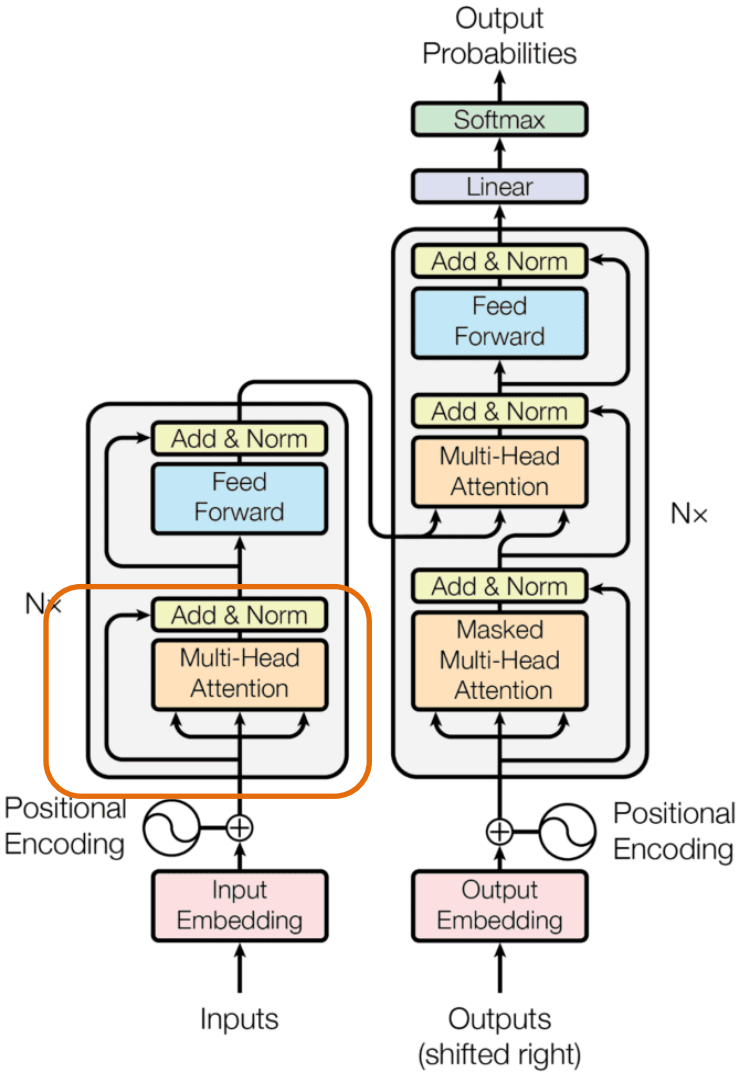
- Positional Encoding은 Transformer 모델에서 입력 시퀀스의 순서를 인식하게 하기 위해 사용하는 방법
- Positional Embedding은 BERT 모델에서 사용하며, 각 위치에 대해 고정된 값을 사용하는 대신 위치 정보를 학습 가능한 임베딩 벡터로 표현

