Explore Transformers

트랜스포머(transformer)는 2017년 구글이 제안한 시퀀스-투-시퀀스(sequence-to-sequence) 모델이다. 최근 NLP에서는 BERT나 GPT 같은 트랜스포머 기반 언어모델이 각광받고 있다. 그 만큼 성능이 좋기 때문이다. 그 좋은 성능의 핵심 동작 원리를 탐구해보자.

시퀀스-투-시퀀스

트랜스포머(Transformer)란 **기계 번역(machine translation)** 등 **시퀀스-투-시퀀스(sequence-to-sequence)**같은 과제를 수행하기 위한 모델입니다. 여기에서 시퀀스란 단어(word) 같은 무언가(something)의 나열을 의미하는데요. **시퀀스-투-시퀀스는** 특정 속성을 지닌 시퀀스를 다른 속성의 시퀀스로 변환하는 작업을 가리킵니다.

기계 번역을 예시로 시퀀스-투-시퀀스가 어떤 태스크인지 알아보자. 기계 번역이란 어떤 언어(소스 언어, source language)의 단어 시퀀스를 다른 언어(대상 언어, target language)의 단어 시퀀스로 변환하는 과제이다. 예를 들어 한글로 된 문장의 소스 시퀀스 길이(단어의 수)와 영어로 된 문장의 소스 시퀀스 길이(단어의 수) 다르다. 그렇기에 시퀀스-투-시퀀스는 Target의 길이가 달라도 해당 과제를 수행하는데 문제가 없어야 한다.

인코더와 디코더

트랜스포머 모델은 시퀀스-투-시퀀스 과제수행에 특화된 모델이다.

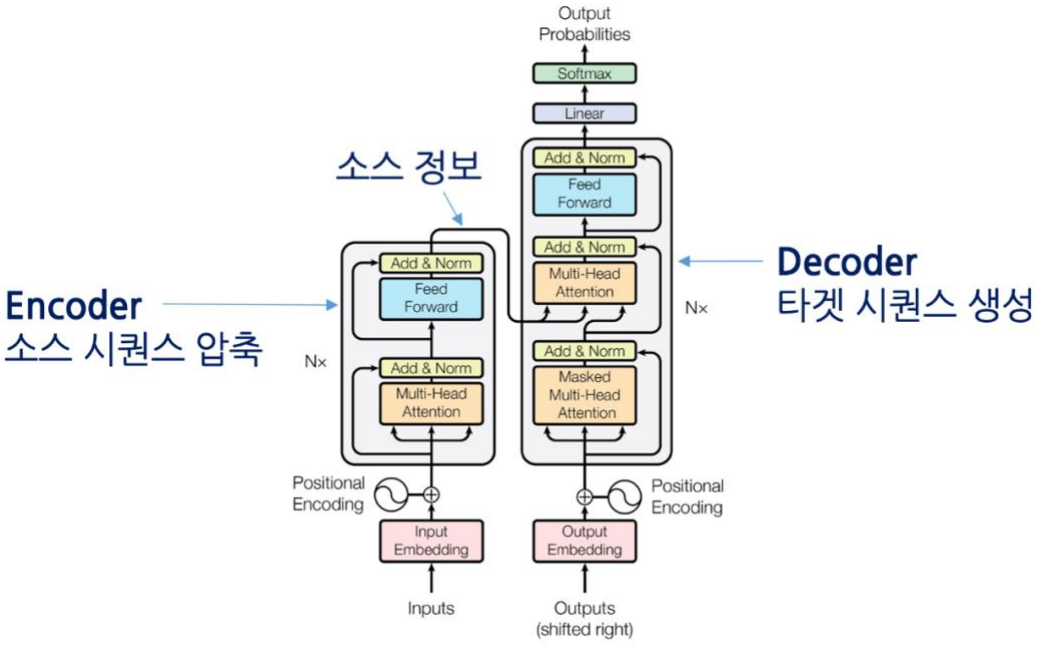
임의의 시퀀스를 해당 시퀀스와 속성이 다른 시퀀스로 변환하는 작업이라면 꼭 기계 번역이 아니더라도 수행할 수 있습니다.

시퀀스-투-시퀀스 과제를 수행하는 모델은 대개 **인코더(encoder)**와 **디코더(decoder)** 두 개 파트로 구성된다.

인코더는 소스 시퀀스의 정보를 압축해 디코더로 보내주는 역할을 담당한다.

인코더가 소스 시퀀스 정보를 압축하는 과정을 **인코딩(encoding)**이라고 합니다. 그리고 디코더는 인코더가 보내준 소스 시퀀스 정보를 받아서 타깃 시퀀스를 생성합니다. 디코더가 타겟 시퀀스를 생성하는 과정을 **디코딩(decoding)**이라고 한다.

트랜스포머 역시 **인코더-디코더** 구조를 따른다.
트랜스포머의 구조는 다음과 같다.



인코더 입력은 소스 전체 시퀀스 이고, 디코더 입력은 타킷 시퀀스를 시작하는 스페셜 토큰으로 시작된다. 인코더는 다음 시퀀스를 압축해 디코더로 보내고, 디코더는 인코더에서 보내온 정보와 현재 디코더 입력을 모두 감안해 다음 토큰을 맞힙니다.
트랜스포머의 최종 출력, 즉 디코더 출력(Output Probabilities)은 타깃 언어의 어휘 수만큼의 차원으로 구성된 벡터(vector)입니다. 이 벡터의 특징은 요솟(element)값이 모두 확률이라는 점입니다.
예를 들어 타깃 언어의 어휘가 총 3만개라고 가정해 보면 디코더 출력은 3만 차원의 벡터이다. 이 벡터의 요솟값 3만 개는 각각은 확률이므로 0 이상 1 이하의 값을 가지며 모두 더하면 1이 된다는 의미이다.

