BERT 연구 자료

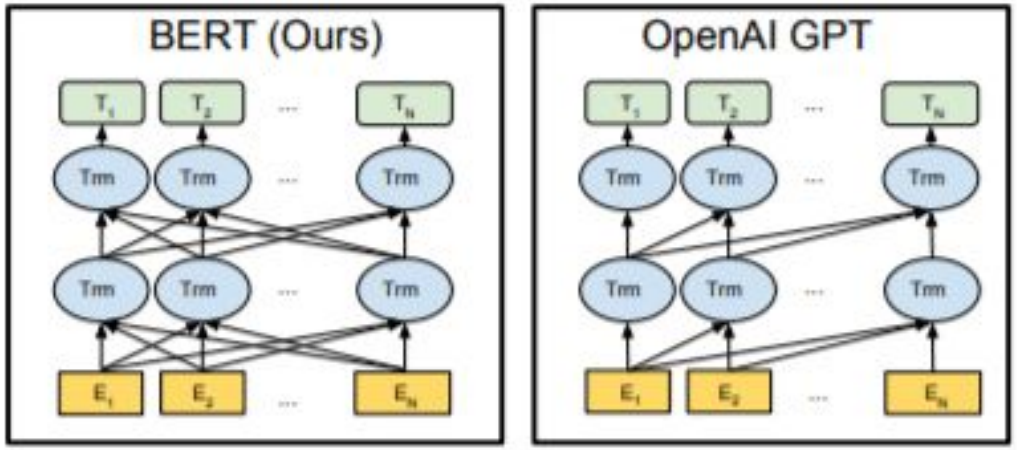
김영석

목차

BERT 무엇인가?

- 토큰을 벡터로 바꾸기 위해서 사용되는 TRANSFORMER 기반의 Pretrain 방법들중 하나

- BERT는 Bidirectional한 구조를 가지고 있다.



TRANSFORMER 무엇인가?

- Google에서 발표한 Attention is all you need 라는 논문에서 제안한 모델 (인용수 약 6000번)

- BERT는 TRANSFORMER의 인코더와 디코더중 인코더만 사용한다.

- 기계번역에서 RNN과 CNN이 가진 문제점을 해결하기 위해 나왔다.

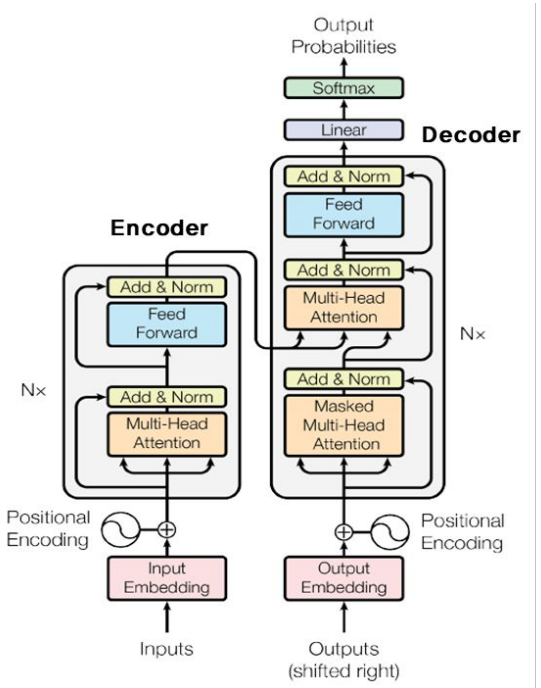
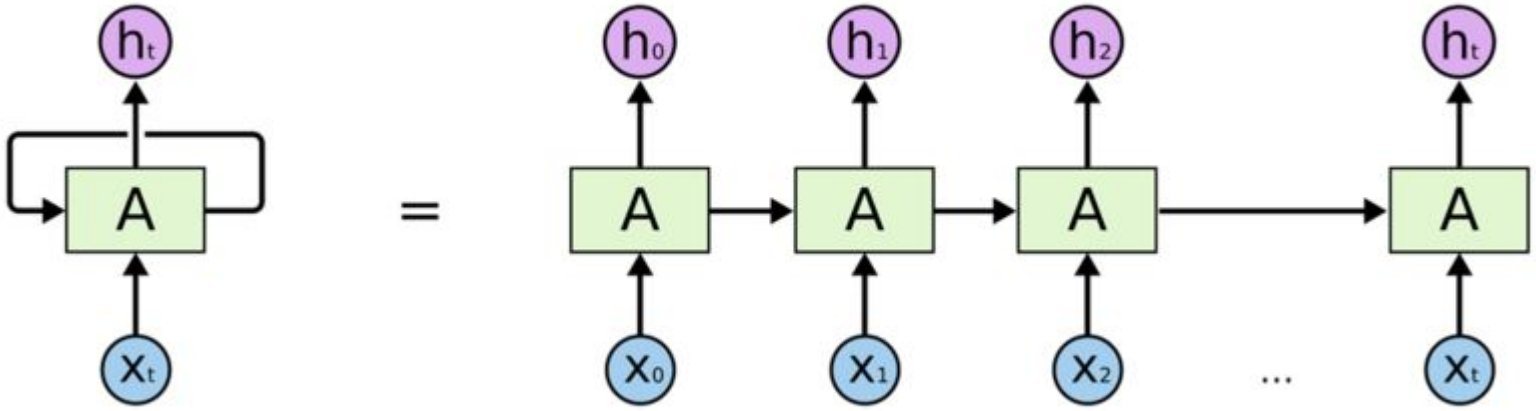


