Attention is all you need

Vaswani, Ashish, et al.

경상국립대학교 석사과정 오서영

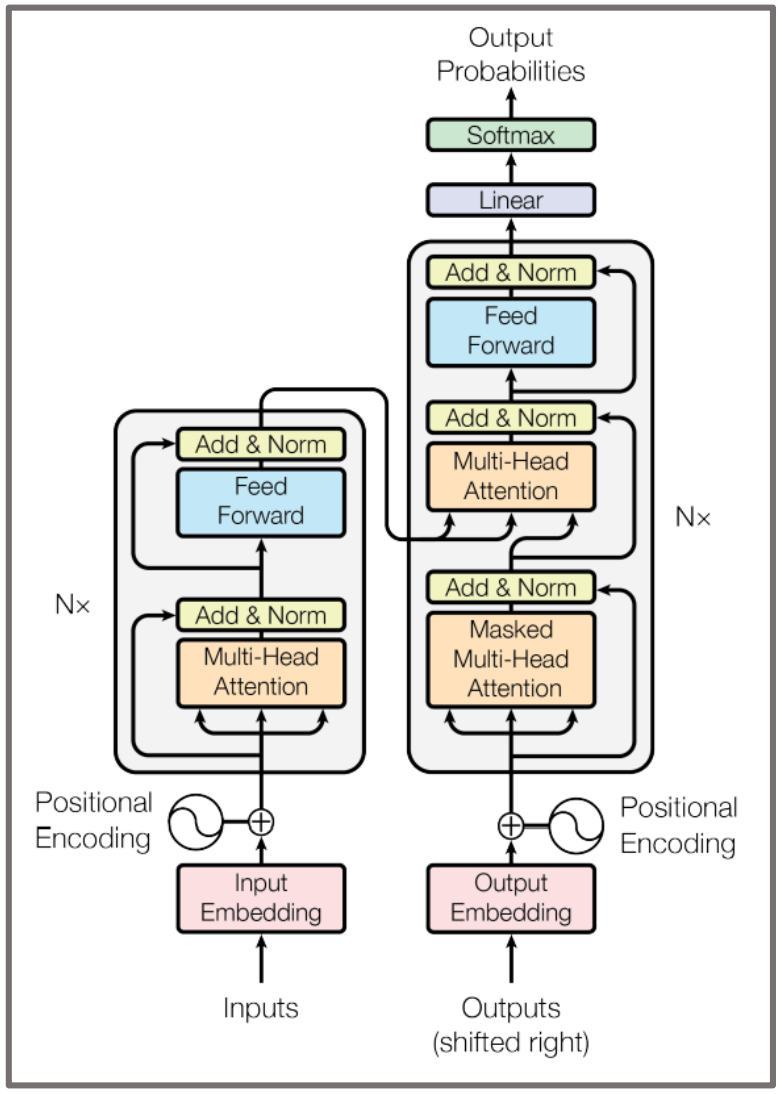
Abstract & Introduction

Transformer : attention 메커니즘 기반 모델, recurrence와 convolution 배제

Recurrent model은 input과 output sequence의 symbol 위치를 따라 계산함
→ 계산 시간 단계에 위치를 정렬하면, 이전 hidden state h_{t-1}와 위치 t의 입력으로 hidden state h_t sequence를 생성
→ 이러한 순차적인 특성은 메모리 제약이 일괄처리를 제한하기 때문에, 더 긴 sequence가 필요한 데이터의 경우 parallelization을 불가능하게함

Transformer는 recurrence를 피하는 대신, input과 output 사이의 global dependency를 이끌어 내는 attention 메커니즘에 의존하는 모델

Encoder & Decoder stacks



Transformer

Encoder

- 6개의 동일한 layer의 스택.
- 각 layer 은 두 개의 sub layer을 가짐
- multi-head self-attention mechanism -> point-wise fully connected feed-forward network
- 각각의 sub layer에 residual connection -> layer normalization