Transformer 리뷰

FAIR Lab

KWANGWOON UNIVERSITY

Multi-Head Attention, Add & Norm



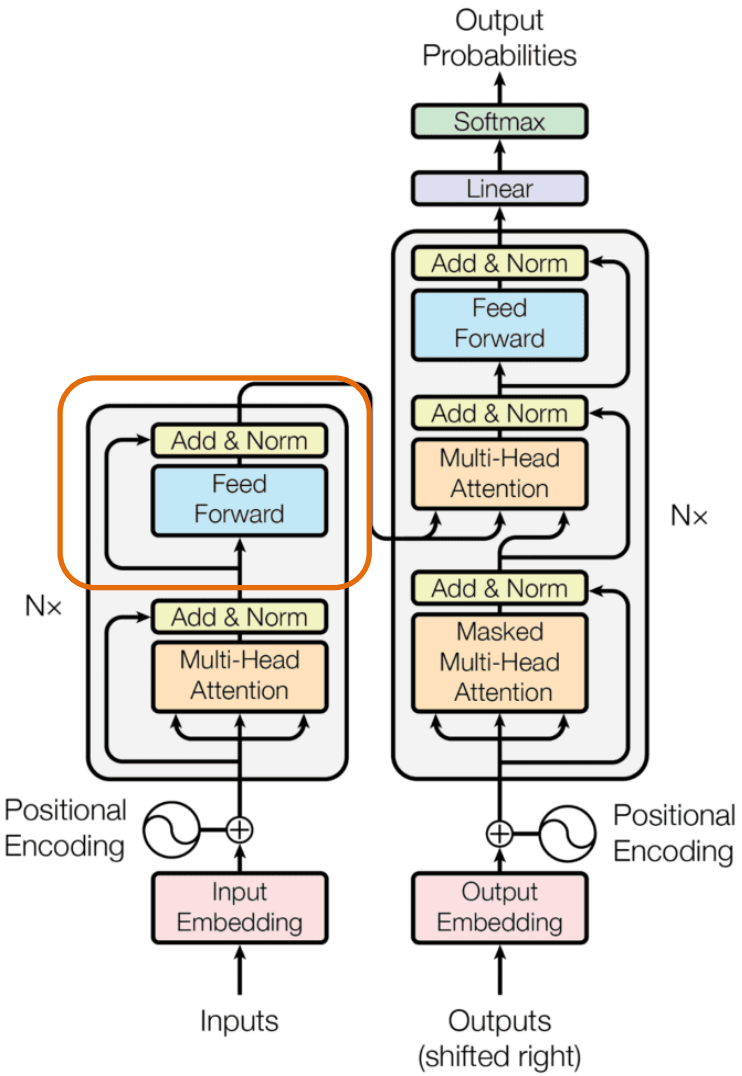
- Multi-Head Attention은 여러 개의 헤드로 Self-Attention이 병렬적으로 동작함
- Add & Norm에서 Attention층 통과 전, 후의 입력 벡터를 합하고 정규화 진행

Transformer 리뷰

FAIR Lab

KWANGWOON UNIVERSITY

Feed Forward 신경망, Add & Norm



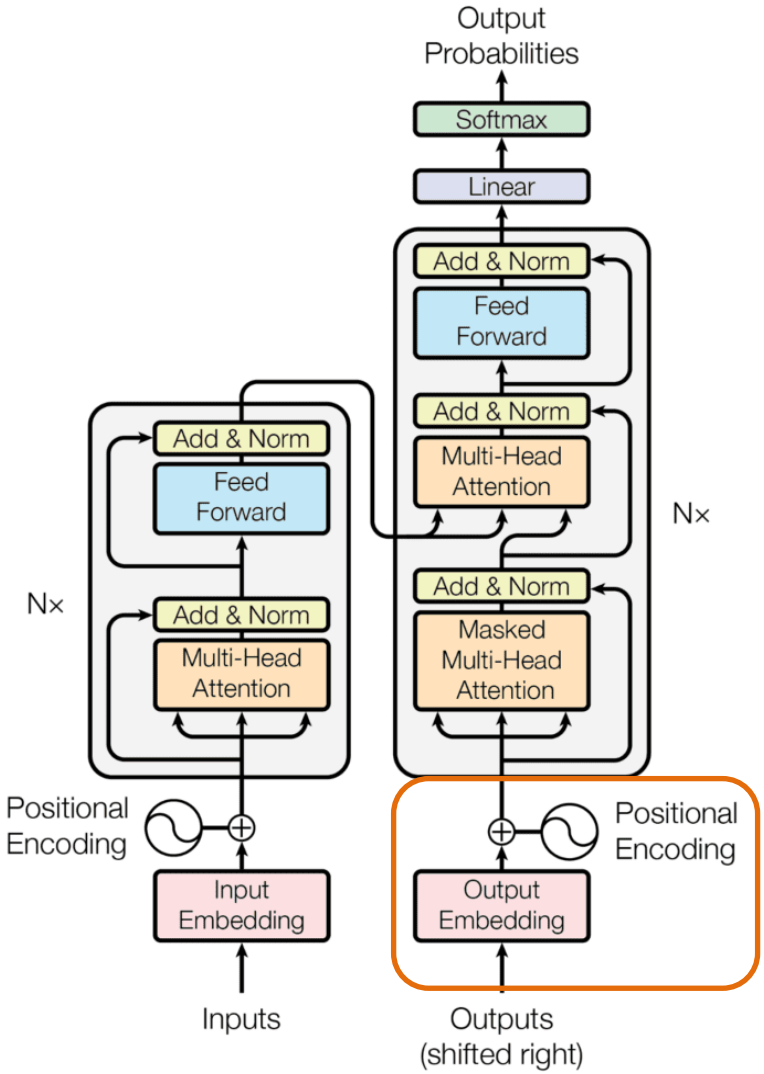
- 피드 포워드 신경망에서 각 벡터를 한 번 더 깊이있게 처리하고 반환함
  이때 비선형성을 추가해 모델의 표현력을 극대화시킴