트랜스포머의 학습(train)은 인코더와 디코더 입력이 주어졌을 때 정답에 해당하는 단어의 확률 값을 높이는 방식으로 수행됩니다. 모델은 이번 시점의 정답 단어 해당하는 확률은 높이고 나머지 단어의 확률은 낮아지도록, 모델 전체를 갱신한다.

RNN 은 단어 시퀀스를 한 언어에서 다른 언어로 매핑하는 기계 번역시스템을 개방할 때 중요한 역활을 했습니다. 이런 종루의 작업은 대게 인코더-디코더 또는 **시퀀스-투-시퀀스** 구조로 처리하며 입력과 출력이 임의의 길이를 가진 시퀀스일 때 잘 맞습니다. 인코더는 입력 시퀀스의 정보를 **"마지막 은닉 상태(last hidden state)**라고도 부르는 수치 표현으로 인코딩 합니다. 그 다음 디코더로 전달되어 출력 시퀀스가 생성된다.

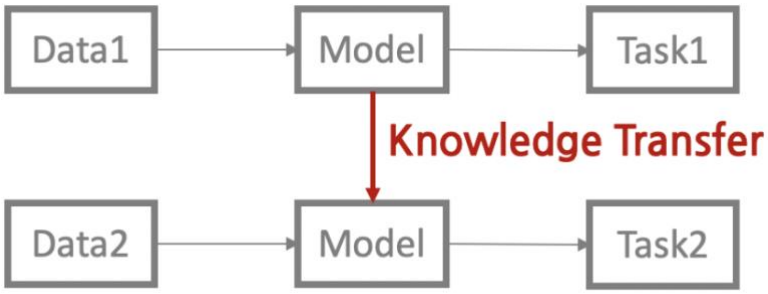
하지만 디코더는 인코더의 마지막 은닉 상태만을 참조해 출력을 만드므로 여기에 전에 입력 시퀀스의 의미가 담겨야한다. 시퀀스가 긴 경우 모든 것을 고정된 하나 표현으로 압축하는 과정에서 시작부분의 정보가 손실된 가능성이 있어 취약하다. 그렇지만 디코가 인코더의 모든 은닉 상태에 접근해 이런 정보 병목현상을 제거한다. 이런 일반적인 메커니즘을 **어텐션(attention)**이라고 한다.

어텐션

어텐션은 입력 시퀀스에서 은닉 상태를 만들지 않고 스텝마다 인코더에서 디코더가 참고할 은닉 상태를 출력한다는 주요 개념에 기초한다. 그렇지만 모든 상태를 동시에 사용하려면 디코더에 많은 입력이 발생하므로 어떤 상태를 먼저 사용할지 우선순위를 정하는 메커니즘이 필요하다. 이로 인해 **어텐션**이 등장한다.
디코더가 모든 타임스켐(time step)마다 인코더의 각 상태에 다른 '어텐션'을 할당한다.

Transfer learning

computer vision 에서는 전이 학습을 사용해 ResNet 같은 합성곱 신경망을 한 작업에서 훈련한 다음 새로운 작업에 적용하거나 fine-tuning 하는 일이 많다. 이를 통해 신경망은 원래 작업에서 학습한 지식을 사용한다.
트랜스퍼 러닝(transfer learning)이란 특정 태스크를 학습한 모델을 다른 태스크 수행에 재사용하는 기법을 가리킵니다. 비유하자면 사람이 새로운 지식을 배울 때 그가 평생 쌓아왔던 지식을 요긴하게 다시 써먹는 것과 같다.



트랜스퍼 러닝을 적용하면 기존 모델 보다 모델의 학습 속도가 빨라지고 새로운 태스크를 더 잘 수행하는 경향이 있습니다. 이 때문에 트랜스퍼 러닝은 최근 널리 쓰이고 있습니다. BERT(Bidirectional Encoder Representations from Transformers)나 GPT(Generative Pre-trained Transformer) 등이 바로 이 기법을 쓰고 있다.
구조적으로 볼때 모델은 바디와 헤드로 나뉘는데 바디의 가중치는 훈련하는 동안 원래 도메인에서 다양한 특성을 학습하고 이 가중치를 사영해 새로운 작업을 위한 모델을 초기화합니다. 전통적인 지도 학습(supervised learning)과 비교하면, 전이 학습은 일반적으로 다양한 작업에서 적은 양의 레이블을 데이터로 훨씬 효과적으로 훈련하는 높은 품질의 모델을 만든다.

사전 훈련

초기 훈련의 목표는 이전 단어를 바탕으로 다음 단어를 예측하는 것이다.
이 작업을 언어 모델링(Language modeling)이라고 한다. 이 작업은 레이블링 된 데이터가 필요하지 않으며 풍부한 텍스트를 활용한다.

도메인 적응

언어 모델을 대규모 말뭉치에서 사전 훈련한 후, 다음 단계로 도메인 내 말뭉치에 적응시킨다. 이 단계에서 여전히 언어 모델링을 사용하지만 이제 모델은 타깃 말뭉치에 있는 다음 단어를 예측한다.

Fine-tuning

이 단계에서는 언어 모델을 타깃 작업을 위한 분류 층과 함께 미세튜닝합니다.

References

"트랜스머를 활용한 자연어 처리" - 루이스 턴스톨, 레안드로 폰 베라, 토마스 울프
"Do it! BERT 와 GPT 로 배우는 자연어 처리" – 이기창