1주차 스터디

텍스트	Attention Is All You Need

참고자료

> 💡 The Illustrated Transformer – Jay Alammar – 한 번에 하나의 개념으로 기계 학습을 시각화합니다.

Introduction

> In this work we propose the Transformer, a model architecture eschewing recurrence and instead relying entirely on an attention mechanism to draw global dependencies between input and output.
> The Transformer allows for significantly more parallelization and can reach a new state of the art in translation quality after being trained for as little as twelve hours on eight P100 GPUs

반복을 피하고 대신 입력과 출력 사이의 전역 종속성을 그리기 위해 주의 메커니즘에 전적으로 의존하는 모델 아키텍처인 Transforme

기존 **순환 신경망(RNN)** 기반 모델, 특히 **LSTM** 및 **GRU**는 자연어 처리(NLP)에서 중요한 역할을 했지만, 순차적 연산으로 인해 병렬화가 어려운 한계가 있었다. 이를 극복하기 위해 **어텐션 메커니즘**이 활용되었으나, 대부분 RNN과 결합된 형태였다.

Transformer는 **순환 구조 없이 어텐션 메커니즘만으로** 문장 내 단어 간 **전역적(dependency-aware)** 관계를 모델링한다. 이로 인해 **병렬 처리 성능이 향상**되며, **8개의 P100 GPU에서 단 12시간 훈련만으로도** 기존 모델을 뛰어넘는 번역 성능을 달성할 수 있다.

Background

ConvS2S의 경우 선형으로, ByteNet의 경우 로그로. 이로 인해 먼 위치 간의 종속성을 학습하기가 더 어려워집니다[12].

트랜스포머에서 이것은 일정한 작동 수로 감소하지만, 평균 어텐션 가중 위치로 인해 유효 해상도가 감소하는 대가를 치르지만, 섹션 3.2에서 설명한 대로 Multi-Head Attention으로 상쇄되는 효과입니다.

> **End-to-end memory networks** are based on a recurrent attention mechanism (instead of sequence aligned recurrence) and have been shown to perform well on simple-language question answering and language modeling tasks

끝 부분을 보고 판단하는 메커니즘이 더 효율적인 결과를 나타냈다.

Transformer는 시퀀스 정렬 RNN 또는 컨볼루션을 사용하지 않고 입력 및 출력의 표현을 계산하기 위해 전적으로 self-attention에 의존하는 최초의 transduction 모델입니다. 다음 섹션에서는 Transformer에 대해 설명하고, 자기 주의를 유발하고, [17, 18] 및 [9]와 같은 모델에 비해 Transformer의 장점에 대해 설명합니다.

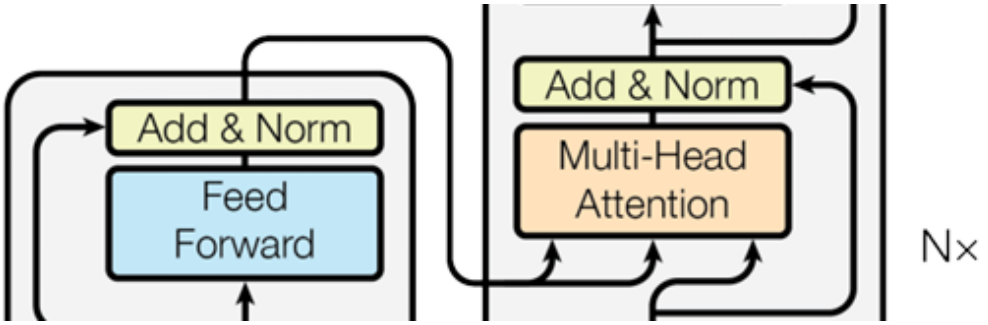
Model Architecture

encoder maps an input sequence of symbol representations (x1,...,xn) to a sequence
of continuous representations z = (z1,...,zn).

z가 주어지면 디코더는 한 번에 한 요소씩 기호의 출력 시퀀스(y1,...,ym)를 생성합니다. 각 단계에서 모델은 자동 회귀 [10]되어 이전에 생성된 심볼을 다음 단계를 생성할 때 추가 입력으로 사용합니다.

인코더: 인코더는 N = 6개의 동일한 레이어 스택으로 구성됩니다. 각 레이어에는 두 개의 하위 레이어가 있습니다. 첫 번째는 다중 헤드 셀프 어텐션 메커니즘이고 두 번째는 간단하고 위치에 따라 완전히 연결된 피드 포워드 네트워크입니다. 두 하위 레이어 각각에 잔류 연결 [11]을 사용한 다음 레이어 정규화[1]를 사용합니다. 즉, 각 하위 계층의 출력은 LayerNorm(x + Sublayer(x))이며, 여기서 Sublayer(x)는 하위 계층 자체에 의해 구현된 함수입니다. 이러한 잔차 연결을 용이하게 하기 위해 모델의 모든 하위 계층과 임베딩 계층은 차원 dmodel = 512의 출력을 생성합니다.

디코더: 디코더는 N = 6개의 동일한 레이어 스택으로 구성됩니다. 각 인코더 계층에 있는 두 개의 하위 계층 외에도 디코더는 인코더 스택의 출력에 대해 다중 헤드 어텐션을 수행하는 세 번째 하위 계층을 삽입합니다.



인코더와 유사하게, 각 하위 레이어 주위에 잔류 연결을 사용한 다음 레이어 정규화를 사용합니다. 또한 디코더 스택의 self-attention 하위 계층을 수정하여 위치가 후속 위치에 주의를 기울이는 것을 방지합니다. 이 마스킹은 출력 임베딩이 한 위치만큼 오프셋된다는 사실과 결합되어 위치 i에 대한 예측이 i보다 작은 위치에서 알려진 출력에만 의존할 수 있도록 합니다

1. Transformer 구조 (Encoder / Decoder)
2. Attention 종류 (Self-Attention, Multi-head Attention)

Self-attention은 트랜스포머가 다른 관련 단어의 "이해"를 현재 처리하고있는 단어에 굽는데 사용하는 방법입니다.

이 논문은 "다중 머리" 주의(multi-headed) 주의(attention)라는 메커니즘을 추가하여 자기 주의 계층을 더욱 개선했습니다. 이는 다른 위치에 초점을 맞출 수 있는 모델의 능력을 확장합니다. Self-attetion은 실제 단어 자체에 의해 지배 될 수 있습니다. "The animal didn't cross the street because it was too tired"와 같은 문장을 번역하는 경우 "it"이 어떤 단어를 가리키는지 아는 것이 유용할 것입니다.

어텐션 레이어에 여러 개의 "표현 하위 공간"을 제공합니다. 다음에 살펴보겠지만, 다중 헤드 어텐션을 사용하면 하나뿐만 아니라 여러 세트의 쿼리/키/값 가중치 매트릭스가 있습니다 (트랜스포머는 8개의 어텐션 헤드를 사용하므로 각 인코더/디코더에 대해 8개의 세트로 끝납니다). 이러한 각 집합은 임의로 초기화됩니다. 그런 다음 훈련 후 각 세트를 사용하여 입력 임베딩(또는 하위 인코더/디코더의 벡터)을 다른 표현 하위 공간에 투영합니다.

3. Positional Encoding