그룹명 : 66기 비전팀
(7)차 주간보고서

활동 현황

작성자	여윤기	장소	오프라인
모임일자	2022년 11월 6일 일요일	모임시간	22:00 ~ 23:00 (총 60분)
참석자	여윤기, 이근호, 정성실	결석자	없음

학습 내용

학습주제 및 목표

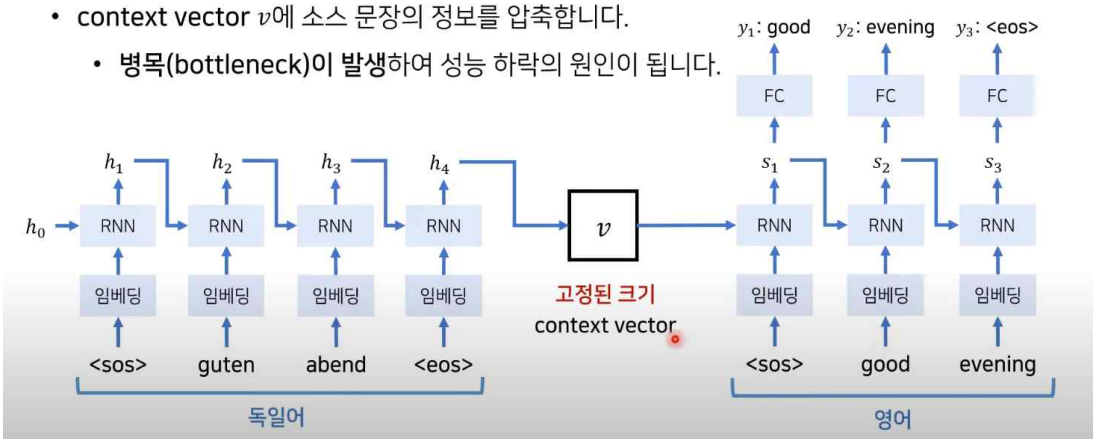
자연어 처리 모델(번역, 챗봇 등)을 학습하기 위해 관련 선행연구인 Transformer - Attention is all you need 논문을 읽고 논문의 핵심 내용과 Transformer 모델 이전 자연어 분야에서 많이 사용된 RNN 모델의 작동 원리에 대해 학습하였음.

학습내용

◇ 기존의 RNN/CNN 모델을 사용하지 않고 Attention 기법을 활용한 인코더-디코더 구조로 설계된 Transformer 모델의 동작원리 학습

< 기존 RNN 모델의 문제점 >

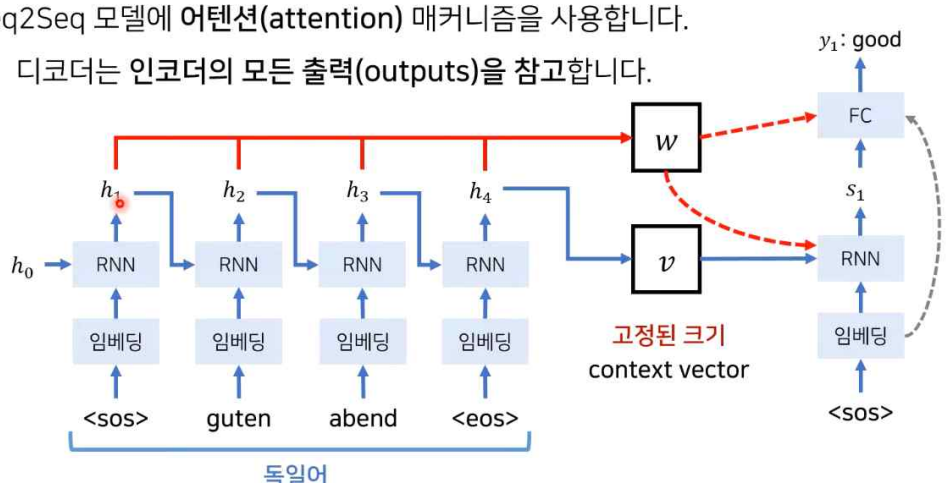
- context vector v에 소스 문장의 정보를 압축합니다.
  - **병목(bottleneck)이 발생**하여 성능 하락의 원인이 됩니다.



