Transformer 2

: 전체적으로 논문 리뷰에 초점을 맞췄다.

트랜스포머의 기본구조

코드로 구현한 트랜스포머는 다음의 Class를 만들어 사용한다.

EncoderDecoder, Generator, Encoder, LayerNorm, SublayerConnection, attention, MultiHeadedAttention, EncoderLayer, Decoder, DecoderLayer, PositionwiseFeedForward, Embeddings

EncoderDecoder는 트랜스포머는 물론 다른 모델들에서도 중요한 기본(Base)이다.

Generator는 기본적인 선형 / 소프트맥스 generation 단계를 정의하는 Class이다.

Class는 아니지만, clones 함수를 통해 Nx를 구현한다.

Encoder를 통해 n개 레이어에 대응하는 인코더를 구현한다.

LayerNorm을 통해 layernorm 모듈을 형성한다. 인코더 내의 Add&Norm 단계에 해당한다,

SublayerConnection는 마지막이 아닌 첫번째로 수행하는데, 이는 코드 간소화를 위해서이다.

Attention 은 scaled dot product attention을 수행한다.

Scaled Dot Product Attention을 n개 채널화 하기 위해 MultiHeadedAttention 클래스를 만든다.

위의 Class들을 통해 전체적인 EncoderLayer를 형성하였다.

Decoder class를 통해 n개 layer로 마스킹을 수행한다.

DecoderLayer는 전체적인 DecoderLayrer를 형성ㅎ난다.

Subsequent\_mask를 통해 인풋에 마스크를 씌운다. 이는 attention을 보다 잘 수행하기 위함이다.

PositionwiseFeedForward class를 통해 비선형성을 더한다. Relu함수를 더하였다.

임베딩 class를 통해 encoder와 decoder에 들어가는 input에 임베딩을 수행한다.

이 때, 임베딩의 방법에는 등장 빈도수 ~ 음절과 어절 중간 젖ㅇ도의 단계로 최소 의미쌍을 형성하는 바이트 페어 인코딩, 띄어쓰기를 표지로 하여 유닛을 만드는 wordpiece 등이 있다. 단, wordpiece는 특성상 한국어에는 부적합할 가능성이 크다.

임베딩 시에는 5개의 제한사항이 있다.

시퀀스 길이나 입력값에 관계없이 동일한 위치값, 입력값에 비해 작은 위치값, 너무 빠르지 않은 positional encoding, 위치 차이에 의한 positional encoding 값 차이의 거리 이용 가능성, 위치에 따라 서로 다른 값을 가지는 positional encoding 값

Positional encoding을 수행ㅇ할 때는 순차적으로 투입을 하며, 각 위치(pos)에 대해 사인 및 코사인 트랜스포메이션을 활용한다.

Before Training

학습 없이 Src를 맞추려는 경우, 가중치가 랜덤이라 엉망인 값을 획득함을 볼 수 있었다.

epoch마다 train\_state 및 steps, accu\_step을 업데이트하는 함수들을 만들었다.

논문에서는 adam방식을 사용하였으나, 세션에서는 learning late warmup방식을 이용하였다.

이 방식을 통해 스텝이 높아질수록 learning rate가 낮아짐을 확인할 수 있었다.

학습과정에서 정답일 확률이 지나치게 높게 나오는 문제를 해결하기 위한 코드가 필요하다. 이 코드는 labelSmooting class에 해당한다.

Copy Model

후반부는 코드만 있어서 이해하기가 어려웠습니다…..

노션에 있는 줌 영상 녹강 볼게요…… 죄송합니ㅏㄷ……