Transformer 2022-03-07 수학과 고승규

No.

Attention → linear combination → skip gram

$$x = (x \cdot v_1)v_1 + (x \cdot v_2)v_2 + \cdots + (x \cdot v_n)v_n$$

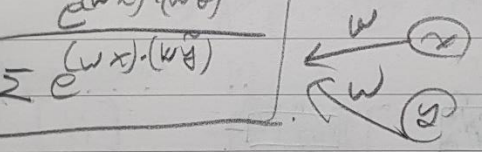
$$\left[(W_Q x)\cdot(W_K v_1)\right](W_V v_1) + \cdots + \left[(W_Q x)\cdot(W_K v_n)\right](W_V v_n)$$

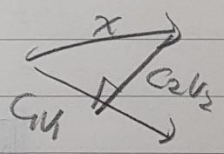
$$a_1 \tilde{v}_1 + a_2 \tilde{v}_2 + a_n \tilde{v}_n$$

$$\tilde{x} = a_1 \tilde{v}_1 + a_2 \tilde{v}_2 + \cdots + a_n \tilde{v}_n$$

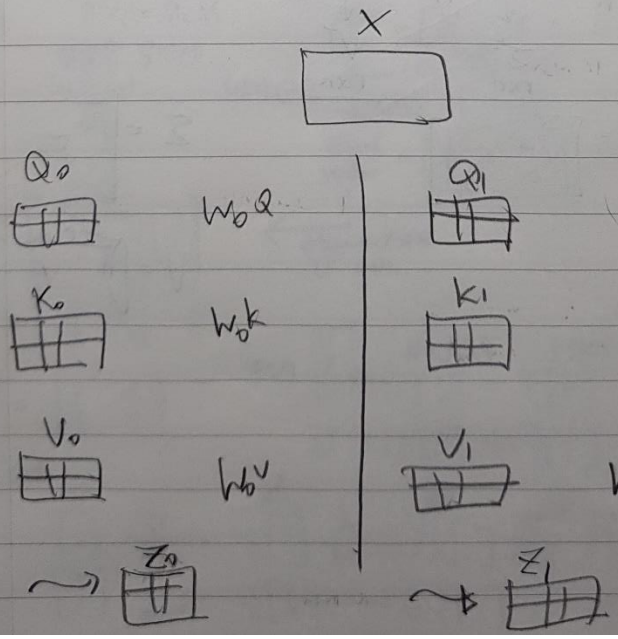
$$\frac{e^{a_1}}{\sum_{i=1}^{n} e^{a_i}} \tilde{v}_1 + \frac{e^{a_2}}{\sum_{i=1}^{n} e^{a_i}} \tilde{v}_2 + \cdots + \frac{e^{a_n}}{\sum_{i=1}^{n} e^{a_i}} \tilde{v}_n$$

$$\frac{e^{(Wx)\cdot(Wy)}}{\sum e^{(Wx)\cdot(Wy)}}$$





Multi-head attention



No.

문장 → subword Tokenizer → 토큰 ID 시퀀스

↓

Embedding layer → 각 토큰 벡터

+

Positional Embedding → 위치 정보를 더한 벡터 예시) $\sin\left(\frac{pos}{10000^{2i/d}}\right)$

↓

Transformer 입력

embedding 단위

(token)

improve
improving
improvement

→ improve
improve

one-hot encoding

$\begin{bmatrix} 0 \\ \vdots \\ 1 \\ 0 \end{bmatrix} = X$

Skip-gram
CBoV

$WX \in \mathbb{R}^{300}$

ELMo

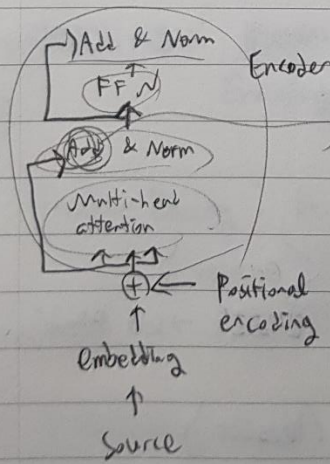
장점

subword → word price, BPE

unaffordable = un + afford + able <subword>

$= [78, 108, 93]$ ← 정수 index

Residual Connection



embedding_dim = num_heads × head_dims

MooKeuk

No.

Masked Attention

$$\text{Masked Attention}(Q,K,V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}} + \text{mask}\right)\cdot V$$

필요한 이유

: transformer는 입력 전체를 동시에 처리, 미래 단어를 봐야함

: Mask로 자기 자신보다 앞의 단어만 보게 제한

Training

입력문장 X → Encoder → context vector

출력문장 y → Decoder → 다음 토큰 예측

1. 입력문장을 토큰화 → Encoder 전달
2. 출력문장을 Decoder에 입력
3. Decoder는 각 위치에서 다음 토큰 예측
4. Cross Entrophy Loss 계산

$$L = -\sum \log P(y_t \mid x, y_{<t})$$

5. Optimizer 통해 파라미터 업데이트

병렬화 학습, Teacher Forcing

병렬화 학습 ; RNN과 다름,

Encoder ; 전체 시퀀스를 한번에 attention으로 처리

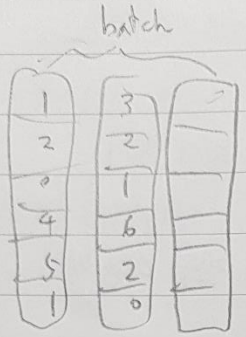
Teacher Forcing ; Decoder 학습시 이전에 생성된 예측값 아닌 정답토큰을 다음 입력으로 사용

Transformer에서 학습하는 parameter

① Embedding Layer
② Attention Weight
③ Position-wise FFNN
④ Layer Norm 파라미터
⑤ 최종출력 Projection

MOOKEUK

Layer Normalization



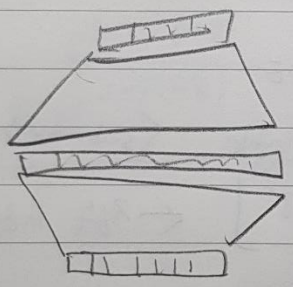
정규화 대상 : 각 샘플의 특성 (dim)

mean 2 3 3

std 2 2 2

Position-wise Feed-Forward Network

$$FFN(x) = \max(0, xW_1 + b_1)W_2 + b_2$$



입력 : 하나의 토큰 벡터

출력 : 같은 차원의 벡터

특징 : Position-wise

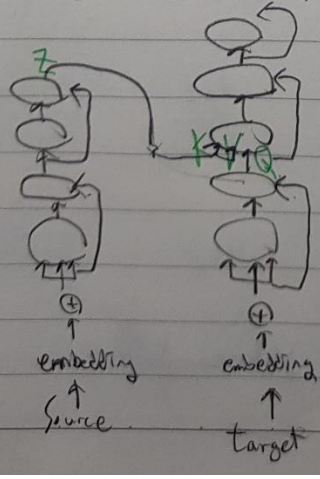
MoE Position-wise FFNN을 여러개의 전문가 네트워크로 확장한 구조

선택은 게이트 네트워크가 결정

$$MoE(x) = \sum g_i(x) \cdot E_i(x)$$

K, V의 역할

in Encoder → decoder



〈Encoder〉

입력토큰들 → self Attention → encoder output → K, V

〈Decoder〉

출력토큰들 → Masked self Attention → Q

↓

Cross Attention (Q, K, V)