• 위 그림과 같이 입력이 들어올 때 마다 hidden state 값을 순서대로 갱신하여, 최종 단어가 들어왔을 때 마지막으로 갱신된 hidden state(위 그림에서는 h4)가 전체 문장의 의미를 함축하고 있으며 이를 고정된 크기를 갖는 context vector화 하여 해당 vector를 참고하여 순서대로 output을 출력하게 된다. 이때 하나의 context vector가 source 문장의 모든 정보를 가지고 있어야 하므로 계산 시 병렬 계산이 불가하고 bottleneck 문제 등이 발생할 수 있으며 이는 성능저하의 원인이 된다.
Transformer 논문에서는 매번 source 문장에서의 출력 전부를 입력으로 받는, Seq2Seq 모델에 Attention 매커니즘을 사용한 기법을 제안하였다.

< Attention 매커니즘 작동원리 >

- Seq2Seq 모델에 **어텐션(attention)** 매커니즘을 사용합니다.
  - 디코더는 **인코더의 모든 출력(outputs)을 참고**합니다.

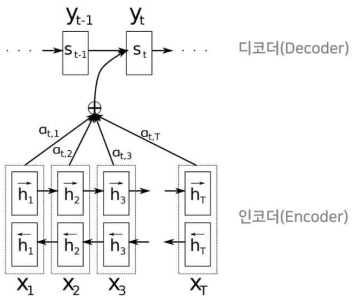


• 기존 Seq2Seq 구조에서 source 문장이 압축된 고정된 크기의 context vector만 참고하는 것이 아닌, source 문장에서 출력된 모든 hidden state 값들을 합한 weight sum 값을 참고하여 최종 output을 출력한다. 이러한 방식으로 기존 Seq2Seq 구조에서 디코더는 인코더의 모든 출력을 참고하게 된다.

- 에너지(Energy) $e_{ij} = a(s_{i-1}, h_j)$
- 가중치(Weight) $\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})}$

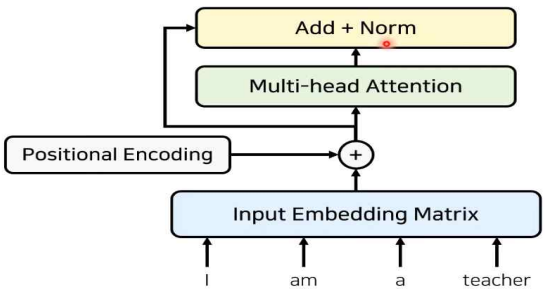
Weighted sum 이용

$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j$$



• 가중치 값은 위와 같이 soucre 문장의 단어별 에너지값들을 softmax 함수에 넣어 단어별로 어떤 단어와 가장 연관성이 있는지를 수치화 한 확률값이며, Weighted sum 값은 가중치와 단어별 각각의 hidden state 값들을 곱한것을 모두 더해준 값이다. Weight sum 값은 디코더의 input이 되어 Attention 가중치를 사용해 디코더의 각 출력이 soucre 문장의 어떤 입력정보를 참고했는지 알 수 있다.

< 인코더 part >



• Transformer 논문에서는 Attention 기법만을 활용하며 RNN과 CNN 기법을 전혀 사용하지 않는다.
• 문장 내 단어의 순서에 대한 정보를 주기 위해 positional encoding을 활용한다. 한 문장 내에서 어떤 단어가 앞에 위치하고, 어떤 단어가 뒤에 위치하는지에 대한 위치 정보를 포함하여 Embedding

한다.

