Transformer 정리

24기 박정양

Transformer는 자연어 처리 분야에서 많은 성과를 거두고 있는 모델로, 이를 활용해서 GPT, BERT 등 다양한 최신 모델들이 나오고 있습니다. 이러한 모델은 자연어 이해와 생성 분야에서 뛰어난 성능을 보여주고 있습니다. 이 모델의 장점은 입력 데이터를 한 덩어리로 계산하기 때문에 계산 속도가 향상되고, 긴 데이터를 효과적으로 처리하는 것이 가능합니다. 또한 모델의 크기와 복잡성을 쉽게 바꿀 수 있어 scaling이 쉽습니다. 다만, 계산이 복잡하고 parameter의 수가 많으며 처리하는 데이터가 길어질수록 필요한 메모리의 양이 급격히 증가한다는 단점이 있습니다.

Transformer에서는 self-attention을 이용해 병렬화 문제와 long distance dependency 문제를 해결했습니다. Self-attention이란 데이터에서 중요한 부분에 더 많은 가중치를 두는 attention을 구할 때, 데이터 스스로만을 이용해서 중요도를 판단하는 방법입니다. Attention은 Query, Key, Value라는 3개의 matrix를 입력받아서 아래 식을 통해 계산합니다.

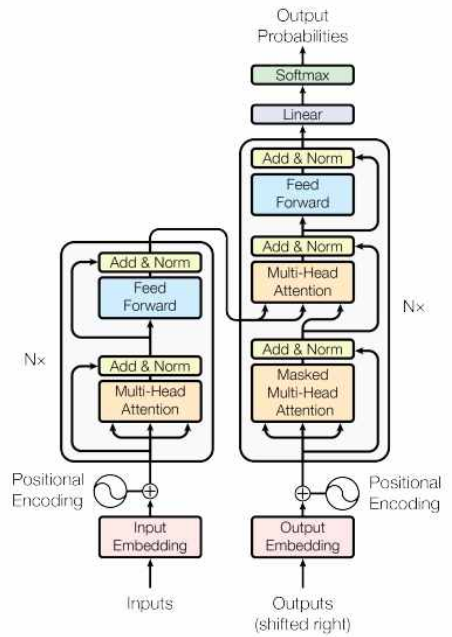
$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

Query와 Key의 유사도를 파악하기 위해 내적을 거치고, key의 차원수 d_k를 이용해 정규화를 진행합니다. 정규화를 거친 값을 softmax 함수에 넣으면 value를 위한 weight를 얻게 됩니다. 이 weight를 value에 곱해주면 우리가 구하고자 하는 attention을 얻을 수 있게 됩니다.

이러한 Attention 계산은 무엇이 중요한지를 파악하는 과정으로, 여러 관점에서 바라본다면 더 정확한 유사도를 파악할 수 있습니다. 따라서 attention을 여러 번 진행하는 방법인 multi-head attention이라는 방법을 이용합니다. multi-head attention은 기존에 입력으로 들어온 key, value, query 행렬을 h번의 projection 연산을 거쳐 h개의 행렬로 만들어서 위에서 설명한 방식으로 h개의 attention을 만들어 합친 것을 최종 출력값으로 이용합니다. Transformer에서는 h=8인 multi-head attention을 이용합니다.

이러한 self-attention은 입력 데이터를 순차적으로 받는 것이 아닌 한번에 받기 때문에 단어의 순서정보를 고려하지 않고 task를 진행합니다. 이러한 점은 단어의 순서가 바뀌면 문장의 뜻이 바뀌거나 아예 틀린 문장이 된다는 점에서 문제가 발생할 수 있습니다. 이러한 문제를 막기 위해 입력 데이터에 순서정보를 만들어주는데, 이 과정을 positional encoding이라고 합니다. 순서정보는 코사인 유사도를 의미하는 아래 식을 통해 계산합니다.

$$f(p) = \begin{cases} \sin\left(\frac{p}{10000^{i/d}}\right) & (i = 2k) \\ \cos\left(\frac{p}{10000^{(i-1)/d}}\right) & (i = 2k+1) \end{cases}$$



Transformer의 전체 architecture은 위의 그림과 같습니다. 왼쪽은 encoder 모듈 6개를 쌓은 encoder, 오른쪽은 decoder 모듈 6개를 쌓은 decoder입니다. Encoder는 입력 데이터를 self-attention 연산을 통해 데이터를 변환합니다. Encoder 모듈은 Multi-head self attention과 feed forward network로 구성되어 있고, sub-layer를 뛰어넘는 residual connection이 존재합니다.

Decoder 모듈은 Masked multi-head self attention, Cross Multi-head attention, feed forward layer로 구성되어 있습니다. Masked attention이란 예측하고자 하는 단어의 앞에 오는 단어들만을 이용해 해당 단어를 예측하는 것으로, 뒤에 오는 단어를 masking한다는 의미입니다. 두번째 오는 cross self attention은 encoder의 결과를 query와 key로, masked attention의 결과 value로 삼아 self attention을 진행합니다. 번역기를 예로 들면, 번역할 문장을 query와 key로, 번역한 문장을 value로 받는 attention이기 때문에 엄격하게 본다면 self-attention은 아닙니다.