Transformer 리뷰

FAIR Lab

KWANGWOON UNIVERSITY

Decoder Input



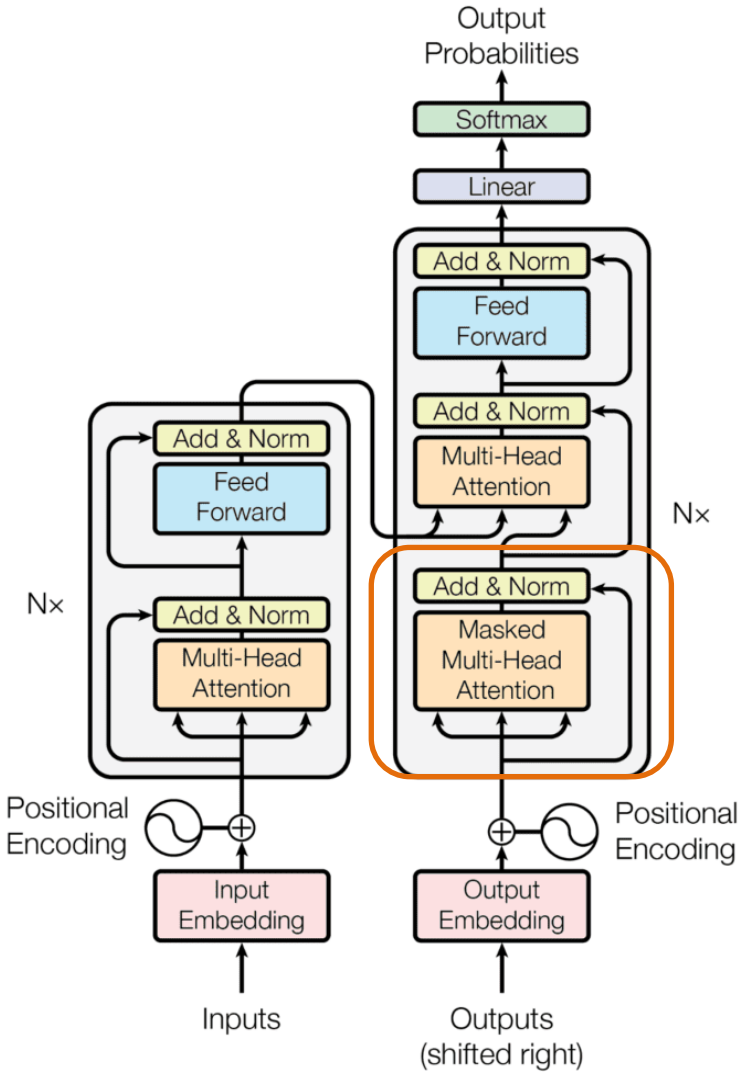
- 입력값이 Input Embedding에 들어가면서 embedding vector로 변환됨
- 이때 입력값은 정답값에서 shifted right된 것이 들어감