Decoder

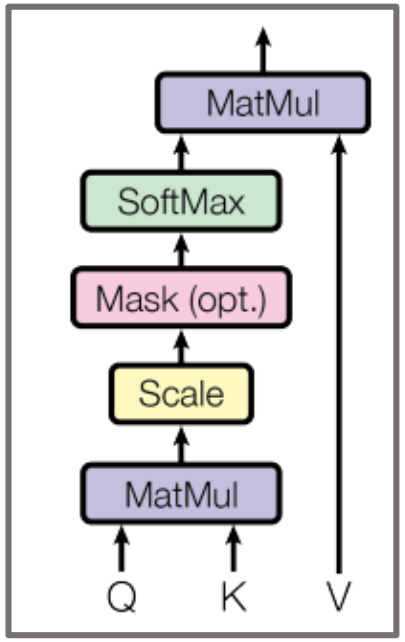
- 6개의 동일한 layer의 스택.
- 각 layer 은 세 개의 sub layer을 가짐
- encoder에 추가로 encoder의 output에 multi-head attention 수행
- 각각의 sub layer에 residual connection -> layer normalization
- Masking : 예측이 i번째 이전의 위치에서 알려진 출력에만 의존 할 수 있게함

Attention

Attention
- Query와 key-value 쌍을 output에 매핑함
- Output은 value의 가중합으로 계산되며, 각 값에 할당된 가중치는 해당 key와 query의 함수에 의해 계산됨

Scaled dot-product Attention
- Input은 d_k 차원의 key와 query, d_v차원의 value

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$



Attention

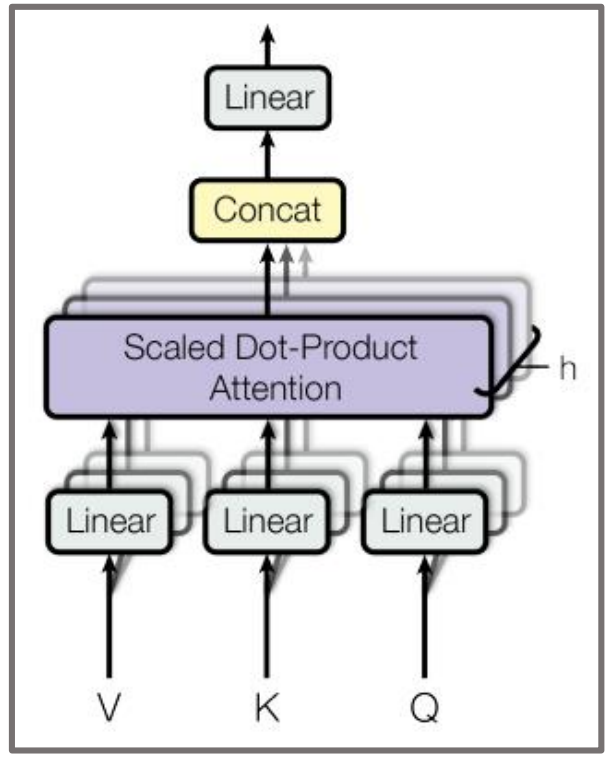
Multi-head Attention

- Query와 key-value 쌍을 output에 매핑함
- Output은 value의 가중합으로 계산되며, 각 값에 할당된 가중치는 해당 key와 query의 함수에 의해 계산됨
- 모델이 서로 다른 위치의 서로 다른 부분 공간의 정보에 동시에 주의를 기울일 수 있음

$$\mathrm{MultiHead}(Q, K, V) = \mathrm{Concat}(\mathrm{head}_1, ..., \mathrm{head}_\mathrm{h})W^O$$

$$\text{where } \mathrm{head}_\mathrm{i} = \mathrm{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

Where the projections are parameter matrices $W_i^Q \in \mathbb{R}^{d_{\text{model}} \times d_k}$, $W_i^K \in \mathbb{R}^{d_{\text{model}} \times d_k}$, $W_i^V \in \mathbb{R}^{d_{\text{model}} \times d_v}$ and $W^O \in \mathbb{R}^{hd_v \times d_{\text{model}}}$.