Figure 1: The Transformer - model architecture.

왜 TRANSFORMER가 나오게 됐나? - RNN이 가지고 있던 문제 (병렬처리 안됨)

만약 Input data (X1, X2, … Xt)에 대한 hidden state 값을 계산하려고 한다면

이전 rnn cell의 hidden state값이 참조 되어야 한다.

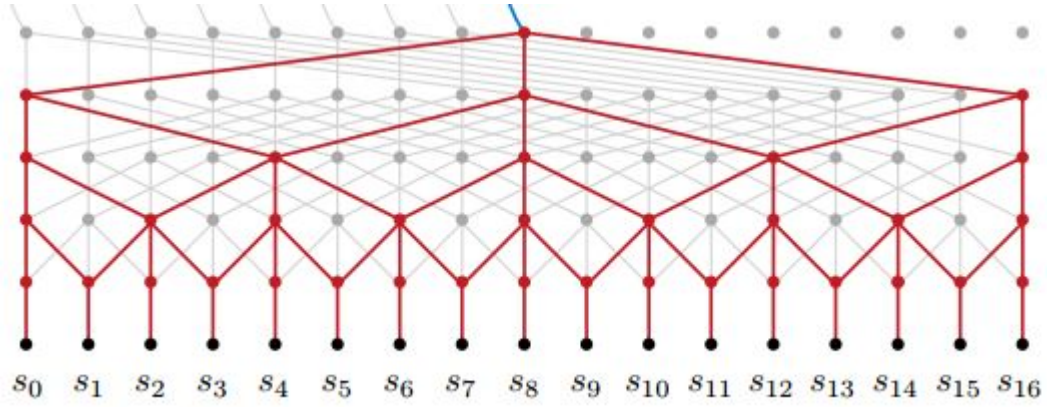


왜 TRANSFORMER가 나오게 됐나? - CNN이 가지고 있던 문제

RNN이 가지고 있던 문제 때문에 Google과 Facebook은

CNN을 기반으로 한 기계번역 모델을 발표하게 된다. (ByteNet, ConvS2S, Extended Neural GPU)

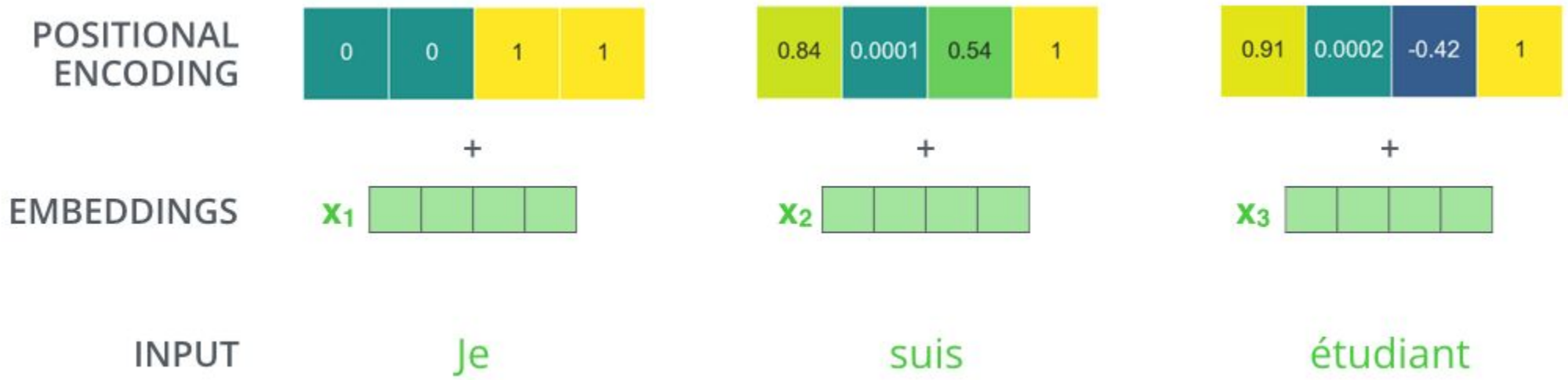
하지만 Gradient vanishing / exploding 문제가 발생하게 된다.