Transformer 리뷰

FAIR Lab

KWANGWOON UNIVERSITY

Masked Multi-Head Attention



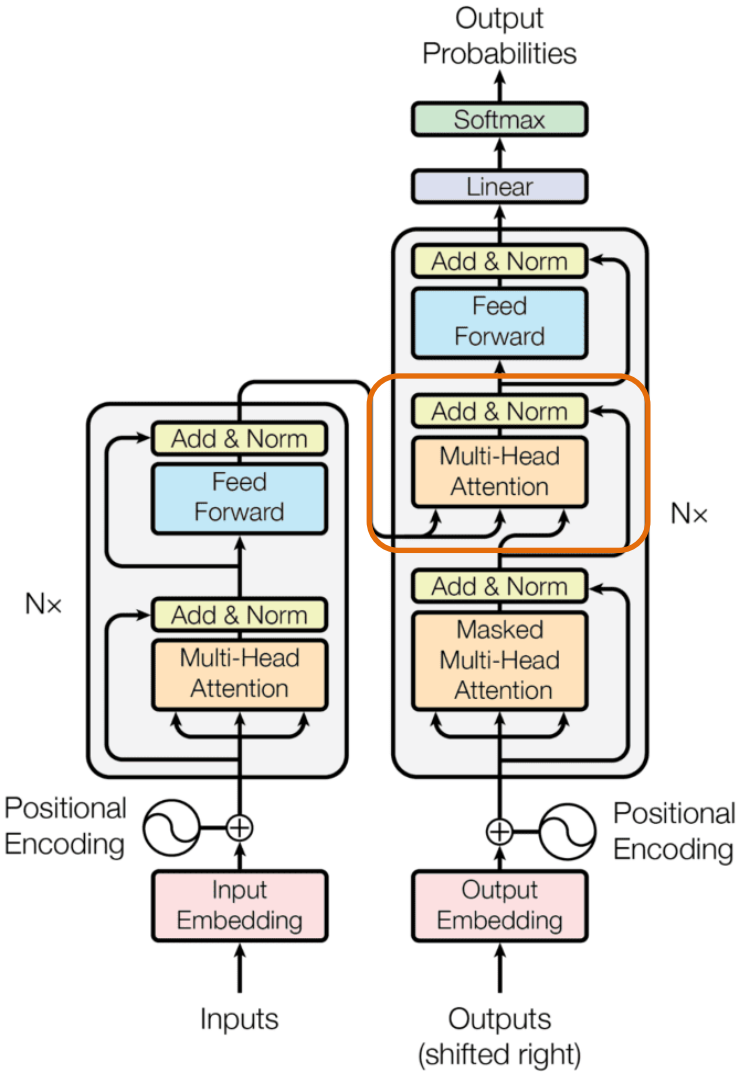
- Masked Multi-Head Attention은 현재 시점 이전의 토큰들만 참고하고 미래 시점 토큰을 참고하지 않도록 마스킹을 사용함

Transformer 리뷰

FAIR Lab

KWANGWOON UNIVERSITY

Multi-Head Cross Attention



- Multi-Head Cross Attention는 인코더에서 나온 최종 출력값 (K, V)와 디코더의 이전 층에서 온 정보(Q)를 함께 사용해 출력의 각 토큰과 입력 문장의 연관성을 파악

- Linear Layer에서 선형 변환으로 차원을 줄여 출력 벡터의 크기를 줄이고, 소프트맥스를 통해 각 토큰의 확률을 계산해 다음 토큰을 예측