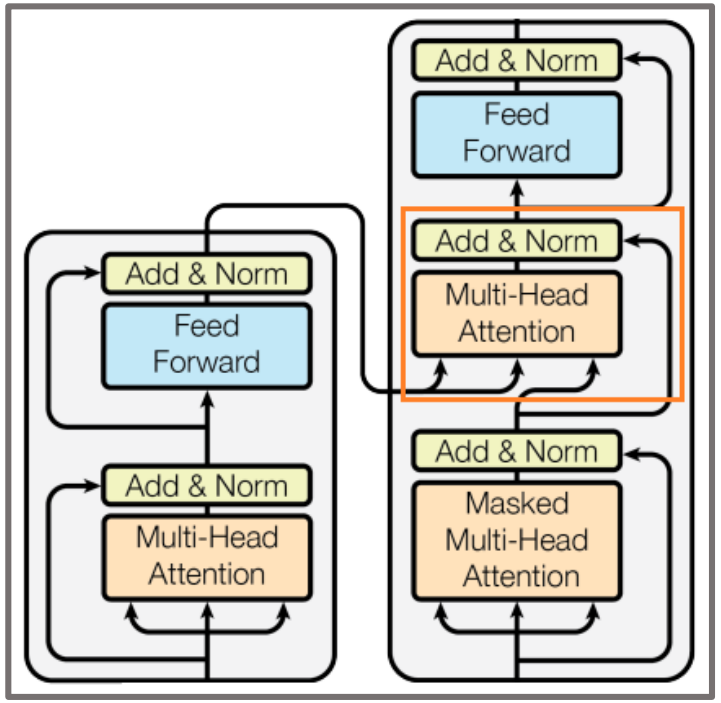


Application of attention in our model

1. Encoder-decoder attention layer

- Query는 이전 decoder에서 나오고 memory key와 value는 인코더의 출력에서 나옴

→ Decoder의 모든 위치에 input sequence의 모든 위치가 attend함

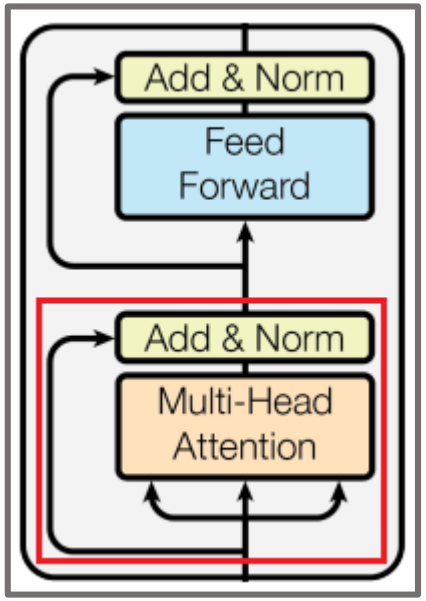


Application of attention in our model

2. The encoder contains self-attention layers

- 모든 key, value, query가 동일한 위치에서 나옴

→ encoder에서 이전 layer의 output
→ Encoder의 각 위치는 encoder의 이전 layer의 모든 위치에서 attend

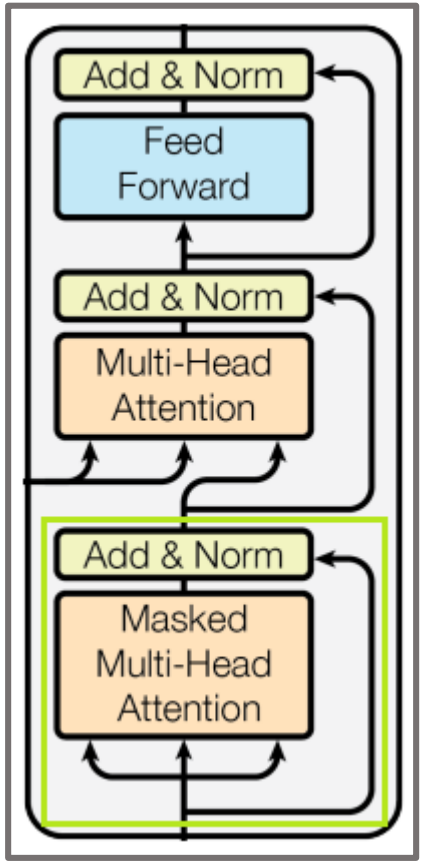


Application of attention in our model

3. The decoder contains self-attention layers

- Decoder의 각 위치는 해당 위치를 포함하여 decode의 모든 위치에 attend함

- Auto-regressive property를 보존하기 위해, input의 모든 값들을 masking 하여 scaled dot-product attention



Point-wise feed-forward networks

Encoder와 decode의 각 layer에는 fully connected feed-forward network가 포함되며,
각 위치에 개별적으로 동일하게 적용됨
→ 두 선형 변환 with ReLU

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

선형 변환은 서로 다른 위치에서 동일하지만 layer마다 다른 파라미터를 사용함

Positional Encoding

이 모델은 recurrence와 convolution이 없기 때문에 모델이 sequence 순서를 활용하기 위해 positional encoding을 활용함
→ 위치 encoding을 encoder 및 decoder 스택의 바닥에 embedding함
→ Embedding과 동일한 차원 d_{model}→ 둘을 더할 수 있음

다른 주파수를 가진 sine 및 cosine 함수를 사용함:

$$PE_{(pos,2i)} = sin(pos/10000^{2i/d_{\text{model}}})$$

$$PE_{(pos,2i+1)} = cos(pos/10000^{2i/d_{\text{model}}})$$