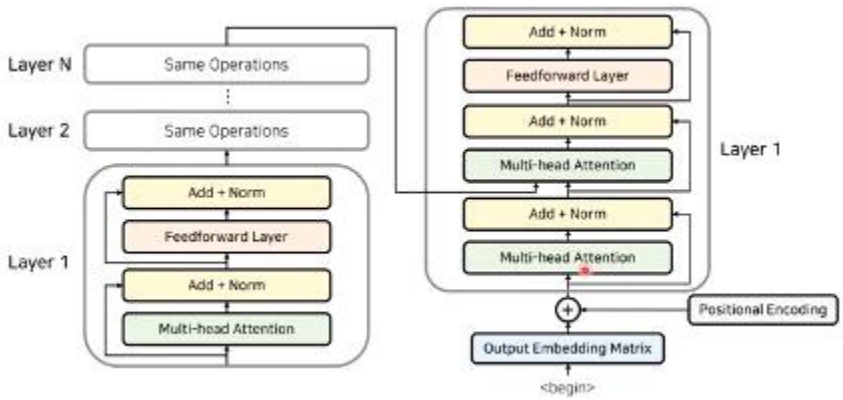
• Positional Encoding 이후 Attention을 수행한다. Source 문장의 각각의 단어가 서로에게 어떤 연관성이 있는지를 Attention score를 통해 수치화하여 구하며, 이를 통해 source 문장에 대한 전반적인 문맥적 의미를 학습할 수 있다.

• 이후, ResNet 에서 제안되었던 방법과 동일하게 성능향상을 위해 특정 layer를 건너뛰고 바로 다음 layer의 입력으로 Skip connection을 추가해주는 Residual Learning 기법을 사용하였다.

• 최종적으로 인코더에서는 Attention 수행 결과와 Residual learning 기법으로 추가한 값을 더해서 Normalize 까지 수행한 후에 결과를 출력하게 된다. 이것이 인코더에서 1개 layer 의 동작 매커니즘이며, 여러 개의 동일한 layer를 중첩시켜 Attention과 Normalization 과정을 반복한다.

  * 이때 각 layer 마다 서로 다른 parameter를 가짐

< 디코더 part >



• 인코더 파트의 가장 마지막 layer에서 나온 출력값은 그대로 디코더 파트의 입력으로 들어가게 된다. 이는 디코더 파트에서 문장 출력 시에 입력 source 문장의 어떤 단어에 중점을 가져야 하는지 알려주기 위함이다. 디코더 또한 단어 정보를 받아 위치정보를 추가하기 위해 Positional Encoding 과정을 거친다.

• 디코더에서 첫번째 Attention 파트는 인코더에서 사용된 Attention 과정과 동일한 Attention 으로, 각각의 단어들이 서로에게 어떤 의미를 주는지 학습하여 출력되는 문장에 대한 전반적 문맥 의미를 학습한다.

• 디코더의 두번째 Attention 에서는 인코더에 대한 정보를 Attention 한다. 인코더 파트에서 마지막 layer의 출력값을 그대로 입력으로 받아, 디코더에서 문장을 출력할 때 source 문장으로부터 인코더 파트에서 Attention 수행한 값을 바탕으로 출력 문장이 source 문장의 어떤 단어와 연관성이 있는지 확인한다.

• 디코더 또한 위와 같이 동작하는 여러 개의 layer를 중첩시키는데, 본 Transformer 논문에서는 인코더의 마지막 layer 출력이 모든 디코더 layer의 입력값으로 들어가게 설계하였음. 또한 일반적으로 인코더의 layer 수와 디코더의 layer 수는 동일하게 맞춰준다.

• 이러한 인코더-디코더 구조의 Seq2Seq 구조에서 Attention 기법을 사용하는 원리로, Transformer 에서는 source 문장의 단어를 한꺼번에 받아 병렬적으로 hidden state & Attention score를 구해줌으로써 번역 문제에서 기존 RNN을 사용하였을 때 보다 속도와 계산 복잡도가 훨씬 줄어들며 정확도를 높일 수 있었다.

그룹 운영 기록사항

-

다음 모임 계획

모임일자	2022년 11월 13일 일요일	모임시간	10:00~12:00 (총 120 분)
역할분담	Transformer 모델을 학습시킬 데이터셋과 모델 학습 진행.	장 소	오프라인