TRANSFORMER - INPUT EMBBEDING , POSITIONAL ENCODING

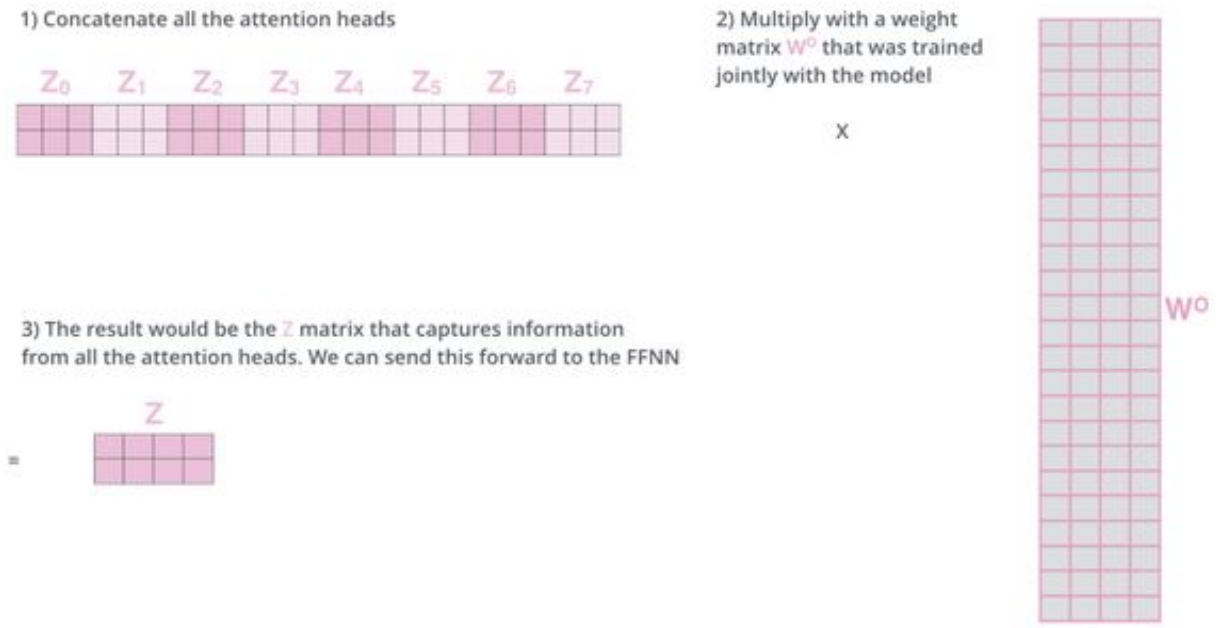
- 모델이 토큰에 대해 여러가지 계산을 하기 위해서 input embedding이 필요하다

- 해당 토큰에 대한 position값을 input embedding에 넣기 위해서 positional encoding이 필요하다



TRANSFORMER - MULTI HEAD ATTENTION

- 토큰에 대해서 Self attention 과정을 여러번 진행한것



TRANSFORMER - SELF ATTENTION

- 하나의 토큰에 문맥 정보를 반영하기 위해서 진행하는 과정

- Convolution은 kernal size 만큼 데이터에서 feature을 추출 할 수 있다.

- self-attention은 더 넓은 범위에서 feature을 추출 할 수 있다.

TRANSFORMER - POSITION WISE FEED FORWARD

- 토큰 하나에 multi head attention이 적용된 값이 feed forward neural net에 들어가게 된다.

- position마다 feed forward neural net이 적용되서 position wise이다

BERT의 학습방법

Masked LM
- bert가 양방향성을 가진 모델이 될 수 있게 하는 학습 방식
- 토큰에 masking을 해서 모델이 masking한 토큰을 맞추게 한다.

Next Sentence Prediction
- 두 문장 사이의 관계를 학습하는데 도움을 준다. (Ex. Question anwsering)

BERT의 FINE TUNING 방법

Pretrain BERT에 Fully Connected Layer 연결

```
#question anwsering task를 수행하는 Layer#
####################fully connected layer####################
logits = tf.matmul(final_hidden_matrix, output_weights, transpose_b=True)
logits = tf.nn.bias_add(logits, output_bias)
logits = tf.reshape(logits, [batch_size, seq_length, 2])
logits = tf.transpose(logits, [2, 0, 1])

unstacked_logits = tf.unstack(logits, axis=0)

(start_logits, end_logits) = (unstacked_logits[0], unstacked_logits[1])
####################fully connected layer####################
return (start_logits, end_logits)
```