- pos는 위치, i는 차수
- 파장은 2π에서 10000 · 2π로 기하학적 진행
- 고정 offset k에 대해 $PE_{(pos,k)}$는 $PE_{(pos)}$의 선형 함수로 표현될 수 있기 때문에, 모델이 쉽게 상대적인 위치에 의해 attend 가능하게 할 것이라고 가정함

Experiments

Dataset : the standard WMT 2014 English-German dataset consisting of about 4.5 million sentence pairs
Optimizer : Adam optimizer & training 과정에서 learning rate를 변화시킴

$$lrate = d_{\text{model}}^{-0.5} \cdot \min(step\_num^{-0.5}, step\_num \cdot warmup\_steps^{-1.5})$$

Regularization : Residual Dropout, Label smoothing

Model	BLEU		Training Cost (FLOPs)	
	EN-DE	EN-FR	EN-DE	EN-FR
ByteNet [18]	23.75			
Deep-Att + PosUnk [39]		39.2		$1.0 \cdot 10^{20}$
GNMT + RL [38]	24.6	39.92	$2.3 \cdot 10^{19}$	$1.4 \cdot 10^{20}$
ConvS2S [9]	25.16	40.46	$9.6 \cdot 10^{18}$	$1.5 \cdot 10^{20}$
MoE [32]	26.03	40.56	$2.0 \cdot 10^{19}$	$1.2 \cdot 10^{20}$
Deep-Att + PosUnk Ensemble [39]		40.4		$8.0 \cdot 10^{20}$
GNMT + RL Ensemble [38]	26.30	41.16	$1.8 \cdot 10^{20}$	$1.1 \cdot 10^{21}$
ConvS2S Ensemble [9]	26.36	**41.29**	$7.7 \cdot 10^{19}$	$1.2 \cdot 10^{21}$
Transformer (base model)	27.3	38.1	**$3.3 \cdot 10^{18}$**	
Transformer (big)	**28.4**	**41.8**	$2.3 \cdot 10^{19}$	

BLEU score table

better BLEU score
than previous state-of-the-art

Conclusion

Transformer는 attention 기반 첫 sequence 변환 모델
- 가장 일반적으로 사용되는 Recurrent layer을 multi-head self-attention으로 대체